aurum

AURUM MÜHENDİSLİK SİSTEMLERİ VE MİMARLIK DERGİSİ
AURUM JOURNAL OF ENGINEERING SYSTEMS AND ARCHITECTURE

Cilt 2, Sayı 2 | Kış 2018
Volume 2, No. 2 | Winter 2018, 29-43

RESEARCH ARTICLE/ARAŞTIRMA MAKALESİ

COMPARISION OF ANT COLONY AND GENETIC ALGORITHMS FOR THE SOLUTION OF TRAVEL SALESMAN PROBLEM

Waled Milad Abulsasem ALASHHEB[1]
[1]Altınbaş University, School of Engineering and Natural Sciences, Information Technologies, Istanbul.
xxxx@altinbas.edu.tr

Adil Deniz DURU[2]
[2]Marmara University, Department of Physical Education and Sports Teaching, Istanbul.
deniz.duru@altinbas.edu.tr ORCID No: 0000-0003-3014-9626

Oğuz BAYAT[3]
[3]Altınbaş University, School of Engineering and Natural Sciences, Electrical and Electronics Engineering,
Istanbul. oguz.bayat@altinbas.edu.tr ORCID No: 0000-0001-5988-8882

Osman Nuri UÇAN[4]
[4]Altınbaş University, School of Engineering and Natural Sciences,
Electrical and Electronics Engineering, Istanbul.
osman.ucan@altinbas.edu.tr ORCID No: 0000-0002-4100-0045

Received Date/Geliş Tarihi: 28/11/2018 **Accepted Date/Kabul Tarihi:** 25/01/2019

Abstract

The Theory of computational complexity is an essential branch of study in the science of theoretical computing and mathematics. The resolution of Polynomial and Non Polynomial problems is one of the main problems that have open solutions, for which no famous efficient algorithm exist. The Problem of Traveling Salesman (TSP) is an example of these problems. Such a problem include, a count of specified cities must be visited by a traveling salesman where both start and end points will be the same city and getting a tour of all cities so that the complete distance or time is minimized will be the aim. The application of Optimization algorithms is one of the famous methods of the solution regarding to the TSP. These algorithms usually simulate the occurring phenomena in nature. Currently there exist several of such algorithms; for example, Genetic Algorithm (GA) and Optimization of Ant Colony (ACO).

This paper aimed to compare two approaches, GA and ACO for solution of TSP. The results obtained from our experiments showed that the ACO is better than GA since it requires less execution time for solving the same problem.

Keywords: Problem of Traveling Salesman, Optimization, Genetic Algorithm, Optimization of Ant Colony.

GEZGİN SATICI PROBLEMİNİN ÇÖZÜMÜ İÇİN KARINCA KOLONİ VE GENETİK ALGORİTMALARININ KARŞILAŞTIRILMASI

Özet

Polinom zamanda çözülebilecek (P) ve polinom zamanda doğrulanabilecek (NP) problemlerin bilinen etkin bir algoritmasının olmaması, hesaplamadaki karmaşıklık teorisinin teorik hesaplama ve matematiğin gerekli bir bilimsel çalışma kolu olmasını sağlamıştır. Gezgin satıcı problemi (GSP) bu tür problemlere örnektir. Bu problemde, satıcı tarafından belli sayıda şehirin ziyaret edilmesi istenir. Başlangıç ve bitiş şehri olarak aynı şehir ele alınır. GSP'nin amacı bir turu en az mesafe ve zamanda bitirmesidir. Evrimsel algoritmalar, GSP çözümü için kullanılan popüler yöntemlerdendir. Bu algoritmalar genelde doğada oluşan olayların benzeşimini temel almaktadır. Günümüzde, karınca kolonisi eniyileştirmesi (KKE) ve genetik algoritma (GA) bu tür algoritmalara örnektir. Bu tez kapsamında, GSP çözümü KKE ve GA ile gerçekleştirilerek sonuçları karşılaştırılmıştır. Deneyler sonucu elde edilen sonuçlar, KKE nun GA dan daha başarılı sonuç verdiği ve aynı problemin çözümü için daha az zaman kullandığı görülmüştür.

Anahtar Kelimeler: Gezgin satıcı problem, Eniyileştirme, Genetik Algoritma, Karınca Kolonisi Eniyileştirme.

1. INTRODUCTION

Several groups of complexity exist that differentiate problems according to their "difficulties". Recently the only identified method that assured to solve optimally the problem of traveling salesman of every size, is through computing every tour that is possible and inspection for the tour with lowest cost. Therefore, in mathematics of computation the TSP is an example of the intensively studied problems [1].

Optimization is consider as an example of the best significant tasks of engineers, which the engineer permanently enquired to design further effective and cheap systems in addition to invent such techniques and plan to progress tasks of running systems in several arenas mainly in the world of industry and scientific [2]. The (TSP) is a polynomial hard and nondeterministic problem in combinatorial optimization studied in researches of operations and algorithms, also studies of theoretical computer science.

The basic problem generally summarized as there are a group of cities with their distances among them or costs, all cities must be visited by a travelling salesman, but on travelling he need to minimize time, therefore we want to find the appropriate series of cities to lessen the distances or traveled costs.

When M different cities are travelled by a salesman. The most essential question is: In what recommended concatenated set of cities must he visited to lessen the cost or traveled total distance [3].

Social insects for example ants, bees, wasps, and termites live in nearly every habitat land on earth. An (ACO) technique is an optimization approach inspired its behaviors from ants to solve problems of combinatorial optimization [4].

(ACO) is an example of the furthermost effective methods in the broader field of swarm intelligence. Environment is used as a medium of communication by the ants and deposits pheromones used in process of interchanging information indirectly among ants, all particularizing the situation of their "action". Algorithm of ACO models the conduct of actual ant's colonies in establishment the shortest route among nests and sources of food. The pheromone is released on the ground by ants while routing from their nest to food and after that return to the nest. The movement of ants is depend to the quantity of pheromones, the richer the trail of pheromone on a route is, the further probable it would be trailed by other ants. So a shorter route has higher quantity of pheromone in probability, ants will tend to select a shorter route [4].

Genetic algorithms (GA) are a relatively novel approach of optimization which can be useful in several problems, containing those that are NP-hard. The approach does not guarantee an ideal solution, however it generally offers good estimates in a reasonable amount of time. This, therefore, would be a good method to try on the TSP, which is an obvious of the most well-known problems that are NP-hard [5].

GA are insecurely constructed on evolution of nature and utilize a technique of "survival of the fittest", where the best solutions survive and are varied until we obtain a result that is good and accepted [6].

Thus, now days there are different solving algorithms to solve TSP, like Genetic Algorithm, ACO and Bee Colony. As a result, for their solution the users must identify the type of approaches and algorithms so well to overawe of these kinds of problems [7].

According to this concept, the analysis of two most common solution algorithms of the TSP are based on (GA) and (ACO) which are studied and applied in this paper.

2. METHODS

2.1. Traveling Salesman Problem

Traveling Salesman Problem is a really significant problem in operational research.

TSP is a kind of common problem which shows one from to solve the further difficult ones. Tour of Hamiltonian starts from a specified city, visiting each of the particular groups of cities and after that return back to the starting city of departure. TSP is the shortest route in a circuit ensuring that it only once will passes all vertices [8].

To represent the TSP as a mathematical formulation, we assume that there is a set of different cities $\{C_1, C_2, \ldots, C_m\}$ and an edge corresponding to each couple of cities $\{C_i, C_j\}$ with a path that is closed $C_a = \{C_{a(1)}, C_{a(2)}, \ldots C_{a(M)}\}$. For the salesman the aim is to obtain a set of cities so that the distance or total cost is decreased. The minimum obtained result is known as the optimal result. Eq. (1) gives the objective function:

$$\sum_{j=1}^{M-1} d(C_{a(j)}, C_{a(j+1)}) + d(C_{a(M)}, C_{a(1)}) \tag{1}$$

It is known as specified group of m cities, named $\{c_1, c_2, \ldots, c_m\}$, and permutations, $s_1, \ldots, s_m$!. The goal is to select σ_i such that the summation of totally distances of Euclidean in a tour among cities is reduced.

TSP topic can be solved through enumerating (m-1)!/2 to calculate number of possible routes, where m represent the cities count and after that choosing the path that has the length which is the shortest among all routes [9].

The optimization algorithms based on population and nature are the methods which are in the class of the recent dependent methods of optimization. GA's and ACO's are considered as the methods of problem solving stimulated depending on nature that could be observed.

Evolutionary Algorithms (EAs) are good search techniques, which are based on search method as exist in the nature, such method ensures that the fittest solution will survive and the bad solution will discarded. To perform a search process in EA, the solutions will represented as population. The repetition of an EA include choosing the best solution among all solutions in the population depending on a competitive which leads to the continuing existence of the fittest and poor solutions are eliminated from the population.

During last 50 years four main paradigms of evolutionary algorithm have been presented. Among those genetic algorithm is an approach of computation, suggested mainly in 1975 by Holland. Besides that, in 1992, Beckers proposed the heuristic of ACO which is founded on the idea that the shortest route between the food and the nest is constructed by real ant. A group of artificial ants are utilized by the algorithm, and collaborate through exchanging knowledge via deposited pheromone to find solution for a problem [10].

2.2. Genetic Algorithm

GA is a heuristic method of search. This algorithm starts with original solution via randomly choosing a population.

Each component of population is known as a chromosome which is a string of symbols. Chromosomes themselves are collection of genes. The actual value of a control parameter, is encoded in a gene [5].

Assuming a first random population generated and evaluated, genetic evolution happen by using essential operators of genetic:

1) Selection of Parent;

2) Crossover;

3) Mutation.

4) Termination

In order to preserve population that develop from one generation to another, the GA utilizes three processes. The initially process is "Selection" process which is accomplished according to by the precept of 'the fittest will survived '. The search starts from a randomly produced population that survive over successive generations (iterations).Among the exist results. Each time two results are selected as parent results by selection process according to fitness function. The second process is the "Crossover" process, which is based on mating in biological populations. The operator of crossover inherits characteristics of well designs from the parent population into the successive population, which will have better value of fitness. The last process is "Mutation" which causes variety in features of population. It randomly reasons local alterations to the novel generation. The novel generation is similar to the parent unless one or more changes created by process of mutation.

Repeating processes of selection, mutation and crossover to generate further novel solutions until the size of population of the novel generation is similar to that of the old solution. Then from this novel population, the iteration process starts. Since best solutions have a higher probability to be chosen for crossover and the new solutions generated carry the characteristics of their parents. It is anticipated that the novel generation will be better than the old generation. The algorithm continues until the solution quality could not be easily enhanced or the number of generations is reached to n [11]. The basic algorithm of getting solution for GA is illustrated in figure 1 [7]:

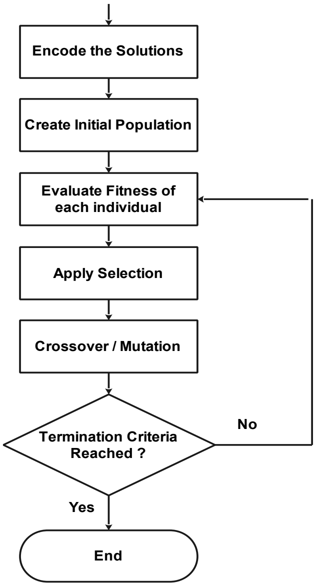

Figure 1. Basic algorithm of GA

2.3. Ant Colony Optimization

The ACO is based on the idea of food search conduct of real ants and their aptitude in finding the optimum routes. It is a common search method that based on population for the solution of hard combinatorial optimization problems and other ants randomly trail one of the routs, likewise leaving trails of pheromone. Because the ants on the shortest route put pheromone trails faster, this route becomes strengthened with more pheromone, making it more attractive to other subsequent ants [12].

The ants become progressively probable to trail the shortest route because it is continually strengthened with a more amount of pheromones. The pheromone trails of the longer routes will vaporize.

Scheme:

Construct solutions of ant

Describe attractiveness τ, according to experience from earlier results.

Define particular function of visibility, η, for an existing problem (e.g. distance) [13].

The essence of ant colony system method is based on idea that pheromone which is a private chemical trail is putted on the ground by ant throughout move, which leads the other ants to the objective solution. In case when more ants go through the tour more pheromone is left, which enhanced the likelihood of other's ants selecting this tour. Moreover, the pheromone has a lessening action since trail will evaporated over time. Besides that, the count of ants using this trail will specify the amount of pheromone putted by ants [14].

ACO algorithm:

Define the visibility, η, and base attractiveness, τ,
for every edge;
for k < MaxIteration do:
for every ant do:
select probabilistically (according to preceding equation) the next state to travel into;
for each ant *add* that travel to the tabu list of it;
repeat until every ant accomplished a route;
end;
for every ant that accomplished a route do:
for every edge that the ant traversed *update the* attractiveness τ;
end;
if (local best route better than global route)
save local best route as global route;
end;
end; [15].

There are several ACO algorithms. The System of Ant (AS) was the first approach of ACO that is utilized to solve Problem of Traveling Salesman [16].

In AS a solution (tour) of the TSP is constructed by every of m (artificial) ants, no local search is exist in AS.

- **Construction of tour**: Initially each ant is placed on a city that is randomly selected. The likelihood with which ant q, presently at city r , selects to go to city s at the τ th repetition of the procedure is calculated by:

$$P_{rs}^{q}(t) = \begin{cases} \frac{[\tau_{rs}(t)]^{\alpha}[n_{rs}]^{\beta}}{\sum_{l \in N_r^q}[\tau_{rl}(t)]^{\alpha}[n_{rl}]^{\beta}} & if \;\; s \in N_s^q \\ 0 & otherwise \end{cases} \quad (2)$$

n_{rs} is the Control of the relative significance of the pheromone versus the information of heuristic, is calculated by $n_{rs} = \frac{1}{d_{rs}}$, where d_{rs} is the remoteness between cities s and r. β and α are two parameters which specify the relative effect of the heuristic information and the pheromone trail, and N_s^q is the favorable neighborhood of ant q, that is the group of cities that ant q not visited yet. If α = 0, the nearest cities are more possible to be chosen, when β = 0, only pheromone strength is at work [17][18].

- **Update of Pheromone**. The process to update the trails of pheromone performed by each ant. This is performed by initially lessening the strength of pheromone on all arcs using a constant factor and after that permitting every ant to increase pheromone value on the arcs it has traversed as in Eq. (3):

$$\tau_{rs}(t+1) \leftarrow (1-\rho).\,\tau_{rs} + \sum_{d=1}^{m} \Delta\tau_{rs}^{q}\,(t) \quad (3)$$

where 1 > p ≥ 0 is the evaporation rate of trail of pheromone. The parameter *p* is utilized to evade limitless accumulation of the trails of pheromone and it allows the system to "forget" formerly selected solutions which it bad. If an arc is not selected by the ants, there will be exponentially decreases on its associated pheromone strength. m refers to the count of ants. $\Delta\tau_{rs}^{q}(t)$ is the quantity of pheromone that ant *q* places on the arcs or edge (r,s) it has visited; it is calculated as following Eq. (4) :

$$\Delta\tau_{rs}^{q} = \begin{cases} \frac{D}{L^{q}(t)} & if \; ant \; q \; used \; edge \; or \; acr(r,s) \; in \; its \; tour, \\ 0 & otherwise \end{cases} \quad (4)$$

where *D* is a constant , and $L^{q}(t)$ is the distance of the tour created by ant q. the better the ant's path is, the further pheromone is leaved by arcs contained in the path. [18].

2.4. Haversine formula

Previously, sailors needed to know the distance from their known longitudinal and latitudinal coordinates to those of their wanted target by utilizing of what became identified as the formula of Haversine. Beginning with the spherical law of Cosines, and carrying out suitable changes utilizing the definition of haversine, and the alteration of cosines identity, then we can get the law of haversines. From there, we can relate it to the particular situation of the earth to reach at the formula of haverine, after that finding a distance between two points through utilizing this formula.

As shown in figure 2, arcs of great-circle form the sides of a triangle of sphere, and where two arcs intersect which made the angle of spherical. On a surface of sphere, a great circle route, is often the shortest path among two points [19].

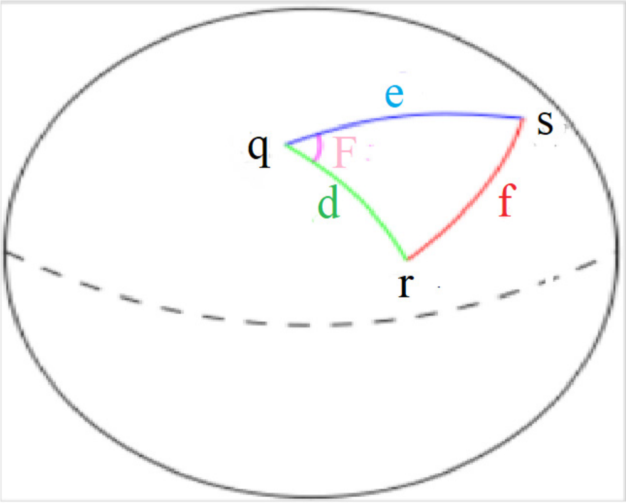

Figure 2. Spherical Triangle

The Cosines Spherical Law:

$$\cos f = \sin d \sin e \cos F + \cos d \cos e \quad (5)$$

Haversine function can also given by

$$\text{haversin}(\theta) = \sin^2(\theta/2) = \frac{1-\cos(\theta)}{2} \quad (6)$$

$$\text{hav}\,(f) = \sin d \sin e \,\text{hav}\,(F) + \text{hav}(d - e) \quad (7)$$

Let the distance of sphere among any two points on the surface of the earth be q, the radius of the earth be r, $\beta 1$ and $\beta 2$ respectively be longitudes of point 1 and point 2 and $\alpha 1$ and $\alpha 2$ be their latitudes respectively. The central angle among the points has a haversine which is [19]:

$$\text{hav}\,\left(\frac{q}{r}\right) = \text{hav}(\alpha 2 - \alpha 1) + \cos(\alpha 1)\cos(\alpha 2)\,\text{hav}\,(\beta 2 - \beta 1) \quad (8)$$

where:

- q is the distance among the two cities.
- r is the radius of the earth. In this situation the radius of the earth is 6.378.137 km.
- $\alpha 1$; $\alpha 2$: latitude of city 1 and city 2 respectively.
- $\beta 1$; $\beta 2$: *longitude* of city 1 and city 2 respectively.

The central angle is q/r on the left side of the equation, presumptuous that angles are measured using radians (α and β can be transformed from degrees to radians by multiplying it by $\pi/180$).

Finally:

$$q = 2 * r * \arcsin(\sqrt{haversin(\alpha_2 - \alpha_1)\cos(\alpha_1) * \cos(\alpha_2)\, haversine(\beta_2 - \beta_1)})$$

$$q = 2 * r * \arcsin(\sqrt{sin^2\left(\frac{\alpha_2 - \alpha_1}{2}\right)\cos(\alpha_1) * \cos(\alpha_2)\, sin^2\left(\frac{\beta_2 - \beta_1}{2}\right)}) \quad (9)$$

3. DATA AND RESULTS

We executed the GA and ACO in two different computers, the first commodity computer is Dell; with processor Intel® Core™ i7 CPU, 2.80 GHz and 4 GB RAM. Under Windows 10 version environment. The second commodity computer is Toshiba; with processor Intel® Core™ i5 CPU, 2.50 GHz and 8 GB RAM. Under Windows 7 version environment. We used 100 of Turkish cities (name, latitude, longitude) as data that are reported in table 1.

Name	latitude	longitude	Name	latitude	longitude
Istanbul	41.00396	28.4516782	Cankir	40.60734	33.544547
Kirklareli	41.73627	27.1828603	Corum	40.5352	34.871863
Edirn	41.66887	26.5383047	Bafra	41.56421	35.870587
Tekirdag	40.95979	27.4879895	Samsun	41.29145	36.243658
Gebze	40.80449	29.3523956	Ordu	40.99345	37.766088
Yalova	40.64358	29.196147	Trabzon	40.99292	39.661219
Bursa	40.22159	28.8922011	Bayburt	40.25911	40.191333
Canakkale	40.13054	26.3570519	Rize	41.03144	40.476569
Balikesir	39.6478	27.8197115	Kars	40.59669	43.066768
Ayvalik	39.3336	26.6737007	Erzurum	39.91188	41.193476
Akhisar	38.92383	27.750777	Erzincan	39.74705	39.433867
Manisa	38.62374	27.3236154	Van	38.50282	43.218466
Izmir	38.41734	26.7995137	Batman	37.8955	41.061696
Urla	38.32663	26.7469164	Mardin	37.47719	39.998102
Cesme	38.32068	26.2268383	Elazig	38.66464	39.128111
Kusadasi	37.8557	27.2257933	Bingol	38.88318	40.483338
Selcuk	37.94868	27.3555784	Diarbakir	37.92277	40.092637
Soke	37.75261	27.3783798	Siirt	37.93037	41.898367
Aydin	37.83576	27.7773754	Patnos	39.23281	42.82282
Didim	37.3723	27.2338948	Ercis	39.02911	43.329135
Milas	37.30901	27.7450818	Hakkari	37.57427	43.699772
Bodrum	37.03579	27.3777541	Yuksekova	37.56806	44.244879
Cine	37.61254	28.0445988	Siverek	37.75368	39.291625

Mugla	37.21014	28.3467873	Sanliurfa	37.16706	38.685736
Marmaris	36.84808	28.2264257	Kilis	36.71955	37.09086
Nazilli	37.91028	28.2915433	Osmaniye	37.33439	35.739747
Denizli	37.78289	29.011605	Nigde	37.88717	34.070876
Fethiye	36.6518	29.0518288	Mus	38.75084	41.436425
Kas	36.20085	29.6322359	Tatvan	38.50373	42.221294
Antalya	37.82368	29.1268648	Sirnak	37.52206	42.438876
Isparta	37.79012	30.479748	Cizre	37.33158	39.944585
Alanya	36.54378	31.9426108	Ahlat	38.74925	42.45206
Beysehir	37.68119	31.6932248	Sakarya	40.74894	30.242206
Konya	37.87817	32.2262788	Duzce	40.8531	31.082558
KKaraman	37.17927	33.1550208	Amasra	41.74388	32.369842
Mersin	36.74263	34.3888169	Bartin	41.62977	32.303097
Tarsus	36.9229	34.8619617	Sinop	42.01411	35.059865
Adana	36.99733	35.147978	Gerze	41.80196	35.169635
Gaziantep	37.05875	37.3100958	Boyabat	41.46586	34.760633
Kahramanaras	37.55536	36.8415508	Tosya	41.01563	34.002652
Kayseri	38.72332	35.3300985	Unye	41.11399	37.167597
Aksaray	38.36072	33.8605277	Fatsa	41.04933	37.401598
Ankara	39.90304	32.4825626	Sungurlu	40.16556	34.353053
Eskisehir	39.76526	30.4047261	Yozgat	39.81521	34.777138
Beypazari	40.16237	31.8909748	Sorgun	39.8125	35.147374
Bolu	40.73572	31.5449295	Polatli	39.57855	32.10304
Zonguldak	41.459	31.7301196	Golbasi	39.78536	32.769855
Karabuk	41.2062	32.5806211	Giresun	40.90424	38.309193
Kastamonu	41.39762	33.7181286	Hopa	41.40239	41.397786
Ardahan	41.11288	42.6866254	Artvin	41.17966	41.80057

Table 1. Cities and their latitudes and Longitudes

3.4. implementation (1st hardware)

3.4.1. Genetic Algorithm

a- With the parameters:

Population Size= 8.

MUTATION RATE = 0.25.

TOURNAMENT SELECTION SIZE = 3.

Generation number	Fitness value	Distance of Shortest Route In kilometers	Execution Time In seconds
48	0.2048	48821.00	3
7	0.1900	52638.00	1
28	0.1936	51647.00	2
29	0.1939	51584.00	1
46	0.1890	52919.00	<1

Table 2. Results of First Computer using GA(a)

b- With the parameters:

Population Size= 10.

MUTATION RATE = 0.5.

TOURNAMENT SELECTION SIZE = 3.

Generation number	Fitness value	Distance of Shortest Route In kilometers	Execution Time In seconds
23	0.1925	51956.00	1
35	0.1888	52957.00	1
5	0.1800	55546.00	< 1
43	0.1907	52446.00	1
11	0.1859	53799.00	< 1

Table 3. Results of First Computer using GA(b)

3.4.2. Ant Colony Optimization

a. With the parameters:

Q = 0.0005; the value is between 0 and 1

RHO = 0.2; the value is between 0 and 1

ALPHA = 0.01; the value is >=0

BETA = 9.5;

Count_OF_ANTS	Ant id	Distance of Shortest Route In kilometers	Execution Time In seconds
176	159	8266.34	1
429	209	8091.02	1
160	154	8301.32	1
37	3	8723.53	<1
313	247	7807.34	1

Table 4. Results of First Computer using ACO(a)

b. With the parameters:

Q = 0.0005; the value is between 0 and 1

RHO = 0.5; the value is between 0 and 1

ALPHA = 0.05; the value is >=0

BETA = 7.5;

Count_OF_ANTS	Ant id	Distance of Shortest Route In kilometers	Execution Time In seconds
496	125	8310.87	2
119	60	8713.85	1
348	322	8472.38	2
35	8	8418.67	<1
298	233	8287.01	1

Table 5. Results of First Computer using ACO(b)

3.5. Implementation (2nd Hardware)

3.5.3. Genetic Algorithm

With the parameters:

Population Size= 10.

MUTATION RATE = 0.5.

TOURNAMENT SELECTION SIZE = 3.

Generation number	Fitness value	Distance of Shortest Route In kilometers	Execution Time In seconds
31	0.2076	48159.00	< 1
3	0.1807	55330.00	< 1
45	0.2028	49303.00	1
28	0.2003	49923.00	< 1
45	0.1911	52330.00	1

Table 6. Results of Second Computer using GA

3.5.4. Ant Colony Optimization

With the parameters:

Q = 0.0005; the value is between 0 and 1

RHO = 0.5; the value is between 0 and 1

ALPHA = 0.05; the value is >=0

BETA = 7.5;

Count_OF_ANTS	Ant id	Distance of Shortest Route In kilometers	Execution Time In seconds
228	13	8441.91	< 1
290	12	8393.07	< 1
236	127	8247.83	< 1
264	109	8258.60	< 1
282	213	7975.19	< 1

Table 7. Results of Second Computer using ACO

4. DISCUSSION AND CONCLUSION

In this paper we examined and compared two population and evolutionary solution approaches of TSP which are Genetic algorithm and Ant Colony Optimization.

This comparison between two approaches depend on the results after experiments on two types of computers with their features and that experiments are performed using 100 Turkish cities.

The results of the experiments show that ACO is the better to find the optimal solution with less time for execution. At the number of ants in these experiments are between 1 and 500 randomly ants.

ACO depends on a number of parameters and very much on the count of ants used on the problem, number of cities, parameter used for adjusting the amount of pheromone deposited, parameter used for varying the level of pheromone evaporation, parameter used for controlling the significance of the pheromone trail and parameter used for controlling the significance of the distance between source and destination

While using Genetic Algorithm, the number of generations are between 1 and 50 randomly generated generations it takes longer time to produce the output.

GA depends also on a several parameters very much on the count of cities, population size, the method include that after encoding the problem the approaches of mutation and crossover and their rates are used.

In [15] different algorithms compared to benchmark the performance for solving the TSP. Ant Colony optimization was better than GA and simulated annealing because of its slow convergence.

While [16] concluded that under the experimental situations, when the population size is between 5 to 15 cities, the ACO for TSP problem is further operative; and when the population size is between 1~2.5 cities, it can get better results by using GA for solving TSP. likewise, in [15] the researchers found that ACO is better than GA with respect to cost of the distances among cities.

The researchers in [18] Observed that it is problematic to find out the best and accurate solution by ACO and GA for TSP, but the iteration depend upon the optimization of both algorithm. The number of iteration in the ACO and GA congregate the solution to optimization. The running time of GA to calculated distance is less than ACO for large no of iteration and cites which might increase gradually. Time taken by the GA is not as much of that the calculated time of ACO, the complexity of ACO is high than GA which is proved by experimental analysis. Whenever no of iteration is less ACO is better than GA and calculate the optimal solution but whenever iteration increases ACO the complexity of ACO increase as compare to GA. Both techniques provide the optimal solution but the ACO is better approach and calculate the optimal distance in the first few iterations when the number of searching agent is greater in number. The results generated by GA approach using less number of iterations is not satisfactory as compare to ant colony optimization for all cases of no of cities.

Thus our findings are agree and are the same as the results obtained by other mentioned researches.

Conflict of Interests/Çıkar Çatışması

Authors declare no conflict of interests/Yazarlar çıkar çatışması olmadığını belirtmişlerdir.

5. REFERENCES

A. Hashim, H. Muhammad , H. Fazl, Ahmadullah, Salman, and S. Yasir. 2016. "Solving Traveling Salesman Problem through Optimization Techniques Using Genetic Algorithm and Ant Colony Optimization", Journal of Applied Environmental and Biological Sciences, 6(4S)55-62, 2016.

B. Kylie. 2000. Genetic Algorithms and the Traveling Salesman Problem, Harvey Mudd college, Department of Mathematics

B.Shuoben, D. Xueshi and M. Yan. 2012. "The Design and Analysis of TSP Problem Based on Genetic Algorithm and Ant Colony Algorithm", I.J. Education and Management Engineering, vol. 9, p. 56-60, 2012.

Ç. Cergibozan. 2013. "Meta-Heuristic Solution Approaches For Traveling Salesman And Traveling Repairman Problems," Dokuz Eylül University Master Thesis, İzmir, Turkey

E. Chen and X. Liu. 2011. "Multi-Colony Ant Algorithm," *Ant Colony Optimization , Methods And Applications*, A. Ostfeld, Ed., InTech Publisher, pp. 3-12.

H. Demez. 2013. "Combinatorial Optimization: Solution Methods of Traveling Salesman Problem," Eastern Mediterranean University Master Thesis, Gazimağusa, Turkish Republic of Northern Cyprus

H.Z.C.S. Su and K.M. Aye. 2011. "An Ant Colony Optimization Algorithm for Solving Traveling Salesman Problem", International Conference on Information Communication and Management IPCSIT vol.16

J. J. Mwemezi and Y. Huang. 2011. "Optimal Facility Location on Spherical Surfaces: Algorithm and Application," *New York Science Journal,* vol. 4, no. 7,

K. PANDEY and P. JAIN. 2015. "Comparison Of Different Heuristic, Metaheuristic, Nature Based Optimization Algorithms For Travelling Salesman Problem Solution", International Journal of Management and Applied Science, ISSN: 2394-7926 Volume-1, Issue-2

M. Alhanjouri, B. Alfarra. 2011. Ant Colony versus Genetic Algorithm based on Travelling Salesman Problem. International Journal of Computer Technology and Applications, 2011. 2(3).

M. R. Bonyadi, M. R. Azghadi and H. Shah-Hosseini. 2008. "Population-Based Optimization Algorithms for Solving the Travelling Salesman Problem," *Travelling Salesman Problem*, I-Tech Education and Publishing, pp. 1-34.

R. Matai, S. P. Singh and M. L. Mittal. 2010. "Traveling Salesman Problem: An Overview of Applications, Formulations,and Solution Approaches," Traveling Salesman Problem*, Theory And Applications*, D. Davendra, Ed., InTech,

R. Stanec. 2011. "Solving of Travelling Salesman Problem for large number of cities in environment with constraints", Mendel University Faculty of Business and Economics Diploma thesis, Brno, Czech Republic

S. Al-hayali, O.N. Ucan, O. Bayat. 2018. "Genetic Algorithm for finding shortest paths Problem", Proceedings of the Fourth International Conference on Engineering & MIS

S. BURILLE. 2016. "Traveling Salesman Problem," Department of Mathematics, Saint Mary's College of California Senior Essay, California

S. Subiyanto, T. Narwadi. 2017. "An Application of Traveling Salesman Problem Using the Improved Genetic Algorithm on Android Google Maps," American Institute of Physics Publishing.

T.A. Mohammed, S. Sahmoud, O. Bayat. 2017. "Efficient hybrid memetic algorithm for multi-objective optimization problems", International Conference on Engineering and Technology (ICET), 2017.

T. Pellonpera. 2014. "Ant colony optimization and the vehicle routing problem"

T. Stautlez and M. Dorigo. 1999. "ACO algorithms for the traveling salesman problem," Evolutionary Algorithms in Engineering and Computer Science,Wiley