A Comparative Study of Metaheuristics Methods for Solving Traveling Salesman

A. Chandra[1]; A. Naro[2]

[1,2]Department of Industrial Engineering, Universitas Mercu Buana

***Abstract*— In this study, we compare 8 (eight) metaheuristics methods: GA, SA, TS, ACO, PSO, ABC, EFOA, and A^3 for solving traveling salesman problem (tsp) in 70 cities in Java island. The result shows ABC algorithm has the best value and the best average value. The results are tested by statistical methods both ANOVA and Tukey test which showed the data of distances are not the same for each method of metaheuristics and 20 pairs are different in means between each pair**

Keywords: Metaheuristics Methods, TSP, ANOVA, Tukey test

I. Introduction

Transportation organization process is one of the most important tasks in organization to deliver end product to customer. The organization efficiency of the transportation process is considered in the context of the effectiveness of the solutions offered by software, procedures and algorithms on basis of solutions offered [1]. There are numerous methods for solving optimization problem, starting from linear programming to modern heuristics. One of the popular and classical optimization problems and widely known is traveling salesman problem, TSP. The goal of TSP is to find the shortest distance by visiting all cities and returning to the starting point. TSP is categorized as a NP-hard, combinatorial optimization problems [2]. TSP is called a hard problem because it is large and complex problem which consists of (n-1)!/2 possible tours for n cities for symmetric TSP and (n-1)! possible tours for asymmetric TSP, then it can not be solved by standard solver [3], [4]. According to [5], there are two methods for optimization, one is an exact method and the other one is an approximate method. Exact methods are capable of precisely finding optimal solution, but they are not applicable for complicated optimization problems and their solution time increases exponentially in such problems. Approximate methods can find close-to-optimal solutions for difficult optimization problems within a short period of time [6]. In approximate methods, there are two categories, one is heuristics and the other one is metaheuristics. Approximate methods do not find the optimal solution or at least do not guarantee the optimality of the found solutions [7], [8]. Heuristics method can be classified into two categories: specific heuristics and metaheuristics. Specific heuristics are designed to specific problems and metaheuristics are more general purposed algorithms which can be used for almost optimization problems. Three kinds of metaheuristics methods: Ant Colony Optimization (ACO), Improved Particle Swarm Optimization (IPSO), Shuffled Frog Leaping Algorithm (SFLA) and the result showed IPSO has the best average values among the others and ACO appears more frequently as the best value among the others [9]. Then other authors do comparative study among ACO, PSO based Genetic Algorithm (GA), and Artificial Bee Colony (ABC), and the result showed ABC and PSO based GA have better results than ACO [10]. A study has reported that GA is better than Standard PSO [11]. Another author states that Improved Fruit Fly Algorithm (EFOA) gives better output of distance than other metaheuristics methods: RAB-NET, HACO, CGAS, ACOTM, HA, DWIO [12]. A comparative study between TS and SA is also reported by [13]. It showed that SA is better than TS for more than 100 cities. Yildirim, A.E. and A. Karci [14], [15], [16] has reported in their comparative study among GA, PSO, ABC, and A^3 (Artificial Atom Algorithm) and shows that A^3 has the best solution for 81 cities in Turkey followed by ABC – GA – PSO. Naro, A and A. Chandra [17], [18] studied GA and TS in solving 57 points of outlets and concludes that GA is better than TS.

In this study, we compare eight optimization methods for solving symmetric, closed traveling salesman problem and almost optimization problems: Metaheuristics by using Genetic Algorithm (GA), Simulated Annealing (SA), Tabu Search (TS), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Artificial Bee Colony (ABC), Improved Fruitfly Optimization Algorithm (EFOA) and Artifiicial Atom Algorithm (A^3).

All of these methods will be tested in 70 cities in Java island and tsplib data based on algorithm's performances: optimality to achieve the shortest distance, then the result of each method will be statistically compared for equality of means by using ANOVA and post-hoc comparison for difference means by applying Tukey test. The reason we take the Java island to be researched is because it is the most populated island and has population more than 50% of total population in Indonesia and has the largest growth in motor vehicles [19]. The growth in motor vehicles will affect the density of public roads and people tend to look for the shortest path that can reduce fuel consumption which can produce greener environment [20].

.

II. Materials and Methods

A. Genetic Algorithm (GA)

Founded by Holland in 1975 and classified as Evolutionary Computation (EC). GA is categorized as population based metaheuristics [21]. Matlab code written by [22]. The main steps are selection, reproduction, evaluation, replacement which crossover and mutation are operator. Crossover should increase the average quality of the population and mutation is needed to explore new states and to avoid local optima. The algorithm for GA is shown in figure 1.

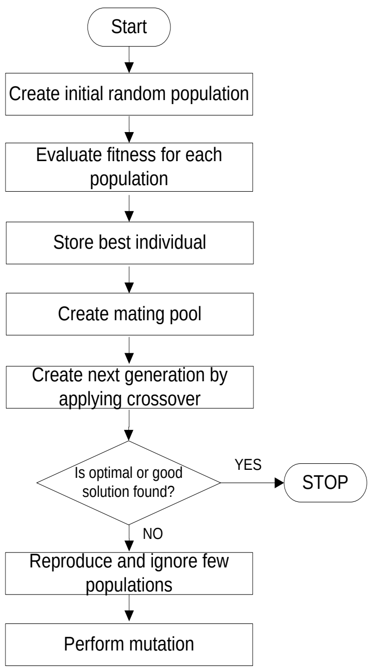

Figure 1. Flowchart of Genetic Algorithm

B. Simulated Annealing (SA)

This algorithm was proposed by Kirkpatrick, Gelatt, and Vecchi in 1983 and is categorized as a single state method metaheuristics. Simulated annealing is an approach that try to avoid entrapment in poor local optima by allowing an occasional uphill move, and this is overcome by random number generator and a control parameter, temperature. SA involves two parameters, cooling ratio, r, 0< r <1 and integer temperature length, L. The algorithms is [23]:

1. Get an initial solution, S
2. Get an initial temperature, T > 0
3. While not yet frozen, do this following:
   a. Pick a random neighbor S' of S
   b. Let Δ = cost (S') – cost (S)
   c. If Δ ≤ 0 (downhill move), then S = S'
   d. If Δ > 0(uphill move), then S = S' with probability $e^{-\Delta/T}$
   e. Set T = rT
4. Return S

In this research, we use cooling rate 0.99 and L is size factor : 70

C. Tabu Search (TS)

Tabu search as a heuristic method was proposed by Glover in 1986 and categorized as a single state method metaheuristics. The basic form of the Tabu search algorithm consists of the following steps [24] :

1. Generating an initial solution
2. Generating a neighboring solution of the current solution by using the move operation between two cities randomly
3. A function that measures each neighboring solution using neighborhood operator
4. A Tabu list in order to prevent cycling and leads the search to unexplored regions of the solution space
5. An aspiration criterion; the tabu move is accepted if it produce better solution than the best obtained
6. Termination criterion; when some number of iterations without improvements exist, then terminate.

D. Ant Colony Optimization (ACO)

ACO was proposed by Dorigo in his dissertation (Ph.D thesis) in 1992 as a natured inspired metaheuristics for the solution of hard combinatorial optimization problem and ACO takes inspiration from foraging behavior of real ants. This algorithm incorporates a mutation operator and tries to include arcs other than the ones with higher pheromone value.
The algorithm of ACO is [25], [26], [27]:

Initialize parameters
For t =1 to number of iterations DO; where t is iteration counter
  For k = 1 to m DO; where m is number of ants
    Repeat until ant k has completed a tour
      Select the city j to be visited next
      With probability p_{ij}
    Calculate the length L_k of the tour generated by ant k
  Update the trail levels τ_{ij} on all edges

In this paper m is equal to number of cities = 70 cities and we are using:
$\alpha = 1;\quad \beta = 1;\quad \rho = 0.05;\quad Q = 1$; where:
α = parameter to regulate the influence of τ_{ij}
β = parameter to regulate the influence of η_{ij}
η_{ij} = visibility of city j from city i
$\rho \in [0,1]$ = parameter to regulate the reduction of τ_{ij}
Q = constant
The matlab code was written by [28].

E. Particle Swarm Optimization (PSO)

PSO is a stochastic optimization method or a biological inspired or natured inspired computational search and optimization method developed by Eberhart and Kennedy in 1995 and categorized as a population methods metaheuristics. It draws the origin of the ecosystem, specifically the social behavior of animals living in swarms, such as schools of fish and grouped flights of birds. Discrete PSO is not as powerful as some specific algorithms, but can easily be modified for any discrete / combinatorial problem. The basic principle is very simple. A set of moving particles (the swarm) is initially thrown inside the search space. Each particle has the following criterions [29], [30], [31]:

- It has a position and a velocity
- It knows its position, and the objective function value for this position
- It knows its neighbors, best previous position and objective position function value
- It remembers its best previous position

Swarm size S equal to N-1 which in this paper N is number of cities = 70 cities and S = 69

The algorithm:

Random initialization of the whole swarm;
Repeat
Evaluate the objective function $f(x_i)$
For all particles i
Update v and move to new position (by using equation 7);
If $f(x_i) < f(pbest_i)$ then $pbest_i = x_i$;
If $f(x_i) < f(gbest_i)$ then $gbest_i = x_i$;
Then update (x_i, v_i);
Endfor
Until Stopping criteria

$$\begin{cases} v_{t+1} = c_1 v_1 + c_2 (p_{ig,t} - x_t) \\ x_{t+1} = x_t + v_{t+1} \end{cases} \quad (1)$$

Where

- v_t = velocity at time step t
- x_t = position at time step t
- $p_{i,t}$ = best previous position at time step t
- $p_{g,t}$ = best neighbour's previous best at time step t (or best neighbour)
- c = confidence coefficient

time step is also recognized as iteration or step

F. Artificial Bee Colony Algorithm (ABC)

This metaheuristics optimization method – population based was developed in 2005 by Dervis Karaboga. This method was inspired by honey bee colonies and based on observing the nourishment behavior of honey bee. There are two types of artificial bees: employed and onlooker bees. ABC algorithm is simple, fast, and easy.

This equation below represents an initial solution which random route between 1 and 70 generated by this equation. At the same time, a scout bee searches the new foods when a source is abandoned.

$$x_{ij} = x_j^{\min} + rand(0,1)(x_j^{\max} - x_j^{\min}) \quad (2)$$

The following equation represents the employed bee phase that performs random modifications on adjacent to a solution.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad (3)$$

Then, the last equation, onlooker bee choose a food source with probability proportional to the quality of food source

$$fitness_i = \begin{cases} \dfrac{1}{1+f_i} & if \quad f_i \geq 0 \\ 1 + abs(f_i) & if \quad f_i \geq 0 \end{cases} \quad (4)$$

In this study, onlooker and employed bees is 15, number of cities is 70, and the manipulating operators are swap, insertion, and reversion

A short algorithm is shown below [32]:

Initialize population
Repeat
Place the employed bees on their food sources and determine their nectar amounts
Calculate the probability value of the sources with which they are preferred by the onlooker bees
Place the onlooker bees on the food sources depending on their nectar amounts
Stop the exploitation process of the sources exhausted by the bees
Send the scouts to the search area for discovering new food sources randomly
Memorize the best food source found so far
Until requirements are met

Table 1. Parameter and selected value

Parameters	Selected Value
Food number / food source positions	70
Employed bee number	15
Onlooker bee number	15
Limit	100
Iteration number	10,000

There are three manipulating operators that used: swap, insertion and reversion for exchanging between two positions.

G. An Improved Fruit Fly Optimization Algorithm (EFOA)

In 2011, Fruit fly algorithm is introduced by Wen Tsao Pan and originated from foraging behavior of fruit flies which have a keen vision and smell to find food quickly by following the odor concentration in the air. This algorithm has a simple structure and easily understood [33]. Because its simple structure then it can easily fall into the local optimum and produces low optimization precision. Then Huang, L., et al in 2016 proposed improved fruit fly algorithm that eliminates some individuals – weak fruit flies and some new individuals are generated in fruit fly foraging process, and this proposed algorithm is called EFOA. There are two operators, a reverse operator and a multiplication operator. The reverse operator is an evolution of the 2-opt algorithm and node is selected randomly. The multiplication operator will generate at least one of the

following three subsequences through the reverse operator: (a_k, a_j), (a_k, a_m),and (a_j, a_k, a_m).
The proposed algorithm is following:

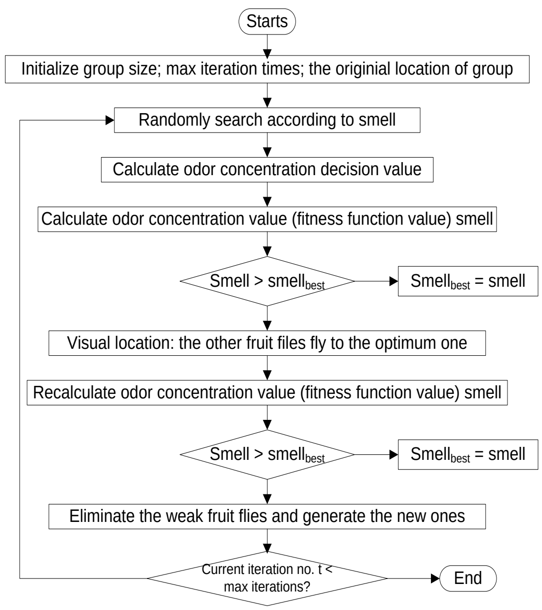

Figure 2. EFOA algorithm

H. *Artificial Atom Algorithm (AAA – A^3)*

A^3 is a new nature inspired metaheuristics optimization method and developed by Yildirim, A.E. in 2018. in his Ph.D thesis. A^3 is inspired by chemical compounding processes and developed by modeling chemical ionic bond and covalent bond processes. The most important feature of A^3 is that A^3 examine the effect of parameter values on the result separately. The algorithm step is following:

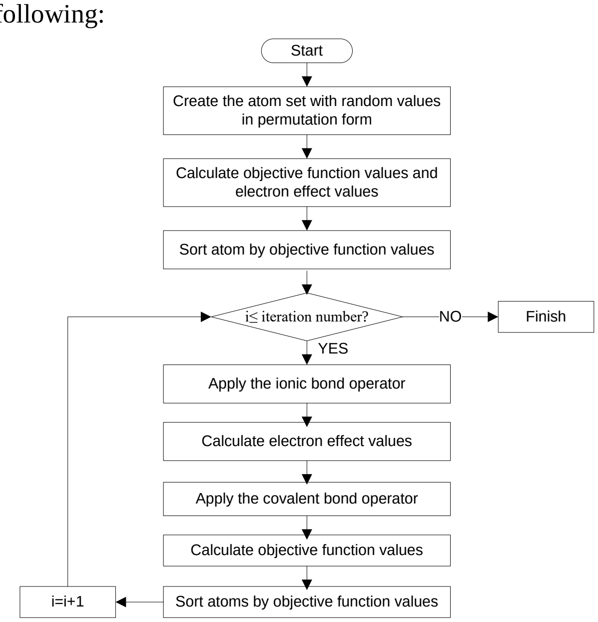

Figure 3. A^3 algorithm steps

In this study, the number of electrons is 70, the number of atoms is 100, covalent of region = 0.5, ionic region = 0.5, and the number of iterations is 10,000.

I. *Research Methodology*

The step by step research is shown in the following figure:

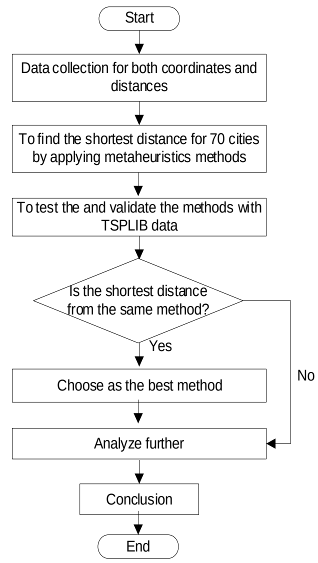

Figure 4. Research Methodology

III. Results and Discussion

To solve the TSP and to get the shortest distance for 70 big cities in Java island, we use MATLAB 2015a, Intel Core i5 7200 U CPU 2.5 GHz, 32 bit ACPI x64 based PC which inputs are: number of cities, city coordinates or distance matrix using Euclidean distance method, number of iterations and output is total distance in kilometer unit.

Table 2. The inputs – No. City and their coordinates:

No	City	Lat. (S)	Long. (E)	X (peta)	Y (peta)
1	Bandung	-6.92	107.62	145	179
2	Banjarnegara	-7.44	109.54	87	394
3	Banten	-6.40	106.02	202	2
4	Banyuwangi	-8.22	114.37	0	932
5	Bekasi	-6.24	106.97	221	108
6	Blitar	-8.10	112.16	14	686
7	Blora	-6.97	111.41	139	602
8	Bogor	-6.59	106.804	181	90
9	Bondowoso	-7.91	113.80	34	868
10	Brebes	-6.86	109.04	152	338
11	Ciamis	-7.32	108.344	100	261
12	Cianjur	-6.80	107.15	158	128
13	Cikampek	-6.39	107.44	203	160
14	Cilacap	-7.69	109.03	59	338
15	Cilegon	-6.02	106.05	245	6
16	Cimahi	-6.88	107.54	149	172
17	Cirebon	-6.73	108.55	166	284
18	Cisarua	-6.68	106.93	171	104

19	Depok	-6.40	106.79	202	88
20	Garut	-7.22	107.918	111	214
21	Gresik	-7.01	112.56	135	730
22	Indramayu	-6.35	108.33	208	259
23	Jakarta	-6.21	106.84	224	94
24	Jember	-8.17	113.68	6	856
25	Jepara	-6.57	110.66	183	519
26	Karawang	-6.29	107.32	214	147
27	Kebumen	-7.67	109.65	61	406
28	Kediri	-7.85	112.02	41	670
29	Kendal	-6.93	110.20	144	467
30	Kudus	-6.81	110.84	157	538
31	Lasem	-6.70	111.44	169	606
32	Lumajang	-8.12	113.23	11	805
33	Madiun	-7.63	111.53	65	615
34	Magelang	-7.48	110.22	82	469
35	Majalengka	-6.83	108.24	154	249
36	Malang	-7.97	112.63	28	738
37	Merak	-5.93	105.99	255	0
38	Ngawi	-7.40	111.41	91	603
39	Pacitan	-8.18	111.10	4	568
40	Pamanukan	-6.29	107.82	215	202
41	Pandeglang	-6.32	106.11	212	12
42	Pasuruan	-7.65	112.89	64	768
43	Pati	-6.76	111.04	163	561
44	Pekalongan	-6.89	109.67	148	409
45	P. Ratu	-6.97	106.56	138	62
46	Ponorogo	-7.87	111.47	39	609
47	Probolinggo	-7.78	113.20	49	801
48	Purwakarta	-6.54	107.45	187	161
49	Purwokerto	-7.42	109.24	89	360
50	Rangkasbitung	-6.37	106.24	206	27
51	Rembang	-6.71	111.34	168	595
52	Salatiga	-7.33	110.51	99	502
53	Semarang	-7.00	110.44	135	495
54	Serang	-6.12	106.15	234	17
55	Sidoarjo	-7.45	112.69	86	745
56	Situbondo	-7.71	113.98	57	888
57	Solo	-7.57	110.82	72	537
58	Sragen	-7.42	111.03	89	560
59	Subang	-6.56	107.74	184	194
60	Sukabumi	-6.93	106.93	144	103
61	Sumedang	-6.85	107.99	152	213
62	Surabaya	-7.25	112.75	108	752
63	Tasikmalaya	-7.35	108.21	97	246
64	Tegal	-6.88	109.12	149	348
65	Tuban	-6.89	112.04	147	672
66	Ungaran	-7.14	110.41	120	491
67	Wonogiri	-7.80	110.93	46	549
68	Wonosari	-7.97	110.60	28	512
69	Wonosobo	-7.37	109.88	94	432
70	Yogyakarta	-7.79	110.37	47	486

Table 3. Methods, Distance, Time of Metaheuristics Methods

No	Metaheuristics Method & Year of Founded	Distance		Run
		Best	Avg	times
1	**GA - 1975**	2,727	2,765	20
2	SA – 1983	2,683	3,041	20
3	TS – 1986	3,190	3,190	20
4	ACO - 1992	2,851	2,891	20
5	PSO - 1995	9,623	10,933	20
6	**ABC - 2005**	**2,447**	**2,458**	20
7	EFOA - 2016	2,727	2,731	20
8	A^3 - 2018	2,727	2,728	20

Note: best value in distance means the shortest path; best value in time means the fastest process

From the above table, we see that the shortest distance resulted from algorithm of ABC is 2,447 Kilometers. The sequence from the shortest to the longest distance is: ABC– SA – [A^3 – EFOA – GA] – ACO – TS – PSO and the sequence from the fastest to the slowest is: GA – SA – PSO – TS - EFOA - ABC – ACO – A^3. The methods in the bracket have the same distance which is 2,727.

Table 4. Distance comparison

No	Metaheuristics Method & Year of Founded	Distance	
		70 cities	Tsplib58
1	**GA - 1975**	2,727	25,902
2	SA – 1983	2,683	28,014
3	TS – 1986	3,190	29,178
4	ACO - 1992	2,851	29,594
5	PSO - 1995	9,623	99,887
6	**ABC - 2005**	**2,447**	**25,400**
7	EFOA - 2016	2,727	28,307
8	A^3 - 2018	2,727	28,306

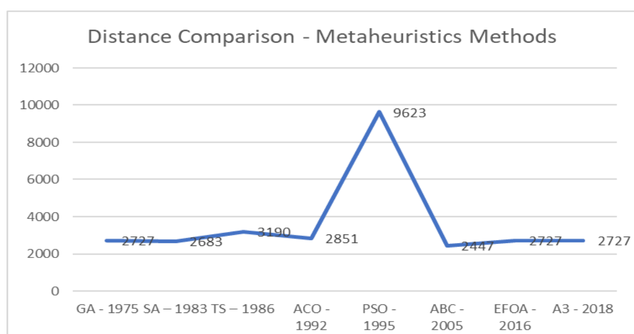

Figure 5. Distance comparison by graph

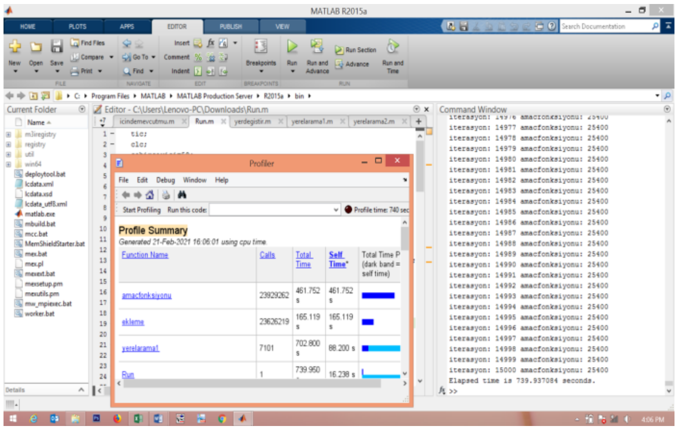

Figure 6. Result of ABC method for tsplib58

Now, we will test this result by statistical methods: ANOVA and Tukey test.

	sum of squares	degrees of freedom	mean square	Fs	p	variance component (%)
among groups	1155992333.894	7	165141761.985	2519.057	1.03E-153	99.21
within groups	9964660.35	152	65556.976			0.79
total	1165956994.244	159				

	GA	SA	TS	ACO	PSO	ABC	A3	EFOA
mean	2765.	3040.5	3190.	2890.8	10932.55	2458.3	2727.6	2730.7
Gabriel comparison interval	128.297	128.297	128.297	128.297	128.297	128.297	128.297	128.297
n	20	20	20	20	20	20	20	20

Figure 7. ANOVA Test

ANOVA test indicates that the data of distances are not the same for every methods of metaheuristics.

Table 5. Tukey – Kramer test

		Tukey-Kramer minimum significant difference							
		GA	SA	TS	ACO	PSO	ABC	A3	EFOA
	GA	-	248.865	248.865	248.865	248.865	248.865	248.865	248.865
actual difference	SA	275.5*	-	248.865	248.865	248.865	248.865	248.865	248.865
('*' if significant)	TS	425*	149.5	-	248.865	248.865	248.865	248.865	248.865
	ACO	125.8	149.7	299.2*	-	248.865	248.865	248.865	248.865
	PSO	8168*	7892*	7743*	8042*	-	248.865	248.865	248.865
	ABC	306.7*	582.2*	731.7*	432.5*	8474*	-	248.865	248.865
	A3	37.4	312.9*	462.4*	163.2	8205*	269.3*	-	248.865
	EFOA	34.3	309.8*	459.3*	160.1	8202*	272.4*	3.1	-

From table 3, the cell contents in the lower left table show 28 pairs which 8 pairs are not different significantly – no label asterisk and 20 pairs are different significantly. The pairs which are not different significantly are: GA – ACO; GA – A3; GA – EFOA; SA – TS; SA – ACO; ACO – A3; ACO – EFOA; A3 – EFOA.

IV. Conclusions

Eight of optimization methods – metaheuristics algorithms have been compared and tested for finding the shortest path in traveling salesman problem for 70 big cities in Java island, and the result shows that ABC optimization method has the shortest distance : 2,447 and the best average value: 2,458 in distance. The ANOVA and Tukey test show that there are 28 pairs in which 20 pairs are different significantly .

References

[1] Olkhova, M., Roslavtsev, D., Matviichuk, O., Mykhalenko, A. "City Delivery Routes Planning Based on The Ant Colony Algorithm. *Science and Technology*," Vol.19 no.4., 2020. https://doi.org/10.21122/2227-1031-2020-19-4-356-362.

[2] Chandra, A., Natalia, C., Naro, A. "Comparative Solutions of Exact and Approximate Methods for Traveling Salesman Problem. *Lampsakos no.25,e-3804, 2020.* DOI: 10.21501/21454086.3804

[3] Saiyed, A.R. "*The Traveling Salesman Problem*." Indiana State University, 2012.

[4] Stidsen, T.J.K. Introduction to Optimization using Metaheuristics, 2011. http://www.imm.dtu.dk/courses/02719.

[5] Talbi, E.G. "*Metaheuristics: From Design to Implementation*". New Jersey: John Wiley & Sons., 2009.

[6] Koopialipoor, M., Noorbakhsh, A. "Application of Artificial Intelligence Techniques in Optimizing Drilling, Emerging Trends in Mechatronics," *Intech Open*, 2020. DOI:105772/intechopen.85398

[7] Oliveira, J.F., and M.A. Carravilla. "*Heuristics and Local Search*". FEUP, 2009.

[8] Hussain, K. "Artificial Intelligence Review: Metaheuristic Research: A Comprehensive Survey," Universiti Tun Hussein On. Malaysia, 2018.

[9] Saud S., Kodaz H., Babaoglu I. "Solving Traveling Salesman Problem by Using Optimization Algorithm," *The 9th International Conference on Advances in Information Technology: IAIT Conference Proceedings*, 2018.

[10] Onder, E., M. Ozdemir, and B.F. Yildrim. "Combinatorial Optimization Using Artificial Bee Colony Algorithm and Particle Swarm Optimization Supported Genetic Algorithm," *KAU IIBF Dergisi* 4(6): 59 – 70, 2013.

[11] Gharib, A., Benhra, J., and Chaouqi, M. "A Performance Comparison of PSO and GA Applied to TSP," *International Journal of Computer Applications.* Vol.130 no.15, 2015.

[12] Huang, L., Wang, G.C., Bai, T., and Wang, Z. "An Improved Fruit Fly Optimization Algorithm for Solving the Traveling Salesman Problem," *Frontiers of Information and Electronic Engineering*, 2016.

[13] Jayaswal, S. "A Comparative Study of Tabu Search and Simulated Annealing for Traveling Salesman Problem". Project Report Applied Optimization MSCI 703. Department of Management Sciences. University of Waterloo, 2018.

[14] Yildirim, A.E., and Karci, A. "Application of Traveling Salesman Problem for 81 Provinces in Turkey Using Artificial Atom Algorithm," *7th International Conference on Advanced Technologies (ICAT 2018).* April 28 – May 1, 2018: 722 – 726. Antalya, Turkey, 2018.

[15] Yildirim, A.E., and Karci, A. "Solutions of Traveling Salesman Problem Using Genetic Algorithm and Atom Algorithm," *In proceedings of 2nd International Eurasian Conference on Mathematical Sciences and Applications (IECMSA-2013).* Sarajevo, Bosnia and Herzegovina, 2013.

[16] Yildirim, A.E., and Karci, A. "Applications of Artificial Atom Algorithm to Small scale traveling salesman problems," *Journal of Soft Computing 22 (22), p.7619-7631*, 2017. DOI:10.1007/s00500-017-2735-z.

[17] Naro, A., and Chandra, A. "Optimasi Jalur Distribusi Dengan Genetic Algorithm," Research Report, Universitas Mercu Buana., 2019.

[18] Naro, A. and Chandra, A. "Optimasi Jalur Distribusi Dengan Tabu Search," Research Report, Universitas Mercu Buana, 2019.

[19] Biro Pusat Statistik Republik Indonesia. "Land Transportation Statistics 2017," Katalog: 8302004. BPSRI, 2018.

[20] Dukic, G., V. Cesnik, T. Opetuk. "Order Picking Methods and Technologies for Greener Warehousing," *Strojarstvo 52 (1)*, p.23 – 31, 2010.

[21] Luke, S. (2015). *Essential of Metaheuristics*, George Mason University. On line version 2.2.

[22] Kirk, J. (2014). *Traveling Salesman Problem: Genetic Algorithm*, Matlab Central File Exchange. Retrieved: January 29, 2019. http://www/mathworks.com/matlabcentral

[23] Johnson, D.S., Aragon, C.R., McGeoch, L.A., and Schevon, C. "Optimization by Simulated Annealing: An Experimental Evaluation Part I, Graph Partitioning," *Operation Research.* Vol.37 no.6, November – December, 1989.

[24] Alkallak, I.N., and Sha'ban, R.Z. "Tabu Search Method for Solving the Traveling Salesman Problem," *Raf. Journal of Comp. & Math.* Vol.5 no.2, 2015.

[25] Dorigo, M., and Stutzle, T. "*Ant Colony Optimization,"* Cambridge: A Bradford Book. The MIT Press, 2004.

[26] Bullnheimer, B., Hartl, R.F., and Strauβ, C. "A New Rank Based Version of The Ant System." Paper presented at Vienna University of Economics and Business Administration. August 2-6, 1997.

[27] Heris, S.M.K. "Ant Colony Optimization for TSP*".* Project code: YOEA103. Yarpiz, 2015.

[28] Monteiro, M.S.R., Fontes, D.B.M.M and Fontes, F.A.C.C. "ACO: A Literature Survey,". University of Porto, 2012.

[29] Clerc, M. (2000). Discrete PSO: Illustrated by the TSP. Accessed April 30, http://www.mauriceclerc.net

[30] Clerc, M. (2012). Standard PSO From 2006 to 2011. Accessed April 30, http://www.mauriceclerc.net

[31] Hanka, M.K.R. "Particle Swarm Optimization for TSP." Paper presented for Metaheuristics course. Institut Teknologi Sepuluh November. April 14, 2012.

[32] Boussaid, I., Lepagnot, J. and Siarry, P. "A Survey On Optimization Metaheuristics.," *Information Sciences* 237: 82-117, 2013.

[33] Iscan, H., and Gunduz, M. "Parameter Analysis on Fruit Fly Optimization Algorithm," *Journal of Computer and Communications,* 2: 137 – 141, 2014.