Emperor Journal of Civil Engineering

Received : November 2022
Accepted : November 2022
Published : December 2022

www.ejceng.mayas.info

Volume - I **Issue – 12** **December - 2022**

An Analysis of Different Metaheuristic Approaches for Solving Travelling Salesman Problems

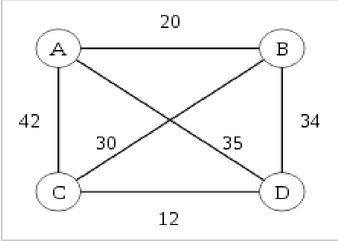
Jaspreet Kaur[1], Ashok Pal[2]
Department of Mathematics
Chandigarh University, Gharuan

ABSTRACT
Travelling Salesman Problem is widely researched optimization problem in computational mathematics as it was originated 6 decades ago. It gives the shortest possible route for the person who is visiting number of cities for a particular purpose so that cost or distance can be minimized. The distances between every two cities will be given. The person has to start from particular city (Home City) and by visiting each and every city exactly once he has to return to his home city. For this purpose a bunch of methods have been proposed since the problem came into existence. Basically there are two main approaches proposed by different authors; Exact Algorithms and Metaheuristic Algorithms. In this article, we will put the light on every metaheuristic approach from different aspects for solving Travelling Salesman Problem (TSM).

Keyword: Travelling Salesman Problem, Exact and Approximate Algorithms, Metaheuristic Approaches.

I. INTRODUCTION

Travelling salesman problem is an issue in optimization in which the person visiting the cities need to find the minimum track to visit n number of cities, who starts from a home city and visit all the cities included in his track only once and arrives to the home city. TSP is one of the most useful techniques of optimization as it gives way out to many industrial optimization problems. These types of problems include order picking in warehouse, workshop scheduling, computer wiring, [3] vehicle routing, drilling of circuit board problems, [7] delivery service like pizza delivery boy, meter reader, postman etc. want to find the minimum track to complete his daily task.



Symmetric TSP with four cities

Fig. 1 Symmetric TSP

TSP was first initiated by Irish mathematician W. R. Hamilton. However Travelling Salesman Problem (TSP) came into existence in 1930 by mathematicians Karl Menger who defined the problem mathematically and Merril M Flood who wanted to find the best route for school bus. Later on name TSP was suggested by Hassler Whitney. In 1950 and 1960 cutting plane method was developed for its solution and applied it on 49 cities. In the same decades branch and bound methods came into existence. In 1964 first heuristics algorithm came into existence [1][3][4][5] and applied to 57 cities. Furthermore many researchers have developed different algorithms and used various approaches for the better solution and efficiency. Some of the metaheuristic approaches have been described in this paper and their comparisons have been showed.

II. MATHEMATICAL FORMULATION

While formulating TSP mathematically in the form of integer programming problem. It can be labeled as the undirected graph G= $< V, E >$ which is complete with set of vertices V vertices having cardinality n i.e. $|\mathrm{V}| = n$ here in TSP n is number of cities, and E contains c_{ij} , the cost of visiting from city I to j.

Now make a notation x_{ij} to denote whether the city i is related to city j or not, by 1 or 0 as follows

$$x_{ij} = \begin{cases} 1 & city\ i\ is\ directly\ connected\ to\ city\ j \\ 0 & city\ i\ is\ notdirectly\ connected\ to\ city\ j \end{cases}$$

Now c_{ij} will denote the distance from city i to j

Then TSP can be written in the following ILPP Form

$$min \sum_{i=1}^{n} \sum_{j \neq i, j=1}^{n} c_{ij} x_{ij}$$

$$0 \leq x_{ij} \leq 1 \qquad i, j = 1, \dots \dots \dots n;$$

$$\sum_{i=1, i\neq j}^{n} x_{ij} = 1 \qquad j = 1,2, \dots \dots \dots \dots . . n;$$

$$\sum_{j=1, j\neq i}^{n} x_{ij} = 1 \qquad i = 1,2, \dots \dots \dots \dots . . n;$$

$$\sum_{i\in Q}\sum_{j\in Q} x_{ij} \leq |Q| - 1 \quad \forall \quad Q \subsetneq \{1,2, \dots \dots . . n\} \quad |Q| \geq 2$$

$$x_{ij} = 0 \; or \; 1$$

Exact Algorithms

Exact Algorithms give exact answers very quickly. But these type of algorithms are applicable only for problems in which number of objects are small. These are not applicable on large samples. For large problem size only metaheuristic approaches can be used which will give approximate solution to the problem.

Metaheuristic Algorithms

These methodologies don't generally locate the genuine ideal arrangement. Rather, [1] they will regularly reliably discover great answers for the issue. These great arrangements are normally viewed as sufficient essentially in light of the fact that they are as well as can be expected be found in a sensible measure of time. Hence, [4] improvement regularly plays the job of finding the most ideal arrangement in a sensible measure of time. Some Metaheuristic approaches for the solution of TSP are briefly discussed as follows;

1 . Ant Colony Optimization

Ant Colony is one of the nature inspired algorithms. It is inspired by ant's behavior in ant colony. Marco Dorigo originated ACO in 1992, furthermore many amendments has been done to get more accurate results. Ant Colony Optimization are applied for the well known Benchmark problems as discussed in [8],[9],[10],[11] showed their results in different scenarios for various parameters like no. of Iteration, Global Best, Average , Time, have seen the behavior of ants while searching of food. They accumulate a chemical substance known as pheromone to communicate other ants about the path of food source from their anthill. When other ants see the trail of pheromone spread by the others they follow their path and also leave the pheromone. In a while we see that the ants following their random path are attracted towards that path and a large number of ants are gathered at that path obtained by their leaders. Simply leaving pheromone on a way isn't adequate to get the perfect course of action. **[8]** Evelia Lizárraga, Oscar Castillo, and José Soria partitioned total ants in 3 subsets such that every ant of the set was examined separately for respective change in ACO. They were used Berlin 52 and bier 127. They showed their method is 64 times faster than conventional methods. Obviously, the "ants" can't simply follow comparative ways unavoidably. Various factors ought to be worked in. The "underground creepy crawly" can't simply take the path with the most pheromone; it must get a chance to self-assertively pick another approach to find a prevalent response for the issue. This is done by applying fundamental probability number juggling to the decision that the "underground creepy crawly" will make. The more pheromone, the practically certain the underground creepy crawly is to pick the way, anyway there is up 'til now an open door that it will take a substitute way. **[9]** Raghavendra BV* applied ACO on symmetric travelling salesman problem for five cities and chosen number of iterations 5 and evidenced that rich pheromone converge to best path. Besides, the "pheromone" can have a "disappearing rate" which would similarly add to the probability that a "creepy crawly" would find a more cutting-edge, better way. The central target is to change the direct of individual ants to convey a perfect response in the settlement lead. This is done with the use of a couple of parameters. These parameters were progressed using a Genetic Algorithm (GA).

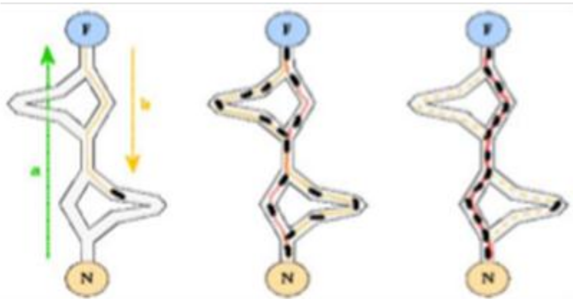


Fig. 2 Ant using path of pheromones

Datasets	ACOhh	ACS	GA	EP	TS	ATS	SA	AG	Best known
Oliver30	427.90	423.74	N/A	423.74	N/A	N/A	424	420	424
Eil51	429.70	427.96	441.46	427.86	445.05	438.12	443	436	426
Eil76	568.89	542.31	548.26	549.18	582.62	540.17	580	561	538
KroA100	21705.70	21285.44	22875.42	N/A	23664.18	21526.56	N/A	N/A	21282

Fig. 3 ACO hyper Heuristics compared to other methods for the best route

2. Genetic Algorithms

A Genetic Approach for solving optimization problems depends on a similar concept as the hypothesis of development. Genetic Algorithm was first originated by John Holland in 1960. In **[14]** Author used GA in Matlab to measure proposed work of cross operators with some traditional operators of measuring path.Firstly it was applied on 7 cities and then applied proposed CX2 on 12 benchmark problems along with traditional crossover operators PMX and OX on some symmetric and some asymmetric TSPs. The parameters used were size of population, mutational probabilities, crossovers and observing that proposed operator perform much better than existing ones and give faster convergence. **[15]** proposed the method for reducing the time for calculations for GA and applied it on benchmarks of sizes 130 to 13thousand cities and showed less time consumed by their proposed work. The advancement work creates a series of data sources (frequently a series of bits that are encodings of the info parameters) at that point approaches the wellness work for a wellness esteem for that string. At the point when a few strings have been doled out a wellness esteem, the advancement work takes the best strings, combines them, some of the time tosses in a "transformation" to the strings and afterward sends the outcomes back as new info strings. The organic relationship is to a people qualities. Actually, an info string is frequently called a chromosome and the bits in the string are alluded to as qualities. The wellness capacity of a hereditary calculation takes in a series of sources of info and runs them through the procedure that is being assessed. In view of the presentation of the information sources, the capacity restores a wellness esteem. On account of the TSP, the wellness work restored the all out length or weight of the way found. In **[13]** author applied atom algorithm and showed better results than GA. The following steps are involved in the standard Genentic Algorithm.

Step 1. Obtaining the starting population

Step 2. Ascertaining wellness.

Step 3.Choosing the best qualities.

Step 4. Traverse.

Step 5. Transforming to present varieties.

Crossover	Optimum	Average value	Best value	Worst value
PMX	17/30	159.7	159	165
OX	14/30	160.3	159	163
CX2	24/30	159.2	159	162

Fig.3 Crossover operators comparison in [14] (30 runs)

3. Simulated Annealing

Simulated Annealing is a nonexclusive probabilistic meta-calculation used to calculate a relative answer (which is not exact) for worldwide problems in optimization. SA was first developed by Kirkpatrick, Gelatt Jr.,Vecchi for solution of Travelling salesman problem in 1983. It came in existence by tempering in metallurgy which is a system of controlled cooling of material to lessen absconds. The Simulated Annealing calculation begins with an arbitrary arrangement. An arbitrary close by arrangement is framed by each cycle. In **[17]** Xiutang Zeng et. al. used Adaptive simulated annealing algorithm with greedy search. They applied ASA-GS on 60 benchmark TSPs using the parameters percentage error and CPU time in seconds and compared it with four developed methods and concluded their algorithm achieves the reasonable time of computation and solution quality. Also CPU time was longer than others therefore further improvement in the method is needed for future work. In the event that this arrangement is a superior arrangement, it will supplant the present arrangement. On the off chance that it is a more terrible arrangement, it might be picked to supplant the present arrangement with a likelihood that relies upon the temperature parameter. As the calculation advances, the temperature parameter diminishes, giving more terrible arrangements a lesser possibility of supplanting the present arrangement. In **[16]** Ai-Hua Zhou, Li-Peng Zhu, Bin Hu, Song Deng , Yan Song, Hongbin Qiu, Sen Pan performed two experiments on 6 benchmark TSPs and with 3 more heuristic algorithms obtained that their method has best quality of solution, error was low, convergence speed was high. We permit most noticeably terrible arrangements toward the starting to stay away from arrangement combining to a neighborhood least instead of the worldwide least. We will utilize the mimicked toughening calculation to take care of Random Traveling Salesman Problem. The sales rep must visit every city just a single time and come back to a similar city wherein he started. The limitations that influence the result of the calculation are the underlying temperature, the rate at which the temperature diminishes and the halting state of the calculation. By modifying these qualities, we run the SA and to see that how calculation reacts.

Instances	Optimal	Algorithm	Best	Worst	Average	Generations	Time (s)
fri26	937	TSP-SAGEP	937	971	941	150	7.0671
		GEP	943	1132	961	800	8.1629
st70	675	TSP-SAGEP	675	699	677	150	12.1135
		GEP	698	822	710	800	14.0931
gr120	6942	TSP-SAGEP	6942	7406	6995	200	19.4612
		GEP	7147	10,035	7624	900	22.3573
pcb442	50,778	TSP-SAGEP	50,811	52,147	50,878	300	51.0964
		GEP	51,092	54,113	51,676	900	57.1428
gr666	294,358	TSP-SAGEP	294,419	305,036	295,542	990	79.7853
		GEP	310,982	410,368	328,459	990	85.1204
rl1304	252,948	TSP-SAGEP	253,104	271,582	254,699	990	160.2458
		GEP	281,296	334,871	290,611	990	165.5572

Fig.4 TSP-SAGEP Compared with traditional GEP Algorithm with four parameters

4. **Particle Swarm Optimization**

PSO is a populace based improvement approach was originated by James Kennedy in social modeling in 1995 (Ref PSO origin) It was initially proposed by Eberhart and Colleagues. In a PSO framework, particles change their situations by flying around in a multidimensional inquiry space until computational impediments are surpassed. A different approach was designed for effective PSO **[18],** applied their results on 5 benchmark problems using three parameters average, time etc. and was extended for various discrete type optimization problems. Author also proposed the method of velocity operator. An uncertain strategy is proposed **[19]** to present discrete particle swarm optimization. Numerical results are shown for algorithm applied on benchmark problems. Hyper Heuristic ACO algorithm was proposed **[2]**, combined low level heuristics, used local and global updates for best rules of pheromones updating. They compared their results with six different existing methods for standard problems. The chief advantage of this method is that, it used heuristics of low level which are easy to implement at every problem.

Instance	Zhong et al. (2007)			PSO-LK		
	Min	Average	T(s)	Min	Average	T(s)
eil51	0.002	1.793	4.06	0	0	0
berlin52	0	0.753	4.12	0	0	0
eil76	0.004	2.550	11.59	0	0	0.01
kroA100	0.001	1.914	23.95	0	0	0.02
kroA200	0.007	3.427	198.55	0	0	0.08

Fig 5 . Comparison between Algorithm of Zhong et al and PSO LK [18]

The PSO procedure is a developmental calculation strategy, yet it contrasts from other notable transformative calculation calculations, for example, the hereditary calculations. Albeit a populace is utilized for looking through the inquiry space, there are no administrators propelled by the human DNA methods applied on the populace. Rather, in PSO, the populace elements reenacts a „bird flock‟s‟ conduct, where social sharing of data happens and people can benefit from the revelations and past experience of the various colleagues during the quest for nourishment. Hence, each friend, called molecule, in the populace, which is called swarm, is expected to „fly‟ over the pursuit space so as to discover promising locales of the scene. For instance, in the minimization case, such areas have lower work esteems than other, visited beforehand. Right now, molecule is treated as a point in a d-dimensional space, which modifies its own „flying‟ as per its flying experience just as the flying experience of different particles (allies). In PSO, a molecule is characterized as a moving point in hyperspace. For every molecule, at the ebb and flow time step, a record is kept of the position, speed, and the best position found in the pursuit space up until this point. The supposition that is an essential idea of PSO. In the PSO calculation, rather than utilizing developmental administrators, for example, transformation and hybrid, to control calculations, for a variable advancement issue, a herd of particles are placed into the d-dimensional hunt space with arbitrarily picked speeds and positions knowing their best qualities up until this point (P best) and the situation in the d-dimensional space.

Problem	*Opt*	Best result	Worst result	*Err* (%)
EIL51	426	427	452	2.5751
BERLIN52	7542	7542	8362	3.8458
ST70	675	675	742	3.3422
EIL76	538	546	579	4.1673
PR76	108159	108280	124365	3.8176

Fig 6. Results obtained by algorithm proposed in [19]

III. CONCLUSION

In this work we observed different metaheuristic approaches to solve Travelling Salesman Problem. Various authors have given their views about different methods. They applied hybrid approaches to benchmark problems to get the better results, compared their results with many existing methods proposed by tremendous authors in the particular field and got outstanding outcomes for their work. As we compared only the methods proposed by others in the present work, hope for our own hybrid approach in the upcoming article and future work.

IV.REFERENCES

1. Suwannarongsri, S., & Puangdownreong, D. (2012, February). Solving traveling salesman problems via artificial intelligent search techniques. In *Proceedings of the 11th WSEAS international conference on Artificial Intelligence, Knowledge Engineering and Data Bases* (pp. 137-141). World Scientific and Engineering Academy and Society (WSEAS).
2. Aziz, Z. A. (2015). Ant colony hyper-heuristics for travelling salesman problem. *Procedia Computer Science*, *76*, 534-538.
3. Matai, R., Singh, S. P., & Mittal, M. L. (2010). Traveling salesman problem: an overview of applications, formulations, and solution approaches. *Traveling salesman problem, theory and applications*, *1*.

4. Abid, M. M., & Muhammad, I. (2015). Heuristic approaches to solve traveling salesman problem. *Indonesian Journal of Electrical Engineering and Computer Science*, *15*(2), 390-396.
5. Gendreau, M., Laporte, G., & Semet, F. (1998). A tabu search heuristic for the undirected selective travelling salesman problem. *European Journal of Operational Research*, *106*(2-3), 539-545.
6. Brucal, S. G. E., & Dadios, E. P. (2017, December). Comparative analysis of solving traveling salesman problem using artificial intelligence algorithms. In *2017IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)* (pp. 1-6). IEEE.
7. Hernández-Pérez, H., Salazar-González, J. J., & Santos-Hernández, B. (2018). Heuristic algorithm for the split-demand one-commodity pickup-and-delivery travelling salesman problem. *Computers & Operations Research*, *97*, 1-17.
8. Lizárraga, E., Castillo, O., & Soria, J. (2013). A method to solve the traveling salesman problem using ant colony optimization variants with ant set partitioning. In *Recent Advances on Hybrid Intelligent Systems* (pp. 237-246). Springer, Berlin, Heidelberg.
9. Raghavendra, B. V. (2015). Solving Traveling Salesmen Problem using Ant Colony Optimization Algorithm. *J Apple Computat Math*, *4*, 260.
10. Huan, H. X., Linh-Trung, N., & Huynh, H. T. (2013). Solving the Traveling Salesman Problem with Ant Colony Optimization: A Revisit and New Efficient Algorithms. *REV Journal on Electronics and Communications*, *2*(3-4).
11. Brezina Jr, I., & Čičková, Z. (2011). Solving the travelling salesman problem using the ant colony optimization. *Management Information Systems*, *6*(4), 10-14.
12. Yang, J., Shi, X., Marchese, M., & Liang, Y. (2008). An ant colony optimization method for generalized TSP problem. *Progress in Natural Science*, *18*(11), 1417-1422.
13. Yildirim, A. E., & Karci, A. (2013, August). Solutions of travelling salesman problem using genetic algorithm and atom algorithm. In *proceedings of 2nd international eurasian conference on mathematical sciences and applications, Sarajevo, Bosnia and Herzegovina* (p. 134).
14. Hussain, A., Muhammad, Y. S., Nauman Sajid, M., Hussain, I., Mohamd Shoukry, A., & Gani, S. (2017). Genetic algorithm for traveling salesman problem with modified cycle crossover operator. *Computational intelligence and neuroscience*, *2017*.
15. Tsai, C. W., Tseng, S. P., Chiang, M. C., Yang, C. S., & Hong, T. P. (2014). A high-performance genetic algorithm: using traveling salesman problem as a case. *The Scientific World Journal*, *2014*.
16. Zhou, A. H., Zhu, L. P., Hu, B., Deng, S., Song, Y., Qiu, H., & Pan, S. (2019). Traveling-salesman-problem algorithm based on simulated annealing and gene-expression programming. *Information*, *10*(1), 7.
17. Geng, X., Chen, Z., Yang, W., Shi, D., & Zhao, K. (2011). Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search. *Applied Soft Computing*, *11*(4), 3680-3689.
18. Geng, X., Chen, Z., Yang, W., Shi, D., & Zhao, K. (2011). Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search. *Applied Soft Computing*, *11*(4), 3680-3689.
19. Shi, X. H., Liang, Y. C., Lee, H. P., Lu, C., & Wang, Q. X. (2007). Particle swarm optimization-based algorithms for TSP and generalized TSP. *Information processing letters*, *103*(5), 169-176.
20. Alhanjouri, M. A. (2017). Optimization Techniques for Solving Travelling Salesman Problem. *Optimization Techniques for Solving Travelling Salesman Problem*, *7*(3).