IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-6, Issue-2 E-ISSN: 2347-2693

A Novel Approach to solve Traveling Salesman Problem (TSP) using Metaheursitic Hybrid Algorithms

Supreet Kaur[1], Kiranbir Kaur[2]

[1] Department of Computer Engineering and Technology, Guru Nanak Dev University, Amritsar, Punjab, India
[2] Department of Computer Engineering and Technology, Guru Nanak Dev University, Amritsar, Punjab, India

**Corresponding Author: supreetk033@gmail.com*

Available online at: www.ijcseonline.org

Received: 25/Jan//2018, Revised: 06/Feb2018, Accepted: 21/Feb/2018, Published: 28/Feb/2018

***Abstract*—** There is a great need for Artificial Intelligence and Nature Inspired Metaheuristic Algorithms for real world problems like Traveling Salesman Problem (TSP) belonging to NP-Hard Optimization problems which are hard to solve using mathematical formulation models. They are also a requirement for fast and accurate algorithms, specifically those that find out a node from start to the goal with the minimum cost, distance, time, money, energy etc. The Traveling Salesman Problem (TSP) is a combinatorial optimization problem which in it's the purest form has a respective application for instance planning, logistics, and manufacture of microchips, military and traffic and so on. Metaheuristic techniques are general algorithmic frameworks including nature-inspired designs to solve complex optimization problems and they are a fast-growing research domain since a few decades. This paper proposes to solve this problem using hybridization of ACO (Ant Colony Optimization) and SA (Simulated Annealing). Ant Colony Optimization (ACO) is a population-based metaheuristic that can be used to find out appropriate approximate solutions to understand difficult NP-Hard optimization problems. Simulated Annealing (SA) is also a population-based metaheuristic that is inspired by annealing process proceeded with higher level temperature rate; it starts position on a first solution to maximum temperature, where the exchange states are accepted with a desired global extreme point is out of sight among many, poor temperature and probability density function, local update extrema. Moreover, MATLAB programming is used to implement the algorithms using solved TSP are three benchmarks on the same platform conditions for ACO, SA and Hybrid ACO-SA.

***Keywords*—** Metaheuristic Hybrids, Ant Colony Optimization (ACO), Simulated Annealing (SA), Traveling Salesman Problem (TSP), NP-Hard Optimization Problems, Global Pheromone Update (GPU), Local Pheromone Update (LPU)

1. INTRODUCTION TO METAHEURISTIC

"Meta" means 'Top level' of the managing the heuristic information and Metaheuristic Algorithms are generally executed as better alternative to simple heuristics. A metaheuristic is a defined as an iterative generation of step-by-step technological development or innovation (Third Generation of Computers) process which guides a subordinate heuristic by compounding intelligently many different parts for exploring and exploiting the search space. **[1]** Learning strategies is used to structure information in order to find out effectively near-optimal best cost solutions. **[2]**

Properties of Metaheuristics Techniques:-

a) Metaheuristics is scheming that conduct the search process.
b) The goal of metaheuristic is to effectively research the search space in order to find out the near optimal solutions.
c) The techniques which comprise metaheuristic algorithms have the scope for simple local searches process of complex learning processes of the solved problem.
d) Metaheuristic algorithms are approximate and specifically non-deterministic.
e) They may integrate mechanisms to avoid acquiring trapped in confined domain of the search space.
f) The concepts of metaheuristics allow an abstract level description.
g) Metaheuristics is not problem-specific to solve the particular problem.
h) Metaheuristics may make use of domain-specific knowledge in the kind of heuristics that are controlled and operated by the top level strategy.
i) Today's many advanced metaheuristics used to search experience (corporate in some kind of memory to guide the search) and domain artificial intelligence.

Advantages of Metaheuristics:-

a) It tends to move flexibility quickly towards very good cost solutions, so it provides a very efficient way of dealing with large complicated problems.
b) It useful in the traditional methods gets stuck at local minima's.
c) The common area of application is a combinatorial optimization problem.

Disadvantages of Metaheuristics:-

a) There is no guarantee that the best cost solution found will be optimal solution.

1.1. Introduction to Hybrid Metaheuristics

A number of algorithms that do not solely as espousing the concepts of one single traditional metaheuristics, because of the compound variously algorithmic estimation, allows for one originating from another branches of optimization (such as cloud computing and the branch of science and engineering). These approaches are also called as metaheuristic hybrids or hybrid metaheuristics. [**3**] The estimation of hybridizing metaheuristics is not a new idea which dates back from their origins of the strict hybrids feedback on the parallel computing. **[4]** In the starting, however, for instance combinations were not much popular since several problems like flexibility head-strongly separated and sometimes competing planned communities of researchers co-existed that extended considerably their favourite classes of metaheuristics generally owing to a lack of superior to another and dogmatically followed up on their specifically philosophies of the particular Metaheuristic Hybrids. The motivation behind the hybridization of differentiable algorithms lexicalized concepts are usually for obtaining better performing to execute systems in that unexploited and unit advantages of the individual sole strategies; such as instance hybrids are believed to derive a benefit from synergic energy system. It is an accomplished fact now that picking an adequate (equal) combination of multiple algorithms concepts is the key to achieving for top performance in the resolution of very difficult problems. There are Classification of Metaheuristic Hybrids are as follows:-

a) Hybridized Algorithms
b) Level of Hybridization
c) Order for Execution
d) Control Strategy

2. RELATED WORK

2.1. Related Work for ACO using TSP:-

Dorigo and Gambardella **[5]** proposed the Ant Colony System (ACS) to solve Traveling Salesman Problem with cooperative learning and a distributed algorithm. In the ACS, a set of co-operating agent system called ants collaborates to find out the good cost solutions to TSP's. Ants collaborate using an indirect kind of communication mediated by a heuristic pheromone they deposit based on the edges of the graph while the building designing cost solutions. The natural metaphors in which ant algorithms are based on are the ant colonies and ACS (Ant Colony System) are a tour or travel construction heuristic similar to Freisleben and Merz's genetic algorithm, which produces a set of feasible shorter solutions after all iterations which take almost sense a mutation of best cost solution.

Randall and Lewis **[6]** applied Parallel ACO to solve traveling salesman problem. ACO is a constructive population based technique on the meta-heuristic search space technique. One of the important applications of parallelization strategies is order to improve its efficiency, particularly for a large real world the particular problem in which that Parallel Independent Ant Colonies, are based on the well best-known master/slave approaches and are thence appropriate for the widely distributed popular based MIMD (Multiple Input, Multiple Data) machines architectures of Parallel Computing.

Musa Peker and Baha Sen **[7]** applied Ant Colony System (ACS) to solve TSP with Taguchi Method. The best solution to the solved problem with daily life in road and route planning, business planning and the identification of a sequence of operations during perforation for printed circuits boards are examples of ACS using TSP.

Pan Junjie and Wang Dingwei **[8]** proposed ACO for solving multiple TSP. Multiple Traveling Salesman Problem (MTSP) includes with regard to the computation complex combinatorial optimization problem, an extended version of the famous TSP. The MTSP can be unspecialized a wide variety of routing and scheduling problems. Thus the MTSP with the capability of the constraint on the real world problems like TSP. Notwithstanding; MTSP is a NP-complete problem in which the optimal solutions can only be found for short or small size problems.

Shu-Chuan Chu, John F. Roddick, Jeng-Shyang Pan **[9]** suggested ACS to solve TSP with communication strategies. The basic motivation of the communication is to update or modify the heuristic pheromone level for each path or route reported to the best route success found by neighbouring groups or in several instances for a given time. The aim at the PACS (Parallel Ant Colony System) is not merely to reduce the computation time. The declared examples of PACS may be applied to solve the quadratic assignment problem, data mining, space planning, data clustering and the combinatorial optimization problems.

Zar Chi Su Su Hlaing and May Aye Khine **[10]** presented the solution of TSP by using Improved ACO Algorithm. An improved extension of ACO algorithm is based on candidate list strategy using database list and also declared dynamic heuristic parameters update based on the entropy and the emergence of local search space solutions is declared.

Chiranjib Patra, Pratyush **[11]** proposed Vector Ant Colony Optimization (VACO), a distributed algorithm which is applied to solve the particular TSP. The proposed system (VACO) is based on the basic ACO algorithm with well distribution strategy in which the full search space domain is initially divided into 2^n number of hyper-cubic quadrants where n is the dimension of search space for updating the heuristic or pheromone parameter in ACO to improve the performance in solving the particular problem using TSP.

Luca Maria Gambardella and Marco Dorigo **[12]** suggested Solving Symmetric and Asymmetric TSPs by Ant Colonies. With Artificial Intelligence Techniques, Ant Colonies (AAC) classes as algorithms are applied to the several instances of combinatorial optimization problems like a symmetric and an asymmetric TSP (TSP and ATSP severally), the quadratic assignment problem and the job shop scheduling problem. The idea about this is to take inspiration behind the reflection of nature phenomena (and in the any way from Genetic Algorithms) and to define as a population of agents systems with different structural parameters of the control system.

Table 1 - Brief Survey of ACO Techniques Used and Details of the System Parameters

Author's Name or Referenced Work	**Technique Used**	**Implementation Parameters**	**Determination Method of System Parameters**
Dorigo and Gambardella **[5]**	ACO (TSP)	Tour Length produced by nearest neighbor heuristic, Number of cities, Number of ants	ACS-3-opt has approximately the same performance as the "large step Markov chain algorithm".
Randall and Lewis **[6]**	Parallel ACO (TSP)	Iterations, Initial City, Next City, Best Solution, Cost, Number of ants	Speedup and efficiency increase as problem size increases.
Musa Peker and Baha Sen **[7]**	Taguchi Method (TSP)	Number of ants, Constant, Tour Length, Pheromone Trace, Desirability Value, Set of cities	Minimize the variability in a product or operation in line with a specified function (Least best, Most best, and Targeted value best) by selecting the most suitable compounded of the controllable factor levels compared to uncontrollable factors that select variability for a specific operation.
Pan Junjie and Wang Dingwei **[8]**	ACO for Multiple Traveling Salesman Problem (MTSP)	Number of Salesman, Number of cities, Maximum number of cities salesman	To evaluate the efficiency of the ACO algorithm for the MTSP with ability constraint, we simulated for the another algorithm available in namely modified genetic algorithm (MGA).
Shu-Chuan Chu, John F. Roddick, Jeng-Shyang Pan **[9]**	Communication Strategies	Current City, Next City, Pheromone level between current city and next city, inverse of the distance between current city and next city	To evaluate the effectiveness of PACS, comparing PACS with ant system (AS) and ant colony system (ACS).
Zar Chi Su Su Hlaing and May Aye Khine **[10]**	Improved ACO (TSP)	Tour length, Iteration, Current Pheromone matrix, Threshold, Heuristic visibility of edge	Its efficiency of best cost solution is higher than ant colony algorithm and the convergence speed is better than that of ant colony system.
Chiranjib Patra, Pratyush **[11]**	Vector ACO (TSP)	Iterations, Pheromone Decay, Cost, Current City, Next City	The performance of VACO algorithm is a set of point is evaluated to find out how tour construction is efficient than another algorithms.
Luca Maria Gambardella and Marco Dorigo **[12]**	Symmetric and Asymmetric TSP by Ant Colonies	Cost, Positive real value associated to edge, Heuristic Function, Random variable, randomly with uniform probability	Slightly Modified version of ACS which integrate a many advanced data structure known as candidate list.

2.2. Related Work for SA using TSP:-

Dasaradh R. Mallampati, Roger L. Wainwright **[13]** proposed Parallel Multi-Phase of Simulated Annealing (SA) to solve TSP. It used a ring topology hypercube arrangement. The algorithm for the TSP using SA has two phases as the following :- In Phase 1, consists of the distribution of the travel to visit tour of the processors and the interaction amongst the Internal Cities within all processor. In Phase 2, several interconnection schemes are used for movement or redistribution the cities of the travelled tour amongst the processors. The algorithms of Parallel Multi-Phase using SA are three phases as the following :- In Phase 1, is 2-opting are executed separately within all of the two tiers of the travelled or tour, In Phase 2, remote-swapping is executed between cities from the two different tiers of the cities tour and In the last phase 3, a synchronization of the cities is completed by all processor shifting transfer a quarter of its cities in a clock-wise direction movement to its neighboring node or state, which is called a quarter-spin.

Husamettin Bayram, Ramazan Sahin **[14]** presented a New SA approach for solving TSP. The main ideas are to improve searching ability of SA heuristic through the integration of two or more neighbourhood mechanisms. Owing to its ease of formulation, to solve the particular problem and various real life important applications are also compared to conventional SA within a swapped neighborhood system region. The main advantages are improved convergence ability by using problem data onto losing inherent stochastically in the SA techniques.

Parham Azimi, Ramtin Rooeinfar **[15]** suggested a New Hybrid Parallel SA Algorithm to solve the TSP with Multiple Transporters. In competitive transportation systems, the passengers try to find out the traveling agencies that are capable of the serve them effectively compared both the traveling time and the transportation costs. The purpose is to determine an optimal sequence of visited travel cities with minimum traveling times for available transporting vehicles for a small-scale budget system, speed and resolution quality. The TSPMT (Traveling Salesman Problem with Multiple Transporters) is much unmanageable than the classical TSP versions extensions, making it a NP-hard problem.

Nitesh M. Sureja, Bharat V. Chawda **[16]** proposed "Random TSP using SA". The computational linguistics SA model is based on the simulating process of slow down cool about molten metal to achieve the minimum probability function value of a minimization problem. Random TSP (RTSP) is a variant of TSP in which SA model turn up good in terms of the length of the travelled tour but really bad in terms of the quality and the convergence time.

David Bookstaber **[17]** proposed an Implementation of an SA algorithm to find out the optimal solutions to the TSP. For comparability, a Genetic Algorithm (GA) is applied to the same TSP and SA problem. In this system, sub-optimal solutions are solved that are even really good. The examples of Monte Carlo Methods like SA and GA's are proven successfully found in generating functional solutions in spite of the size of the problems.

M.A. Rufai, R.M. Alabison **[18]** suggested a solution to the Traveling Salesperson Problem using SA Algorithm. The salesperson wants to find out the shortest feasible tours for instance road (path or route) in which the total distance or total time will have minimum values. The major problems are how to travel salesperson problem to accomplish his or her military mission effectively and efficiently.

Lin Xiong, Shunxin Li **[19]** tried to solve TSP based on the Improved SA Algorithm with Sequential Access Restrictions. Sequential Access Restrictions compounded with the important properties of convex hull, give a loose access order for the TSP. The access restrictions on the route or path planning algorithm of TSP, clustering analysis, combined to the clustering with SA algorithm, and improving the transformed rules of cost solutions to SA, an Improved SA algorithm is proposed. The proposed algorithm reduces the operation time, improves the overlapped effectives of the SA algorithm.

Zicheng Wang, Xiutang Geng **[20]** proposed an effective SA algorithm for solving the TSP. It is a well known solution as TSP is one of the almost NP-Complete problems and evolutionary technique. SA is many seen used to solve the various particular NP-Complete problems. The two stage SA algorithm is represented which are following :- In the first stage, a simple SA algorithm is declared to obtain appropriate solutions or closed travel tours. In the second stage, an effective SA algorithm is declared to obtain cost solutions to a good quality based on the best solutions or closed tours or cities problems obtained by the easy SA algorithm.

Table 2 - Brief Survey of SA Techniques Used and Details of the System Parameters

Author's Name or Referenced Work	Technique Used	Implementation Parameters	Determination Method of System Parameters
Dasaradh R. Mallampati, Roger L. Wainwright **[13]**	Parallel Multi-Phase or Parallel TSP	Number of cities, Initial Tour Distance, Final Tour Distance	Improvement on his random spin around.
Husamettin Bayram, Ramazan Sahin **[14]**	Genetic Algorithm or Simulated Annealing	Cooling Rate, Neighbor Solution, Fitness value, current solution	Increase it is running time and Improve the quality of solutions.
Parham Azimi, Ramtin Rooeinfar **[15]**	Hybrid Parallel SA for TSP or Multiple Transporters	Population Size, Temperature decreasing rate	Lower time consuming is more expensive and flexibility.
Nitesh M. Sureja, Bharat V. Chawda **[16]**	Random TSP	City Problem, Length of Tour	Computational SA Model.
David Bookstaber **[17]**	SA	Temperature, Iteration, Distance, Fitness Value, Cities	Beginning the temperature is very high and the behavior is more random.
M.A. Rufai, R.M. Alabison **[18]**	Branch and Bound Algorithm	Best Tour, Initial Tour, Cities, Distance	Aim is to minimize the total distance travelled to a salesperson during his/her visits to all the regions.
Lin Xiong, Shunxin Li **[19]**	Improved SA with sequential Access Restrictions	Initial Temperature, Termination Temperature, Cooling rate, Iterations, Optimal Solutions	To test the feasibility and effectiveness of method of solving the TSP problem.
Zicheng Wang, Xiutang Geng **[20]**	Greedy Search or 2-Stage SA	Temperature, Size, Numbers of Tour, Cooling Times	2-Stage SA, the cool co-efficient of temperature is slow.

2.3. Related Work for Hybrid ACO-SA using TSP:-

Seyed Ahmad Sheibat Alhamdy **[21]** proposed TSP using ACO algorithm and compared with Tabu Search, SA and Genetic Algorithm. It is demonstrated that against reckoning to the restriction of resource system and better reduction in the cost of production actions, service activities and other actions of this form. Decision makers are seeking quest for new methods along with accurate and speedy results to access the optimum tools. A variety of problems and acceleration in finding out a cost solution has direct affect on the success of classical methods in finding out the solution for these problems.

Adewole A.P. **[22]** compared SA and GA for solving the TSP. It concluded that SA runs faster than GA and runtime of GA grows exponentially with the number of cities involved. Yet, in the terms of a good quality solution GA is better than SA. GA nonetheless can provide the quality solutions if an adequate population bases is setting marked, but larger population increase its run time. The powerful systems with the parallel computing ability can be used to operate these algorithms for larger number of cities. To a better extent, both algorithms are very good at/for solving the problem and can find out the optimal solutions if the right sets of parameters are set. If the solution quality important, decrease or increase from cool rate close to the one should be set for SA but maximum number of iterations the algorithm will be need to perform.

Mateusz Borowski and Rafal Machon **[23]** proposed a modified TSP. A modified TSP includes hybrid metaheuristics combination of the delivery time and the fuel consumptions are taken to compare two algorithms which are the based of ACO approaches and SA approach. The main targets are to select a metaheuristic algorithm which can find out the best path or route in the defined scenario. The classical task consists of finding out an optimal route or path regarding either time or distance. It could be important to decrease cost of company's maintenance of the system. Normally, the fuel consumption function is defined as an accepted best or worse solution to be the non-linear system in advance for specific vehicle routes used by the courier's messenger of the modified TSP using ACO.

Hashim Ali and Yasir Shah **[24]** tried to Solve Traveling Salesman Problem with optimization techniques using GA and ACO. Swarm robotic is a new research area in the domain of Artificial Intelligence Techniques. Especially, the swarm robot concept is adopted from the Mother Nature that compound small robots in a group to solve the particular problem. It also solves the TSP using free and easy parameters; i.e. number of cities, number of iteration and number of ants committed in a search for by the solution searches space. Running time of GA and ACO is calculated in the two scenarios by fixing or altering number of iteration, fixing or altering the number of cities to nodes in a solution search space as input methods and keeping other parameters constant or infinite.

M. Fikret Erean and Xiang Li **[25]** proposed to hybridize Particle Swarm Optimization (PSO) and SA. PSO algorithm is the widely used for the various engineering problems on account of its simplicity and effectively. Yielding, stated that PSO has a problem of early convergence, because of the lack of diversity. The Standard PSO is a stochastic search algorithm with multimodal search space problem, rising from simultaneous use of dynamic systems such as bird flocks and fish swarms. Typical application of PSO-SA hybrid includes a first search for PSO and refines results of the parameter system with SA. SA presents a higher probability function to avoid local minima.

Muhammed Basheer Jasser and Mohamad Sarmini **[26]** presented an approach to solve TSP with ACO and a variation of Bee Colony Optimization (BCO). TSP finds out the Hamiltonian path or route with the minimum cost travelled or visited. It has good applications, which include enhancing transportations, finding out the best path or route in the network and enhancing routing algorithm. One of the variations is integrating the primary BCO with 2-opt local search heuristic to ensure better cost solutions to TSP. It is found that ACO algorithms costs less or take less time compared to an enhanced BCO when the search spaces problem and number of iteration are small, but when the search space and iterations number to increase, the computational time for both ACO and enhanced BCO becomes approximately similar.

Younis Elhaddad and Omar Sallabi **[27]** applied a new hybrid Genetic and SA Algorithm to solve the TSP. The basic motivation is to the declare Hybrid Genetic and Simulated Annealing Algorithm (HGSAA) to get the better results than GA. When it deposits any early local minima, by switching to SA, which has a better opportunity in jumping all over this problem with its hill-climbing approach. When it appears that the GA is stuck after 20 consecutive generations of the hybrid algorithm switches to the population of the SA which provided uphill jumps to a higher cost solution in order to avoid being trapped in the local minima. After several SA iterations; the new population comes to GA, more researcher's idea of combining two or more algorithms in order to improve the cost solution quality and reduces execution time to be performed.

Huiling Bao **[28]** presented a solution to TSP. The ACO and SA is used to declare a two-phase hybrid optimization (TPASHO) algorithm. In declared TPASHO algorithm, the advantages of parallel, collaborative and positive feedback of the ACO algorithm are used to implement a procedure for global search in the present temperature. Adaptive adjustment threshold strategy is used to improve the search spaces exploration and balance the local exploitation. When the calculation process proceed, the ACO algorithm drops into the stagnation state of inactivity, the SA algorithm is used to make a local optimal solution and obtain best cost solution of the ACO algorithm. Then, the SA algorithm is executed to search out the neighbourhood. It is a best search space precision and has faster convergence speed, high efficiency and high effectiveness to the specific problem.

Table 3 - Brief Survey of Hybrid ACO-SA Techniques Used and Details of the System Parameters

Author's Name or Referenced Work	Technique Used	Implementation Parameters	Determination Method of System Parameters
Seyed Ahmad Sheibat Alhamdy **[21]**	ACO, SA, GA, Tabu Search (TSP)	Number of cities, Iterations, Cost	It worth remarking that the shortest path or route for all city has been solved by Fogel Method.
Adewole A.P. **[22]**	SA, GA (TSP)	Population Mutation Rate, Cut length temperature, Cooling Rate, Absolute Temperature	The TSP used to the Chromosomes "Mate" method to reproduce new offspring from favored population.
Mateusz Borowski and Rafal Machon **[23]**	SABA (Simulated Annealing Based Algorithm), ACOBA (Ant Colony Optimization Based Algorithm)-TSP	Temperature, Repetition, Iteration, Initial Pheromone, Number of Ants	The perform resulting is path or route, total distance, total traveling time, fuel consumption, average fuel consumption.
Hashim Ali and Yasir Shah **[24]**	GA, ACO	Number of cities, square distance, Iteration, Number of ants	The problem assumes different complexity because it's a type of metaheuristics algorithm.
M. Fikret Erean and Xiang Li **[25]**	PSO, SA	Temperature Current, Temperature end, best solution, weight start, weight end, Iteration	Initial Search with PSO and refining with SA.
Muhammed Basheer Jasser and Mohamad Sarmini **[26]**	ACO, BCO (TSP)	Pheromone, Fitness population, current city, next city, iteration	Both ACO and enhanced BCO algorithms have been array data structure where the journey paths, set of moves, pheromone trails, waggle dances are represented using 2D arrays.
Younis Elhaddad and Omar Sallabi **[27]**	Hybrid Genetic and SA Algorithm (TSP)	Iteration, Cost, Cities, Population Size, Error, Optimal	Switches the population of SA in order to allow for uphill jumps to a higher cost solution in order to avoid getting trapped in local minima when the GA stuck after consecutive generations.
Huiling Bao **[28]**	2-Phase Hybrid Optimization (TPASHO) Problems such as ACO-SA (TSP)	Iteration time, population size, Initial temperature, temperature cooling, pheromone factor, heuristic factor, pheromone amount, evaporation co-efficient	Performance of the proposed TPASHO Algorithm is evaluated on 18 library TSP and is favorably compared with SA and ACO algorithms.

3. METHODOLOGY

3.1. Traveling Salesman Problem (TSP)

Given a set of cities and the cost of travel (or distance) between all possible pairs (X co-ordinates and Y co-ordinates), the TSP is to find out the better possible ways of travelling to all the cities precisely once and backward to the beginning point that minimizes the travel cost (or travel distance). The TSP is a NP-Hard optimization problem with combinatorial optimization; analyze the system performance in theory-based branch of computer science and engineering. In other applications, extra constraints say that limited resources or time windows create the problem substantially harder. Deducting the constraint on travel to each city precisely one time also did not reduce the complexity of the problem (output will be runtime). Despite the computational difficulty of the problem, a large number of heuristics or metaheuristics and exact methods are well-known which can solve the case with thousands of cities.

States of the TSP :- The TSP reported to a salesman who must be traveling through N cities in which the order in which he does so is something he does not worry about, as long as he visits all at once throughout his trip, and end goals where he had started. One city is linked to other cities, or nodes, by airplanes or by road or railway. One of those links between the cities has one or more weights (or the cost) assigned. The cost represents how difficult is it to traverse city on this edge of the graph such as post-order, pre-order and in-order, and may run, for instances, by the cost of an airplane ticket or train ticket or possibly by the length of the edge, or time needed to traversal it. The Salesman needs to keep some the travel costs, a part from a distance he travels as small as theoretical analyze workable.

The TSP is a typically among numerous NP-hard optimization problems that have fascinated mathematicians and computer scientists for years. It has its applications for science and engineering. **[29]** For example, to fabricate a circuit board, it is important to find out the best cost solution orders in which a laser will drill thousands of holes and an effective best cost solution to this problem cut down production costs significantly for the manufacturer.

Complexity of TSP :- Given N as the number of cities to be visited, the total number of executable routes covering all cities can be given by a set of shorter feasible solutions to the TSP and is given formula by,

$$\frac{(N-1)!}{2}$$

[30]

Difficulty of the TSP :- The TSP is considered as difficult to be solved, if there is a way to break this problem with smaller element problems, the elements will be at least as hard as the primary one. This is what computer scientists and technologies call as NP-Hard optimization problems.

The easiest and the most valuable resolution is to only try all possibilities specific to the problem. With this, is for N cities you have (N-1) factorial number of possibilities and means that for only 10 cities there are over 180 thousand combinations to try (since the begin city is defined, there can be permutations system on the remaining nine). We only have half the number since one route has an equal route or path in invert with the same length or cost or time.

$$\frac{(10-1)!}{2} = \frac{(9)!}{2} = 181440$$

Mathematical Formulation Description for the TSP :- In Traveling Salesman Problem (TSP), a set of N cities

$$N = \{N_1, N_2, \ldots\ldots\ldots\ldots, N_m\}$$

and a set of edges (E) of path or routes linked to the cities with each other cities are given.

In the other words, we have an undirected graph G = (N, E), where N is the set of nodes and

$$E = \{E_{ij}\}$$

i = 1, 2, 3_ _ _ _ _, m and j = 1, 2, 3_ _ _ _ _, m, $i \neq j$, is a set of edges. One edge $E_{i\,j}$ (link nodes N_i and N_j) has a values $D_{i\,j}$ – the distance between cities N_i and N_j, the task is to find out the shortest feasible route or path to all the possible cities, such that one city has to be visited just once.

3.2. Ant Colony Optimization (ACO) using TSP

Ant Colony optimizations (ACO) belong to the population based optimization and group of metaheuristic techniques which are used to the established the existent behaviour of ants in nature. The working of an ant colony allows for indirect communication with the help of local pheromones updates, which ants run or walk. **[31]** The Pheromone Updates is chemical substances in attracting other ants inquiring into food. The attractiveness of an acknowledged path or node relies on the quantity of pheromones that an ant. The Pheromone Update elimination is governed by some regulation and has not always the same strength. The quantity of pheromone updates to affect the attractiveness of the path. The uses of many attractive paths ensure that the ant executes much pheromone on search space on its way backward and so that the path also becomes attractive to other ants. A significant characteristic of local pheromones updates is the evaporation rate, during the given time. When the path is no longer used, local pheromone updates are evaporated and the ants start using other paths.

A large number of artificial intelligent ants tries to the construct solutions to the given problem with the help of the heuristic pheromone update deposited to the knowledge cognition and some another required information correlated to the specific particular problem. **[32]**

Edge Selection :- An ant is a plain computational agent in the ant colony optimization (ACO) algorithm which includes maximum iterations constitutes the best cost solution to the problem in hand. The intermediate cost solutions are known as solution states. During each iteration of the algorithm, all ants move from a state x to state y, corresponding to a more full intermediate cities problem of the solution. Thus, every ant k computes a set $A_k(x)$ of feasible shorter path expanding its present state in each iteration and moves to one of these in probability density function. For ant k, the probability density function F^k_{xy} of moving from state x to state y (for instance X Co-ordinates and Y Co-ordinates) depends upon the combination of two values of the attractiveness η_{xy} of the move, as calculated by some heuristic function indicating the a priori desirability of that move and the trail level τ_{xy} of the move, indicating how proficient it has been in the past to create that special move. **[33]**

The trail level includes a posterior indication of the desirability of that path to the cities problem and the trails are updated commonly when all ants have completed their best cost solution, increasing or decreasing the level of trails related to moves that are part of the good or bad solutions. Specifically, the k^{th} ant movement towards state x to state y with probability density function F is given by,

$$F^k_{xy} = \frac{(\tau^{\alpha}_{xy})(\eta^{\beta}_{xy})}{\sum_{z \in y}(\tau^{\alpha}_{xz})(\eta^{\beta}_{xz})}$$

where,

i. τ_{xy} is the amount of pheromone update spread for transition to state x to y, $0 < \alpha$ is a parameter to control the influence of τ_{xy}
ii. η_{xy} is the desirability of state transition xy (a priori knowledge, generally $1 / d_{xy}$, where d is the distance) and $\beta > 1$ is a parameter to control the influence of η_{xy}
iii. τ_{xz} and η_{xz} represent the attractiveness and trail level for the possible state transitions

Local Pheromone Update :- When all the ants have completed a good or bad solution, the trails are updated by given formulas :-

$$\tau_{xy} \leftarrow (1-\rho)\tau_{xy} + \Sigma_k \Delta\tau_{xy}^k$$

where,

i. τ_{xy} is the amount of pheromone update for transition xy
ii. ρ is the pheromone evaporation rate coefficient
iii. $\Delta\tau_{xy}^k$ is the amount of pheromone update situated by k^{th} ant

Algorithms of ACO Using TSP :-

```
Set Parameters and Initialize Pheromone Trails;
while <condition not met>
{
        Construct Solutions;
        Local Pheromone Update;
        Heuristic Solutions;
        Update Trails;
}
Global Pheromone Update;
Evaporation Rate;
```

3.3. Simulated Annealing (SA) using TSP

Simulated Annealing (SA) is a metaheuristic and probabilistic technique which is inspired by annealing process having the combination of the maximum iteration and reduces the number of sub-iterations at the higher temperature with probability distribution functions resulting in increased speed and accuracy. The Simulated Annealing (SA) algorithm is a method that simulates the material annealing process of metallurgy such as the science and technology of metals. For solving the TSP, SA begins from an initial tour (at high temperature), and the temperature is step-by-step decreased to achieve a minimum tour length.

An Annealing is also called as tempering which includes alloys of metal, glass or crystal by heating over its melting point, maintaining its temperature and then the cool it very slowly until the metal solidifies into a perfect (clear) crystalline structure. This reduces the defects of the alloy. This physical/chemical process produces high-quality materials. **[34]** The simulation is sometimes done on a smaller scale system of this process called as Simulated Annealing (SA). A physical material state represents the problem cost solutions, the energy of a state to best cost of a solution, and the high temperature to a control absolute parameter.

The state of the physical systems and the function E(S) to be the minimum values is the correspondent to the internal free energy of the system in that states. The goal of Simulated Annealing using TSP is to bring the system from an absolute first state, to a state with the minimum workable energy.

Maximum Iterations :- At all the steps, the SA heuristic views some neighbouring state S' of the present state S, and decides after comparing probabilistic function whether to move system to state S' or stay in state S. These probability distribution functions ultimately control the system to execute moves to states of lower energy level. Usually this step is repeated till the system reaches a both present state and new state that is sufficiently good for the application, or until a given computation budget has been exhausted.

Neighbour States :- The Optimization of a best cost solution evolves to appraise the neighbours of a state of the problem, which are the new states created by altering a granted state. For instance, in the TSP all states are typically defined as a permutation of the cities to be visited, and its neighbours are the set of permutations created by inverting the order of any two sequential cities and well-defined way in which the states are altered to create neighbouring states is called a “move” and different movements yield different sets of neighbouring states. They are state movements are resulted in minimal alterations of the last state and an attempt to growingly improve the resolution to iteratively improving its component parts such as the city links up the TSP.

Acceptance Probability Density Functions :- The probability density or distribution function F of making the transition states from the present state S to a candidate new state S' is specified by an acceptance probability density function (PDF) F(en, en', Temp), that depends upon the energies en = E(S) and en' = E(S') of the two states, and on a global update time-varying parameter Temp called the initial temperature. **[35]** States with least energy are better than those with a larger energy. The probability density function F must be positive even when en' is larger than en as well as the characteristic prevents the method of becoming cursed upon a local minimum that is worse than the global one.

When Temp is given value zero, the probability density function F(en, en', Temp) must become zero if en' > en and to a positive value else wise. For sufficiently short values of Temp, this system will then increase favour moved to the travel that goes “downhill” (i.e. to the lower energy values), and avoid those that go “uphill”. With Temp = 0 the operation reduces to the greedy algorithm, which makes only the downhill transitions.

The probability density functions F(en, en', Temp) is equal to one when en' < en, i.e. the operation always travels through downhill when it finds a way to do so, irrespective of the temperature.

The probability density functions F is usually marked so that the probability density functions of accepting a move decreases while the difference en' – en increase that is small uphill moves travel is more likely than large ones. Still, this requirement is not strictly essential, provided that the necessities are met.

Given these dimensions, the initial temperature Temp plays an important role of controlling the development of the present state S of the system considering its sensitivity to the variations on system energies parameter. To be accurate, for a large Temp, the evolution of present state S is sensitive to rough-cut energy variations, while it is sensitive to finer energy variations when Temp is short.

"Repetitions Schedule" defined as the number of maximum iterations performed at all temperature. Temperature is held constant or infinite for a considerable number of maximum iterations such that the stochastic matrix can be created for stationary (steady state) distribution also known as Constant Temperature repetitions (nrep).

Algorithms of SA Using TSP

```
Set Parameters and Initialize Temperature (Temp), Constant Temperature Repetitions (nrep);
while (sub_iteration <= MaxIt)
{
    sub_iteration = sub_iteration + 1;
    Temp = 0;
    while (Temp <= nrep)
    {
            Temp = Temp + 1;
            Select a new value from the neighborhood state;
            Compute current_cost (new value);
            delta = current_cost – previous_cost;
            if (delta < 0)
            {
                    Accept with Neighbor States;
            }
            else
            {
                    Accept with Probability Density Function {exp(-delta / Temp)};
            }
    }
      Temp = α * Temp;          (0 < α < 1)
}
```

3.4. Hybrid ACO-SA using TSP

Solving the TSP is an absorbing problem with recent years. Almost all new approaches from solving the engineering and optimization problems are tested for the TSP as a general test workbench.

The Metaheuristics Techniques which represented a family of approximate optimization techniques that had gained a lot of popularity in the past two decades is the tools used for solving the specific TSP. They are providing "Acceptable" best cost solutions to a sensible time for solving hard and complex problems in the branch of Science and Engineering.

The Population Based Metaheuristics, i.e. ACO and SA have been proposed in this paper for solving the Traveling Salesman Problem (TSP). The idea of Hybrid ACO-SA techniques is that by walk amid (among) their colony and the food source, ants leave heuristic pheromones on the way from where execution of SA starts beginning with highly absolute temperature and the temperature is step-by-step decreased to achieve minimum tour length at considering accuracy of the system.

The most instinctive way of the hybridizing two optimization algorithms is probably density function to engraft one algorithm into another algorithm either for beginning with the best solutions or for possibly improving intermediate solutions.

A Population Based Metaheuristic manipulates and controls a collection of candidate solutions to the database table such as list of a given time. There are possible solutions which are not independent (meaning that they are not parallel hill-climbers):- they regulate the way the other candidate solutions shut down with the optimum solutions. The main advantage of a population based method of the single-state metaheuristic results from the definition – in last bit-by-bit, many (near) optimal best solutions can be retrieved. Moreover, if difficult search space is the case, good solutions with higher probability function are not confused and new candidate solutions with special characteristics are to be generated from other solutions in the aggregation.

This paper proposes the development of a new Hybrid algorithm ACO-SA, that merges or unifies or unite two most popular techniques to solve the most complex population based techniques, the TSP.

The Aim at Hybrid ACO-SA algorithm is to increase the performance of ACO by combination of Number of Ants and Number of sub-iteration, temperature by SA for TSP.

Algorithms of Hybrid ACO-SA using TSP

```
Set Parameters and Initialization;
k = 1;
while <condition not met>
{
        for each ant until <condition not met>
        {
                Construction Solution;
                Local Pheromone Update;
          Heuristic Solutions;
                Update Trails;
        }
        SA intensification;
        for maximum sub-iteration sub-optimal solution until <condition not met>
        {
                Global Pheromone Update;
```

```
                Create Neighbours States;
                Sort Neighbours States;
                Evaporation Rate;
                Update Temperature;
                k = k + 1;
        }
}
```

4. Implementation and Results

4.1. Implementation and Results of Solved TSP using ACO

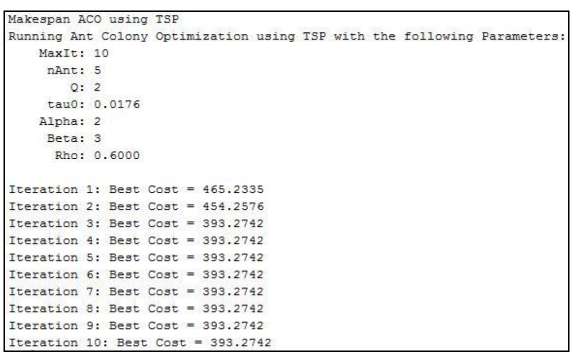
The solved TSP with ACO algorithms is implemented using the MATLAB Software (MATLAB version R2010a). In the Input system parameters of solved TSP with ACO algorithms are:-

a) **MaxIt –** Maximum Number of Iterations
b) **nAnt –** Number of Ants (Population Size)
c) **tau0 –** Initial Pheromone Update
d) **Alpha (α) –** Pheromone Exponential Weight
e) **Beta (β) –** Heuristic Exponential Weight
f) **Rho (ρ) –** Evaporation Rate
g) **Q –** Constant that determines how to Initial Pheromone Update

In the Output system parameters of solved TSP with ACO algorithms are:-

a) **BestCost –** Array to Hold Best Cost Value

In the Input and Output values of the system parameters using Makespan of Solved TSP with ACO are as Figure 1 as shown.



```
Makespan ACO using TSP
Running Ant Colony Optimization using TSP with the following Parameters:
    MaxIt: 10
     nAnt: 5
        Q: 2
     tau0: 0.0176
    Alpha: 2
     Beta: 3
      Rho: 0.6000

Iteration 1: Best Cost = 465.2335
Iteration 2: Best Cost = 454.2576
Iteration 3: Best Cost = 393.2742
Iteration 4: Best Cost = 393.2742
Iteration 5: Best Cost = 393.2742
Iteration 6: Best Cost = 393.2742
Iteration 7: Best Cost = 393.2742
Iteration 8: Best Cost = 393.2742
Iteration 9: Best Cost = 393.2742
Iteration 10: Best Cost = 393.2742
```

Figure 1 - Makespan solved TSP with ACO

In the Execution Time of Solved TSP with ACO algorithms as Figure 2 shown, the cities problem of X Cities and Y Cities in values using MATLAB implemented are :-

X = [82 91 12 92 63 9 28 55 96 97 15 98 96 49 80 14 42 92 80 96];

Y = [66 3 85 94 68 76 75 39 66 17 71 3 27 4 9 83 70 32 95 3];

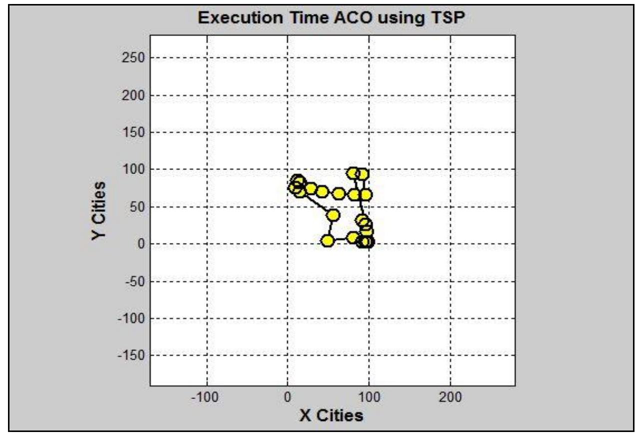


Figure 2 - Execution Time solved TSP with ACO

In the representation of the graph, there is a Calculate Cost of Solved TSP with ACO algorithms as Figure 3 as shown. The Inputs parameters number of iterations was set to 10 (MaxIt = 10) simultaneously with the graph and the Output parameters Best Cost Values are Solutions (Result Value) in the graph.

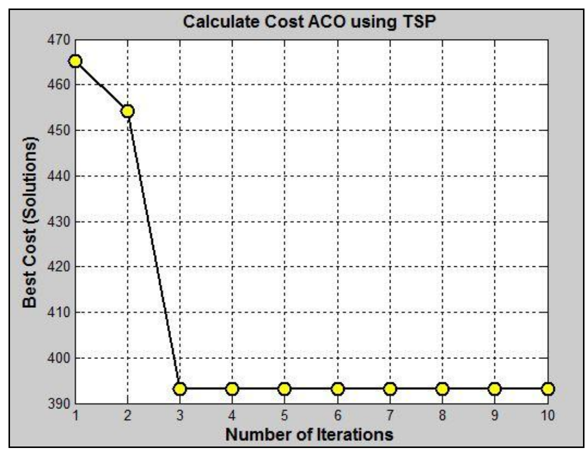


Figure 3 - Calculate Cost Solved TSP with ACO

In the results of Values Parameters with Solved TSP using ACO algorithms as Table 4 as shown, the values of all parameters inputs are different values and all parameters outputs are different values of the Best Cost (Solution).

Table 4 - Results of Values Parameters with Solved TSP using ACO

Input							Output
MaxIt	*nAnt*	*Q*	*tau0*	*α*	*β*	*P*	*Best Cost*
50	40	1	0.0088	1	1	0.05	372.2972
50	33	1	0.0088	2	5	0.5000	362.038
100	66	2	0.0176	3	2	0.1000	368.4315
200	70	3	0.0264	4	1	0.20	364.038
1000	22	0.3158	0.0028	1	1	0.682	364.9591
40	30	2	0.0176	2	3	0.6	371.4359
60	15	2	0.0176	2	3	0.45	370.3737
90	55	3	0.0264	2	4	0.6	368.4315

70	65	0.1	8.794	5	6	0.2	365.897
40	30	2	0.0176	6	2	0.4	377.9295

4.2. Implementation and Results of Solved TSP using SA

The solved TSP with SA algorithms is implemented using the MATLAB Software (MATLAB version R2010a). In the Input system parameters of solved TSP with SA algorithms are :-

a) **MaxIt –** Maximum Number of Iterations

b) **MaxSubIt –** Maximum Number of Sub-Iterations sub-optimal solutions

c) **T0 –** Initial Temperature System

d) **Alpha (α) –** Temperature Reduction Rate

e) **nPop –** Population Size

f) **nMove –** Number of Neighbour state per individual. In the Output system parameters of solved TSP with SA algorithms are:-

a) **BestCost –** Array to Hold Best Cost Value

In the Input and Output values of the system parameters using Makespan of Solved TSP with SA are as Figure 4 as shown.

```
Makespan SA using TSP
Running Simulated Annealing using TSP with the following Parameters:
       MaxIt: 10
    MaxSubIt: 5
          T0: 4
       Alpha: 1
        nPop: 3
       nMove: 3

Iteration 1: Best Cost = 917.4952
Iteration 2: Best Cost = 669.2478
Iteration 3: Best Cost = 662.1606
Iteration 4: Best Cost = 614.1086
Iteration 5: Best Cost = 614.1086
Iteration 6: Best Cost = 614.1086
Iteration 7: Best Cost = 598.649
Iteration 8: Best Cost = 536.4562
Iteration 9: Best Cost = 536.4562
Iteration 10: Best Cost = 536.4562
```

Figure 4 - Makespan solved TSP with SA

In the Execution Time of Solved TSP with SA algorithms as Figure 5 shown, the cities problem of X Cities and Y Cities in values using MATLAB implemented are :-

X = [82 91 12 92 63 9 28 55 96 97 15 98 96 49 80 14 42 92 80 96];

Y = [66 3 85 94 68 76 75 39 66 17 71 3 27 4 9 83 70 32 95 3];

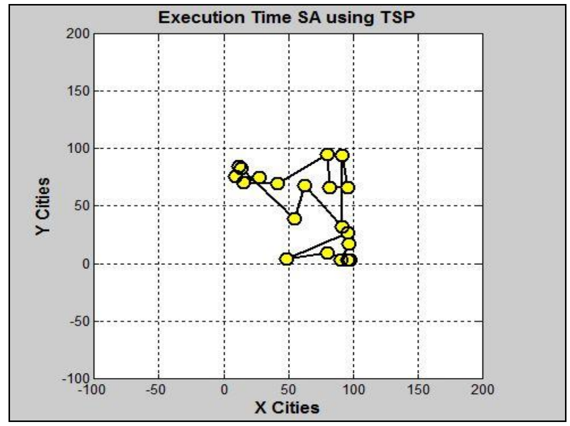


Figure 5 - Execution Time solved TSP with SA

In the representation of the graph, there is a Calculate Cost of Solved TSP with SA algorithms as Figure 6 as shown. The Inputs parameters number of iterations was set to 10 (MaxIt = 10) simultaneously with the graph and the Output parameters Best Cost Values are Solutions (Result Value) in the graph.

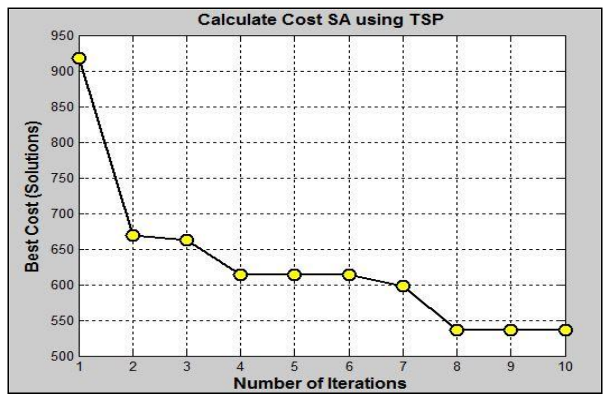


Figure 6 - Calculate Cost Solved TSP with SA

In the results of Values Parameters with Solved TSP using SA algorithms as Table 5 as shown, the values of all parameters inputs are different values and all parameters outputs are different values of the Best Cost (Solution).

Table 5 - Results of Values Parameters with Solved TSP using SA

Input						**Output**
MaxIt	***MaxSubIt***	***T0***	***α***	***nPop***	***nMov***	***Best Cost***
100	10	0.0250	0.99	3	5	362.038
50	30	0.1	2	1	4	474.3771
90	30	0.56	3	4	5	392.9072
40	5	0.78	1	6	8	369.2946
60	15	0.99	4	2	3	529.8129
70	66	2	0.5	3	6	362.038
200	100	4	5	1	7	384.9372
66	25	0.2	4	5	1	778.2068
30	25	3	0.1	4	2	362.038
10	5	4	1	3	3	532.1193

4.3. Implementation and Results of Solved TSP using Hybrid ACO-SA

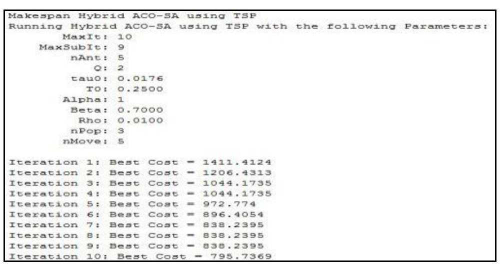
The solved TSP with Hybrid ACO-SA algorithms is implemented using the MATLAB Software (MATLAB version R2010a). In the Input system parameters of solved TSP with Hybrid ACO-SA algorithms are:-

a) **MaxIt –** Maximum Number of Iterations
b) **MaxSubIt –** Maximum Number of Sub-Iterations sub-optimal solution
c) **nAnt –** Number of Ants (Population Size)
d) **tau0 –** Initial Pheromone Update
e) **T0 –** Initial Temperature System
f) **Alpha (α) –** Pheromone Exponential Weight and Temperature Reduction Rate
g) **Beta (β) –** Heuristic Exponential Weight
h) **Rho (ρ) –** Evaporation Rate
i) **Q –** Constant that determines how to Initial Pheromone Update
j) **nPop –** Population Size
k) **nMove –** Number of Neighbours state per individual

In the Output system parameters of solved TSP with Hybrid ACO-SA algorithms are:-

a) **BestCost –** Array to Hold Best Cost Value

In the Input and Output values of the system parameters using Makespan of Solved TSP with Hybrid ACO-SA are as Figure 7 as shown.


```
Makespan Hybrid ACO-SA using TSP
Running Hybrid ACO-SA using TSP with the following Parameters:
      MaxIt: 10
   MaxSubIt: 9
       nAnt: 5
          Q: 2
       tau0: 0.0176
         T0: 0.2500
      Alpha: 1
       Beta: 0.7000
        Rho: 0.0100
       nPop: 3
      nMove: 5

Iteration 1: Best Cost = 1411.4124
Iteration 2: Best Cost = 1206.4313
Iteration 3: Best Cost = 1044.1735
Iteration 4: Best Cost = 1044.1735
Iteration 5: Best Cost = 972.774
Iteration 6: Best Cost = 896.4054
Iteration 7: Best Cost = 838.2395
Iteration 8: Best Cost = 838.2395
Iteration 9: Best Cost = 838.2395
Iteration 10: Best Cost = 795.7369
```

Figure 7 - Makespan solved TSP with Hybrid ACO-SA

In the Execution Time of Solved TSP with Hybrid ACO-SA algorithms as Figure 8 shown, the cities problem of X Cities and Y Cities in values using MATLAB implemented are :-

X = [82 91 12 92 63 9 28 55 96 97 15 98 96 49 80 14 42 92 80 96];

Y = [66 3 85 94 68 76 75 39 66 17 71 3 27 4 9 83 70 32 95 3];

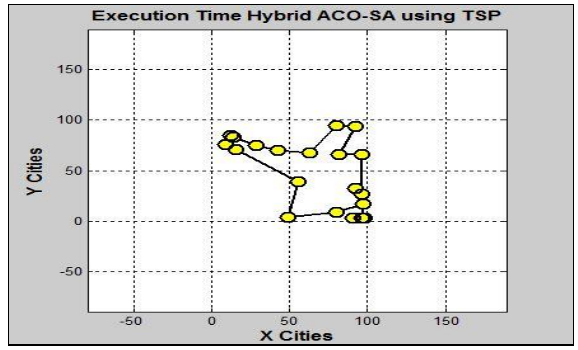


Figure 8 - Execution Time solved TSP with Hybrid ACO-SA

In the representation of the graph, there is a Calculate Cost of Solved TSP with Hybrid ACO-SA algorithms as Figure 9 as shown. The Inputs parameters number of iterations was set to 10 (MaxIt = 10) simultaneously with the graph and the Output parameters Best Cost Values are Solutions (Result Value) in the graph.

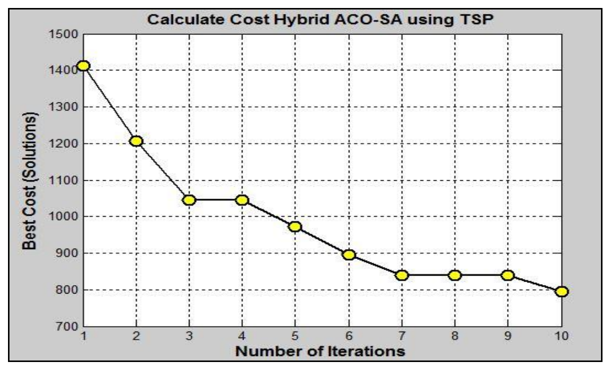


Figure 9 - Calculate Cost Solved TSP with Hybrid ACO-SA

In the results of Values Parameters with Solved TSP using Hybrid ACO-SA algorithms as Table 6 as shown, the values of all parameters inputs are different values and all parameters outputs are different values of the Best Cost (Solution).

Table 6 - Results of Values Parameters with Solved TSP using Hybrid ACO-SA

Input											**Output**
MaxIt	***MaxSubIt***	***nAnt***	***Q***	***tau0***	***α***	***β***	***ρ***	***T0***	***nPop***	***nMov***	***Best Cost***
50	10	15	0.1	8.7940	0.99	2	0.45	0.02	2	1	736.8629
70	50	30	0.8	0.0070	3	2	0.46	0.9	3	3	773.9437
100	60	30	0.9	0.0079	0.3	1	0.5	0.6	4	2	724.076
200	100	55	0.07	6.1558	1	3	0.05	1	1	4	724.076
60	50	35	1	0.0088	2	0.1	0.56	0.15	2	4	871.4839
40	20	5	2	0.0176	0.2	3	0.96	0.95	2	3	724.076
140	90	65	3	0.0264	2.2	0.3	1.96	2	5	6	728.2784
20	10	7	1	0.0088	1	2	0.05	2	4	4	848.3315
80	50	30	4	0.0352	2	3	0.7	0.1	6	10	724.076
90	30	15	2	0.0176	1	0.7	0.01	0.25	3	5	781.5954

5. Conclusion and Future Scope

The Population Based Metaheuristics handle in all iterations with a set (i.e. a population) of best cost solutions rather than instead with a single best cost solution. The Metaheuristic Hybrids is consisting of the importantly many difficult than classical strict methods in which is used to the requirement specification of the development time and tuning attempted to be to endeavour may be substantially higher than the applying a straightforward out-of-the-box strategy necessary of the system. The key idea or intellect of the hybrids techniques in which is used to a many difficult hybrid algorithm or metaheuristic hybrid does not automatically execute operation to be performed better than an adequate equal design hybrid systems and appropriate approximate optimization tuning is development time compulsory and the attempted was to effort increases with the system's parameters are complexity its reduces operations performance to be run-time. Thence, a designing of the hybrid systems for NP-hard optimization problems is the presenting a natural action or activity. For the NP-hard optimization problems, as many real world problems are the traditional mathematical formulation methods proof to be unequal; the main understanding that is their computational expensiveness, even infeasibility, in the returning optimal best cost solutions. Rather than Metaheuristics, even if do not guarantee to undertake to find out the optimal solution(s) and near optimal solution(s) in a highly reduced performed to be executed computational time. For the many NP-hard optimization problems, this form of solving is a good sufficient manner.

Based on the solved TSP with ACO Algorithms, it can be reasoned out that the quality of best cost solutions depends upon the number of ants and the lower number of ants evolves the individual stage (bit-by-bit) to change the route much accurate and faster. The higher number of ants in population based technique causes effort the higher accumulation of local pheromone updates on the edges, and therefore an individual bit-by-bit keeps the route with higher absorption of pheromone updated with a high probability distribution function.

Based on the solved TSP with SA Algorithms, it is a metaheuristic inspired by annealing process of high temperature rate and SA begins with a first best cost solution at higher temperature, where the exchanges are accepted with higher probability density functions, so that the systematic consideration capability of the algorithm diversified is high and the local search space can be explored widely. As the algorithm to be proceeding to continue to run, the temperature decreases bit-by-bit (step-by-step), like the annealing process and the acceptance probability density function of non-successful movements, decrease.

Based on the solved TSP with Hybrid ACO-SA Algorithms, the main features of the strict ACO component of ACO-SA, which mainly different from the algorithms in the following expressions:-

a) It uses a new local pheromone update (LPU) rules in whose of the pheromone updated values is autonomous of the problem cost (or quality) probability distribution function and they are delimited within at random select and fixed interval.

b) It uses a new global pheromone update (GPU) rules in which makes the pheromone values asymptotically increase (decrease) to the upper (lower) delimited without requiring any explicit cut-off as in the Ant Colony Optimization (ACO).

c) It uses a diversification strategy based depends upon the impermanent disturbance system of the pheromone values executed by a local pheromone update (LPU) ruled within any individual values maximum iteration.

References

[1] Divya Gupta, "*Solving TSP using various Meta-heuristic Algorithms*", International Journal of Engineering Science, Vol. 1, pp. 26-30, 2013

[2] Anas Arram, Masri Ayob, "*Comparative Study of Metaheuristic Approaches for solving Traveling Salesman Problems*", Asian Journal of Applied Sciences, Vol. 7, pp. 662-670, 2014

[3] Gunther R. Raidl, Jakob Puchinger, "*Metaheuristic Hybrids*", International Series in Operations Research and Management Science, Vol. 146, pp. 469-496, 2010

[4] Teodor Gabriel Crainic, Michel Toulouse, "*Parallel Meta-Heuristics*", International Series in Operations Research and Management Science, Vol. 146, pp. 497-541, 2010

[5] Marco Dorigo, Luca Maria Gambardella, "*Ant Colony System: A cooperative learning approach to the Traveling Salesman Problem*", IEEE Transactions on Evolutionary Computation, Vol. 1, pp. 53-66, 1997

[6] Marcus Randall, Andrew Lewis, "*A Parallel Implementation of ACO*", Journal of Parallel and Distributed Computing, Vol. 62, pp. 1421-1432, 2002

[7] Musa Peker, Baha Sen, "*An efficient solving of the TSP: The Ant Colony System having parameters optimized by the Taguchi Method*", Turkish Journal of Electrical Engineering and Computer Sciences, Vol. 21, pp. 2015-2036, 2013

[8] Pan Junjie, Wang Dingwei, "*An ACO Algorithm for Multiple Traveling Salesman Problem*", Proceedings of the First International Conference on Innovative Computing, Information and Control, Vol. 1, pp. 210-213, 2006

[9] Shu-Chuan Chu, John F. Roddick, "*Ant Colony System with communication strategies*", Information Sciences, Vol. 167, pp. 63-76, 2004

[10] Zar Chi Su Su Hlaing, May Aye Khine, "*Solving Traveling Salesman Problem by using Improved ACO Algorithm*", International Journal of Information and Education Technology, Vol. 1, pp. 404-409, 2011

[11] Chiranjib Patra, Pratyush, "*Vector Ant Colony Optimization and Traveling Salesman Problem*", Computer Science and Information Technology, Vol. 4, pp. 31-39, 2014

[12] Luca Maria Gambardella, Macro Dorigo, "*Solving Symmetric and Asymmetric Traveling Salesman Problems by Ant Colonies*", IEEE Conference on Evolutionary Computation, Vol. 6, pp. 622-627, 1996

[13] Dasaradh R. Mallampati, Pooja P. Mutalik, "*A Parallel Multi-Phase Implementation of Simulated Annealing for the Traveling Salesman Problem*", Proceedings of the Sixth Distributed Memory Computing Conference, Vol. 6, pp. 488-491, 1991

[14] Husamettin Bayram, Ramazan Sahin, "*A New Simulated Annealing Approach for Traveling Salesman Problem*", Mathematical and Computational Applications, Vol. 18, pp. 313-322, 2013

[15] Parham Azimi, Ramtin Rooeinfar, "*A New Hybrid Parallel Simulated Annealing Algorithm for Traveling Salesman Problem with Transporters*", Journal of Optimization in Industrial Engineering, Vol. 15, pp. 1-13, 2014

[16] Nitesh M. Sureja, Bharat V. Chawda, "*Random Traveling Salesman Problem*", International Journal of Emerging Technology and Advanced Engineering, Vol. 2, pp. 621-624, 2012

[17] David Bookstaber, "*Simulated Annealing for Traveling Salesman Problem*", Springer, Vol. 1, pp. 1-9, 1997

[18] M.A. Rufai, R.M. Alabison, "*Solution to the Traveling Salesperson Problem using Simulated Annealing Algorithm*", Electronic Journal of Mathematical Analysis and Applications, Vol. 5, pp. 135-142, 2017

[19] Lin Xiong, Shunxin Li, "*Solving Traveling Salesman Problem Based on the Improved SA Algorithm with Sequential Access Restrictions*", Advances in Intelligent Systems Research, Vol. 130, pp. 610-616, 2016

[20] Zicheng Wang, Xuitang Geng, "*An Effective Simulated Annealing Algorithm for solving the Traveling Salesman Problem*", Journal of Computational and Theoretical Nanoscience, Vol. 6, pp. 1680-1686, 2009

[21] Seyed Ahmad Sheibat Alhamdy, Mohammad Majdara, "*Solving Traveling Salesman Problem using Ant Colony Optimization Algorithm and comparing with Tabu Search, Simulated Annealing and Genetic Algorithm*", Journal of Applied Sciences Research, Vol. 8, pp. 434-440, 2012

[22] Adewole A.P., Otubamowo K., "*A Comparative Study of Simulated Annealing and Genetic Algorithm for solving the Traveling Salesman Problem*", International Journal of Applied Information Systems, Vol. 4, pp. 6-12, 2012

[23] Mateusz Borowski, Rafal Machon, "*A Modified Traveling Salesman Problem: Algorithms and Experimentation System*", Journal of Intelligent Computing, Vol. 6, pp. 59-67, 2015

[24] Hashim Ali, Yasir Shah, "*Solving Traveling Salesman Problem through Optimization Techniques using Genetic Algorithm and Ant Colony Optimization*", Journal of Applied Environmental and Biological Sciences, Vol. 6, pp. 55-62, 2016

[25] M. Fikret Ercan, Xiang Li, "*Particle Swarm Optimization and its Hybrid*", International Journal of Computer and Communication Engineering, Vol. 2, pp. 52-55, 2013

[26] Muhammed Basheer Jasser, Mohamad Sarmini, "*Ant Colony Optimization and a variation of Bee Colony Optimization (BCO) in solving Traveling Salesman Problem, a comparative study*", International Journal of Computer Applications, Vol. 96, pp. 1-8, 2014

[27] Younis Elhaddad, Omar Sallabi, "*A New Hybrid Genetic and Simulated Annealing Algorithm to solve the Traveling Salesman Problem*", Proceedings of the world congress on Engineering, Vol. 1, pp. 11-14, 2010

[28] Huiling Bao, "*A Two-Phase Hybrid Optimization Algorithm for solving complex optimization problem*", International Journal of Smart Home, Vol. 9, pp. 27-36, 2015

[29] Elena Simona, Nicoara, "*Population Based Metaheuristics : A Comparative Analysis*", International Journal of Science and Engineering Investigations, Vol. 1, pp. 84-88, 2012

[30] Rajesh Matai, Surya Prakash Singh, Murari Lal Mittal, "*Traveling Salesman Problem: An Overview of Applications, Formulations, and Solution Approaches*", Traveling Salesman Problem, Theory and Applications, pp. 1-24, 2010

[31] Ivan Brezina Jr., Zuzana Cickova, "*Solving the Traveling Salesman Problem Using Ant Colony Optimization*", Management Information Systems, Vol. 6, pp. 010-014, 2011

[32] Zar Chi Su Su Hlaing, May Aye Khine, "*An Ant Colony Optimization Algorithm for Solving Traveling Salesman Problem*", International Conference on Information Communication and Management, Vol. 16, pp. 54-59, 2011

[33] Marco Dorigo, "*Ant Algorithms Solve Difficult Optimization Problems*", Springer, Vol. 1, pp. 11-22, 2001

[34] V. Cerny, "*Thermodynamical Approach to Traveling Salesman Problem: An Efficient Simulated Annealing*", Journal of Optimization Theory and Applications, Vol. 45, pp. 41-51, 1985

[35] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, "*Optimization by Simulated Annealing*", Science, Vol. 220, pp. 671-680, 1983

Authors Profile

"Supreet Kaur" received her under-graduate degree from Amritsar College of Engineering and Technology (ACET), Amritsar (Punjab), India and currently she is the candidate of M.Tech. (Computer Science and Engineering) at Department of Computer Engineering and Technology (CET), Guru Nanak Dev University (GNDU), Amritsar (Punjab), India. Her interests are in Cloud Computing, Artificial Intelligence and Nature Inspired Metaheuristic Algorithms.

"Kiranbir Kaur" is currently working as an Assistant Professor, Department of Computer Engineering and Technology (CET), Guru Nanak Dev University (GNDU), Amritsar (Punjab), India. Currently, she is pursuing Ph.D. at Department of Computer Engineering and Technology (CET), Guru Nanak Dev University (GNDU), Amritsar (Punjab), India. She is UGC-NET qualified. Her interests are in Cloud Computing.

