Optimization of Ant Algorithms: A Meta-Heuristic Approach to Complex Problems

Falak Singhal
Student, Dept. of Computer Science,
Guru TeghBahadur Institute of Technology, GGSIPU
Delhi, India

Amrit Pal Singh
Asst. Professor, Dept. of Computer Science
Guru TeghBahadur Institute of Technology, GGSIPU
Delhi, India

***Abstract*—this paper studies three algorithms Ant system, Ant Colony System and Min-Max Ant System that are applied to travelling salesman problem Berlin52 benchmark function. Three comprehensive experimental studies that involve simulations of algorithms at author's machine have been carried out that investigates the behavior of these algorithms & compare their relative strengths and weaknesses. The first study establishes the optimal number of computational agents which will result in best solution of the benchmark function. The second study emphasizes on finding the minimum number of tours allowed for each ant such that number of iteration required for that solution are minimum. The experiments also verify the established optimal relationship between alpha and beta parameter of the Ant System given by Marco Dorigo.**

Keywords—Ant System,Ant Colony System,Min-Max Ant System, Travelling Salesman Problem, optimization

I. INTRODUCTION

Meta heuristics can be efficiently used in order to provide a sufficiently good solution to an optimization problem. Swarm Intelligence, which is a sub-category of meta-heuristics, refers to the behavior of decentralized and self-organized systems that may be artificial or natural. Ant algorithms come under swarm intelligence.

These algorithms use probabilistic techniques to solve computation problems that can be applied to finding optimal paths through graphs. The first of this class of algorithms The Ant System (AS) was proposed by Marco Dorigo in his research paper in 1991[1].This algorithm introduces the basic foraging behavior of ants that uses branching factor to choose between various paths. This technique has seen many variants since its inception.

The main disadvantage of AS is its high run time [2,3]. Thus a new algorithm was introduced called Ant Colony System (ACS). The main differences between ACS and AS are the trail update schemes and choice of next node. In ACS only the global best ant is allowed to update the trails.

Experiments with AS indicated that a more greedy approach improves the performance of the algorithm and hence lead to the development of Min-Max Ant System (MMAS) which differs from AS in three major aspects: only a single ant is allowed to update the trail in each iteration, the trail intensities are bounded in between τ_{min} and τ_{max} and the trails are initialized in a specific way giving the solution space a specific interpretation.

The vehicle routing problem is a classic combinatorial optimization problem, which is NP-hard. It addresses the problem of a salesman who has to travel 'n' cities and return back to the starting city and should cover minimum total distance while doing so.

TSP has to comply with certain constraints that are, the salesman has to start at an initial city $\boldsymbol{x}$ and he must visit each city exactly once and must return back to the starting city $\boldsymbol{x}$. The resulting route should be optimal that is it should incur minimum cost.

II. METHEDOLOGY USED

A. *The Ant System*

Ant System [2,3,4] algorithm uses a combination of both a priori and post priori information while computing an optimal solution. Here an ant is a computational agent whose behavior is stochastic. The ant inherently wanders randomly in search of food. Upon finding the food it returns to the colony laying down a pheromone trail in the path. Once other ants find this trail they are more likely to follow the path instead of wandering and will reinforce the trail in the process. But the pheromone evaporates with time and the trail become fainter over time.

The idea is that a shorter path will get marched upon by more ants than a longer one and hence balances the deposition and evaporation of pheromone trail and persists. The longer paths on the other hand vanish over time. So in principle the system should converge to a shortest possible route speaking in the terms of graph [13].

1) *Construction of Tour*

Defining the system in mathematical terms let an ant move from state $\boldsymbol{x}$ to $\boldsymbol{y}$in each iteration corresponding to a partial solution. Each ant $\boldsymbol{k}$ then contributes to its set of partial solution given as A_k(x). Then the probability p_{xy}^{k} , of ant, moving from state $\boldsymbol{x}$ to $\boldsymbol{y}$ is a function of two values :

a) The attractiveness of the route η_{xy} as computed using some heuristics indicating the a priori information

b) And the trail level τ_{xy} *of the move* indicating the pheromone level on the path, a post priori information.

In general$\boldsymbol{k}^{\text{th}}$ ant move from sate $\boldsymbol{x}$ to state $\boldsymbol{y}$ with the probability [14]

$$p_{xy}^{k} = \frac{(\tau_{xy}^{\alpha})(\eta_{xy}^{\beta})}{\Sigma_{\boldsymbol{y} \,\epsilon\, allowed \quad \boldsymbol{y}} (\tau_{xy}^{\alpha})(\eta_{xy}^{\beta})} \qquad (1)$$

Where 0≤α and β≥ 1 are constants that control the influence of τ_{xy} and η_{xy} respectively.
The desirability of the move $\boldsymbol{xy}$ is typically $\frac{1}{d_{\boldsymbol{xy}}}$ where $d_{\boldsymbol{xy}}$ is the distance between states $\boldsymbol{x}$ and $\boldsymbol{y}$.The larger the $d_{\boldsymbol{xy}}$ the smaller the attractiveness of the route.

2) *Trail Update in AS*

Once the ant has completed its tour the trail strengths are updated as

$$\tau_{xy}{}^{new} \leftarrow \rho.\, \tau_{xy}{}^{old} + \sum_{k=1}^{n} \Delta\tau_{xy}^{k} \qquad (2)$$

Where ρ is the pheromone persistence coefficient, τ_{xy} is the pheromone deposited for the transition $\boldsymbol{xy}$ and $\Delta\tau_{xy}^{k}$ is amount of pheromone deposited by $\boldsymbol{k}^{\text{th}}$ and typically given by

$\Delta\tau_{xy}^{k} = \frac{Q}{L_k}$ if $\boldsymbol{k}^{\text{th}}$ ant uses the curve ($\boldsymbol{x}$,), 0 otherwise.

Where Q is a constant quantity of pheromone laid and L_k is the cost of $\boldsymbol{k}^{\text{th}}$ ant tour.

B. *The Ant Colony System*

The AS was shown as an exemplary approach to solve the TSP which gave satisfactory results. However due to high computational time, modifications to the original algorithms were proposed. Based on such variant called Ant-Q algorithm [5] some authors proposed Ant Colony System [2].

The Key aspect of difference is the pheromone update heuristics in which only the global best ant is allowed to update the tail. So equation (2) is modified and the trail is chosen with the probability of (1-ρ) and the selection is made as the equation (1).

1) *Trail Update in ACS*

$$\tau_{xy} \leftarrow (1\text{-}\rho).\, \tau_{xy} + \sum_{k=1}^{n} \Delta\tau_{xy}^{k} \qquad (3)$$

This rule results allows for a balance between exploration and exploitation strategies.

C. *The Min-Max Ant System*

Min-Max Ant System [7] is a greedier approach than both of the above algorithms. It was observed that for larger systems without a greedy approach the above algorithms degrades rapidly in performance and converge to a less optimal solution. To limit the stagnation of search the trail update rule is modified and the trail level is bounded in interval (τ_{min} ,τ_{max})[3,12].

1) *Trail Update in Min-Max Ant System*

In MMAS two types of updates can be considered. One, in which after each iteration the best ant updates the trail level and second, in which the global best found solution is used to update the trails. Both of these approaches differ in greediness. The modified trail update rule as per second approach is given as

$$\tau_{xy}(i) \leftarrow (1\text{-}\rho).\, \tau_{xy}\;(i-1) + \Delta\tau_{xy}^{best} \qquad (4)$$

Where $\Delta\tau_{xy}^{best} = \frac{1}{L_{best}}$, L_{best} is the tour corresponding to global best found solution in i^{th}iteration.

III. TRAVELLING SALESMAN PROBLEM

The traveling salesman problem is the problem faced by a salesman who, starting from his home town, wants to find the shortest possible trip through a given set of customer cities, visiting each city once before finally returning home. The TSP is a NP-hard in combinatorial optimization [8].

The traveling salesman problem (TSP) is then the general problem of finding a minimum cost Hamiltonian circuit in a weighted graph, where a Hamiltonian circuit is a closed walk (a tour) visiting each node of graph G(V,E) exactly once.

In order to solve the TSP, one needs to find a route such that the total distance travelled by the salesman is minimum. Hence the input needed to solve such a problem is-

a) A matrix of 2-d coordinates of various locations in a map.

b) Thenumber of cities.

The Starting city can be selected randomly or can be provided as input.

After taking the input, the distance matrix is calculated that gives the Distance, d_{ij}between the ith and the jth city. This is provided by the value corresponding to the ith row and jth column.
The optimal solution then to the TSP is a permutation Ω of the node indices {1, 2…..n} such that the length f(Ω) is minimal [9] , where f(Ω) is given by

$$f(\Omega)=\sum_{i=1}^{n-1} [d_{\Omega(i)\,\Omega(i+1)}] + d_{\Omega(n)\,\Omega(1)} \quad (5)$$

Travelling Salesman problem has been classified in two categories namely Symmetric and Asymmetric. In the symmetric TSP, the distance between two cities is the same in each opposite direction, forming an undirected graph. This symmetry halves the number of the possible solutions of the graph [11].

In asymmetric TSP, given two nodes (or cities) path may not exist in both the directions or may have different values (or weights) hence forming a directed weighted graph.

IV. EXPERIMENTAL STUDY

The author implemented the three ant algorithms (Ant System, Ant Colony Optimization, Min-Max Ant) and investigated their relative strengths and weaknesses by experimentation. Simulations were run to collect statistical data for this purpose.

A. Variation of optimal solution as a function of polpulation size

The parameters considered here are those that affect directly or indirectly the computation of the best tour resulted by these two algorithms include

*1)*n: The number of computational agents (Ants) that simultaneously explore the problem space and yield their respective partial solution sets.
*2)*k: The number of tours each agent is allowed to perform for completing its own set of optimal solution.

The number n of the ants is varied in an increasing fashion up to the close proximity of number of cities used in the TSP problem space. During this time the number if iterations (k) are kept to a constant value of 100. In this study author considered berlin52 benchmark function from TSP Library for our experimentation.

To compare the two models simulations were run to yield the optimal number of computational agent for a fixed number of iterations.

TABLE I. VARIATION OF OPTIMAL SOLUTION AS A FUNCTION OF POPULATION SIZE

Algorithm		Ant System	Ant Colony System	Min-Max Ant System
S.no.	**Population Size**	**Best Solution**	**Best Solution**	**Best Solution**
1	5	9380	8423	8132
2	10	8621	7734	8077
3	20	8325	7657	8038
4	30	8307	7542	7662
5	40	8203	7542	7662
6	45	8043	7542	7689
7	50	7895	7542	7662
8	51	7938	7542	7596
9	52	7663	7542	7859
10	55	7790	7547	7910

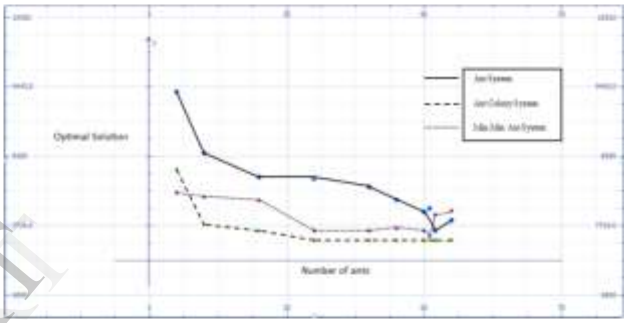

Figure 1. Number of Ants vs. Optimal Solution

The graphical plots clearly show that the algorithms perform better and yield a more optimal solution when the number of computational agents reaches the number of nodes in the problem domain.

B. Variation of number of iterations to optimal solution as a function of number of tours

This study involves finding how the algorithms behave when the number of tours allowed for one ant is varied.

TABLE II. NUMBER OF TOURS VS. NUMBER OF ITERATIONS TILL OPTIMAL SOLUTION

Algorithm		AS ▬ ▬	ACS ▪▪▪▪	MMAS▬
S.No.	Tours	No. of iterations	No. of iterations	No. of iterations
1	10	4	3	8
2	20	3	3	10
3	30	2	4	27
4	40	2	2	24
5	50	2	3	47
6	60	9	4	8
7	70	9	10	47
8	80	11	12	19
9	90	12	15	60
10	100	15	16	35

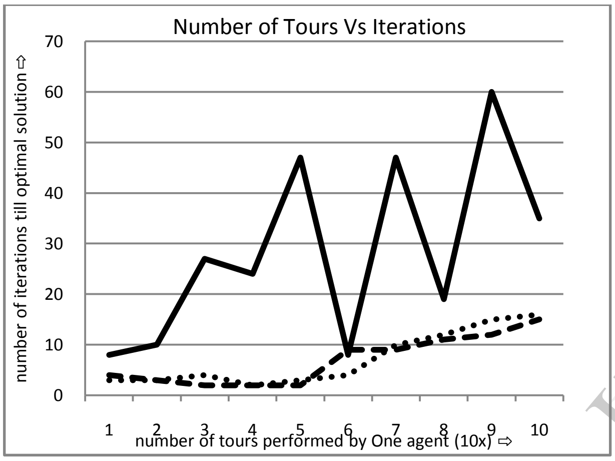

Figure 2. Number of tours vs. Number of iterations till optimal solution

From this experimental study it was observed that the best number of tours (k) allowed for each computational agent to minimize the number of iteration lies within the close proximity of the number of nodes (n) for Ant System and Ant Colony System. When the number of tours k is increased till the value n, the number of iterations required for optimal solution decreases slowly, attains a minimum value at k=n and then increases sharply.

However the variation of number of iterations with number of tours for Min Max Ant was different. It was observed that for the optimal number of ants, i.e. n=52 , the number of computation till optimal solutions are highest. For the entire variation period the number of iterations fluctuates greatly between minimum and maximum value.

C. Variation of optimal solution as a function of alpha and beta

The parameters considered here are those that affect directly the computation of the best tour resulted by the Ant System algorithm include

1) Alpha α: the relative importance of the trail, α > 0
2) Beta β: the relative importance of the visibility, β > 0

To compare the three models we first experimentally determined the parameters best values for each algorithm, and then we ran each algorithm five times using the best parameters set. The results of the study are as follows

TABLE III. VARIATION OF OPTIMAL SOLUTION AS A FUNCTION OF ALPHA AND BETA

Algorithm	**Exp. No.**	**α**	**β**	**Tour Length**	**Average Value**	Best Value	% Error
Ant System (Berlin52No. of ants = 52, No. of Tours =100)	1(α=β)	3	3	8092	8099.4	8021	7.3
		3	3	8058			
		3	3	8230			
		3	3	8021			
		3	3	8096			
	2(α<β)	1	5	7872	7748.2	7674	2.6
		1	5	7674			
		1	5	7794			
		1	5	7722			
		1	5	7679			
	3(α>β)	3	1	8837	8939.8	8606	18.4
		3	1	9191			
		3	1	8606			
		3	1	8782			
		3	1	9283			

V. RESULTS OF EXPERIMENTAL STUDIES

A. Number of ants vs. best found solutions

It was observed that as the number of computational agents 'n' increases both the algorithms converge to an optimal solution. As 'n' approaches to the number of nodes in the problem domain the change in solution (best found tour length) for both the algorithms is abrupt.

However for Ant Colony System optimal solution attains a constant value till number of ants are in proximity of number of nodes (here 52) in the problem. This is in contrast with the Ant System algorithm in which the value for optimal solution decreases till a minimum value. Moreover it was found that the Ant Colony System algorithm gives a better optimal solution for the same number of ants.

For the same set of conditions the reported optimal solution for berlin52 benchmark problem by Ant Colony System was 7542 by the simulation software which is equal to the established optimal value by TSP Library in comparison to Ant System which reported an optimal solution of 7663 which has +1.604 % error and Min Max Ant solution 7596 with +0.71 % error.

TABLE IV. ACCURACY OF ALGORITHMS

	AS	ACS	MMAS
Calculated Value	7663	7542	7596
Established Value	7542	7542	7542
% Error	+1.60%	0%	+0.71%

B. Variation of number of iterations to optimal solution as a function of number of tours

For the algorithms AS and ACS the number of iterations required to an optimal solution decreases linearly as the number of tours by each agent (k) increases and approaches the number of nodes in the problem domain.

It was observed that the optimal number of tours each agent is allowed to perform such that the iterations are minimum lies within the close proximity of number of nodes in the problem domain for these algorithms.

For Ant System the minimum observed iterations were 2 when the number of tour 'k' approaches 'n' and for the Ant Colony System the minimum number of iterations observed for an optimal solution were 3.

The fact that the Ant System algorithm outperforms the ACS algorithm can be attributed to the accuracy of ACS which demands more computation which is an acceptable tradeoff for ACS as it produces an established optimal solution for berlin52 problem.

For both the algorithms the iterations to optimal solution increases exponentially when the number of tours 'k' is increased beyond 'n'.

However this was not the case with Min Max Ant System. The number of iterations in this scenario fluctuates between a minimum and maximum value of (8, 47).

C. Variation of optimal solution as a function of alpha and beta

For the Ant System algorithm it was observed that the optimal solution occurs when the value of **α = 1 & β = 5.** This was in agreement of the results found by M.Dorigo in his empirical study of ant system algorithm. For both the cases **(α=β) & (α>β)**, large deviations of + 7.3 % and 18.4 % respectively were observed in the average solution from the established solution of 7547 for berlin52 problem.

The +2.6 % error in **(α=1,β=5)** can be attributed to random errors that can be incurred due to runtime environment limitations etc.

Also the graphical study of tour length vs. iterations while conducting the experiments shows that the initial optimal solution for the case **(α>β)** are way off the optimal solution. Hence in this case the computations towards an optimal solution are large as compared to other two cases. This effect can be inferred from the steep drop in tour length vs. iterations graph during initial iterations.

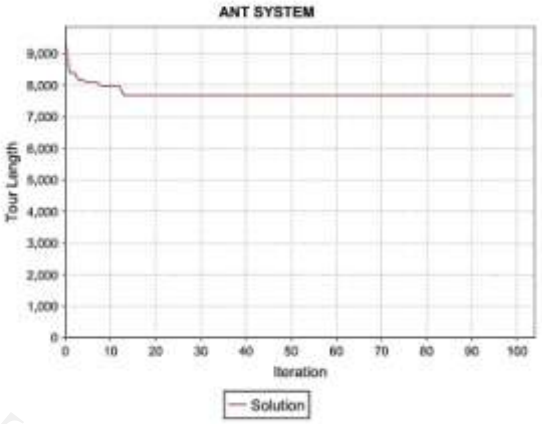

Figure 3. Iterations of AS when α < β

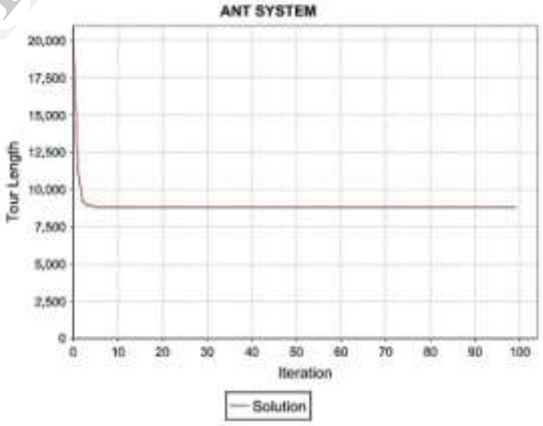

Figure 4. Iterations of AS when α > β

The experiment was conclusive of that the best combination of alpha and beta is **(α=1, β=5)** for Ant System Algorithm.

The case of **(α>β) is the worst** which resulted in not only large computational complexity but a large deviation of result from established value also.

VI. CONCLUSION AND FUTURE WORK

The studies were conclusive of the result that both ACS and MMAS performs better than originally proposed AS algorithm. The optimal solution occurs when the number of computational agents is equal to the number of nodes in the problem domain. Also the number of iterations to optimal solution were minimum when the number of tours performed by each ant are in close proximity of number of nodes for AS and ACS algorithms. The optimal values of alpha and beta parameters are experimentally found out to be 1 and 5 respectively.

The future work of this study involves the behavior of Eigen ant and more recent algorithms in comparison to the initial algorithms like AS and ACS. Other artificial intelligence and swarm intelligence algorithms like Artificial Bee colony etc. can be studied and further analyzed.

REFERENCES

[1] M.Dorigo,"Ant colony optimization: artificial ants as computational intelligence technique",IEEE Computational Intelligence Magazine,2006.

[2] Alberto Colorni, Marco Dorigo,and Vittorio Maniezzo. "Distributed optimizationbyant colonies", Elsevier Publishing, 1991.

[3] Marco Dorigo. "Optimization, learning,and natural algorithms". PhDthesis,Politecnico di Milano, 1992.

[4] M. Dorigo, T. Stützle. "The ant colony optimization metaheuristic: algorithms, applications and advances." In F. Glover and G. Kochenberger, editors, Handbook of Metaheuristics . Kluwer Academic Publishers, 2002

[5] Luca Gambardella and Marco Dorigo. "Solving symmetric asymmetric TSPs by ant colonies." In IEEE Conference on Evolutionary Computation (ICEC'96), 1996

[6] Luca M. Gambardella and Marco Dorigo. "Ant-Q: A reinforcement learning approach to the traveling salesman problem." In Proceedings of the Twelfth International Conference on Machine Learning, page 252. Morgan Kaufmann, 1995.

[7] Thomas Stutzle and HolgerHoos, "Improvements on ant-system: introducing MAX-MIN ant system", Technical University of Darmstadt,2012

[8] M.R.Garey and D.S.Johnson. "Computers and intractability: A guide to the theory of NP-completeness." in Freeman, San Francisco, CA, 1979.

[9] Marco Dorigo&Thomas Stutzle, "Ant colony optimization," in Library of Congress Publications,The MIT Press, Cambridge, Massachusetts London, England, 2002

[10] Aardal, K. I., van Hoesel, S. P. M., Koster, A. M. C. A., Mannino, C., &Sassano, A. "Models and solution techniques for the frequency assignment problem." Technical report 01-40, Konrad-Zuse-Zentrum,2002

[11] Bonabeau, E., Sobkowski, A., Theraulaz, G., &Deneubourg, J.-L.Adaptive task allocation inspired by a model of division of labour in social insects. In D. Lundha, B. Olsson, & A. Narayanan (Eds.), "Bio-computation and emergent computing" (pp. 36–45). Singapore, World Scientific Publishing, 1997

[12] Clark, P., & Boswell, R. Rule induction with CN2: Some recent improvements. In Proceedings of the European Working Session on Learning (EWSL-91), vol. 482 of Lecture Notes in "Artificial Intelligence" Berlin, Springer-Verlag,1991

[13] Deb, K. "Multi-objective optimization using evolutionary algorithms." Chichester, John Wiley & Sons Publications , UK, 2001

[14] Dorigo, M. "Optimization, learning and natural algorithms" [in Italian]. PhD thesis, Dipartimeno di Elettronica, Politecnico di Milano, Milan,1992.