

An Adaptive Ant System using Momentum Least Mean Square Algorithm

Abhishek Paul
ECE Department
Camellia Institute of Technology
Kolkata, India

Sumitra Mukhopadhyay
Institute of Radio Physics and
Electronics
University of Calcutta
Kolkata, India

ABSTRACT

In this paper, a novel model has been proposed for pheromone updation of the Ant-System, entitled as Momentum Adaptive Ant System (MAAS). MAAS exploits the properties of Adaptive Filters. The proposed algorithm is implemented using momentum-LMS (Least Mean Square) based algorithm. It imparts information about the previous occurrence of the system so as to keep the system active even in the region close to the minimum (i.e., minimum optimal) solution. MAAS modifies its properties in accordance to the requirement of surrounding realm and for the betterment of its performance in dynamic environment. The proposed algorithm overcomes stagnation and offers better searching capability. Also it helps the ants (i.e., co-operating agents) not to get stuck at local optima. The results of experimental study are well described and it establishes the usefulness of the new strategy. Proposed algorithm shows effective performance when applied to the Traveling Salesman Problem (TSP).

General Terms

Evolutionary Algorithm

Keywords

Ant System (AS), Momentum Least Mean Square (momentum-LMS) Algorithm, Momentum Adaptive Ant System (MAAS), Traveling Salesman Problem (TSP).

1. INTRODUCTION

Bio-inspired optimization plays a vital role in today's world and is very obliging in the study of searching optimal solution for any problem. Researchers are fervent in finding the most favorable solutions for different systems and problems with different constraints using bio-mimic optimization techniques. Till date different nature inspired algorithms are contributing a large in the field of optimization. Some of such widely used techniques are Genetic Algorithm (GA) [1], Evolutionary Strategies (ES) [2], Particle Swarm Optimization (PSO) [3], Ant Colony Optimization (ACO) [4], Artificial Bee Colony (ABC) Optimization [5], etc.

This paper is based on Ant Colony Optimization (ACO), which was first introduced by Dorigo as Ant System (AS) [6]. This technique mimics the food searching behavior of ants. Ant algorithm has been widely exploited to solve NP-hard combinatorial optimization (CO) problems. Traveling Salesman Problem (TSP) is one of the widely used benchmark problem in this domain. Being an effectual and dominating algorithm, there still exist shortcomings of consuming time, easy to stagnation, falling to local optima and slow convergence speed. Here, the algorithm is modified and a novel scheme of pheromone updating rules is introduced, by exploring the

properties of adaptive filter, in the domain of ant colony optimization. This novel proposal is based on momentum-LMS algorithm, entitled as **Momentum Adaptive Ant System (MAAS)**. The characteristics of each and individual ant system has been implemented in terms of an adaptive filter [8] and its performance is investigated by changing the correction function of the Least Mean Square (LMS) filter using Momentum Least Mean Square (LMS) algorithm. In this paper, the entire parametric adaptation has been done using Momentum Least Mean Square algorithm. The expediency of using momentum LMS algorithm in this domain is to make the Ants (co-operative agents) active even in the close vicinity of the minimum optimal solution. This accelerates the searching towards the optimal solution and helps the co-operating agents, not trapped in local minima. Hence, the search for optimal solution will be successfully explored. The past information of the pheromone updation makes the Ants not idle (i.e., not stagnate) and keeps on searching for the optimal result.

The reliability and the properties of any optimization technique are widely tested in literature by means of, Traveling Salesman Problem (TSP) [4] [6-7]. TSP is a paradigmatic NP-hard combinatorial optimization problem. In TSP, a salesman travels different cities (every city visited once) in a way that the total travelling distance is minimal. This is an important optimization problem to test the reliability of any proposed optimization algorithm and the proposed method has been tested using the same approach. Simulation studies and the comprehensive analysis demonstrate that an efficient naturally inspired model can be achieved by the proposed approach.

The paper is organized as follows: Section II represents the Traveling Salesman Problem (TSP). Section III describes the basics of Adaptive Filter and henceforth illustrates the Momentum Least Mean Square (MLMS) algorithm with which our proposed ant system has been implemented. Basics of the Ant System are explained in Section IV. In Section V, the newly designed ant system, entitled as Momentum Adaptive Ant System (MAAS), has been introduced and it is based on the pheromone adaptation inspired by the momentum LMS algorithm. Simulations and results are elaborately described in Section VI. And finally the paper concludes in Section VII.

2. TRAVELLING SALESMAN PROBLEM (TSP)

Traveling Salesman Problem is a well-known optimization problem, which is used to test the effectiveness of the optimization algorithms. It is to find the shortest tour distance of a closed tour graph. This problem is a NP-hard problem and researchers have an immense interest in it.

Traveling Salesman Problem: Given a set of cities representing a graph $G = (V, E (d_{ij}))$, where, $V =$ set of cities, $E =$ set of edges between cities and d_{ij} is the Euclidean distance between i^{th} and j^{th} city, it is the problem of finding the tour that visits each city of the graph exactly once and minimize the total distance travelled.

Traveling Salesman Problem discussed in this section is mainly of two types: *Symmetric TSP* and *Asymmetric TSP*. Let us consider a set of two cities i, j , the distance from i to j is d_{ij} and from j to i is d_{ji} . In *Symmetric TSP*, the distance remains same, i.e., $d_{ij} = d_{ji}$. But for *Asymmetric TSP* the above said distance is different, $d_{ij} \neq d_{ji}$. This suggests that only one path exists between two cities in *Symmetric TSP* and for *Asymmetric TSP* there exists more than one path. This makes *Symmetric TSP* easier to implement than its counterpart. In this paper, we will refer to *Symmetric TSP* for its simplicity and we call it simply 'TSP'.

The proposed algorithm in this paper is being tested using different TSP benchmark problems; such as oliver30, ulysse16 and others. These benchmark problems are available at TSPLIB benchmark library [9].

3. ADAPTIVE FILTER AND MOMENTUM LMS ALGORITHM

Filtering is a mean of signal processing in order to manipulate the information contained in the signal. A filter performs three basic information processing tasks. They are filtering, smoothing, and prediction. Adaptive Filter [8] is one such nonlinear filter and is attractive due to its reliability, accuracy, and flexibility.

The adaptive filter is primarily controlled by the cost function (error function), which determines the performance of the filter. Main usefulness of this filter is to reduce the cost function, so the required output approaches to its desired value.

This cost function is given by

$$e(n) = d(n) - d'(n) \quad (1)$$

The designed process and its modification require the minimization of the cost function.

The basic adaptive filter equation is given by

$$w(n+1) = w(n) + \Delta w \quad (2)$$

Where, Δw is the correction factor, and can be modified using different updation algorithm of the adaptive filter.

Different updation algorithms are formulated to minimize the correction factor Δw , to reduce the cost and also to ease the hardware implementation. In this paper, we use the well-known updation algorithm known as momentum-LMS algorithm that has been exploited for the proposed implementation.

3.1 Momentum LMS Algorithm

The Momentum LMS algorithm is a popular variant of Least Mean Square (LMS) algorithm and was first proposed by Proakis [10]. This derivative of LMS is useful where error explosion is a huge problem [11].

The updation rule for momentum LMS is

$$w(n+1) = w(n) + \mu e(n)x(n) + \alpha' [w(n) - w(n-1)] \quad (3)$$

Where, the correction factor, $\Delta w = \mu e(n)x(n)$, $\mu =$ step size which controls the gradient information of the updation, $e(n) =$ cost function, $x(n) =$ input signal or input function and α' , the momentum factor lies between $[-1, 1]$.

The momentum factor α' controls the gradient of the previous filter weight. It is seen from theoretical and simulation analysis that momentum LMS shows good convergence rate for $\alpha' > 0$ [11]. If the previous gradient is large then the additional gradient term will enforce the system toward its global optima. This algorithm offers good tracking capabilities and helps to accelerate the searching towards the optimal solution.

4. ANT SYSTEM

Ant Colony Optimization (ACO) [4] was first introduced by Marco Dorigo in his PhD thesis in 1992. ACO imitates the foraging behavior of real ants. We focus our attention to the basic Ant-System (AS) [6], which is the first successful technique to emphasize the swarm nature of the real ants. The properties of Ant System can be modeled from the real ant colonies. Ant Algorithms are useful in solving different problems such as traveling salesman problem (TSP) [4] [6-7], assignment problems [12], and scheduling problems [13].

The ant system in TSP helps a salesman to find out his global tour length out of n -cities, where m -ants dedicate themselves to find the optimal path. Here, we are considering a 2-D Euclidean TSP, where, d_{ij} is the Euclidean distance between the two cities i and j . $\tau_{ij}(t)$ is the intensity of trail on edge (i, j) at time t . During initialization, ants are placed on different cities with some positive initial value $\tau_{ij}(0)$. The pheromone updation formula [5] depends upon the inverse of the tour length L_k traversed by the k^{th} ant and the updation formula is defined as,

$$\tau_{ij}(t+n) = \rho \tau_{ij}(t) + \Delta \tau_{ij} \quad (4)$$

Where,

$$\Delta \tau_{ij} = \sum_{k=1}^m \Delta \tau_{ij}^k = \sum_{k=1}^m \frac{Q}{L_k} \quad (5)$$

Q is a constant, that represents the amount of pheromone the ant releases after going through the edge (i, j) ; and ρ ($0 \leq \rho < 1$) is the pheromone evaporation (decay) parameter, $(1-\rho)$ is the pheromone residual parameter.

During their tour, an ant k in city i will move to city j with the probability given by

$$P_{ij}^k = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{k \in \text{allowed}_k} [\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}; \text{ if } j \in \text{allowed}_k \quad (6)$$

$$= 0$$

Where, 'allowed_k' are the cities that are not visited, η_{ij} is the visibility factor and α, β are the control parameters for τ_{ij}, η_{ij} respectively.

The choice of Ant-System (AS) depends on the pheromone trail. Therefore, Marco Dorigo put forward three models of AS,

say, *ant-density*, *ant-quantity* and *ant-cycle* [6] model. *Ant-cycle* uses global information and it functions better than the other two models. Therefore, researchers adopt this model for their study on ACO.

5. MOMENTUM ADAPTIVE ANT SYSTEM (MAAS)

During the iterative process, it has been observed that a difference in the pheromone concentration gradient of the basic Ant-System leads the ants (co-operating agents) to fall in local optima. Therefore the concentration to specific edges increases, reducing the global searching capability of the ants. To solve this problem, we propose to fit the behavior of biological ant system in a structured framework of adaptive filter using momentum LMS algorithm, which will guarantee the searching capability and convergence by virtue of its property. Also, block diagram model is proposed in Figure 1. This may help us in the study of stability analysis and global convergence.

5.1 Momentum Adaptive Ant System (MAAS)

The main idea of our novel MAAS model is to emulate the characteristics of ant system using LMS algorithm with momentum adaptation, which includes recursion of previous pheromone trail.

The above mentioned AS can be thought of as a filter process as modeled in Figure 1 and the detailed representation of the proposed approach has been depicted. Initially, it is required to identify the inputs and outputs of the system. Here, the locations of cities are considered as inputs and best global tour length is the desired output. The parameters of the ant system are acclimatizing during the processing time. Therefore, the AS can be best represented in terms of adaptive filters rather than general filters. The adaptive process [8] of the filter is primarily controlled by the cost function (error function), which determines the performance of the filter.

After the initialization process of the algorithm, the system starts iteration. The optimal value is recorded as L_{min} in respective iteration. $ITime$ is the number of iterations. The optimal minimum tour length is recorded as $GMinL$ (Global Minimum Length) and the optimal iteration as $FRIT$ (Favorable Iteration). The pheromone on the optimal path is updated after all the ants completed their tours for each time. Before the pheromone update, the tour lengths TL is calculated for each ant in each iterations as given below,

$$TL = [L_1, L_2, \dots, L_k, \dots, L_m] \quad (7)$$

where, L_k is the tour length of k -th ant in a specific iteration. $L_{min}(t)$ is calculated as the minimum tour lengths of ants for respective iterations as given below,

$$L_{min}(t) = \min(L_1, L_2, \dots, L_k, \dots, L_m) = \min(TL) \quad (8)$$

$L_{min}(t)$ is compared with the previous best minimum tour length $L_{min}(t-1)$. The best minimum optimal desired tour length is stored as $L_{min}(t)$, given by (9).

$$L_{min}(t) = \text{minimum} [L_{min}(t), L_{min}(t-1)] \quad (9)$$

In ant-system, the pheromone update rule inversely depends on the tour length of the ants. Hence, the cost (error) function, as shown in equation (2) for each ant in MAAS is modeled as (10).

$$\text{Error}(k) = (1/L_{min}(k)) - (1/L_k) \quad (10)$$

5.2 Pheromone Correction factor, Total Tau (TAU)

The total tau (TAU) given by (11), is calculated having an analogy with the correction factor of LMS algorithm. Here, the gradient information μ is made dynamic as in (11) to control the pheromone concentration and ensures faster search speed.

$$TAU = \Delta\tau_{ij} = \sum_{k=1}^m [\mu \cdot TL(k) \cdot \text{Error}(k)] \quad (11)$$

The proposed equations are helpful in finding the optimal path, for the closed tour of the cities and do not let ants fall in local optima, as μ changes with iteration.

5.3 Modified Pheromone Update Rule

Pheromone update rule in (12) is the final proposed update rule. This proposed update rule uses the global information to improve the convergence speed.

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + TAU + \alpha'[\tau_{ij}(t) - \tau_{ij}(t-1)] \quad (12)$$

' α' ' is the momentum factor which modulates the previous pheromone trails. It shows that if the previous pheromone trail gradient becomes large, then the searching capability close to optimal minima increases and stagnation problem can be solved. Moreover, as the agents become active even close to minima, it prevents the ants not get stuck into local optima. The pheromone update rule finds the optimal solution quickly and improves the performance of the algorithm.

5.4 Schematic representation of the Momentum Adaptive Ant System (MAAS)

Figure 1 represents the detailed block diagram of the proposed MAAS. City (node) coordinates are the inputs to the system, which are used to calculate the total tour length. The proposed algorithm in this paper updates the TSP problem according to the adaptation equations from (4) to (12). These are detailed in the proposed block diagram representation. Pheromone update rule using (12) is generated using the output TAU and the feedback $\rho\tau_{ij}(t)$ and the momentum adaptation part. The μ , shown as the μ - block in Figure 1, is also proposed in (11). Also α' , the momentum constant, as in (12), is also detailed in Figure 1. The estimated error (cost) function given by (10) is the difference between the reciprocal of L_{min} and L_k . It is shown in Figure 1 as $\text{Error}(k)$. L_{min} is the output and is generated by passing the output of MAXTERM block to the INVERSE block. It is the desired output (Global minimum length). The desired optimal solution in the Figure 1 is dynamic in nature and is found from the ant's tour lengths, TL .

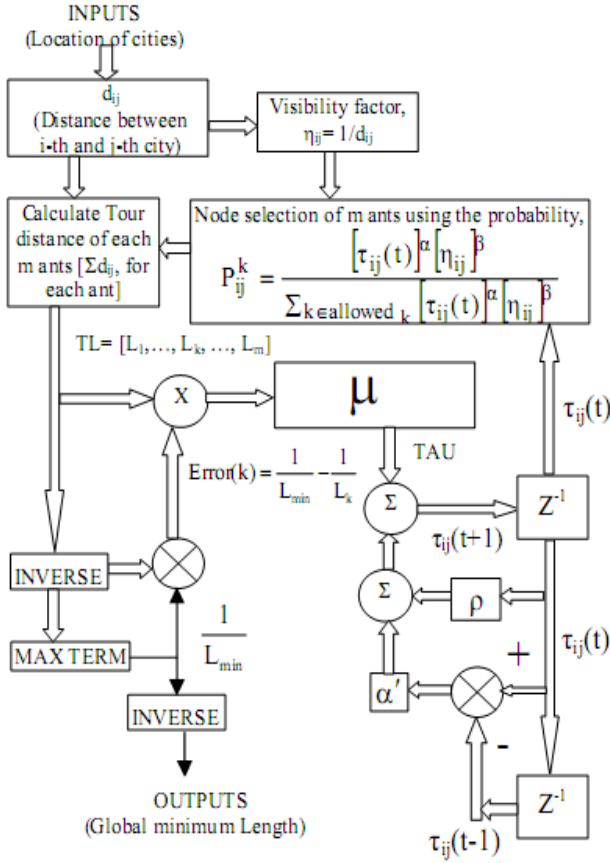


Fig1: Schematic Block diagram of Momentum Adaptive Ant System (MAAS)

5.5 Description of the MAAS Algorithm

- Given the city coordinates, initializing parameters α , β , ρ , Q , $\tau_{ij}(0)$, C and maximum iterations MaxITime .
- The m ants are randomly placed into n cities and the cities are put into the ants tabu list Tabu .
- Ants select the next city to visit according to (6), and the selected cities are put into their Tabu list correspondingly to complete the n -city tour.
- Calculate the shortest tour length, L_{\min} of the current iteration and all iterations when the ants complete its iteration. If $\text{ITime} > \text{MaxITime}$, then goto 8). Or else goto 5).
- The total tau, $\Delta\tau$ is calculated in the following steps:
 - ❖ If previous found $L_{\min} >$ present L_{\min} , then present L_{\min} is stored, else previous is retained. This is the desired optimal tour length, L_{\min} .
 - ❖ The cost (error) function is calculated in terms of distance inverse defined by (10) to find out the difference in the pheromone accumulation concentration for each ant in iteration.
 - ❖ Total tau, $\Delta\tau_{ij}$, is calculated using the new pheromone updation rule defined by (11).
- The pheromone is then updated by applying the improved global updation technique by (12).
- Set the tabu list, Tabu to Null, set $\text{ITime} = \text{ITime} + 1$, jump to 2).
- Output is the optimal minimum tour length (i.e., $\text{GMinL} = \text{Global Minimum Length}$) and the optimal iteration (i.e., $\text{FRIT} = \text{Favorable Iteration}$).

➤ End of Search.

6. EXPERIMENTAL STUDY

The simulation work has been carried out, based on the Ant-cycle model [6], as discussed in the previous sections. Ant-cycle uses global information and gives good solutions for the Ant-system. The simulation parameters are: $\alpha=0.5$; $\beta=5$; $\rho=0.5$; $\text{MaxITime}=6000$ (Maximum Iteration Time). The gradient factor in (11) is taken as $\mu=0.5$ and the momentum factor as $\alpha' = 0.1$ in (12). For simplicity, we have adopted number of ants equal to the number of cities (for e.g., number of ants = 22, for 22-city problem).

Experiments were carried out for oliver30.tsp, Ulysses16.tsp and Ulysses22.tsp problems and were averaged over 20 subsequent trials. Table 1 analyses the performance of the proposed Momentum Adaptive Ant System (MAAS). Table 2 gives the comparative study of the proposed MAAS algorithm, with other existing Ant algorithms. The novel Ant Algorithm proposed in this paper shows better results than the basic AS [6] while considering the performance on optimal solution, optimal iterations, and average solution. Compared with ACS [14] and a proposed algorithm from reference1 [14], the optimal best tour (also known as $\text{GMinL} = \text{Global Minimum Length}$) is same but the average tour and the worst tour has been improved. The optimal iteration (also called $\text{FRIT} = \text{Favorable Iteration}$) is comparable with the other two algorithms. Best results are marked in bold in Table 2.

Figure 2 shows the best tour path obtained by the proposed MAAS algorithm for Oliver30.tsp problem. Figure 3 gives the iterative best costs over the total run and the average node branching for the same problem. The best tour length obtained is 423.7406 ($\text{GMinL} = \text{Global Minimum Length}$) and on iteration 1555 ($\text{FRIT} = \text{Favorable Iteration}$), with average tour length of 423.9175. Figure 4 gives the variation of best iterative tour found at 20 successive run of the proposed algorithm. Figure 5, 6 and 7 gives the Average Node Branching and iterative best tour, best tour graph and best tour for 20 successive run for Ulysses22.tsp problem. Figure 8, 9 and 10 gives the same for Ulysses16.tsp problem. The TSP problems discussed here show effective and satisfactory results with the proposed MAAS algorithm. From Table 1 it is seen that the iteration time ($\text{FRIT} = \text{Favorable Iteration}$) for the best tour length are 1555, 2057, and 4830. So, time required to converge to the optimal (minimal) solution is large but the agents are still active, search for better optimal solution and keep on searching even close to optima. The time requirement is more, as ants (co-operating agents) searching ability increases even at the close vicinity of its optimal tour. This is important as the ants do not get trapped at local minima and so it takes more time to converge. This algorithm is effective where high resolution solutions are required for high precision system.

Table1: Analysis of Performance of the Proposed MAAS Ant System using different TSP Problems.

TSP PROBLEMS	MAAS Ant System				
	Best tour (GMinL)	Best tour iteration time (FRIT)	Average tour	Optimal tour	Relative Deviation (%)
OLIVER30	423.7406	1555	423.9175	423.7406	0
ULYSSES16	74.1087	2057	74.2078	73.9876	0.1637

TSP PROBLEMS	MAAS Ant System				
	Best tour (GMinL)	Best tour iteration time (FRIT)	Average tour	Optimal tour	Relative Deviation (%)
ULYSSES22	75.3984	4830	76.3472	75.3	0.0984

Table2: Analysis and Comparison of the Proposed MAAS with the Existing Ant System for Oliver30.tsp Problem.

Performance	MAAS	AS [6]	ACS [14]	Referencel [14]
BEST	423.7406	423.74	423.74	423.74
WORST	424.4643	----	434.63	425.27
AVERAGE	423.9175	424.25	426.74	424.19
OPTIMAL ITERATION	1555	----	1259	521

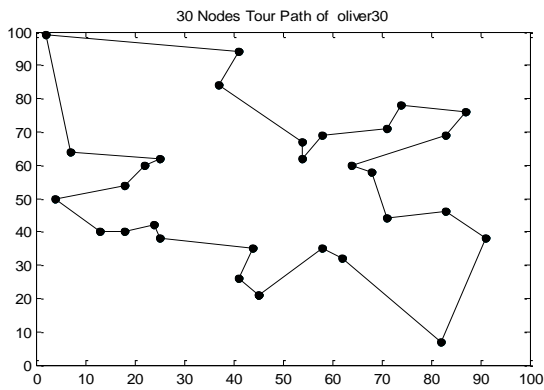


Fig2. Best Tour Path obtained by MAAS using Oliver30.tsp

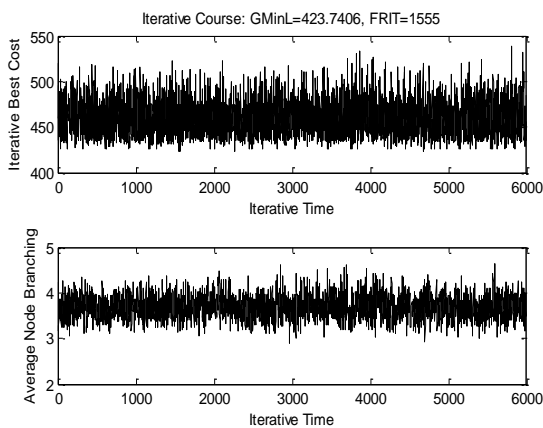


Fig3: Iterative Best Cost and average node branching of MAAS using Oliver30.tsp problem

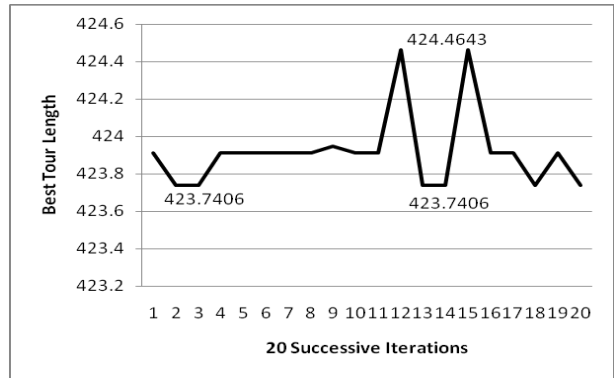


Fig4: Best Tour of MAAS over 20 successive iteration using Oliver30.tsp problem

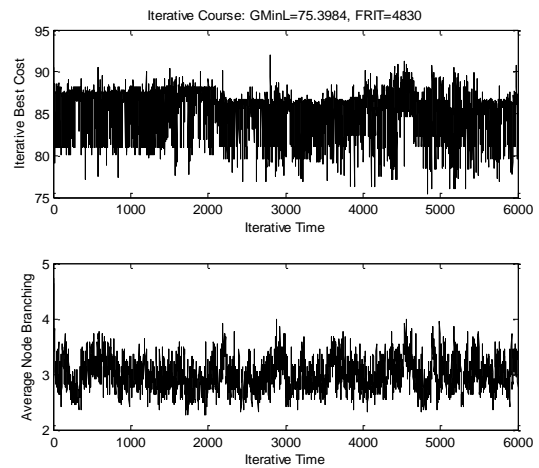


Fig5: Iterative Best Cost and average node branching of MAAS using Ulysses22.tsp problem

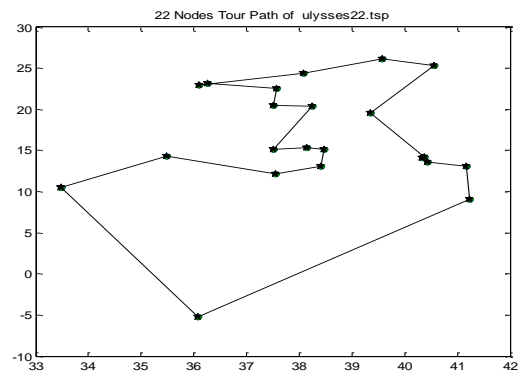


Fig6: Best Tour Path obtained by MAAS using Ulysses22.tsp

7. CONCLUSION

In this paper, an adaptive model of Ant System has been introduced based on the pheromone updation using adaptive filter entitled as Momentum Adaptive Ant System (MAAS). The main idea was to make the Ant System simpler to design and cost effective in terms of both speed and hardware. This adaptation minimizes the cost function and guides the system towards a satisfactory optimal solution. But, it takes longer time and more iteration to converge as the system is active even at the close environs of its optimal solution. MAAS gives comparatively favorable results as shown in Table 1. As the number of city increases more, i.e., in case of large city

problems, it takes more iteration to converge. However, high precision system some time requires meticulous results while sacrificing more convergence time and this algorithm is effective in that context. This paper is a step forward of incorporating Adaptive Filter in the domain of optimization.

Future analysis can be made on the study of stability criterion and searching for a good convergence. Also, we intend to apply the same on ATSP (Asymmetric TSP) problem and other optimized problems. Moreover, MAAS can be explored and analyzed using the properties of momentum-LMS algorithm.

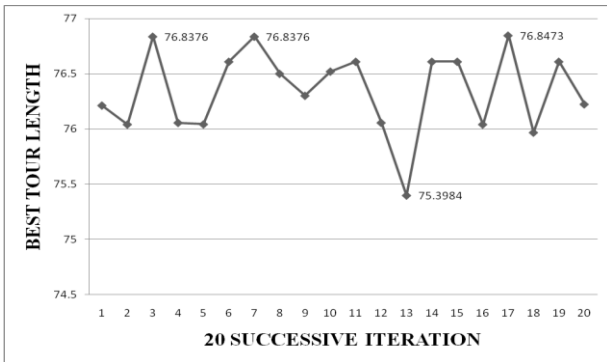


Fig7: Best Tour of MAAS over 20 successive iteration using Ulysses22.tsp problem

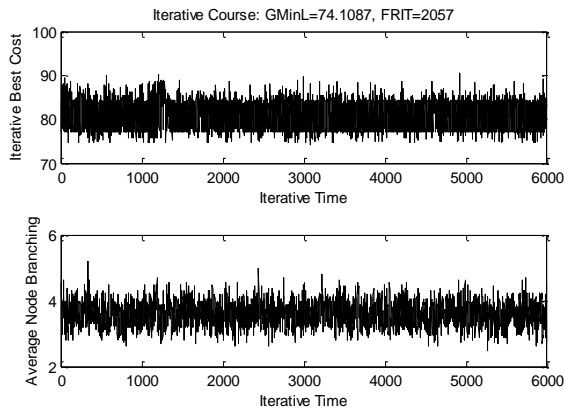


Fig8: Iterative Best Cost and average node branching of MAAS using Ulysses16.tsp problem

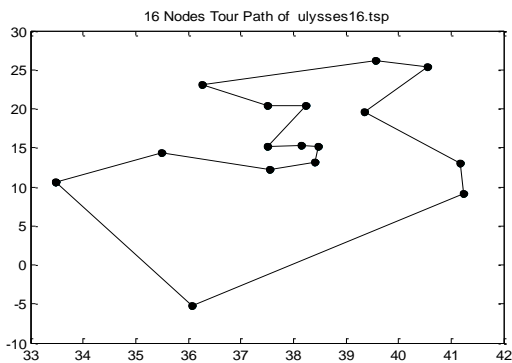


Fig9: Best Tour Path obtained by MAAS using Ulysses16.tsp

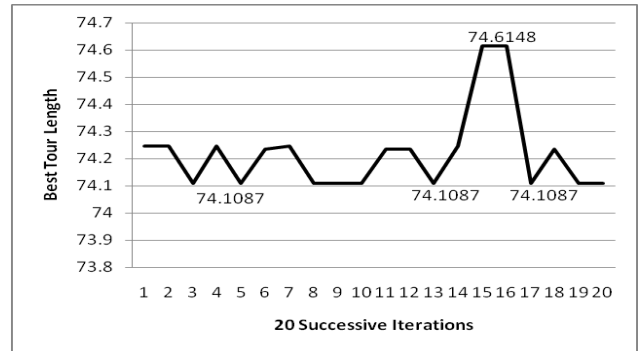


Fig.10: Best Tour of MAAS over 20 successive iteration using Ulysses16.tsp problem

8. References

- [1] D.G. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning", Addison- Wesley, Reading, MA, 1989.
- [2] H.G. Bayer and H.P. Schwefel, "Evolutionary Strategies: A Comprehensive Introduction", *Journal of Natural Computing*, 2004, pp. 3-52.
- [3] J. Kennedy and R.C. Eberhart, "Particle Swarm Optimization", in Proc. *IEEE, Int. Conf. On Neural Networks*, Piscataway, NJ, 1995, pp. 1942-1948.
- [4] M. Dorigo and L.M. Gambardella, "Ant Colony System: A cooperative learning approach to the travelling salesman problem", in *IEEE Trans. Evol. Comput.*, 1997, pp. 53-66.
- [5] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm", *J. Glob. Optim.*, 2007, vol.39, pp. 459-471.
- [6] K. M. Dorigo, V. Maniezzo, and A. Colorni, "The ant system: Optimization by a colony of cooperating agents", in *IEEE Trans. Syst., Man Cybern. Part B*, 1996, pp. 29-41.
- [7] A. Paul and S.Mukhopadhyay, "An Adaptive Pheromone Updation of the Ant System using LMS Technique", in Proc. *Int. Conf. on Modelling, Optimization and Computing (ICMOC-2010)*, AIP, 2010, vol. 1298, pp. 498-503.
- [8] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, 3rd edition, 1996.
- [9] [online] [TSPLIB]: <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp/>.
- [10] J. G. Proakis, "Channel identification for high speed digital communications," *IEEE Trans. Automat. Contr.*, 1974, vol. AC-19, pp. 916-922.
- [11] R. Sharma, W.A. Sethares, and J.A. Bucklew, "Analysis of Momentum Adaptive Filtering Algorithm", *IEEE Trans. of Signal Processing*, 1998, vol.46, no. 5, pp. 1430-1434.
- [12] V. Maniezzo and A. Colorni, "The Ant System applied to the quadratic assignment problem", *IEEE Trans. of Data Knowledge Engrg.*, 1999, vol.11, no. 5, pp. 769-778.
- [13] A. Colorni, M. Dorigo, V. Maniezzo, and M. Trubian, "Ant System for job-shop scheduling", *JORBEL-Belgian J. Operations Res., Stat. Comput. Sci.*, 1994, vol. 34, no. 1, pp.39.
- [14] Guo Shao-sheng and Meng You-xin, "An improved entropy-based ant colony optimization algorithm," in Proc. *Int. Conf. on Computer Application and System Modeling*, 2010, vol.15, pp. 1548-1550.