I.J.Modern Education and Computer Science, **2013, 4, 25-35**
Published Online May 2013 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2013.04.04

Bio-inspired Ant Algorithms: A review

Sangita Roy
Department of Electronics and Communication Engineering, Narula Institute of Technology, WBUT,
Indiaroysangita@gmail.com

Sheli Sinha Chaudhuri
Department of Electronics and Telecommunication Engineering, Jadavpur University, India
shelism@rediffmail.com

***Abstract*—** Ant Algorithms are techniques for optimizing which were coined in the early 1990's by M. Dorigo. The techniques were inspired by the foraging behavior of real ants in the nature. The focus of ant algorithms is to find approximate optimized problem solutions using artificial ants and their indirect decentralized communications using synthetic pheromones. In this paper, at first ant algorithms are described in details, then transforms to computational optimization techniques: the ACO metaheuristics and developed ACO algorithms. A comparative study of ant algorithms also carried out, followed by past and present trends in AAs applications. Future prospect in AAs also covered in this paper. Finally a comparison between AAs with well-established machine learning techniques were focused, so that combining with machine learning techniques hybrid, robust, novel algorithms could be produces for outstanding result in future.

***Index Terms*—** Stigmergic, Forage, Pheromone, Combitorial Optimisation, GA, Artificial Ant, AI.

I. INTRODUCTION

Optimization problem is the problem of finding the best solution from all feasible solutions in mathematics and computer science. Optimization problems can be divided into two categories depending on whether the variables are continuous or discrete. Optimization problem with discrete variables is known as combinatorial optimization problem. An object such as an integer, permutation or graph from a finite set is looking for. Optimization problem are applicable in the field of commercial field, engineering, scientific etc.Subset of optimisation problem is combinatorial optimisation (CO) [12]. CO problem requires the optimal solution with great cost especially when the problem is
NP-Hard. InCO problems, NP-Hard CO problems computational time become too high for any practical use. Subsequently approximation algorithms developed to attain accuracy with minimum time and cost. Ant algorithms, recent approximate optimisation methods, inspired by the behavior of real ants in the wild[2] where communication between ants within the colony via the secretion of chemical pheromones . In artificial intelligence (AI) , ant algorithms are subset of SWARM Intelligence (SI)[13].SI is a problem solving tool for intelligent multi-agent systems inspired from the behavior of real insect swarms. Other such algorithms are developed from the behavior of swarms of wasps and bees ,even the behavior of birds specially cuckoos are observed and developed algorithms[14].From the success of practical implementation of ant algorithms, ant colony optimisation (ACO) which later catagorised as metaheuristic, were proposed [15]. Lists of other metaheuristics are simulated annealing[16], tabu search[17] and [18] and local search [19]. It was first introduced in 1992 and list of publications are given below in Table I.

II. THE IDEA &INSPIRATION BEHIND THE ANT ALGORITHMS (ANT IN THE WILD)

The observation in the foraging behavior of ants in the wild which is known as stigmargyis the inspiration behind the ant algorithms. In 1959 Grasse introduced the term stigmargywhich means indirect communication among a self –organising emergent system via individuals modifying their local environment. Denopboury, Asron, Goss, and Pasteels (1990) narrated the stigmergy nature of ants colonies. Ants communicate indirectly by pheromone trails of the previous ants laid down paths and former ants follow the paths. Subsequently, it was observed that ants follow the path of high pheromone concentration deposit which obviously is the optimized path [20].

A. The Analogy

'Shortest bridge' experiment[20] in Fig. 1 shows two possible paths of different length between a nest and a food source, upper path is shorter and lower path is longer. Initially, half of the ants follow the upper path and remaining half follow the lower path for foraging (Fig. 1a). The ants follow the upper path reach the food source sooner (Fig. 1b). As it is shorter path. On the returning time those ants follow the same path laid by them by tracing the pheromone and again deposited pheromone on the same path, hence strengthen the pheromone trail. (Fig.1c).

TABLE I. PIONEERING REFERENCE WORKS ON ANT ALGORITHMS

Serial Number.	Author	Paper
1.	Ant System	M. Dorigo, V. Maniezzo& A. Colorni, Ant System: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics-Part B, 26(1):29-41, 1996.
2.	System Elitist Ant	M. Dorigo, Optimization, Learning and Natural Algorithms. Ph.D.Thesis, Politecnico di Milano, Italy, [in Italian], 1992.; M.Dorigo, V. Maniezzo and A. Colorni, Ant System: Optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man, and Cybernetics-Part B, 26(1), 29-41, 1996.
3.	Ant-Q	L.M. Gambardella and M. Dorigo, Ant-Q: A Reinforcement Learning Approach to the Traveling Salesman Problem. Proceedings of ML-95, Twelfth International Conference on Machine Learning, Tahoe City, CA, A. Prieditis and S. Russell (Eds.), Morgan Kaufmann, 252-260,1995.
4.	Ant Colony System	M.Dorigo&L.M. Gambardella, Ant Colony System: A CooperativeLearning Approach to the Traveling Salesman Problem. IEEE Transactions on Evolutionary Computation, 1(1):53-66, 1997.
5.	Max-Min Ant System	T. Stützle and H. H. Hoos, MAX-MIN Ant System. Future Generation Computer Systems. 16(8):889--914, 2000.
6.	Rank-based Ant System	B. Bullnheimer, R. F. Hartl and C. Strauss, A New Rank Based Version of the Ant System: A Computational Study. Central European Journal for Operations Research and Economics, 7(1):25-38, 1999. pom-wp-3-97.ps .
7.	Ants	V. Maniezzo, Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem, INFORMS Journal on Computing, 11(4), 358-369, 1999.
8.	Hyper Cube -ACO	C. Blum, A. Roli, and M. Dorigo. HC-ACO: The hyper-cube framework for Ant Colony Optimization. In Proceedings of the Fourth Metaheuristics International Conference, volume 2, pages 399-403, 2001. Later, a strongly extended version of this paper has been published in IEEE Transactions on Systems, Man, and Cybernetics -- Part B, 34(2):1161-1172, 2004.

This half of the ants will return to the nest with minimum time and reinforcing the pheromone trails. From now ants will follow the upper path from nest to food due to higher pheromone concentration. Eventually most, if not all, of the ants end up following the upper shortest path (Fig. 1d). Stigmergy of ants in the wild can be used in a more computational way as an intelligent multi-agent system solution to a shortest route optimisation problem. These observations inspired the first ant algorithm [2] which was applied originally to the well-known "Travelling Salesman 'Benchmark' Problem".

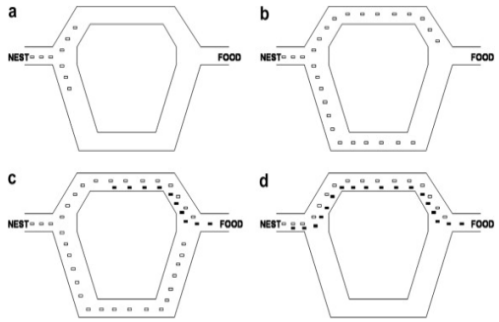

Fig. 1. A Schematic illustration of the "Shortest bridge" experiment (Cardon et al.,2002)[1]

III. FROM NATURE TO COMPUTERS

In case of artificial ant colony, simple computational agents, act as artificial ants, work cooperatively, collectively, communicating through artificial pheromone trails. In an iterative fashion, each ant travels from state S_i to state S_j influenced by two factors: i) Heuristic Information: It is a measure of the heuristic preference (which is application based) travelling from state S_i to the state S_j. This is a priori information to the algorithm run, and is not modified during the algorithm run and ii) Artificial Pheromone Trail(s): It is measure of pheromone deposit from the previous transition from the state Si to state S_j. It can also be defined as the measure of the "so far" learned preference. This information is modified during the algorithm run by the artificial ants.

A. Properties of Artificial Ants

The main properties associated with the artificial ant(Cordon, Herrera, &Stutzle, 2002): i)Each artificial ant has an internal memory which is used to store the path followed by the ant (i.e. the previously visited path),ii)Starting in an initial state $S_{initial,}$ each ant tries to build a feasible solution to the given problem, moving in an iterative fashion through its search space/environment, iii)The guidance factor, movement of ants, takes the form of a transition rule which is applied before every move from S_i to state S_j. This transition rule may also include additional problem specific constraints and may utilise the ants' internal memory, iv) the amount of pheromone, each ant deposits, is governed by a problem specific pheromone update rule, iv) Deposition of pheromone of ants is governed/associated with states or state transitions, v) during solution construction, pheromone deposition may occur at every state transition. This is known as online step-by-step pheromone trial update and vi)Alternatively ants may retrace their paths once a solution has been constructed and only then deposit pheromone, all along their individual paths. This is known as online delayed pheromone update.

B. Additional Characteristics

Apart from the above properties, artificial ants may also have some other characteristics to improve their performance which their biological counterpart does not have. Widely used additional characteristics are local search [19] and [5] and candidate list [21] and [5].Daemon actions may be introduced into the algorithm depending on the problem. Daemon actions influence the guidance of the ants during algorithm runtime and may be used for the speed up convergence. By these daemon actions, extra pheromone may be added to the best solution trail at the end of each iteration.

C. The original Ant system

The first ant algorithm "Ant System" (AS) developed by Dorigo et al., 1996 used benchmark "Travelling Salesman Problem"(TSP) as its test-bed. There are M towns and a sales man has to travel each town once and ending up back to the starting town with shortest route. In Euclidian TSP, the path between any two given towns i and j is given by Euclidean distance:

$$d_{ij} = = [(x_i - x_j)^2 + (y_i - y_j)^2]^{\frac{1}{2}}$$

Each ant in the system has the characteristics as follows:i) An ant decides which town to go by using a transition rule that is a function of the distance between the ant and the decided town and the amount of pheromone deposited along the connecting path, ii) Transitions to already visited towns are added to a tabu list and are not allowed to visit and iii) Once a tour is complete, the ant lays a pheromone trail along each path visited in the tour. An iteration is defined here as the interval in (t, t+1) where each of the N ants moves once. Then an epoch is defined as the collection of N iterations, when each ant has completed a tour. After each epoch the pheromone intensity trails are updated according to the following formula:

$$\tau_{ij}\ (t+n) = (1-\rho).\tau_{ij}(t) + \sum_{k=1}^{N} \Delta\tau_{ij}^{k} \quad (1)$$

Where ρϵ (o, 1] is the evaporation rate and $\Delta\tau^{k}_{ij}$is the quality of pheromone laid on path (i, j) by the k^{th} ant between the time t and t+n, and is given by :

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{L_k}, & if\ ant\ k\ used\ edge\ (i,j)\ in\ its\ tour, \\ 0, & otherwise, \end{cases} \quad (2)$$

Where Q is a constant and L_k is the tour length of the k^{th}ant.

The heuristic information in this case is called the visibility,η_{ij}, and is defined as the quantity $1/d_{ij}$. This quantity is not modified during the algorithm run contrary to the pheromone trail. Atabu list of growing vector of previous and current visited towns is maintained where $tabu_k(s)$ is the s^{th}visited town by the k^{th}ant in the current tour. The probability of the k^{th}ant making the transition from town i to town j is given by:

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{k \in allowed_k} [\tau_{ik}]^{\alpha} \cdot [\eta_{ik}]^{\beta}} & if\ j \in allowed_k \\ 0, \quad otherwise, \end{cases} \tag{3}$$

Where $allowed_k$ = {M-$tabu_k$} and α and β control the relative importance trail and visibility respectively.

IV. ANTALGORITHMS ASCOMPUTATIONAL TECHNIQUES

Ant colony optimisation (ACO) metaheuristic [13] & [15] was development after the successful application of AS on classical TSP.The ACO combinatorial problems by approximate solutions based on the generic behavior of natural ants. Therefore, it can be described that all the applications of the ACO metaheuristic may be grouped as Ant Algorithms, but ant algorithms are not ACO metaheuristic.

A. The ACO metaheuristic

ACO is composed of three main functions (algorithm 1) i) AntSolutionsConstruct() performs the solution construction process where artificial ants travel from one state to another according to transition rule and iteratively reaching the solutions. ii) PheromoneUpdate() performs pheromone trial updates .This pheromone trails update may be completed after complete solutions or after each iteration update. Apart from pheromone trails reinforcement, pheromone trail evaporation is also considered. Pheromone trail evaporation helps the ant to 'forget' bad solutions on the process of algorithm run. This is incorporated reducing all pheromone trails by a set of amount after each epoch. Iii) DaemonActions () is an optional step in the algorithm which provides additional updates from a global perspectives with no natural counterpart available. Pheromone reinforcement to the best solution is one example and known as offline pheromone trail update.

Algorithm 1. The Ant Colony OptimisationMetaheuristic

Parameter Initialisation
WHILE termination conditions not met do
ScheduleActivities
AntSolutionsConstruct()
PheromoneUpdate()
DaemonActions() optional
END ScheduleActivities
END WHILE

B. Developed ACO Algorithms

Ant algorithms collectively form the main ACOalgorithms, generated from original ant system and described below:

1) MAX-MIN Ant System

TheMAX-MINAnt System(MMAS)Algorithm differs from the AS in two ways: i) only the best ant updates the pheromone trails, and ii) the pheromone update function is bound [6].The modified pheromone update is as:
Where,

$$\tau_{ij}(t+n) = \left[(1-\rho) \cdot \tau_{ij}(t) + \sum_{k=1}^{N} \Delta\tau_{ij}^{best} \right]_{\tau_{min}}^{\tau_{max}} \tag{4}$$

$$\Delta\tau_{ij}^{best} = \begin{cases} \dfrac{1}{L_{best}}, if(i,j) belongs to\ the\ best\ tour \\ 0, \quad otherwise \end{cases} \tag{5}$$

And τ_{max} and $\tau_{min}$are the upper and lower pheromone bounds respectively. If x be the upper function with upper and lower bounds incorporated as below:

$$[x]_b^a = \begin{cases} a, if\ x > a \\ b, if\ x > b \\ x, otherwise \end{cases} \tag{6}$$

$L_{best}$is the length of the tour by the best ant. This can be the iteration-best, the best-so-far or a combination of the two.

2) Ant Colony System

The main difference between the Ant Colony System (ACS) [5] & the Ant System [2] is the pheromone update function. In ACS like the natural behavior of ants, a local pheromone update is employed in addition to the pheromone update at the end of each epoch (offline pheromone update). Each ant performs the pheromone update after each construction setup according to the formula:

$$\tau_{ij} = (1-\varphi) \cdot \tau_{ij} + \varphi \cdot \tau_0 \tag{7}$$

Where 错误！未找到引用源。 is the pheromone decay co-efficient and τ_0 is the initial value of the pheromone. The offline pheromone update is applied at the end of each epoch by the iteration –best or the best-so-far ant only similar to the MMAS. The update function for the offline pheromone update is given by:

错误！未找到引用源。 (8)

As in MMAS, $\Delta\tau_{ij}$= 1/ L_{best}, where$L_{best\ can}$ either be the iteration –best or best-so-far. Apart from the differences in the pheromone updateprocedure, the ACS also uses a different transition rule, called the pseudorandom proposition rule. K be an ant located at state i,q_0 Є [0,1] be a parameter , and q a random value in[0,1] then the next node j, is described as:

If q> q_0 :

$$p_{ij}^{k} = \begin{cases} 1, if\ j = \ argmax_{j \in N_k(i)} \tau_{ij} \cdot \eta_{ij}^{\beta} \\ 0, \quad otherwise \end{cases} \tag{9}$$

else (q>q_0) and equation 3 is used.

3) Rank Based Ant System

The idea of ranking in the pheromone update procedure was introduced in the rank-based Ant System (AS_{rank}) [22]. The N-ants are ranked according to the decreasing quality of their solutions, i.e., tour length ($L_1 \leq L_2 \leq L_3 \leq .....\leq L_N$) .The pheromone trails are updated offline in the form of a daemon action, only the paths traversed by the σ -1 best ants receive any pheromone, and the amount deposited depends directly on the ant's rank, μ, and on the quality of its solution. In addition, L_{gb}, the global best solution path traversed, achieves an extra amount of pheromone which depends on the quality of that solution, weighted by parameter σ. The parameter update function is:

$$\tau_{ij} = \rho \tau_{ij} + \Delta \tau_{ij}^{gb} + \Delta \tau_{ij}^{rank} \quad (10)$$

where

$$\Delta\tau_{ij}^{rank} = \begin{cases} \sum_{\mu=1}^{\sigma-1} (\sigma-\mu).\frac{Q}{L_{ij}} & ,if\ \mu^{th}\ best\ ant\ travels\ on\ edge\ (i,j) \\ 0\ ,otherwise \end{cases} \quad (11)$$

and

$$\Delta\tau_{ij}^{gb} = \begin{cases} \sigma\frac{Q}{L_{gb}} & ,if\ edge(i,j)is\ part\ of\ the\ best\ solution\ found \\ 0\ ,\ otherwise \end{cases} \quad (12)$$

V. ANT ALGORITHMS COMPARISONS

A performance comparative study on the four ant algorithms [6] and tabulated in the table II. These observations are performed on the test–bed of TSP application (instances). The table contains six columns & four rows. First column depicts instances---ei151, kroA100, d198.Second column gives their optimum values and rest of the columns gives MMAS, ACS, AS_{rank}& AS respectively. These results show that MMAS algorithm enumerates the best close the optimised value, next ACS, then $AS_{rank}$and AS at the last, or worst. This table is included here to decrease to describe an idea of the performance of these algorithms which can be implemented in practical cases. These results can be improved by changing different parameters chosen [6].

TABLE II. PERFORMANCE STUDY OF ANT ALGORITHMS

Instance	Opt	MMAS	ACS	AS_{rank}	AS
ei151	426	427.6	428.1	434.5	437.3
kroA100	21282	21320.3	21420.0	21746.0	22471.4
D198	15780	15780	16054.0	16199.1	16702.1

VI. PAST AND PRESENT APPLICATIONS

Ant Algorithms now-a-days have been deployed all fields, such as bio-medicines, robotics, engineering, bio-engineering, economics, telecommunications etc. Remarkable changes are occurring by implementing these algorithms. Ant algorithms are well adapted to NP-hard combinatorialoptimisationproblems, and original AS was applied on the bench-mark test bed TSP algorithm. The TSP has characteristics of shortest path/route where AS can be well suited. After TSP, AS was implemented to another NP-hard problem i.e., Quadratic Assignment Problem (QAP) & the Job-Shop Scheduling Problem (JSP) . QAP & JSP were utilised as to prove the robustness of the original AS. Subsequently, improved AS were proposed to solve QAP & JSP individually with sufficient care. Data networking –dynamic problem and routing are of immense importance in modern technology and improved ant algorithms can be imposed to solve the shortest problem where node availability fluctuates randomly with high frequency. Set covering [23], graph colouring [24], and the vehicle routing problem [25] and [26] are the other most important applications of AAs. Among them, vehicle routing problem using AAs implemented successesfully in traffic control, industry.DyvOil is used to support the sales and distribution of fuel distribution. An *offline module* DyvOil is still in use for solving static vehicle routing problem using ACO algorithms every evening for the next day vehicle routing problem.apainaPetroli , a leading swiss fuel oil company tested the offline planning module and 20% to 30% increase in vehicle routing problem performance over the man-made plan [27]. Performance efficiency showed the same improvement by using ACS algorithms for dynamic vehicle routing problems. The Switzerland supermarket, Migros, has been using AntRoute for their supply chain. The tour of the distribution vehicles to the supermarkets across Switzerland is computed by MACS-VRPTW, ant based algorithm and again outperformed human planning [26].The spread of applications of ant algorithms(AAs) has been expanded enormously in recent years with lots of novel approach as well as broader areas of study like continuous optimisation [28],[29] & parallel processing [30] & [31]. Originally AS was implemented on discrete optimization [15], as a result of which the ACO metaheuristic was developed, but later it gained popularity on continuous optimisation.

VII. DEVELOPMENT IN ANTALGORITHMS

In the last decade novel algorithms and novel approach have been crystalised and applied in a wide range .Following are the few examples.

A. Ant Algorithms for Digital Image Processing (DIP)

For the last twenty years DIP gains popularity as fast processors became available and cheaper.DIP has established as one of the most utilised tool for

engineering and scientific research areas. By using the essence of machine learning and AI, DIP can be increased its performance and efficiency. AS are also introducing as a new technique to improve its performance by using basic low level image segmentation via boundary detection methods [32],[33], [34], & [35] and via clustering methods [36] & [37]. Novel approach of boundary detection methods deploy the pheromone aspects of AS , here the ants are considered as pixels within the image and move within the image in a discretised pixel-wise fashion whose aim is to extract and map boundary's within the image . A heuristic information is proposed by which probability of an ant tours from its current location (pixel) to a surrounding node (pixel) which has the greatest boundary characteristics (greatest change in image gradient for example). Again each ant deposits pheromone with each move from one pixel to another and this pheromone evaporation which is fixed per iteration. Therefore the transition rule is a function of the heuristic information and the pheromone map. Artificial Ants start at random position within the image and resultant pheromone trails mapping out the boundaries between image segments as shown in fig. 2 .As the number of iteration increases quality of the extracted picture increases as the pheromone concentration increases of high gradient and decreases with low gradient. The original and final pheromone field map is shown in fig. 3.The premise of this algorithm (basic low level image) has two major differences from the "Traditional" types of ant algorithms i) individual ants never construct complete solutions of their own, and ii) the pheromone is not only used to guide the ants movement but also represents the final solution in its entity. In clustering methods, a more standard AA is employed as a tool to optimise the pixel mapping as clusters within image which does not employ pheromone as a visual solution. Image threshold [38] is another low level segmentation Technique where ants search the image in low gray scale regions, and in an area for image compression [39].

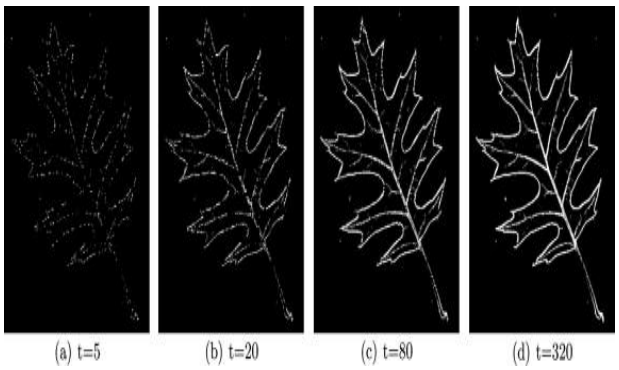

Fig. 2. Emerging pheromone map at different time-steps in the algorithm run. Brighter pixels equal higher pheromone concentration at that point [1].

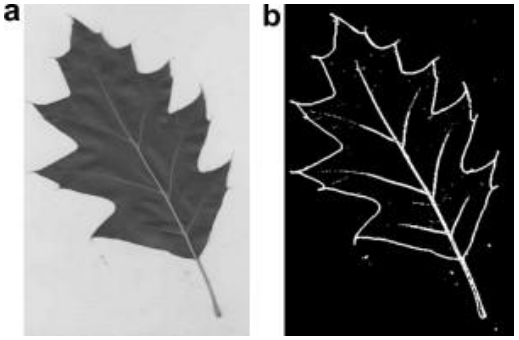

Fig. 3. Original leaf image (a) and final pheromone field map (b)[1].

B. Ant Algorithms and Protein folding problem

Computational biology encounters serious problem relating the prediction of protein structure from its linear sequence. It is NP-hard and computationally very expensive .Numerous models have been come out for simplifying the search space of prediction and conformation. Hydrophobic–polar(HP) model [40] explores the primary amino acid sequences of a protein of 24 letter alphabet representation to a sequence of hydrophobic (H) and Polar(P) resides that reduces to Aar string of alphabets H & P. The protein conformations of this sequence are restricted to self-avoiding paths on a lattice. AA have been proposed for both the 2D [41] & 3d HP folding problem [42] considering 2D and 3D lattice respectively .The HP folding [43] is based on the thermodynamics hypothesis where native state of protein with the lowest Gibbs free energy and the energy is calculated from the number of topological contacts between hydrophobic amino-acids that are not neighbors in a given sequence. Example: for a conformation C with exactly n of H-H contacts, will have free energy level E(C) = N. (-1). For the 2D case, candidate conformation is expressed by the relative folding directions: straight(S), left (L), and right (R) .These representations indicate the present amino-acid location with respect to previous sequence. In the construction phase of the algorithm, each ant is placed randomly within the problem sequence. Sequences are then built up by summing one amino-acid symbol at a time with respect to relative direction where conformation extends chosen probabilistically by familiar Alas both heuristic and pheromone construction trail/concentration) in heuristic nature, the construction process conforms towards maximum number of H-H contacts, thus minimises the Gibbs free energy. ii). In the pheromone update, "standard" form is eq (1) and relative solution quality is measured by E(C))/E, where E is the known or approximate minimal energy for the given protein sequence. In this way solutions are built/ developed as in classical TSP application. Here, the solution components are the folding directions, as opposed to paths between cities, as with TSP.

C. Ant Algorithms and Data Mining

The goal of data mining is to extract knowledge from data [44]; discovering previously unknown, valid

patterns and relationships in large data sets[45]. The classification, clustering and forecasting are the application of data mining where the aim is to analyse the data and make some information or prediction out of them. The application of classification task of data mining on AAs[46],[47],[48] and [50] is to ascertain each case (object, record, or instance) to one class , out of a set of predefined classes , based on the values of some attributes for the case and this can be expresses as:

IF< term1 AND term2 AND ……..> THEN<class>,
Where each term is a triple

<Attributes, operator, value>i.e.< gender=female>
where value being a value belonging to the domain of attributes and the operator is the relational operator.

Ant-Miner [48] is a sequential covering approach to discover most of the training cases. Initially, list of discovered rule is empty and the training set is full. Once a classification rule is discovered for each iteration of the algorithm is added to the empty discovered rule list and the training set covered by the mentioned rule is deleted from the training set list. The algorithm repeated until it reaches a threshold specified by the uncovered training cases. Ants build solution ineach iteration via three steps: i) rule construction, ii) rule pruning, and iii) pheromone updating. An Ant starts with an empty rule, and builds up partial rules by adding one term at a time, corresponding to the path or segment of path taken by the ant through the search space of terms.The choice of direction to take to extend the current path. This choice depends on i) problem dependent heuristic information, ii) the amount pheromone deposition with each term. The ant keeps adding terms to the rule until either all attributes have been used, or the minimum number of cases covered per rule has been met. Once the rule has been constructed by the ant, it is then pruned of any irrelevant terms and the pheromone trails are updated .The next ant then begins constructing its rule, using the updated pheromone trails as guidance. The process is repeated until all ants have constructed the rule, or convergence has happened. When the process is halted, iteration is complete, then the best rule constructed and the process is repeated. In Ant-Miner, heuristic information is the measure of entropy (amount of information) associated with each term. The pheromone update rule involves increasing the amount of pheromone associated with each term occurring after pruning, in arule found by an ant, where the increase is proportional to the quality of that rule. Moreover, pheromone evaporation is applied to each term that does not occur in the rule.

VIII. ANTALGORITHMS AND OTHER MACHINE LEARNING TECHNIQUES

AAs exhibit similar characteristics with many well established machine learning techniques, in this section comparative study of AAs with machine learning techniques have carried out, with hybrid novel algorithms proposals.

A. Ant Algorithms as an alternative to other Machine Learning Techniques

Machine learning techniques –evolutionary computing and neural networks [51] and [15] have some similarities with some of the ant algorithms, especially ACO algorithms. Both evolutionary computing and the ACO met heuristic deploy a population of individuals to incrementally build more suitable solutions to a given problem, by building on the solutions of the previous populations. Evolutionary computing deals with the information or knowledge of the problem contained only within the current population,whereas the ACO met heuristic explores information from the current as well as previous generation populations as the knowledge base in the form of pheromone trails. There are some forms of specific evolutionary computing algorithms which are more close to ACO met heuristics. Characteristics [51] & [15]. Problem solved by Artificial Neural Network (ANN) [52], biologically inspired learning systems and AAs may be often represented graphically by a set of connecting nodes. “States” in AAs is synonymous to “neurons” in neural networks (NNs), and equivalently the local neighborhood structure around a given state is equivalently to the set of synaptic –line links in neuron & [15].The ants are collectively equivalently to the input signals that propagate through the NNs. AAs connecting paths receive more pheromone as more they are used, equivalently strength of synapse increases as more it is used. Both the AAs connecting paths and synaptic links in ANNs uses paths and synapses links to create better solution and these reinforcement is more than others. Due to these similarities, such algorithms are used to solve same type of problems. But they differ in their performance for different specific tasks. From the point of view of convergence for optimum solution, and fast operation , original AS [2] and ACS[3] ,AAs tested over test-bed benchmark problem TSP show better performance over other heuristic approaches . By introducing certain elements especially the stigmergic effect through synthetic pheromone increase their performance over the original learning techniques, Quality learning (Q-learning) [1] & [53].

B. Ant Algorithms as a counterpart to other learning techniques

Q-learning (Watkins, 2001) and GAs ([Holland, 1992], [10] & [11]) are two machine learning techniques which are used in conjunction with AAs.

1) Ant Algorithms & Q-learning

Q-learning hails from reinforcement learning and reinforcement learning is a subset of machine learning which could be correlated to AAs. Reinforcement learning leads to agents learning by trial and error to achieve goals in their current environment taking necessary best actions [52].In the training phase , every time an agent develops an action in its environment, it gets reward or penalty reflecting the necessity of the outcome performed. The goal of an agent is to choose

sequences of actions so as to maximize the cumulative reward. Q-learning involving learning an action values function where action values are called as Q-values. , measures the utility of taking a given in a given list /sequence within the environment. At each time-step "t", an agent in state S_i takes an action 'a' and changes to an agent in state $S_{i+t.}$ The agent receives a reward depending on the new state. The Q-values for each state –action pair are updated at each time-step until convergence between successive Q-values approaches zero by equation follows:

$$Q_n(s_t, a) \leftarrow (1 - \alpha_n) Q_{n-1}(s_t, a) + \alpha_n [r_t + \gamma_{maxa} \cdot Q_{n-1}(s_{t+1}, a)] \quad (13)$$

and

$$\propto_n = \frac{1}{1 + visit\ S_n\ (s, a)} \quad (14)$$

where γ is the discount factor, a is the action that maximizes Q, and visits (S_t, $a_)$ is the total number of times the given state-action pair visited previously.

Ant-Q an Algorithm inspired by original AS, [5] & [4]) has many similarities with the Q-learning, Ant-Q agents communicate in the form of AQ-values. Ant-Q likeAS, algorithm was developed for the test-bed of TSP bench mark algorithm. AQ(r, s) is the Ant-Q value for the path (r, s) between the cities HE(r, s) is the heuristic value for the path (r, s) where as for TSP is the inverse of distance. K is the agent to complete a closed tour of all cities and J_k (r) is the list of all cities still to cover where r is the current city. This list is acting as a memory. This is another difference between Ant-Q and Q-learning. The state transition rule for an agent K in the city r is as follows:

$$s = \begin{cases} argmax_{\mu \in Jk(r)} \{[AQ(r, \mu)]^{\alpha} \cdot [HE(r, u)]^{\beta}\}, if\ q \leq q_0 \\ s, \quad otherwise \end{cases} \quad (15)$$

where α and β parameters which weigh the relative importance of the learned AQ-values and the heuristic values , q is a uniform probability randomly chosen value in [0,1] , q_0 $(0 \leq q_0 \leq 1)$ inversely changes with probability and s is a random variable of probability distribution AQ(r,u) and HE(r,u) with u € J_1 (r)

$$AQ(r, s) \leftarrow (1 - \alpha). AQ(r, s) + \alpha(\Delta AQ(r, s) + \gamma. MAX_{z \epsilon jk(z)} AQ(s, z)) \quad (16)$$

Where α and γ are the learning step and discount factor respectively. J_k (S) is a function of the previous history of agent K and the set of available actions in the set s.The update rule in Ant-Q is same as in Q-learning. Ant-Q does not make use of artificial pheromones. A different idea has been proposed by [54] & [53]) where Q-learning idea adopted using artificial pheromone in multi-agent Q-learning. The pheromone Q-learning (Phe-Q) algorithm uses the same Q-value update function (equation 13) with an additional factor, the belief factor, to be minimized. The belief factor is a function of artificial pheromone concentration on the trail and how extracts laid down by other agents of the same group. The belief factor is the ratio of the sum of the actual pheromone concentrations in the current states and surrounding states to the sum of maximum pheromone concentrations in the current states and surrounding states:

$$B(s, a) = \frac{\sum_{s \epsilon N_a} \varphi(s)}{\sum_{s \epsilon N_a} \varphi_{max}(\sigma)} \quad (17)$$

where φ(s) is the pheromone concentration at state 's' in the environment, and N_a is the set surrounding states for a chosen action a.Q-learning update function becomes after adding belief factor:

$$Q_n(s_t, a) \leftarrow (1 - \alpha_n) Q_{n1}(s_t, a) + \alpha_n \{\gamma_t + \gamma max_a [Q_{n+1}(s_{t+1}, a) + \xi B(s_{t+1}, a)]\} \quad (18)$$

where ξ is a sigmoidal function of time epoch ≥ 0 and increases as the number of agents successfully complete the task.Pheromone improvement in Ant-Q where Ant algorithm is coupled with Q-learning machine learning technique has been shown.Then compared with without ant algorithm [53]. It can then be concluded that "Hybrid Algorithm" produce superior performance.

2) Ant Algorithms with Genetic Algorithms:

Genetic algorithms evolved from evolutionary computing which is a subset of machine learning, Evolution computing, started early 1950s, inspired by biological evolutions which deploy a population of candidate solutions to a given problem, uses operators based on natural selection and natural genetic variation [55]. The basic GA arises updating a population of hypothesis iteratively. Each individual hypothesis is examined by a problem specific fitness function. A new population is then created, firstly by probability selection a portion of the fit individual and the remaining of the new problem are then created by probability selecting pairs of the fit individual, and creating new offspring hypothesis by applying genetic crossover operators. Moreover Genetic mutation operators can also be implemented to random members of the new population before carrying it to the next iteration. A GA [10], [11] for automatically adapt the control parameter s of AAs, picks up appropriate control parameters which is vital to the success of the algorithms for the given application or problem, so that an optimum balance is achieved between exploration of unexplored areas of the search space /environment, and exploration of previously learned preferences for states to visit, within the context of the problem.. The GAs can be used as automated alternative against often used trial and error process of selecting appropriate control parameters as wells potentially adaptable online balance exploration and exploration with i the search space/environment. The control parameters of ant algorithm ACS are genetically adapted for the TSP[10]. According to section IV.B.2, each ant is encoded by the parameters β, ρ, and q_0 in the form of bit strings (chromosomes) which are randomised. At each iteration, four ants are chosen via a tournament selection method

and they construct their TSP tours. Then algorithm checks out for new selected tours. The pheromone trial of an ant that produced the best overall tour is used for the global update using ρ parameter. The fitness of each ant is calculated as the length of the tour found by the beat ant. Best two ants of the selected list are crossed over to produce two offspring, which are then muted. The worst two selected ants are then replaced by the two offspring. The basic philosophy here is to impose pressure on the population to improve its performance with maintaining population size constant. It has explored that some problems emerged out of GAs in AAs [10]. Many ants trapped in local minima with pheromone trails for good solutions to worst performing ants with no guarantee of optimal solution due to immense variability of population of ants. Still due to merging of GAs into AAs the speed of convergence increased towards a solution where desirable characteristics for large problems which is not desired, or indeed may not be possible to find the optimal solution. Choosing appropriate control parameters is an important factor a given AA. Some characteristics may change over time or problems may change due to different circumstances, there fine tuning of the control parameters is very advantageous to adapt with the changing characteristics. Any other GA can be utilised to improve the speed of convergence of the solution problem as well as different applications of AAs can be implemented by this GA as those of dynamic nature.

IX. CONCLUSION

AAs explore a vast knowledge field of complex computational problem during last 20 years inspired by the nature social insects. They are used theoretically as well as practically in science and industry for social benefits. Original AS through ACO metaheuristic travelled a long journey and many new algorithms developing from them inspired from the adaptations and stigmergic behavior of ants in the wild. This paper briefly studied AAs from the original AS to ACO metaheuristics i.e., the most successful variants and reinforces the success of this biological adaptation and stigmergic approach to complex computational problem solution. A brief comparative study of the most successful ant variants for speed of convergence solution is drawn. Concise theoretical and mathematical expressions of AAs have been discussed. In the last ten years AAs become a hot topic because of its distributed nature, direct or indirect interaction between small agents, flexibility and robustness. The numbers of successful users are growing exponentially in the field of combinatorial optimization, communications networks, and robotics. More and more researchers are bending in this form of artificial intelligence as world is becoming complex and lots of informational threat is acting upon human civilization. In this vulnerable situation ant algorithm intelligent systems with autonomy, emergent and distributed functioning can replace control, preprogramming and centralization.

REFERENCES

[1] R.J.Mullen, D. Monekosso, S. Barman, P.Remagnino, “A review of ant algorithms”, R.J.Mullen, D. Monekosso, S. Barman, P.Remagnino. DOI:10.1016/j.eswa.2009.01.020

[2] M. Dorigo, V. Maniezzo, *Ant System* & A. Colorni, *Ant System:optimization by a colony of cooperating agents*. IEEE Transactions on Systems, Man, and Cybernetics-Part B, 26(1):29-41, 1996. DOI:10.1109/3477.484436

[3] M. Dorigo, System Elitist Ant, Optimization, Learning and Natural Algorithms. Ph.D.Thesis, Politecnico di Milano, Italy, [in Italian], 1992.; M.Dorigo, V. Maniezzo and A. Colorni, Ant System: Optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man, and Cybernetics-Part B, 26(1), 29-41, 1996. DOI:10.1109/3477.484436

[4] L.M. Gambardella and M. Dorigo, Ant-Q: A Reinforcement Learning Approach to the Traveling Salesman Problem. Proceedings of ML-95, Twelfth International Conference on Machine Learning, Tahoe City, CA, A. Prieditis and S. Russell (Eds.), Morgan Kaufmann, 252-260,1995

[5] M. Dorigo, Ant Colony System& L.M. Gambardella, Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. IEEE Transactions on Evolutionary Computation, 1(1):53-66, 1997. DOI:10.1109/4235.585892

[6] T. Stützle, Max-Min Ant Systemand H. H. Hoos, MAX-MIN Ant System, Future Generation Computer Systems. 16(8):889--914, 2000. DOI:10.1016/S0167-739X(00)00043-1

[7] B. Bullnheimer, R. F. Hartl, Rank-based Ant System, and C. Strauss, A New Rank Based Version of the Ant System: a Computational Study,Central European Journal for Operations Research and Economics, 7(1):25-38, 1999.

[8] V. Maniezzo, Ants, Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem, INFORMS Journal on Computing, 11(4), 358-369, 1999. DOI:10.1287/ijoc.11.4.358

[9] C. Blum, A. Roli,Hyper Cube –ACO and M. Dorigo. HC-ACO: The hyper-cube framework for Ant Colony Optimization. In Proceedings of the Fourth Metaheuristics International Conference, volume 2, pages 399-403, 2001. Later, a strongly extended version of this paper has been published in IEEE Transactions on Systems, Man, and Cybernetics -- Part B, 34(2):1161-1172, 2004. DOI:10.1109/TSMCB.2003.821450

[10] M. L. Pilat,T. White, Using Genetic Algorithms to Optimise ACS-TSP,Proceedings of Ant Algorithms,Third International workshop,2002. DOI:10.1007/3-540-45724-0_28

[11] White, T., Pagurek, B. and Oppacher, F. (1998). Connection management using adaptive mobile agents. Proceedings of the International

Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA '98) (Arabnia, H. R., ed), pp. 802{809.CSREA Press.

[12] C.H. Papadimitriou, K.Steiglitz, Combitorial Optimisation: Algorithm and complexity, Prentice Hall, 1982.

[13] E. Bonabeau M., Dorigo, & G. Theraulaz, Swarm Systems: From Natural to Artificial Systems, Oxford University Press,1999.

[14] Xin-She Yang and Suash Deb, Cuckoo search via levy flight, Proc. of f IEEE, 2009. DOI:10.1109/NABIC.2009.5393690

[15] Dorigo M., G. Di Caro and L. M. Gambardella. Ant Algorithms for Discrete Optimization. Artificial Life, 5, 2, pp. 137-172, 1999. DOI:10.1162/106454699568728

[16] S. Kirkpatrick; C. D. Gelatt; M. P. Vecchi, Optimization by Simulated Annealing,Science, New Series, Vol. 220, No. 4598. (May 13, 1983), pp. 671-680.

[17] Fred Glover (1990). "Tabu Search - Part 2". ORSA Journal on Computing 2 (1): 4–32. DOI:10.1287/ijoc.2.1.4

[18] Fred Glover (1989). "Tabu Search - Part 1". ORSA Journal on Computing 1 (2): 190–206. DOI:10.1287/ijoc.1.3.190

[19] Helena R. Lourenco, Olivier C. Martin, Thomas Stuzle,Handbook of Metaheuristics (2002), pp. 321- 353.

[20] S. Gross, S.Aron, J. L. Deneubourg, J. M. Pasteels,Self- organised shortcuts in Argentine ANTS, Springer-Verlag, 1989. DOI:10.1007/BF00462870

[21] G. Di Caro and M. Dorigo. AntNet: A mobile agents approach to adaptive routing.Technical Report 97-12, Universit é Libre de Bruxelles, 1997.

[22] Bernd Bullnheimer, Richard F. Hartl, Christine Strauß, A New Rank Based Version of the Ant System A Computational Study ,1996.

[23] Lessing, Dumitrescu, & Stutzle, A Comparison Between ACO Algorithms for the Set Covering Problem, ANTS Workshop'04. DOI:10.1007/978-3-540-28646-2_1

[24] Costa D.; Hertz A, Ant can colour graphs, Journal of the operation research society, 1997.

[25] Bernd Bullnheimer, Richard F. Hartl, Christine Strauss, Applying the Ant System to the Vehicle Routing Problem, 1999. DOI:10.1007/978-1-4615-5775-3_20

[26] L. M. Gambardella, É. Taillard, G. Agazzi, A multiple ant colony system for vehicle routing problems with time windows, Mcgraw-Hill'S Advanced Topics In Computer Science Series, 1999.

[27] Luca Maria Gambardellaa, b, Andrea E. Rizzolia, b, Fabrizio Oliveriob,Norman Casagrandea, Alberto V. Donatia, Roberto Montemannia, Enzo Lucibellob, Ant Colony Optimization for vehicle routing in advanced logistics systems,IEEE Transactions2003.

[28] Krzysztof Socha: ACO for Continuous and Mixed- Variable Optimization, ANTS Workshop 2004: 25-36. DOI:10.1007/978-3-540-28646-2_3

[29] Krzysztof Socha, Christian Blum: An ant colony optimization algorithm for continuous optimization: application to feed-forward neural network training. Neural Computing and Applications 16(3): 235-247 (2007). DOI:10.1007/s00521-007-0084-z

[30] Max Manfrin, Mauro Birattari, Thomas St ¨utzle, and Marco Dorigo, Parallel Ant Colony Optimization for the Traveling Salesman Problem, ANTS 2006, LNCS 4150, pp. 224–234, 2006.@ Springer-Verlag Berlin Heidelberg 2006. DOI:10.1007/11839088_20

[31] Talbi E-G., Roux O., Fonlupt C., Robillard D., Parallel ant colonies for combinatorial optimization problems, BioSP3 Workshop on Biologically Inspired Solutions to Parallel Processing Systems, in IEEE IPPS/SPDP'99. DOI:10.1007/BFb0097905

[32] Mas, J. and Fernandez, G. (2003). Video shot boundary detection based on color histogram, In TRECVid Workshop.

[33] Fernandez, D.C., 2005. Delineating Fluid-filled region boundaries in optical coherence tomography images of the retina. IEEE Trans. Med. Imag., 24: 929-945. DOI:10.1109/TMI.2005.848655

[34] H. Nezamabadi-Pour, S. Saryazdi, and E. Rashedi, "Edge detection using ant algorithms," Soft Computing, vol. 10, pp. 623–628, May 2006. DOI:10.1007/s00500-005-0511-y

[35] Ramos, V. and Almeida, F. (2000). Artificial ant colonies in digital image habitats: A mass behavior effect study on pattern recognition, In Dorigo, M., Middendorf, M.,and Stuzle, T., editors, From Ant Colonies to Artificial Ants - 2nd Int. Wkshp. on Ant Algorithms, pages 113–116.

[36] Channa, A. H., Rajpoot, N. M., & Rajpoot, K. M. (2006). Texture segmentation using ant tree clustering. In *2006 IEEE international conference on engineering of intelligent systems, ICEIS 2006* (pp. 1–6). Piscataway: IEEE Press. DOI:10.1109/ICEIS.2006.1703192

[37] Salima Ouadfel, Mohamed Batouche, Unsupervised Image Segmentation Using a Colony of Cooperating Ants, Biologically Motivated Computer Vision Lecture Notes in Computer Science Volume 2525, 2002, pp 109-116 .DOI:10.1007/3-540-36181-2_11

[38] Malisia, A. R. and Tizhoosh, H. R. (2006). Image thresholding using ant colony optimization. In Proceedings of the 3rd Canadian Conference on

Computer and Robot Vision (CRV'06). DOI:10.1109/CRV.2006.42

[39] Antonio Plaza, David Valencia, Javier Plaza, Pablo Martinez, Commodity cluster-based parallel processing of hyperspectral imagery,Journal of Parallel and Distributed Computing,Volume 66, Issue 3, March 2006, Pages 345–358. DOI:10.1016/j.jpdc.2005.10.001

[40] Kit Fun Lau , Ken A. Dill, A lattice statistical mechanics model of the conformational and sequence spaces of proteins,Macromolecules, 1989, 22 (10), pp 3986–3997. DOI:10.1021/ma00200a030

[41] Shmygelska A, Hoos H, An ant colony optimisation algorithm for the 2D and 3D hydrophobic polar protein folding problem, *BMC Bioinformatics* 2005. DOI:10.1186/1471-2105-6-30

[42] Fidanova S. ,3D HP Protein Folding Problem using Ant Algorithm, BioPS'06, October 24-25, III.19-II.26.

[43] Shmygelska A, Hoos H, An Improved Ant Colony Optimisation for the 2D HP Protein folding problem, Springer Berlin, 2003. DOI:10.1007/3-540-44886-1_30

[44] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth, From Data Mining to Knowledge Discovery in Databases, American Association for Artificial Intelligence. All rights reserved. 0738-4602-1996.

[45] Herbert A.Edelstein (1999): Introduction to Data Mining and Knowledge Discovery, Third dition. U. S. A:Two Crows Corporation, pp: 8-9, 2005.

[46] A. A. Freitas, R. S. Parpinelli, and H. S. Lopes, Ant colony algorithms for data classification, in encyclopedia of Information Science and Technology, M. Khosrou-Pour, Ed., pp. 420–424, IGI Publishing, Hershey, Pa, USA, 2nd edition, 2005. DOI:10.4018/978-1-60566-026-4.ch027

[47] Ji X, Bailey J, Dong G, Mining minimal distinguishing subsequence patterns with gap constraints, In: Proceeding of the 2005 international conference on data mining (ICDM'05), Houston,TX, pp 194–201. DOI:10.1109/ICDM.2005.96

[48] PARPINELL, R.S., LOPES, H.S. and FREITAS, A.A., 2002, Data mining with an ant colony optimization algorithm.IEEE Transaction on Evolutionary Computation, 6, pp.321–332. DOI:10.1109/TEVC.2002.802452

[49] James Smaldon, Alex Alves Freitas, A new version of the ant-miner algorithm discovering unordered rule sets. GECCO 2006: 43-50. DOI:10.1145/1143997.1144004

[50] Smaldon, J. & Freitas, A.A. (2006). A new version of theAnt-Miner algorithm discovering unordered rule sets.Proc. Geneticand Evolutionary Computation Conf. (GECCO-2006), 43-50. San Francisco, CA: MorganKaufmann. DOI:10.1145/1143997.1144004

[51] Cordon,O. et al. (2002) Linguistic modeling by hierarchical systems of linguistic rules.IEEE Trans. Fuzzy Syst., 10, 2–20. DOI:10.1109/91.983275

[52] T. Mitchell, Machine Learning, Handbook on Machine Learning,1997.

[53] Monekosso, N. and Remagnino, P. (2004), The analysis and performance evaluation of the pheromone-Q-learning algorithm. Expert Systems 21(2):80-91. DOI:10.1111/j.1468-0394.2004.00265.x

[54] D. N. Monekosso, P. Remagnino, Chapter *"Phe-Q : A pheromone based Q-Learning"* in 'AI 2001: Advances in Artificial Intelligence', Lecture Notes in Computer Science Edited by Stumptner, M.;Corbett, D.;Brooks, M., Springer, December/10, Adelaide, Australia, pp. 345-356. DOI http://dx.doi.org/10.1007/3-540-45656-2_30 (2001).

[55] Mitchell-Olds T. Genetic constraints on life-history evolution: Quantitative-Trail Loci influencing growth and flowering in *Arabidopsis thaliana*. Evolution. 1996; 50: 140–145.

Sangita Roy is an Assistant Professor at ECE Department of Narula Institute of Technology under WBUT .She has a teaching experience of more than sixteen years. She was in instrumentation industry for two years and in administration for two years. She completed her Diploma (ETCE), A.M.I.E (ECE) and M-Tech (Comm. Engg.).Currently perusing her PhD under Dr. Sheli Sinha Chaudhuri at ETCE Department of Jadavpur University. She is a member of IEI, IETE, FOSET, and IEEE.

Dr. SheliSinhaChaudhuri is an Associate Professor at ETCE Department of Jaduvpur University. She completed her B-Tech, M-Tech, and PhD at Jadavpur University. She has a vast teaching experience of 12 years. She has large number of papers in International and national level journals as well as conferences. Currently research scholars are pursuing PhD under her guidance. She is a member of IEEEand IEI.