International Journal on Applications of Information and Communication Engineering
Volume 1: Issue 2: February 2015 , Pages:35-40 www.aetsjournal.com

---

A Review on Optimization Techniques used in Mobile Ad-hoc Networks

S.Sridhar, R.Baskaran

Abstract**—** MANETs (Mobile Ad-Hoc Network) are networks that consist entirely of wireless nodes, placed together in an ad hoc manner (not engineered) and without the support of a fixed communication infrastructure. All nodes are mobile, and can enter or leave the network at any time. One of the major issues in MANET is routing due to the mobility of the nodes. Failure rate and security attack rate of processes running on mobile hosts in MANET is high. Evolutionary algorithms (EAs) are stochastic search methods that mimic the natural biological evolution and/or the social behavior of species. Such algorithms have been developed to arrive at near-optimum solutions to large-scale optimization problems. Besides conventional approaches, many new researches have proposed the adoption of Swarm Intelligence for MANET routing. Swarm Intelligence (SI) refers to complex behaviors that arise from very simple individual behaviors and interactions, which is often observed in nature, especially among social insects such as ants, bees, fishes etc. Nature-inspired algorithms (swarm intelligence) such as ant colony optimization (ACO) algorithms have shown to be a good technique for developing routing algorithms for MANETs.

Keywords**—** Mobile Ad-Hoc Network , Swarm Intelligence, Evolutionary algorithms, ant colony optimization etc

I. INTRODUCTION

A mobile ad-hoc network consists of a collection of mobile nodes which can communicate with each other with the help of wireless links without the help of any pre-existing communication infrastructure. In Mobile Ad hoc Networks [1] [2], nodes are self-organized and use wireless links for communication between themselves. Nodes within each other's radio range communicate directly via wireless links, while those that are far apart rely on intermediate nodes to forward their messages. Mobile ad hoc networks are the self organizing and self-configuring wireless networks consisting of multiple numbers of hops. MANET is made up of a set of mobile hosts which can freely movable and helps in delivering or relaying packets on behalf of one another. Due to the lack of infrastructure in these type of networks, nodes itself can act as a routers and relay the packets from source to destination. For deploying MANETs, there is no need of infrastructure resulting into the use of MANETs in the scenarios like crowd control, search and disaster rescue operations and battlefields.

One of the major issues that affects the performance of an ad hoc network is the way routing is implemented in a network. Routing algorithms used in conventional wired networks is impractical in ad hoc networks due to its inability to adapt to the changing topology in a mobile environment. Generally, routing is the process of discovery, selecting, and maintaining paths from a source node to destination node deliver data packets. The goal of every routing algorithm is to direct traffic from sources to destinations, maximizing network performance whilst minimizing costs. This is a main challenge in MANET. Nodes move in an arbitrarily manner and at changing speed, often resulting in connectivity problems. The high mobility and the arbitrarily movement of nodes in MANET causes links between hosts to break frequently.

Evolutionary algorithms (EAs) are stochastic search methods that mimic the metaphor of natural biological evolution and/or the social behavior of species. Examples include how ants find the shortest route to a source of food and how birds find their destination during migration. The behavior of such species is guided by learning, adaptation and evolution [3]. Few evolutionary-based algorithms: genetic algorithms, memetic algorithms, particle swarm, ant-colony systems, and shuffled frog leaping.

Swarm intelligence (SI) [4] is a type of artificial intelligence based on the collective behavior of decentralized, self-organized systems. SI systems are typically made up of a population of simple agents interacting locally with one another and with their environment. The agents follow very simple rules, and although there is no centralized control structure dictating how individual agents should behave, local, and to a certain degree random, interactions between such agents. One such SI based optimization is ANT colony optimization. The phenomenon of emergence found in natural systems show how simple behavioral patterns from participants give rise to complex self-regulatory behavior of the entire system[5].

Ant Colony Optimization (ACO) [6] is a branch of a newly developed form of Artificial Intelligence called Swarm Intelligence. The amount of pheromone deposited varies in quantities. An ant chooses a trail depending on the amount of pheromone deposited on the ground. The larger the concentration of pheromone in a particular trail, the greater is the probability of the trail being selected by an ant. The ant then reinforces the trail with its own pheromone. The idea behind this technique is that the more the ants follow a particular trail, the more attractive is that trail for being followed by other ants. These ants use the notion of stigmergy to communicate indirectly among the ants. They dynamically find a path on the fly. The ACO is an inherently parallelizable technique. The ACO technique is quite amenable to ad hoc

S.Sridhar[1] is with Department of Computer Applications, S.A. Engineering College, Chennai, India. (e-mail : .ssridharmca@yahoo.co.in)

R.Baskaran is with Department of Computer Science & Engineering, CEG, Guindy, Anna University, Chennai, India.

networks due to their similar characteristics. An artificial ant acts like a mobile agent or a node in an ad hoc network. An ant creates a path dynamically on the fly as the routing protocols in MANETs. The communication between ants is very minimal as in MANETs. Since the communication links may change dynamically, node routes packets depending on the link conditions. Similarly, an ant can also exploit the link conditions in the amount of pheromone it deposits on a trail.

II. Evolutionary Algorithms

To mimic the efficient behavior of these species, various researchers have developed computational systems that seek fast and robust solutions to complex optimization problems. The first evolutionary-based technique introduced in the literature was the genetic algorithms (GAs) [7]. GAs was developed based on the Darwinian principle of the 'survival of the fittest' and the natural process of evolution through reproduction. Other developments in EAs include four other techniques inspired by different natural processes: memetic algorithms (MAs) [8], particle swarm optimization (PSO) [9], ant colony systems [10], and shuffled frog leaping (SFL) [11]. A schematic diagram of the natural processes that the five algorithms mimic is shown in Fig. 1.

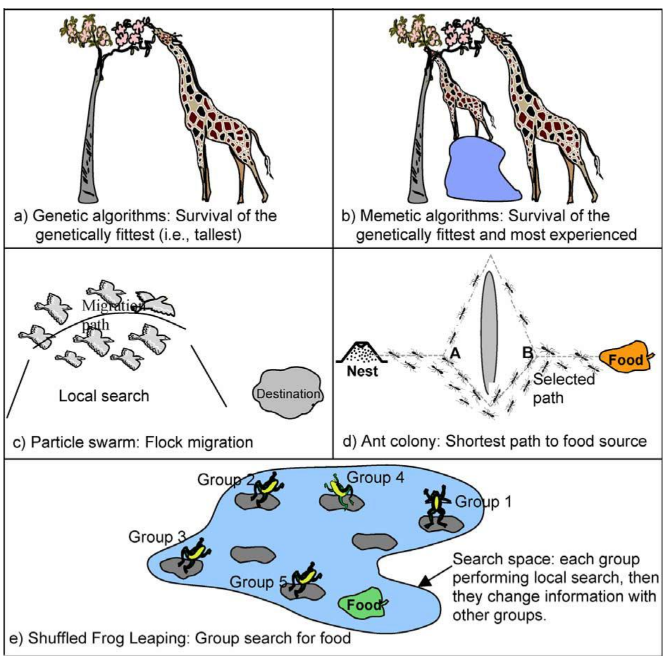

Fig. 1. Schematic diagram of natural evolutionary systems

A. Genetic Algorithms

GAs is inspired by biological systems' improved fitness through evolution [12]. A solution to a given problem is represented in the form of a string, called 'chromosome', consisting of a set of elements, called 'genes', that hold a set of values for the optimization variables [7]. GAs work with a random population of solutions (chromosomes). The fitness of each chromosome is determined by evaluating it against an objective function. To simulate the natural survival of the fittest process, best chromosomes exchange information (through crossover or mutation) to produce offspring chromosomes. The offspring solutions are then evaluated and used to evolve the population if they provide better solutions than weak population members. Usually, the process is continued for a large number of generations to obtain a best-fit (near optimum) solution. Four main parameters affect the performance of GAs: population size, number of generations, crossover rate, and mutation rate. Larger population size (i.e. hundreds of chromosomes) and large number of generations (thousands) increase the likelihood of obtaining a global optimum solution, but substantially increase processing time.

B. Memetic Algorithms

MAs are inspired by Dawkins' notion of a meme [13]. MAs are similar to GAs but the elements that form a chromosome are called memes, not genes. The unique aspect of the MAs algorithm is that all chromosomes and offsprings are allowed to gain some experience, through a local search, before being involved in the evolutionary process [14]. As such, the term MAs is used to describe GAs that heavily use local search [15]. Similar to the GAs, an initial population is created at random. Afterwards, a local search is performed on each population member to improve its experience and thus obtain a population of local optimum solutions. Then, crossover and mutation operators are applied, similar to GAs, to produce offsprings. These offsprings are then subjected to the local search so that local optimality is always maintained. The parameters involved in MAs are the same four parameters used in GAs: population size, number of generations, crossover rate, and mutation rate in addition to a local-search mechanism.

C. Particle Swarm Optimization

PSO was developed by Kennedy and Eberhart [9]. The PSO is inspired by the social behavior of a flock of migrating birds trying to reach an unknown destination. In PSO, each solution is a 'bird' in the flock and is referred to as a 'particle'. A particle is analogous to a chromosome (population member) in GAs. As opposed to GAs, the evolutionary process in the PSO does not create new birds from parent ones. Rather, the birds in the population only evolve their social behavior and accordingly their movement towards a destination [16]. Physically, this mimics a flock of birds that communicate together as they fly. Each bird looks in a specific direction, and then when communicating together, they identify the bird that is in the best location. Accordingly, each bird speeds towards the best bird using a velocity that depends on its current position. Each bird, then, investigates the search space from its new local position, and the process repeats until the flock reaches a desired destination. It is important to note that the process involves both social interaction and intelligence so that birds learn from their own experience (local search) and also from the experience of others around them (global search). The main parameters used in the PSO technique are: the population size (number of birds); number of generation cycles; the maximum change of a particle velocity.

D.Ant-Colony Optimization

Similar to PSO, ant-colony optimization (ACO) algorithms evolve not in their genetics but in their social behavior. ACO was developed by Dorigo et al. [10] based on the fact that ants are able to find the shortest route between their nest and a source of food. This is done using pheromone trails, which ants deposit whenever they travel, as a form of indirect communication. As shown in Fig. 1d, when ants leave their nest to search for a food source, they randomly rotate around an obstacle, and initially the pheromone deposits will be the same for the right and left directions. When the ants in the shorter direction find a food source, they carry the food and start returning back, following their pheromone trails, and still depositing more pheromone. As indicated in Fig. 1d, an ant will most likely choose the shortest path when returning back to the nest with food as this path will have the most deposited pheromone. For the same reason, new ants that later starts out from the nest to find food will also choose the shortest path. Over time, this positive feedback (autocatalytic) process prompts all ants to choose the shorter path [17]. The main parameters involved in ACO are: number of ants ; number of iterations ; exponents a and b; pheromone evaporation rate ; and pheromone reward factor.

E. Shuffled Frog leaping Algorithm

The SFL algorithm, in essence, combines the benefits of the genetic-based MAs and the social behavior-based PSO algorithms. In the SFL, the population consists of a set of frogs (solutions) that is partitioned into subsets referred to as memeplexes. The different memeplexes are considered as different cultures of frogs, each performing a local search. Within each memeplex, the individual frogs hold ideas, that can be influenced by the ideas of other frogs, and evolve through a process of memetic evolution. After a defined number of memetic evolution steps, ideas are passed among memeplexes in a shuffling process [18]. The local search and the shuffling processes continue until defined convergence criteria are satisfied [11]. The main parameters of SFL are: number of frogs ; number of memeplexes; number of generation for each memeplex before shuffling; number of shuffling iterations; and maximum step size.

III. SWARM INTELLIGENCE

In nature several animals tend to live in large swarms like insect colonies, bird flocks or fish schools. The reason is that in the swarm each animal is more effective for evolution than single animals. Many social insects like ants, bees, termites, or wasps live in colonies or hives. They exhibit an astonishingly well-developed social behavior and are able to self-organize, even in the absence of a central leader like a queen. Honey bees communicate locations of food sources by the language of dance that is understood by all nearby honey bees. On the other hand, many insects use a form of indirect communication called stigmergy. Stigmergy works by leaving traces in the environment that can be understood by other insects. Termites use stigmergy to build complex nests by simple rules. A termite constructing a nest deposits material like a mud ball and invests it with pheromones, a chemical that can be smelled by other termites. The smell of pheromones encourages other termites to deposit their material close to freshly deposited pheromones. This way, a group of termites can manage to synchronize so that they all work on the same spot. Swarm Intelligence (SI) is an Artificial Intelligence technique based on the study of collective behavior in decentralized, self-organized systems. The expression "swarm intelligence" was introduced by Beni & Wang in 1989, in the context of cellular robotic systems. It gives rise to complex and often intelligent behavior through simple, unsupervised interactions between a total numbers of autonomous swarm members. Usually there is no centralized control structure dictating how the individual agents should behave, but local interactions between such agents often lead to the emergence of a global behavior. Swarm is considered as biological insects like ants, bees, wasps, fish etc. The quick coordinated flight of a group of birds with very little visual communication and the concerted effort of an ant colony in gathering food, building nests, etc are some of the vivid examples of emergence in natural world. SI has found immense applicability in fields like Robotics, Artificial Intelligence, process optimization, telecommunications, routing, software testing, networking etc.

A. Ant-Colony Optimization

1.1 The Source Of Inspiration: The Ants

Ant as a single individual has a very limited effectiveness. But as a part of a well-organized colony, it becomes one powerful agent, working for the development of the colony. The ant lives for the colony and exists only as a part of it. Each ant is able to communicate, learn, cooperate, and all together they are capable of develop themselves and colonies a large area. They manage such great successes by increasing the number of individuals and being exceptionally well organized. The self organizing principles they are using allow a highly coordinated behavior of the colony. Stigmergy is defined as a method of indirect communication in a self-organizing emergent system where its individual parts communicate with one another by modifying their local environment. Ants communicate to one another by laying down pheromones along their trails, so where ants go within and around their ant colony is a stigmergic system. In many ant species, ants walking from or to a food source, deposit on the ground a substance called *pheromone*. Other ants are able to smell this pheromone, and its presence influences the choice of their path, that is, they tend to follow strong pheromone concentrations. The pheromone deposited on the ground forms a pheromone trail, which allows the ants to find good sources of food that have been previously identified by other ants. Using random walks and pheromones within a ground containing one nest and one food source, the ants will leave the nest, find the food and come back to the nest. After some time, the way being used by the ants will converge to the shortest path.

1.2 The Pheromones

International Journal on Applications of Information and Communication Engineering
Volume 1: Issue 2: February 2015 , Pages:35-40 www.aetsjournal.com

Pheromones represent in some ways the common memory. The fact that it is external and not a part of the ants / agents, confers to it an easy access for everyone. The memory is saved in without regarding the configuration of the ground, the number of ants, etc. It is totally independent, and still remains extremely simple. During implementation two different types of pheromones are used. The first one is represented in red and is let by the ants which do not carry the food. We will call it the *Away* pheromone, as it means that the ant is going away from the nest. Oppositely, the ants which carry the food to bring it back to the nest let a blue trace behind them, the *Back* pheromone. Pheromones just proceed to one task: nature will take care of it in the real life, although it is a simple process in algorithms. In course of time, a global reduction of the pheromones by a certain factor is applied, simulating the *evaporation* process. Thus the non-succeeding path will see their concentration of pheromones reduced, although good solutions will stay full of pheromones as the ants keep using it.

1.3 ACO Algorithm

Ant-colony optimization (ACO) algorithms evolve not in their genetics but in their social behavior. ACO was developed by Dorigo et al. [10] based on the fact that ants are able to find the shortest route between their nest and a source of food. This is done using pheromone trails, which ants deposit whenever they travel, as a form of indirect communication. As shown in Fig. 2, when ants leave their nest to search for a food source, they randomly rotate around an obstacle, and initially the pheromone deposits will be the same for the right and left directions. When the ants in the shorter direction find a food source, they carry the food and start returning back, following their pheromone trails, and still depositing more pheromone.

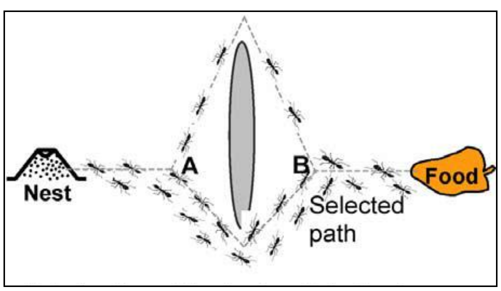

Fig. 2. Ants choosing path

As indicated in Fig. 2, an ant will most likely choose the shortest path when returning back to the nest with food as this path will have the most deposited pheromone. For the same reason, new ants that later starts out from the nest to find food will also choose the shortest path. Over time, this positive feedback (autocatalytic) process prompts all ants to choose the shorter path[17]. Implementing the ACO for a certain problem requires a representation of S variables for each ant, with each variable i has a set of ni options with their values lij, and their associated pheromone concentrations { T_{ij} }; where i=1, 2,., S, and j=1, 2,., ni. As such, an ant is consisted of S values that describe the path chosen by the ant. In the ACO, the process starts by generating m random ants (solutions). An ant k (k=1, 2,., m) represents a solution string, with a selected value for each variable. Each ant is then evaluated according to an objective function. Accordingly, pheromone concentration associated with each possible route (variable value) is changed in a way to reinforce good solutions, as follows:

$$T_{ij}(t) = P\,T_{ij}(t-1) + D\,T_{ij};\; t = 1,2,\ldots J \qquad (1)$$

where J is the number of iterations (generation cycles); T_{ij} (t) is the revised concentration of pheromone associated with option lij at iteration t, T_{ij} (t-1) is the concentration of pheromone at the previous iteration (t-1); D T_{ij} = change in pheromone concentration; and P is the pheromone evaporation rate (0–1). The reason for allowing pheromone evaporation is to avoid too strong influence of the old pheromone to avoid premature solution stagnation. In Eq. (1), the change in pheromone concentration DT_{ij} is calculated as :

$$D\,T_{ij} = \sum_{k=1}^{m} \frac{R}{\text{fitness } k} \qquad (2)$$

where R is a constant called the pheromone reward factor; and fitness k is the value of the objective function (solution performance) calculated for ant k. It is noted that the amount of pheromone gets higher as the solution improves. Therefore, for minimization problems, the Eq. (2) shows the pheromone change as proportional to the inverse of the fitness. In maximization problems, on the other hand, the fitness value itself can be directly used. Once the pheromone is updated after an iteration, the next iteration starts by changing the ants' paths (i.e. associated variable values) in a manner that respects pheromone concentration and also some heuristic preference. As such, an ant k at iteration t will change the value for each variable according to the following probability:

$$P_{ij}(k.t) = [\,T_{ij}(t)]^{\alpha}\ X\ [n_{ij}]^{\beta}\ /\ \sum_{Tij} [T_{ij}(t)]^{\alpha}\ X\ [n_{ij}]^{\beta} \qquad (3)$$

where Pij(k, t) = probability that option lij is chosen by ant k for variable i at iteration t; T_{ij} (t) = pheromone concentration associated with option lij at iteration t; n_{ij} =heuristic factor for preferring among available options and is an indicator of how good it is for ant k to select option lij (this heuristic factor is generated by some problem characteristics and its value is fixed for each option lij); and a and b are exponent parameters that control the relative importance of pheromone concentration versus the heuristic factor [19]. Both α and β can take values greater than zero and can be determined by trial and error. Based on the previous discussion, the main parameters involved in ACO are: number of ants m; number of iterations t; exponents α and β; pheromone evaporation rate r; and pheromone reward factor R.

IV. ACO BASED ROUTING PROTOCOLS

The nature of swarms largely resembles mobile ad-hoc networks (MANETs) and that is why ideas from swarm animals like ants and bees are used for creating suitable routing protocols for MANETs as well as wireless sensor networks. They are more efficient, more robust and are able to discover multiple paths. There exist a number of swarm intelligence based protocols but the most important are ant based protocols.

A. Ant-AODV

Ant- AODV is a hybrid protocol that is able to provide reduced end-to-end delay and high connectivity as compared to AODV. AODV does the reactive part and an ant-based approach does the proactive one. The main goal of the ant algorithm here is to continuously create routes in the attempt to reduce the end-to-end delay and the network latency, increasing the probability of finding routes more quickly, when required [20]. Ant-AODV's artificial pheromone model is based on the number of hops and its goal is to discover the network topology, without any other specific functions, as opposed to most ACO algorithms. Route establishment in conventional ant-based routing techniques is dependent on the ants visiting the node and providing it with routes. The nodes also have capability of launching on-demand route discovery to find routes to destinations. The use of ants with AODV increases the node connectivity (the number of destinations for which a node has un-expired routes), which in turn reduces the amount of route discoveries and also the route discovery latency. This makes Ant-AODV hybrid routing protocol suitable for realtime data and multimedia communication. Ant-AODV uses route error messages (RERR) to inform upstream nodes of a local link failure similar to AODV. Routing table in Ant- AODV is common to both ants and AODV. Frequent HELLO broadcasts are used to maintain a neighbor table.

B. Ant-DSR

Ant Dynamic Source Routing (Ant-DSR) is a reactive protocol that implements a proactive route optimization method through the constant verification of cached routes [21]. This approach increases the probability of a given cached route express the network reality. Mobile nodes are required to maintain route caches that contain the source routes of which the mobile is aware. Entries in the route cache are continually updated as new routes are learnt. The protocol consists of two major phases: route discovery and route maintenance. In Ant DSR (ADSR) the Forward ant (FANT) and backward ant (BANT) packets are added in the route request and route reply of DSR respectively. Forward ants are used to explore new paths in the network. Ants measure the current network state for instance by trip times, hop count or Euclidean distance travelled. Backward ants the information collected by the forward ant.

C. Ant-DYMO

Ant-DYMO is a hybrid protocol that uses an ant-based approach in its proactive phase while DYMO is the bass for the reactive one [22]. Ant-DYMO is a hybrid and multi hop algorithm. Nodes acquire information on their neighborhood by the limited flooding of Hello messages. Each node creates its routing probability table similar to ACO's pheromone table. Ant-DYMO defines two types of artificial ants: explorer ant (EANT), responsible for creating routes to its source and search ant (ARREQ), responsible for searching for a specific destination. The EANTs carry the information on the destination node and create (or enforce) pheromone trails along the way. The EANTs carry the address of the source node and also a list containing every intermediate node it has passed by. ARREQ has main goal to search for a specific destination, and it inherits the format of DYMO's RREQ, adding a probabilistic search mechanism that takes into account the level of pheromones on the paths.

D. HOPNET

The HOPNET algorithm also involves characteristics of Zone Routing Protocol, a hybrid protocol which combines benefits of proactive and reactive protocols. HOPNET [23] is a hybrid routing algorithm for MANETs which involves Swarm Intelligence to solve routing problems. The algorithm has features extracted from ZRP and DSR protocols. The network is divided into zones which are the node's local neighborhood. The size of the zone is not determined locally but by the radius length measured in hops. Boundary nodes are at a distance from the central node. All other nodes less than the radius are interior nodes. In order to construct a zone, a node, and determining border nodes, a node needs to know its local neighbors. It has two routing tables, Intrazone Routing Table (IntraRT) and Interzone Routing Table (InterRT). IntraRT is a routing table maintained proactively by HOPNET. Its goal is to map the deposit of pheromone on each node within its zone. InterRT is a responsible routing table for storing routes to a destination out of its zone i.e. when a node fails to find the destination within its zone in the IntraRT table. The route discovery within a zone is accomplished by using IntraRT. There are four elements in the routing table for a particular(row, column) pair: Pheromone, Visited times, Hops, SeqNum. The pheromone value gets updated by the ants as they traverse the links. The ants change the concentration of the pheromone value on their journey to the destination and on their way back to the source. The data structure of the ant contains six fields: Source, Destination, SeqNum, Type, Hops and Path.

E. AD-ZRP

A self-configuring reactive routing protocol for Wireless Sensor Networks based on HOPNET AD-ZRP also consists of ZRP similar to HOPNET, but it is based on dynamic zones which, acting together with ACO, deals with the restrictions of WSNs and yet improves the route discovery and the route maintenance through pheromone [24]. It helps us to handle important routing problems in ad hoc networks such as route discovery and broken routes. But HOPNET is not a suitable routing protocol for WSNs. AD-ZRP is proposed as a reactive routing protocol to avoid sending ants periodically into their zones and thus bringing additional overhead to the sensor network. Different from HOPNET, which uses fixed-sized zones defined by the zone radius, our approach uses dynamic zones to minimize the latency while reducing the network overhead. They are defined by the presence of pheromone on routes between the source nodes and any other node in the network. Initially all zones in the network are empty. After each data packet transmission to an unknown destination, a new route is added to the zone. OPNET uses two routing tables to perform intrazone and interzone routing separately, whereas Ad-ZRP uses only IntraRT as routing table structure. To accomplish these routing operations, a new collection of ants is presented: internal transport ant (ITA) and exploratory transport ant(ETA).

F. SDVR

Swarm Distance Vector Routing Protocol is a unicast ondemand routing algorithm based on optimization of three QoS parameters delay, jitter and energy [25]. This avoids the overhead of having three independent routing algorithms, one for each QoS metric. The mechanism was based on information obtained from periodically transmitted backward ANTs resulting in reinforced path-pheromone levels. The main purpose of QoS routing is to find a feasible path that has sufficient resources to satisfy the constraints. A fundamental problem in QoS routing is to find a path between a source and destination that satisfies two or more end-to-end QoS constraints. The source nodes maintain a routing table that contains entries of neighboring nodes to reach destination nodes. When the source receives the BANT, it has an entry for reaching the destination through one of its neighbors. Since duplicate FANTs are not discarded, the destination node may send multiple BANTs back to the source. Once the destination receives the FANT, it sends a BANT back to the source using the same path the FANT has travelled.

V. CONCLUSION

A review on the evolutionary algorithms like GA, MA, PSO, ACO and SFL is made, which are used as computational systems that seek fast and robust solutions to complex optimization problems. Swarm Intelligence, an Artificial Intelligence technique based on the study of collective behavior in decentralized, self-organized systems is also discussed which can be the emergent collective intelligence of groups of simple agents. A detailed discussion of Ant Colony Optimization concepts that explains how shortest and optimized path can be chosen in Mobile Adhoc network routing is presented. Description on Ants, Pheromones and ACO algorithm were also discussed. Finally Routing in MANETs using various types of ANT routing protocols like SDVR, AD-ZRP, HOPNET, Ant-DYMO, Ant-DSR, Ant-AODV were discussed. The above analyses clearly shows that MANET routing can be made more efficient and can increase the Quality of service by introducing Ant colony optimization techniques along with ANT based routing protocols. The extended work of this article is to implement a trust based MANET routing protocol along with ANT optimization incorporating energy and security in routing.

REFERENCES

[1] Charles E. Perkins, Ad Hoc Networking, Addison-Wesley, 2001.
[2] C.K. Toh, Ad Hoc Mobile Wireless Networks: Protocols and Systems, Prentice Hall, 2001.
[3] Lovbjerg M. Improving particle swarm optimization by hybridization of stochastic search heuristics and self-organized criticality. Masters Thesis, Aarhus Universitet, Denmark; 2002.
[4] E. Bonabeau, M. Dorigo, and G. Theraulaz, Swarm Intelligence – From Natural to Artificial Systems. New York: Oxford University Press, 1999.
[5] G. Serugendo, N. Foukia, S. Hassas, A. Karageorgos, S. K. Mostefaoui, O. F. Rana, M. Ulieru, P. Valckenaers, C. Van Aart, "Self-organisation: Paradigms and applications", Engineering Self-Organizing Systems, Vol. 2977, 2004
[6] Dorigo, M., & Caro, G.D 1999. Ant Algorithms for Discrete Optimization. Artificial Life.
[7] Holland J. Adaptation in natural and artificial systems. Ann Arbor, MI: University of Michigan Press; 1975.
[8] Moscato P. On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms. Technical Report Caltech Concurrent Computation Program, Report 826, California Institute of Technology, Pasadena, CA; 1989.
[9] Kennedy J, Eberhart R. Particle swarm optimization. Proceedings of the IEEE international conference on neural networks (Perth, Australia), 1942–1948. Piscataway, NJ: IEEE Service Center; 1995.
[10] Dorigo M,Maniezzo V,ColorniA.Ant system: optimization by a colony of cooperating agents. IEEE Trans SystMan Cybern 1996;26(1):29–41.
[11] Eusuff MM, Lansey KE. Optimization of water distribution network design using the shuffled frog leaping algorithm. J Water Resour Plan Manage 2003;129(3):210–25.
[12] Goldberg DE. Genetic algorithms in search, optimization and machine learning. Reading, MA: Addison-Wesley Publishing Co; 1989.
[13] Dawkins R. The selfish gene. Oxford: Oxford University Press; 1976.
[14] Merz P, Freisleben B. A genetic local search approach to the quadratic assignment problem. In: Ba¨ck CT, editor. Proceedings of the 7th international conference on genetic algorithms. San Diego, CA: Morgan Kaufmann; 1997. p. 465–72.
[15] Moscato P, Norman MG. A memetic approach for the traveling salesman problem—implementation of a computational ecology for combinatorial optimization on message-passing systems. In: Valero M, Onate E, Jane M, Larriba JL, Suarez B, editors. International conference on parallel computing and transputer application. Amsterdam, Holland: IOS Press; 1992. p. 177–86.
[16] Shi Y, Eberhart R. A modified particle swarm optimizer. Proceedings of the IEEE international conference on evolutionary computation. Piscataway, NJ: IEEE Press; 1998. p. 69–73.
[17] Dorigo M, Gambardella LM. Ant colonies for the traveling salesman problem. Biosystems, Elsevier Sci 1997;43(2):73–81.
[18] Liong S-Y, Atiquzzaman Md. Optimal design of water distribution network using shuffled complex evolution. J Inst Eng, Singapore 2004;44(1):93–107.
[19] MaierHR, Simpson AR, ZecchinAC, FoongWK, PhangKY, SeahHY, et al. Ant colony optimization for design of water distribution systems. J Water Resour Plan Manage 2003;129(3):200–9.
[20] S. Marwaha, C. Khong, and T. Srinivasan, "Ant-Aodv – Mobile agents based routing protocol for mobile ad hoc networks," in Proceedings *of IEEE Globecom*, 2002.
[21] M. Aissani, M. Fenouche, H. Sadour, and A. Mellouk, "Ant-DSR: cache maintenance based routing protocol for mobile ad-hoc networks," *AICT 2007. The Third Advanced International Conference* in *Telecommunications*, pp. 35–35. 2007.
[22] J. Alex Pontes Martins, S. Luis O. B. Correia, J. Celestino J´unior, "Ant-DYMO: A Bio-Inspired Algorithm for MANETS", 2010 *7th International Conference on Telecommunications*.2010.
[23] J. Wang, E. Osagie, P. Thulasiraman, R. Thulasiram, Hopnet: a hybrid ant colony optimization routing algorithm for mobile ad hoc network, *Ad Hoc Networks* 7 (4) (2009) 690–705
[24] A. Massayuki Okazaki and A. Augusto Frohlich, "AD-ZRP: Antbased Routing Algorithm for Dynamic Wireless Sensor Networks," 2011, *18th International Conference on Telecommunications.*
[25] R. Asokan, A. M .Natarajan, and A.Nivetha, "A Swarm-based Distance Vector Routing to Support Multiple Quality of Service (QoS) Metrics in Mobile Adhoc Networks," *Journal of Computer Science* 3 (9): 700-707, 2007.