COMPARATIVE STUDY OF ROUTING ALGORITHMS BASED ON OPTIMIZATION TECHNIQUES

H.Fathima[*]

Abstract

A new agent-based routing algorithm using optimization techniques is implemented in this paper. The different optimization techniques are Ant, Bee, Ant Bee, Ant GA, Ant PSO, GA, PSO, Ant Dijkstra are the combinations used in the packet delivery between the networks. Routing is the process of selecting best paths in a network. The Routing is also used to forward network traffic among networks. Otherwise routing is a process of carrying the data from source to destination in the network. The outputs of these algorithms are determined by the simulation time and throughput. The experiments are implemented with the NS2 software platform, which is based on the basics of C, C++, and TCL Scripting Language. The results of these algorithms show that Ant-PSO is much better than the other algorithms in the packet delivery between the networks.

Keywords

Routing, Networks, Genetic Algorithm, Ant Bee, PSO Algorithm, Packet Delivery.

[*] Assistant professor, Department of computer science, Bharathidasan College of arts and science, Ellispettai, Erode-16

INTRODUCTION

The Routing is the process of selecting paths in a network by which a packet travels from a source to a destination. Routing plays a critical role in communication networks in determining the overall network performance in terms of throughput and transmission delay. Routing algorithms are currently implemented through the information contained in routing table independently available at each node of the network (Kurose, 2005). Traditional routing algorithms are not intelligent and do not have enough flexibility to satisfy new routing demands and they need human assistance in order to adapt themselves to failure and changes. Insects that live in colonies, ants, bees, wasps, and termites, have fascinated researchers: every single insect in a colony seems to have its own agenda, and yet an insect colony looks so organized. This characteristic in social insects not only has consequences on the study of social insects, but also can bring much insight into the design of algorithms and provides us with powerful tools to transfer knowledge about social insects to the field of intelligent system design. This approach emphasizes distributed ness, direct or indirect interactions among relatively simple agents, flexibility, and robustness. The best known and most popular swarm algorithms are the ant algorithms. The number of successful applications is exponentially growing in combinatorial optimization, communications networks, and robotics. Another swarm algorithm family, known as Bee algorithms, tries to model the natural behavior of real honeybees. The algorithm, which is proposed in this study, is inspired by earlier study on ant and bee colonies (DiCaro and Dorigo, 1998; Teodorovic et al., 2006). Real ants have been shown to be able to find shortest paths using only the pheromone trail deposited by other ants. The study by Schoonderwoerd et al. (1996, 1997) was the first attempt to apply it to a routing problem. Their algorithm called Ant-Based Control (ABC) was applied to the case of virtual circuit based on symmetric networks. The Ant Net algorithm introduced by DiCaro and Dorigo (1998) for routing in packet switching networks outperformed all conventional algorithms on several packet-switched communications networks in their simulations. The foraging metaphor has been put to work successfully in solving the problem of telecommunications routing.

ANT ROUTING ALGORITHM

Deneubourg showed that self-organization enables ants and other social insects to succeed in complex activities using simple algorithms and minimal individual intelligence. It was therefore a natural step to apply antlike approaches to solving antlike problems, specifically the very general AI problem of search. The foraging metaphor has been put to work successfully in solving the problem of telecommunications routing. Rued Schoonderwoerd of Hewlett-Packard labs collaborated with a group of scientists to create the world's first ant-based routing system. As anyone who has used the Internet will know, communications networks are characteristically unpredictable. Sudden interest in a particular web site or a local crisis will lead to surges of network activity, which must somehow be routed efficiently, minimizing both delays and congestion. The network must therefore dynamically route calls/requests through quieter sections of the network. Congestion in a particular section of the network can be seen as analogous to depletion of a food source near an ant colony, causing the ants to search for new routes, dynamically updating the virtual pheromone trail between nodes. In the system developed by Schoonderwoerd et al., antlike agents are sent randomly between nodes from time to time, updating each node's routing table as they go with information regarding how long the journey from their origin took, and which nodes they have used on the way. The routing table contains a list of the node's immediate neighbors, and probabilities associated with using those neighbors as the next step on the journey to each target node on the network. The fastest ants will have a positive effect on the probability scores of the nodes they have used, while slow ants will have a negative effect. A more recent algorithm for Internet routing designed by Dorigo and DiCaro [8] has, in simulation, outperformed all other routing methods, including the current standard routing protocol of the Internet, Open Shortest Path First.

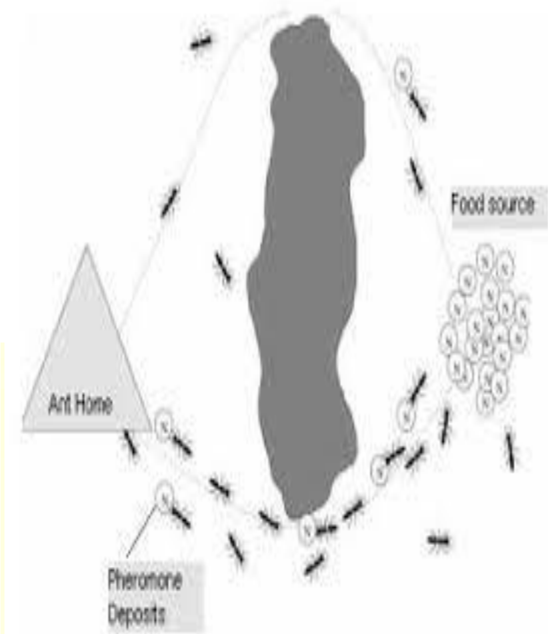


Fig 1: Ant Algorithm

BEE ROUTING ALGORITHM

A colony of honeybees can extend itself over long distances (up to 14 km) and in multiple directions simultaneously to exploit a large number of food sources. A colony prospers by deploying its foragers to good fields. In principle, flower patches with plentiful amounts of nectar or pollen that can be collected with less effort should be visited by more bees, whereas patches with less nectar or pollen should receive fewer bees. The foraging process begins in a colony by scout bees being sent to search for promising flower patches. Scout bees move randomly from one patch to another. During the harvesting season, a colony continues its exploration, keeping a percentage of the population as scout bees. When they return to the hive, those scout bees that found a patch which is rated above a certain quality threshold (measured as a combination of some constituents, such as sugar content) deposit their nectar or pollen and go to the "dance floor" to perform a dance known as the waggle dance. This dance is essential for colony communication, and contains three pieces of information regarding a flower patch: the direction in which it will be found, its distance from the hive and its quality rating (or fitness). This information helps the colony to send its bees to flower patches precisely, without using guides or maps. Each individual's knowledge of the outside environment is gleaned solely from the waggle dance. This dance enables the colony to evaluate the relative merit of different patches according to both the quality of the food they provide and the amount of energy needed to harvest it. After waggle dancing inside the hive, the dancer (i.e. the scout bee) goes back to the flower patch with follower bees that were waiting inside the hive. More follower bees are sent to more promising patches. This allows the colony to gather food quickly and efficiently. While harvesting from a patch, the bees monitor its food level. This is necessary to decide upon the next waggle dance when they return to the hive. If the patch is still good enough as a food source, then it will be advertised in the waggle dance and more bees will be recruited to that source. Waggle dance is a term used in beekeeping and ethnology for a particular figure-eight dance of the honeybee. By performing this dance, successful foragers can share with their hive mate's information about the direction and distance to patches of flowers yielding nectar and pollen, to water sources, or to new housing locations. [1][2] Thus the waggle dance is a mechanism whereby successful foragers can recruit other bees in their colony to good locations for collecting various resources. It was once thought that bees had two distinct recruitment dances — round dances and waggle dances — the former for indicating nearby targets and the latter for indicating distant targets, but it is now known that a round dance is simply a waggle dance with a very short waggle run (see below). Austrian ethnologist and Nobel laureate Karl von Frisch was one of the first who translated the meaning of the waggle dance.[3]

Fig 2: Bee Algorithm

ANT-BEE ROUTING ALGORITHM

The ants try to find the suitable solution from one node to another then bees update the routing table based on ants collected data. In the ABR algorithm, each node generates an ant at regular intervals. The ant goes in the random destination. The forward ants watches and collect the network traffic condition such as node identifiers and trip time. When a forward ant reaches a node, the ant will continue its trip to the destination. But if the node is the destination, the ant will be killed and a bee starts working. In the destination, a bee is created and all data which were collected by the forward ant are delivered to the newborn bee. Then the artificial bee goes back to the source node by moving along the same path as before but in the opposite direction. A colony prospers by deploying its foragers to good fields. In principle, flower patches with plentiful amounts of nectar or pollen that can be collected with less effort should be visited by more bees, whereas patches with less nectar or pollen should receive fewer bees. The foraging process begins in a colony by scout bees being sent to search for promising flower patches. Scout bees move randomly from one patch to another. During the harvesting season, a colony continues its exploration, keeping a percentage of the population as scout bees. When they return to the hive, those scout bees that found a patch which is rated above a certain quality threshold (measured as a combination of some constituents, such as sugar content) deposit their nectar or pollen and go to the "dance floor" to perform a dance known as the waggle dance. This dance is essential for colony communication, and contains three pieces of information regarding a flower patch: the direction in which it will be found, its distance from the hive and its quality rating (or fitness). This information helps the colony to send its bees to flower patches precisely, without using guides or maps. Each individual's knowledge of the outside environment is gleaned solely from the waggle dance. This dance enables the colony to evaluate the relative merit of different patches according to both the quality of the food they provide and the amount of energy needed to harvest it.

Fig 3: Ant and Bee Routing

GENETIC ROUTING ALGORITHM

John Holland, from the University of Michigan began his work on genetic algorithms at the beginning of the 60s. A first achievement was the publication of Adaptation in Natural and Artificial System[7] in 1975. Holland had a double aim: to improve the understanding of natural adaptation process, and to design artificial systems having properties similar to natural systems[8]. The basic idea is as follow: the genetic pool of a given population potentially contains the solution, or a better solution, to a given adaptive problem. This solution is not "active" because the genetic combination on which it relies is split between several subjects.Only the association of different genomes can lead to the solution. Simplistically speaking, we could by example consider that the shortening of the paw and the extension of the fingers of our brachiosaurus are controlled by 2 "genes". No subject has such a genome, but during reproduction and crossover, new genetic combination occurs and, finally, a subject can inherit a "good gene" from both parents.It's important to understand that the functioning of such an algorithm does not guarantee success. We are in a stochastic system and a genetic pool may be too far from the solution, or for example, a too fast convergence may halt the process of evolution. These algorithms are nevertheless extremely efficient, and are used in fields as diverse as stock exchange, production scheduling or programming of assembly robots in the automotive industry.

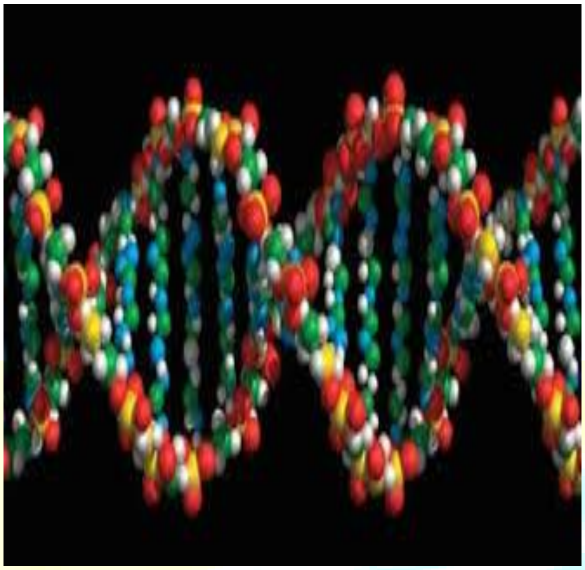


Fig 4: Genetic Routing Algorithm

ANT-GA ROUTING ALGORITHM

The GA and ACO are both widely used evolutionary optimization algorithms, which have their own advantages and disadvantages. GA performs crossover and mutation operations to recombine individual (chromosomes).It has strong global search capability. But its convergence speed is slow. ACO, which mimics the collective behavior of ant colony and adopts the positive feedback based on pheromone, behaves well in local search capability and convergence rate. But it is sensitive to initial parameters and easy to get into the local optimum, especially when the population is large. Many hybrid methods of GA and ACO are developed.

Fig 5.1: Ant Genetic Routing

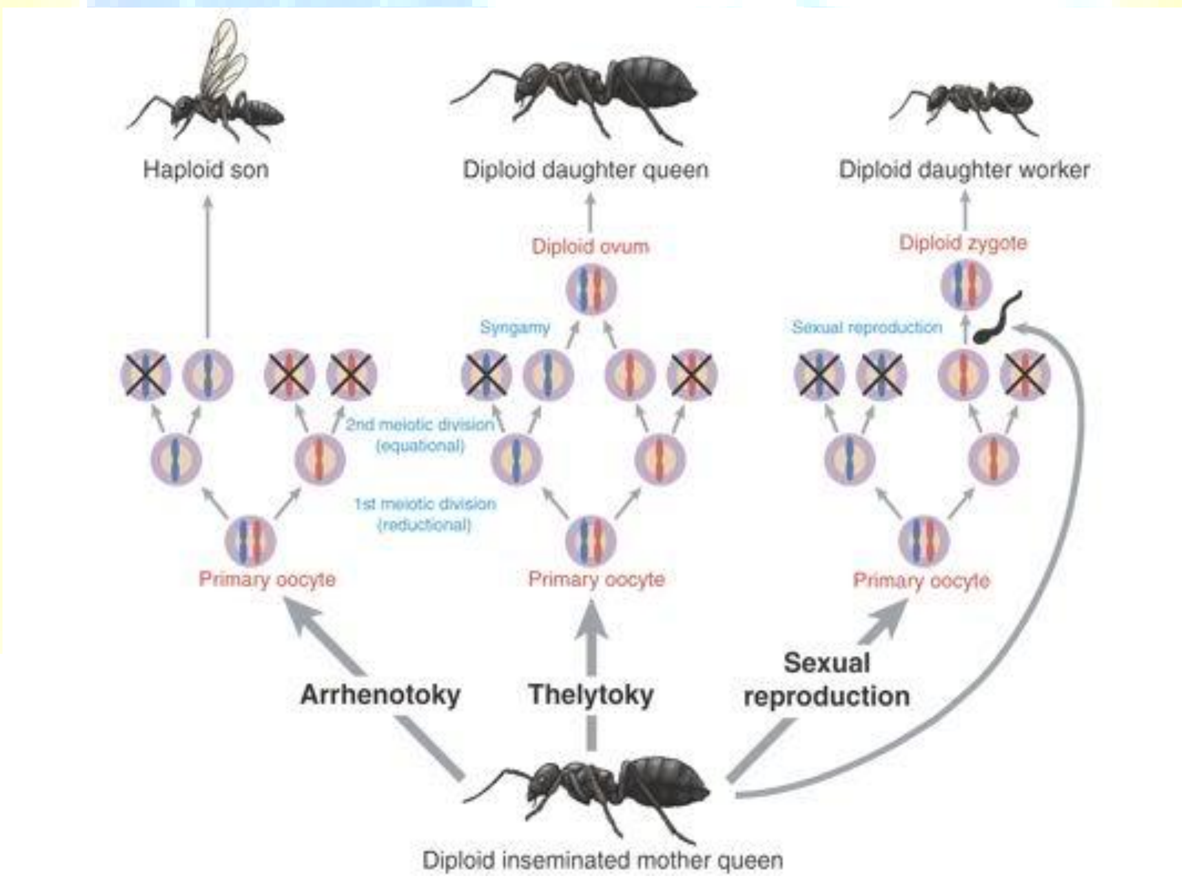


Fig 5.2 Ant Genetic Routing Algorithm

PSO ROUTING ALGORITHM

Particle Swarm Optimization (PSO) algorithm is a new bionics algorithm. Because of its good characteristics, the algorithm has aroused many researchers' interests. Most researches were centered on the control abilities of the algorithm on the strongly nonlinear and naturally unstable objects. The PenduBot is the most typical experiment object in the field of automatic control. The thesis concentrated on the position and velocity evolution equations and fitness functions selection. Compared with the simulation results on PenduBot by mature algorithm LQR, the PSO algorithm performed better control abilities. Referring to the parameters from simulation, the two-link PenduBot can stabilize itself in the physical experiment. It was a reference to the algorithm convergence and anti-interference properties. Particle Swarm Optimization is an optimization technique, which provides an evolutionary based search. This search algorithm was introduced by Dr Russ Eberhart and Dr James Kennedy in 1995. James is a social psychologist and is from the Bureau of Labor Stats, Washington DC. Russ is an electrical engineer from Purdue School of engineering and technology, Indianapolis. The term PSO refers to a relatively new family of algorithms that may be used to find optimal or near to optimal solutions to numerical and qualitative problems. It is implemented easily in most of the programming languages. Since the core of the program can be written in a single line of code and has proven both very effective and quick when applied to a diverse set of optimization problems. PSO algorithms are especially useful for parameter optimization in continuous, multi-dimensional search spaces. PSO is mainly inspired by social behavior patterns of organisms that live and interact within large groups. In particular, PSO incorporates swarming behaviors observed in flocks of birds, schools of fish, or swarms of bees. We explain the PSO algorithm in detail and demonstrate its performance on one-, two- and multi-dimensional continuous search problems. PSO is originally attributed to Kennedy, Eberhart and Shi [1] [2] and was first intended for simulating social behavior [3]. The algorithm was simplified and it was observed to be performing optimization. The book by Kennedy and Eberhart [4] describes many philosophical aspects of PSO and swarm intelligence. An extensive survey of PSO applications is made by Poli [5] [6].

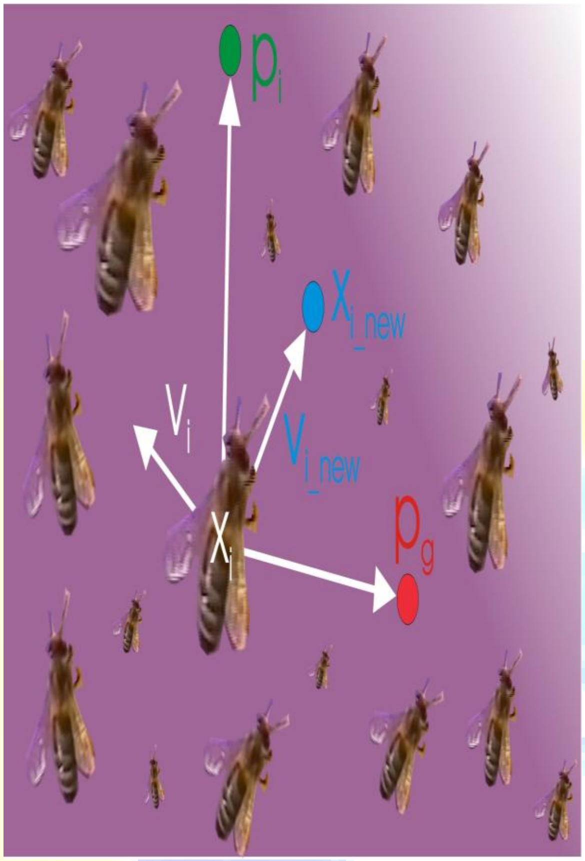


Fig 6: PSO Routing Algorithm

ANT-DIJKSTRA'S ALGORITHM

Ant systems are a population based approach. In this respect it is similar to genetic algorithms. There is a population of ants, with each ant finding a solution and then communicating with the other ants. Ant colony optimization algorithms have been applied to many combinatorial optimization problems, ranging from quadratic assignment to protein folding or routing vehicles and a lot of derived methods have been adapted to dynamic problems in real variables, stochastic problems, multi-targets and parallel implementations. It has also been used to produce near-optimal solutions to the travelling salesman problem. They have an advantage over simulated annealing and genetic algorithm approaches of similar problems when the graph may change dynamically; the ant colony algorithm can be run continuously and adapt to changes in real time.

This is of interest in network routing and urban transportation systems.The first ACO algorithm was called the Ant system [8] and it was aimed to solve the travelling salesman problem, in which the goal is to find the shortest round-trip to link a series of cities. The general algorithm is relatively simple and based on a set of ants, each making one of the possible round-trips along the cities.

Dijkstra's algorithm, conceived by computer scientist Edsger Dijkstra in 1956.It is a graph search algorithm that solves the shortest path problem producing a shortest path tree. This algorithm is often used in routing and as a subroutine in other graph algorithms.For a given source vertex (node) in the graph, the algorithm finds the path with lowest cost (i.e. the shortest path) between that vertex and every other vertex. For example, if the vertices of the graph represent cities and edge path costs represent distances between pairs of cities connected by a direct road, Dijkstra's algorithm can be used to find the shortest route between one city and all other cities. As a result, the shortest path algorithm is widely used in network routing protocols, most notably IS-IS and OSPF (Open Shortest Path First).

Fig 7:Ant Dijkstra Algorithm

ANT-PSO ROUTING ALGORITHM

The quadratic crossover operator is a nonlinear multiparent crossover operator, which makes use of three particles of the swarm to produce a particle, which lies at the point of minima of the quadratic curve passing through the three selected particles. The new particle is accepted in the swarm irrespective of the fact whether it is better or worse than the worst particle present in the swarm. In this way the search is not limited to the region around the current best location but is in fact more diversified in nature.

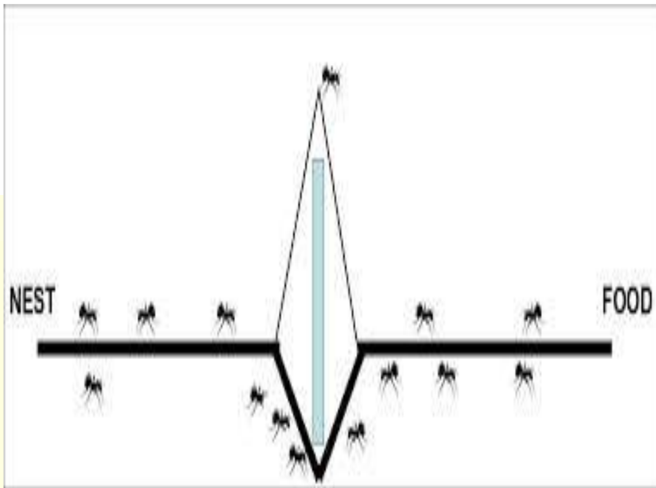
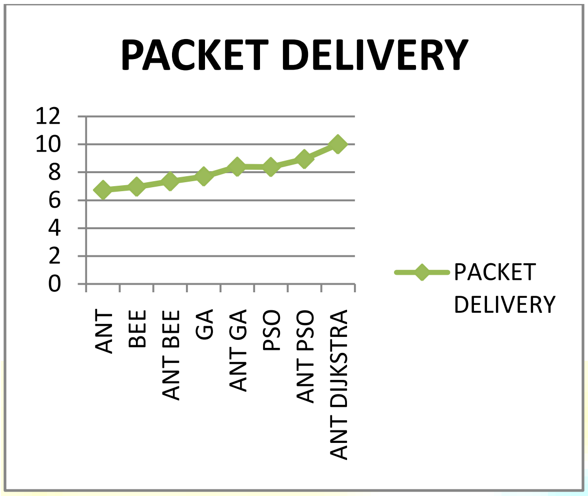


Fig 8: Ant PSO Routing Algorithm

EXPERIMENTAL RESULTS

ALGORITHM	PACKET DELIVERY
ANT	6.721
BEE	6.956
ANT BEE	7.341
GA	7.693
ANT GA	8.386
PSO	8.367
ANT PSO	8.934
ANT DIJKSTRA	9.999

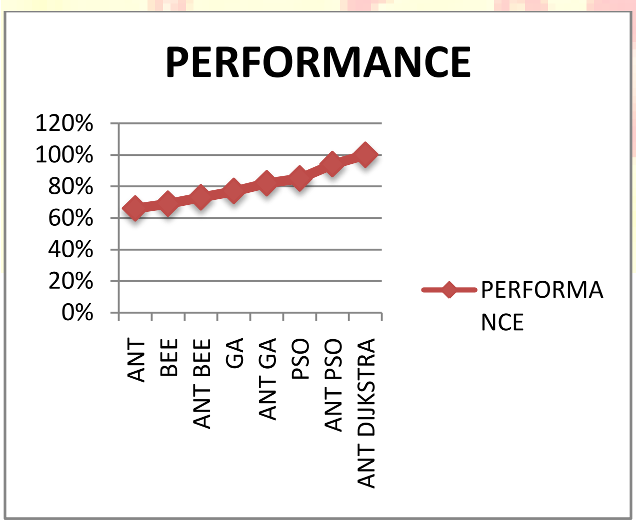
Table 1: Packets Dropped Ratio



Graph1:Packets Dropped Ratio

ALGORITHM	PERFORMANCE
ANT	66%
BEE	69%
ANT BEE	73%
GA	77%
ANT GA	82%
PSO	85%
ANT PSO	94%
ANT DIJKSTRA	100%

Table 2:Performance Analysis Of Routing Algorithms



Graph 2: Performance of Algorithms

CONCLUSION

The Ant PSO algorithm combines ant with bees to create fault tolerant routing algorithm. The other algorithm uses forward ants and backward bees to update the routers in the computer network. The Ant PSO algorithm is tested in both the static and dynamic routing in the networks. In the static network all the routers work properly The Ant PSO algorithm performance much better than all other algorithms. The Ant PSO algorithm delivers the packets with much higher speed while keeping the throughput almost at the same level like other algorithms. The Ant PSO algorithm is much suitable for dynamic and real computer networks where the failures of some routers are anticipated.

REFERENCES

1. Marco Dorigo and Thomas Stutz. The Ant Colony Optimization Metaheuristic: Algorithms, Applications and Advances. of International Series in Operations Research and Management Science. Kluwer Academic Publishers, 2003.
2. Ibrahim H. Osman and James P. Kelly, editors, Proceedings of the Meta-heuristics Conference, pages 53-62, Norwell, USA, 1995. Kluwer Academic Publishers.
3. N. Holden and A.A. Freitas. Hierarchical Classification of Protein- Coupled Receptors with a PSO/ACO Algorithm. In: Proc. IEEE Swarm Intelligence Symposium (SIS-06), pp. 77-84. IEEE, 2006.
4. [8] J. Kennedy and R. Mendes, Population structure and particle swarm performance. Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2002), Honolulu, Hawaii USA. 2002
5. J. Kennedy and R. C. Eberhart, with Y. Shi. Swarm Intelligence, San Francisco: Morgan Kaufman/ Academic Press, 2001
6. R.S. Parpinelli, H.S. Lopes and A.A. Freitas. Data Mining with an Ant Colony Optimization Algorithm, IEEE Trans. on Evolutionary Computation, special issue on Ant Colony Algorithms, 6(4), pp. 321-332, Aug 2002.
7. T. Sousa, A. Silva, A. Neves, Particle Swarm based Data Mining Algorithms for classification tasks, Parallel Computing 30, pp. 767-783, 2004.
8. Kennedy,J.; Eberhart, R.C(2001). *Swarm Intelligence*. Morgan Kaufmann.

9. Poli, R. (2007). "An analysis of publications on particle swarm optimisation applications". *Technical Report CSM-469* (Department of Computer Science, University of Essex, UK).
10. Poli, R. (2008). "Analysis of the publications on the applications of particle swarm optimisation". *Journal of Artificial Evolution and Applications* 2008: 1–10. doi:10.1155/2008/685175.
11. Eberhart, R.C.; Shi, Y. (2000). "Comparing inertia weights and constriction factors in particle swarm optimization". *Proceedings of the Congress on Evolutionary Computation* 1. pp. 84–88.
12. Carlisle, A.; Dozier, G. (2001). "An Off-The-Shelf PSO". *Proceedings of the Particle Swarm Optimization Workshop*. pp. 1–6.
13. van den Bergh, F. (2001). *An Analysis of Particle Swarm Optimizers* (PhD thesis). University of Pretoria, Faculty of Natural and Agricultural Science.
14. Clerc, M.; Kennedy, J. (2002). "The particle swarm - explosion, stability, and convergence in a multidimensional complex space". *IEEE Transactions on Evolutionary Computation* 6 (1): 58–73. doi:10.1109/4235.985692.
15. Trelea, I.C. (2003). "The Particle Swarm Optimization Algorithm: convergence analysis and parameter selection". *Information Processing Letters* 85 (6): 317–325.
16. Bratton, D.; Blackwell, T. (2008). "A Simplified Recombinant PSO". *Journal of Artificial Evolution and Applications*.
17. Evers, G. (2009). *An Automatic Regrouping Mechanism to Deal with Stagnation in Particle Swarm Optimization* (Master's thesis). The University of Texas - Pan American, Department of Electrical Engineering.
18. Meissner, M.; Schmuker, M.; Schneider, G. (2006). "Optimized Particle Swarm Optimization (OPSO) and its application to artificial neural network training".
19. Pedersen, M.E.H. (2010). *Tuning & Simplifying Heuristical Optimization* (PhD thesis). University of Southampton, School of Engineering Sciences, Computational Engineering and Design Group.
20. Pedersen, M.E.H.; Chipperfield, A.J. (2010). "Simplifying particle swarm optimization". *Applied Soft Computing* 10 (2): 618–628.

21. Pedersen, M.E.H. (2010). "Good parameters for particle swarm optimization". *Technical Report HL1001* (Hvass Laboratories).
22. Mendes, R. (2004). Population Topologies and Their Influence in Particle Swarm Performance (PhD thesis). Universidade do Minho.
23. Das and B. K. Chakrabarti (Eds.), *Quantum Annealing and Related Optimization Methods*, Lecture Note in Physics, Vol. 679, Springer, Heidelberg (2005)
24. Press, WH; Teukolsky, SA; Vetterling, WT; Flannery, BP (2007). "Section 10.12. Simulated Annealing Methods". *Numerical Recipes: The Art of Scientific Computing* (3rd ed.). New York: Cambridge University Press.