Distributed Biobjective Ant Colony Algorithm for Low Cost Overlay Network Routing

Benjamin McBride
Kansas State University
2061 Rathbone Hall
Manhattan, KS 66506
Email: bdm7935@ksu.edu

Caterina Scoglio
Kansas State University
2061 Rathbone Hall
Manhattan, KS 66506
Email: caterina@ksu.edu

Sanjoy Das
Kansas State University
2061 Rathbone Hall
Manhattan, KS 66506
Email: sdas@ksu.edu

Abstract— **In this paper we introduce a biobjective ant colony algorithm for constructing low cost overlay routing networks. The ant colony algorithm is distributed and adaptive in finding shortest paths from source to destination nodes while also constructing a low cost overlay routing network. Additionally, we define a cost model for overlay network construction that includes network traffic demands. The proposed ant colony algorithm was applied to a randomly generated 100-node network with an average node degree of 10.2. The results show that the algorithm quickly converges to the shortest path between nodes while converging on a low cost overlay routing network topology, despite changing traffic demands.**

I. INTRODUCTION

Ant colony algorithms get there inspiration from the behavior of real ants foraging for food. Individual ants act as simple agents that forage for food by following pheromone trails and depositing pheromone along the path taken. An ant is more likely to follow a trail with a high concentration of pheromone. Pheromone on shorter paths gets increasingly reinforced as more ants follow the higher concentration of pheromones. This is due to more ants being able to travel a shorter path than a longer path over a given period of time. Eventually, the pheromone concentration is greatest along the shortest path to the food source. This results in the ants converging on the shortest path to the food source. While each ant acts independently, the collective behavior of the ant colony is cooperative. This emergent behavior has inspired a host of stochastic optimization algorithms called ant colony algorithms.

Ant colony algorithms, as described in [1], [2], and [3], have been quite popular for a wide variety of discrete optimization problems such as the traveling salesman problem, quadratic assignment problem, job-shop scheduling, vehicle routing, and graph coloring. This is widely due to the ease of implementation and the inherent balance between exploration and exploitation found in the ant colony algorithm. The adaptive multi-agent characteristics of ant colony algorithms are also attractive for distributed and dynamic network routing problems. Ant colony algorithms have previously been used in connection-oriented network routing, connectionless network routing, and wireless sensor network routing [4] [5] [6] [7] [8].

The growth of peer-to-peer and other multimedia applications such as video conferencing and internet telephony has created the need for increased quality of service over the physical network. Providing the required quality of service and performance requirements of these applications over a packet switching network has been elusive.

In recent years, overlay routing networks have been seen as a possible solution [9] [10] [11] to the quality of service problem. An overlay routing network is an application-layer, logical network created on top of the physical, or underlay network. The overlay routing network can be used to improve performance and provide quality of service by routing data through intermediate nodes in the logical network, rather than directly through the underlay network. In this manner, an overlay routing network can respond to changing traffic congestion and performance degradation in the underlay network.

In this work, overlay routing networks are created using an ant colony algorithm inspired by the Ant Colony System algorithm and the AntNet algorithm described in [2] and [5]. As data is routed through the network the algorithm converges on the shortest path between source and destination nodes, while also constructing a low cost overlay routing network. This biobjective ant colony algorithm is both distributed and adaptive. While each individual node is selfish in the construction of a low cost overlay, the emergent behavior of the ant colony system results in a cooperative system. This differs from other selfishly constructed networks [12].

The proposed algorithm was tested on a randomly generated network with changing traffic demand patterns. It was found that the routing quickly converges to the shortest path between source and destination nodes while also constructing a low cost overlay routing network. As the traffic demands reach equilibrium the cost of the network converges on a single overlay topology.

The paper is organized as follows. In Section II we discuss background information related to the ant colony algorithm and the specific overlay cost model used. Section III gives the detailed algorithms implemented. Simulation results are presented in Section IV. Concluding remarks and future research directions are outlined in Section V.

II. BACKGROUND

A. Ant Colony Optimization

The ant colony algorithm described in section III is based on the Ant System and AntNet presented in [1] and [5]. In Ant System, individual agents, or ants, transition from state i to a neighbor state j, according to the pheromone table entry $\tau_{i,j}$ and the desirability $\eta_{i,j}$ of transitioning from i to j. The pheromone table can be seen as a way of ants communicating, while the desirability can be viewed as a heuristic. Pheromone trails are updated based upon the quality of the solution found. Pheromone values along good paths are reinforced greater than values along poor paths. While each ant acts autonomously and with limited information, the collective behavior that emerges causes ants to follow good paths with a higher probability than poor paths.

Formally, the probability of an ant moving from state i to state j is defined as:

$$P_{i,j} = \frac{\tau_{i,j}^{\alpha}\eta_{i,j}^{\beta}}{\sum_{k\in(allowed)}\tau_{i,k}^{\alpha}\eta_{i,k}^{\beta}} \tag{1}$$

where α and β are parameters controlling the relative importance of pheromone trails and desirability, $allowed$ is the set of states in the neighborhood of i less the states that the ant has already visited. Once a complete solution is found the pheromone table is updated according to:

$$\tau_{i,j}(t+1) = (1-\rho)\tau_{i,j}(t) + \Delta\tau_{i,j} \tag{2}$$

where ρ represents the evaporation of the trail since time t and $\Delta\tau_{i,j}$ is directly proportional to the relative goodness of the solution found if the solution contains state transition (i, j), otherwise $\Delta\tau_{i,j} = 0$.

The ant agent model presented in Ant System tends to fall apart in a distributed environment. AntNet solves these problems by distinguishing between forward and backward traveling ants. Each state contains a separate pheromone table, or routing table, and desirability information. A forward ant at a given state determines the next state by equation (1) and the local routing table. Upon arrival at the destination state, the quality of the path is known and the ant traverses its path in the opposite direction, updating the pheromone tables at each state [5].

B. Overlay Network Creation

We assume that each node needs to select its neighbors in a distributed fashion, without knowing the other node neighbors selection. Let $G = (N, L)$ be the graph representing the overlay network and $G_u = (N, E)$ the graph representing the underlay, or physical, network. N is the set of nodes that are in both the overlay and physical network, while the set of logical links L can be different from the set of physical links E. A logical link $l \in L$ is setup on a path composed by physical links $e \in E$. Each node $i \in N$ has a traffic demand toward a node subset $S_i \subset N$. Let $d_{i,j}$ be the traffic demand between node i and node j in the subset S_i. The objective for the node is to create logical links to be connected to all nodes in S_i such that the total cost is minimized. The cost model defined is an extension of the cost models given in [12] and [13] to include traffic demand between nodes. We define cost using two cost components:

1) Cost to create a logical link $l_{i,j}$ from node i to node j proportional to the length of $l_{i,j}$ in number of hops on the physical network.
2) Cost to transport the traffic demands proportional to the distance $u_{i,j}$ between node i and node j, and the amount of traffic demand.

The cost for node i to connect to each node $j \in S_i$ and carry traffic demand $d_{i,j}$ is defined:

$$C_i = \sum_{j\in B_i} c_h h_{i,j} + \sum_{j\in S_i} c_t t_{i,j} d_{i,j} \tag{3}$$

where B_i is the set of neighbors torward which node i has a logical link $l_{i,j}$, c_h is the overlay cost coefficient, $h_{i,j}$ is the number of hops on the physical network in $l_{i,j}$, c_t is the underlay transit coefficient, and $t_{i,j}$ is the number of transit virtual links in the path to node j. The total cost of the overlay network is consequently defined as:

$$C(G) = \sum_{i\in N} C_i \tag{4}$$

It is important to note that the C_i is a function of both the location of the logical link $l_{i,j}$ and the demand $d_{i,j}$. Changing traffic demands in the network will cause $C(G)$ to change over time.

III. APPROACH

Like the ant colony approach taken in [5] and [7], forward ants are used to traverse the network, constructing a path from the source to the destination, and backward ants retrace the path, updating pheromone tables and routing information. However, in addition to routing through the network, a low cost overlay network topology is also constructed. Forward ants use the pheromone table and an estimate of the distance to the destination in determining the next node to visit. Additionally, if there exists an overlay at the current node for the destination, then the ant takes the overlay link to the next node. The complete algorithm is given in Algorithm 1.

Backward ants follow the path taken by a forward ant, updating the pheromone entries at each node along the path as given in Algorithm 2. At each node the backward ant calculates the cost of establishing an overlay to this node, storing the minimum cost along the path. Additionally, the backward ant updates the estimate of the number of hops remaining to the destination. In this manner, the heuristic function for each node is improved iteratively. Once the backward ant arrives at the source node, the source node establishes a temporary logical link to the minimum cost node along the ant's path if the node does not already have a logical link established towards the destination node. Nodes also perform periodic daemon activities such as deleting expired logical links and applying a global evaporation rule to the pheromone table:

$$\tau_{i,j}(dst) = (1-\rho_g)\tau_{i,j}(dst)$$

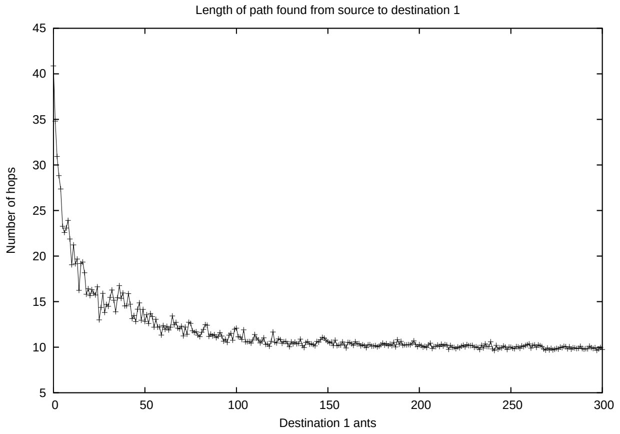

Fig. 1. Path length found from source to destination averaged over 100 runs.

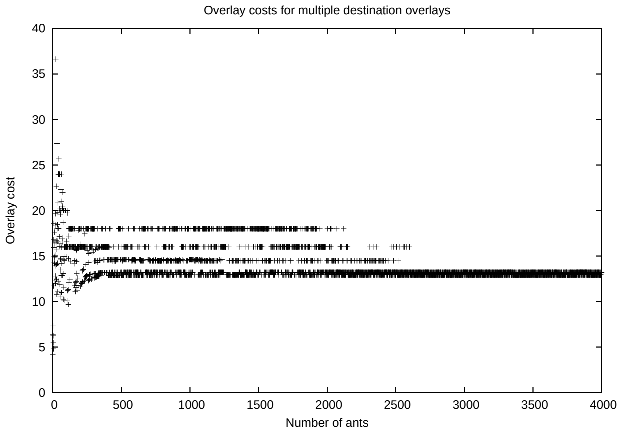

Fig. 2. Overlay cost for a typical simulation

where ρ_g is a global evaporation constant.

IV. RESULTS

The system was implemented as specified in Section III and tested with a randomly generated 100-node network having average node degree of 10.2. The following parameter values were used: $\alpha = 1$, $\beta = 3$, $\rho = 0.05$, $\rho_g = 0.02$, $Q = 100$, $Q' = 100$, $\sigma_0 = 0.1$, $c_h = 2$, and $c_t = 1$. The case with a single source node and multiple destination nodes was considered. As can be seen in Figure 1 the path length of paths found from the source to a destination node quickly converges to a minimal path. The length of the paths found to other destination nodes behaves similarly.

The overlay cost of a typical simulation, using a single source and two destinations, is shown in Figure 2. The cost oscillates initially while the traffic demand to both destinations grows asymptotically. During this transient phase, several low cost overlay network topologies are found as the cost is a function of the traffic demand. Once the traffic demand reaches equilibrium, several candidate overlay networks emerge before the algorithm converges on the final, low cost topology. The resulting overlay network has two cost values, one for each destination, with the total overlay cost being the sum of the two cost values as given in Equation 3.

V. CONCLUSION

In this paper we introduced a biobjective ant colony algorithm for constructing low cost overlay routing networks. The results show the distributed nature of ant colony algorithms is ideal for quick construction of low cost overlays. Additionally, we show the adaptive behavior of the ant colony algorithm makes it well suited for the dynamic behavior of traffic demands in the physical network.

This preliminary investigation is promising. Future work will model overlay cost in the entire network and will include additional resource constraints, such as non-uniform edge cost and bandwidth. As noted in [12], the distance in the cost function (equation (3)) can represent any metric such as latency or bandwidth. However, in a multiobjective case multiple metrics should be considered congruently. Additional investigations will extend the algorithm to a multiobjective case considering latency, bandwidth, and quality of service metrics in overlay routing network construction.

ACKNOWLEDGMENT

The authors would like to thank Kamran Qadir for his help in analyzing optimal cost overlay construction in an offline environment.

REFERENCES

[1] M. Dorigo, V. Maniezzo, and A. Colorni, “The ant system: Optimization by a colony of cooperating agents,” *IEEE Transactions on Systems, Man, and Cypernetics*, vol. 26, no. 1, 1996.

[2] M. Dorigo and L. M. Gambardella, “Ant colony system: A cooperative learning approach to the traveling salesman problem,” *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, April 1997.

[3] M. Dorigo, G. D. Caro, and L. M. Gambardella, “Ant algorithms for discrete optimization,” *Artificial Life*, vol. 5, no. 2, 1999.

[4] R. Schoonderwoerd, O. Holland, and J. Bruten, “Ant-like agents for load balancing in telecommunications networks,” in *Proceedings of the First International Conference on Autonomous Agents*. ACM Press, 1997, pp. 209–216.

[5] G. D. Caro and M. Dorigo, “Antnet: Distributed stigmergetic control for communications networks,” *Journal of Artificial Intelligence Research*, vol. 9, pp. 317–365, 1998.

[6] D. Subramanian, P. Druschel, and J. Chen, “Ants and reinforcement learning: A case study in routing in dynamic networks,” in *IJCAI (2)*, 1997, pp. 832–839.

[7] G. Singh, S. Das, S. Gosavi, and S. Pujar, *Recent Developments in Biologically Inspired Computing*. Idea Group Publishing, 2005, ch. Ant Colony Algorithms for Steiner Trees: An Application to Routing in Sensor Networks, pp. 181–206.

[8] G. D. Caro and M. Dorigo, “Extending antnet for best-effort quality-of-service,” 1998, unpublished presentation at ANTS’98-From Ant Colonies to Artificial Ants: First International Workshop on Ant Colony Optimization. http://iridia.ulb.ac.be/ants98/ants98.html.

[9] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, “Resilient overlay networks,” in *SOSP ’01: Proceedings of the eighteenth ACM symposium on Operating systems principles*. New York, NY, USA: ACM Press, 2001, pp. 131–145.

[10] A. Rowstron and P. Druschel, *Lecture Notes in Computer Science*. Springer-Verlang, 2001, vol. 2218, ch. Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems.

[11] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz, “Tapestry: A resilient global-scale overlay for service deployment,” *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, January 2004.

[12] B.-G. Chun, R. Fonseca, I. Stoica, and J. Kubiatowicz, “Characterizing selfishly constructed overlay routing networks,” in *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, 2004.

[13] A. Fabrikant, A. Luthra, E. Maneva, C. H. Papadimitriou, and S. Shenker, “On a network creation game,” in *PODC ’03: Proceedings of the twenty-second annual symposium on Principles of distributed computing*. New York, NY, USA: ACM Press, 2003, pp. 347–351.

APPENDIX

Algorithm 1 $ForwardAnt(src, dst, volume)$

```
{Initialize data structures}
$path \leftarrow []$ {Set the path to an empty list}
$hopCount \leftarrow 0$
$current \leftarrow src$
while $current \neq dst$ do
    {Pick next node}
    if there exists an overlay $j$ for $current$ to $dst$ then
        $next \leftarrow j$
    else
        Pick a node $j$ that has not been visited, from the neighbors of the current node $i$ with probability given by:
            $P_{i,j} = \frac{\tau_{i,j}(dst)^{\alpha}\eta_{i,j}^{\beta}}{\sum_{k \in (neighbors-path)} \tau_{i,k}(dst)^{\alpha}\eta_{i,j}^{\beta}}$
        where $\tau$ is a pheromone table recording the cost of hopping from $i$ to $j$ when going to $dst$, $\eta$ is inversely proportional to the estimate of the distance from the neighbor node to the destination node and where $\alpha$ and $\beta$ are parameters that control the relative importance of the pheromone trail to the heuristic.
    end if
    {Move to the next node}
    Append $next$ to $path$
    $hopCount \leftarrow hopCount + hops_{current,next}$
    $current \leftarrow next$
    {Update the node demand and hops from $src$}
    $d_{current,dst} \leftarrow (1-\rho)d_{current,dst} + (\rho)volume$
    $hops_{current,src} \leftarrow min(hopCount, hops_{current,src})$
end while
```

Algorithm 2 $BackwardAnt(forwardAnt)$

```
{Initialize data structures}
$src \leftarrow forwardAnt.src$
$dst \leftarrow forwardAnt.dst$
$path \leftarrow forwardAnt.path$
$hopCount \leftarrow forwardAnt.hopCount$
$previous \leftarrow current \leftarrow dst$
$minOverlayCost \leftarrow \infty$
while $current \neq src$ do
    {Update node routing parameters}
    Set the number of hops from the current node to the destination node as:
        $hops_{current,dst} \leftarrow hopCount - hops_{current,src}$
    and set the desirability $\eta$ of the current node with regards to the destination node:
        $\eta_{current,dst} \leftarrow max(\eta_{current,dst}, \frac{1}{hops_{current,dst}})$
    Update the node pheromone entry from the current node $i$ to the previous node $j$ for ants traveling to $dst$ according to:
        $\tau_{i,j}(dst) \leftarrow (1-\rho)\tau_{i,j}(dst) + \rho\frac{Q}{hopCount}$
    where $\rho$ is the local evaporation rate and $Q$ is a scaling parameter. The cost of establishing an overlay to the current node is defined as:
        $cost \leftarrow c_h hops_{current,src} + c_t d_{current} hops_{current,dst}$
    and is used in setting the value of establishing an overlay to the current node as follows:
        $\sigma_{current} \leftarrow max(\sigma_0, (1-\rho)\sigma_{current} + \rho\frac{Q'}{cost})$
    where $\rho$ is the local evaporation rate and $Q'$ is a scaling parameter.
    {Check if current node is best overlay node}
    if $cost \leq minOverlayCost$ then
        $minOverlayCost \leftarrow cost$
        $overlay \leftarrow current$
        $\sigma_{overlay} \leftarrow \sigma_{current}$
    end if
    {Move to the next node on path}
    $previous \leftarrow current$
    $current \leftarrow$ next node from $path$
end while
```