

COMP 4520: Ant Colony Optimization for Congestion-Sensitive Traffic Routing

Project Report

Taylor Cox

Supervisor: Dr. Parimala Thulasiraman

2017-07-01

Introduction

Rapid population growth in recent decades has led to unprecedented density in urban areas. The scale and complexity of modern urban centers has introduced new challenges in a wide variety of research areas. These challenges include problems in city planning, government services and transportation. The latter of these areas is of primary interest to this work. This area is formally described as the field of *Intelligent Transportation Systems* (ITS).

ITS covers a wide range of topics, all of which aim to improve the efficiency of urban transportation. ITS concerns systems in which technologies are applied to road transportation. These applications include infrastructure, vehicles and users, and traffic management or mobility management [1]. Traffic management is the aspect of ITS which this work concerns. Traffic management problems (and their approaches) sit at the intersection of computer science, mathematics and logistics.

This project concerns urban traffic congestion, a specific problem in traffic management. In this project, the feasibility of improving traffic flow in road networks will be investigated. Primarily, Ant Colony Optimization will be demonstrated as a strategy for enabling traffic-sensitive routing. Existing literature that uses ACO for traffic congestion reduction will be evaluated. Improvements upon the literature will then be identified and tested.

Motivation for this project

This project specifically address the problem of traffic congestion in urban centers. This is a highly impactful problem in socioeconomics and environmental policy which affects billions of individuals every day [2]. Improving the flow of vehicles in urban networks will reduce the economic and environmental costs of traffic congestion, while avoiding the infrastructural costs of constructing additional roads.

This project aims to evaluate the performance of existing Ant Colony Optimization strategies for traffic congestion control. This project was also undertaken to identify venues for improving upon the existing literature in this area. These venues include improvements to the accuracy of a pheromone deposit/evaporation model, and computational performance improvements.

This project begins with a literature review in order to evaluate existing ACO mechanisms for reducing traffic congestion. This portion of the project enabled further investigation as the literature survey shows that i) traffic-sensitive routing leads to lower travel-times over traffic-agnostic routing and ii) there are aspects of the current literature which may be improved upon.

This project is focused on a specific problem within the area of traffic management. This is the problem of urban traffic congestion and its reduction. Traffic congestion poses significant

economic and ecological impact when scaled to large urban centers. The problem of urban traffic congestion (or simply “traffic congestion”) is as follows:

Problem Statement

In urban traffic networks, drivers are assumed to use the fastest path to their destination. Without prior knowledge, this is the shortest path corresponding to the minimum travel time. The minimum travel time of a specific road corresponds to the road’s distance d and its maximum speed v . Using the classical formula for speed ($v = d/t$), the minimum travel time of a road is d/v . Therefore, vehicles embarking on a route to their destination attempt to optimize the following equation:

$$path = \min(\sum_{i,j \in P} \frac{d_{ij}}{v_{ij}}), P \in E$$

Where P is the subset of edges E on a graph G which forms a path to the vehicle’s destination. V and d denote the distance and velocity (speed limit) of the ij th edge. This optimization is determined via the execution of a shortest-path algorithm against G .

If all drivers use their shortest path, critical arteries will inevitably congest. To avoid excessive congestion, drivers must also account for traffic flow when embarking on a route. However, hypersensitivity to traffic congestion may result in inappropriately long detours. An approach must be established for optimal path discovery which considers both route distance and traffic flow.

This project examines the usefulness of ant colony optimization for this optimal path discovery problem. ACO is used for modelling traffic on specific roads of a network. This is accomplished via the deposit of *pheromone* onto each road by vehicles which inhabit that road at a given moment in time. Pheromone is then evaporated at a rate proportional to the maximum possible speed of that road.

It is assumed that the static attributes of the road network (such as speed limits and physical distances) are known. Without a traffic-pheromone system, drivers embark on the shortest path considering only the static travel time. Such pathfinding is accomplished with standard algorithms such as Dijkstra’s Algorithm and Bellman-Ford-Moore. When the aspect of traffic-pheromone is introduced, graph theory algorithms may still be applied. However, these algorithms must consider the *entire* cost of a road/edge (the physical length *and* the current traffic density) when establishing an optimal path.

In the traditional Ant Colony Optimization system, pheromone is introduced onto a graph in order to attract incoming ants. In the traffic congestion problem, a higher number of vehicles

on a road corresponds to a higher pheromone deposit rate. Therefore, in order to avoid congested routes, pheromones serve as a *repellent* to incoming ants (drivers) in this scenario. The reverse-pheromone concept has proven to be fruitful for optimal vehicle path discovery both in the current literature and in this project.

Current Literature

In recent years, there have been two key works which explore the use of ACO as a traffic modelling tool. Both of these works also use pheromones as a means to repel incoming vehicles from highly populated arteries. These works are *An Inverted Ant Colony Optimization Approach to traffic* by Capela, Machado, Silva and Abreu (IACO by Capela et al.) [3], and *Dynamic Travel Path Optimization System Using ACO* by Kponyo, Kuang and Zhang (DTPOS by Kponyo et al.) [4].

The first of the two relevant works is IACO by Capela et al. IACO proposes a two-phase approach to modelling urban traffic. The first phase (the *initial* phase) involves vehicles executing Dijkstra's algorithm on the road network without any traffic considerations. These first agents deposit pheromone on their routes, initializing the traffic-density model.

Sometime after the first phase, IACO initiates its second phase (the *pheromone* phase). In this phase, agents both read from and write to the pheromone model. When embarking on a route, vehicles select the path which corresponds to the minimum cost, where the cost of a road (edge) is its distance added to its current pheromone level.

In the pheromone phase of IACO, agents determine the optimal path to their destination every time they reach an intersection on their current path. Agents in the pheromone phase execute the modified Dijkstra's algorithm at each intersection, while agents in the initial phase execute Dijkstra's algorithm from their source exactly once.

As shown in the paper itself, the IACO algorithm reduces travel time and wait-time compared to a naïve shortest-path algorithm (referred to as *shortest-time* by the authors). IACO also introduces the novel contribution of varying the amount of agents which comply with the system. The authors evaluate the performance of their proposed algorithm when applied to various portions of the total population.

The first disadvantage of IACO is its reliance on constants in its pheromone deposit model. All vehicles in IACO deposit pheromone onto their corresponding roads at a constant rate. In this system, a large number of vehicles moving quickly through an edge (such as a freeway) will deposit the same amount of pheromone as a large number of slowed or stopped vehicles.

Another setback of IACO is the unnecessary complexity introduced by its two-phase approach. In IACO, agents in the pheromone phase determine the cost of an edge by adding its pheromone to its distance. If all edge pheromones initialize at zero, there is no need to

populate pheromone values in an initial phase. The initial phase of IACO is an avoidable simplification of the pheromone phase.

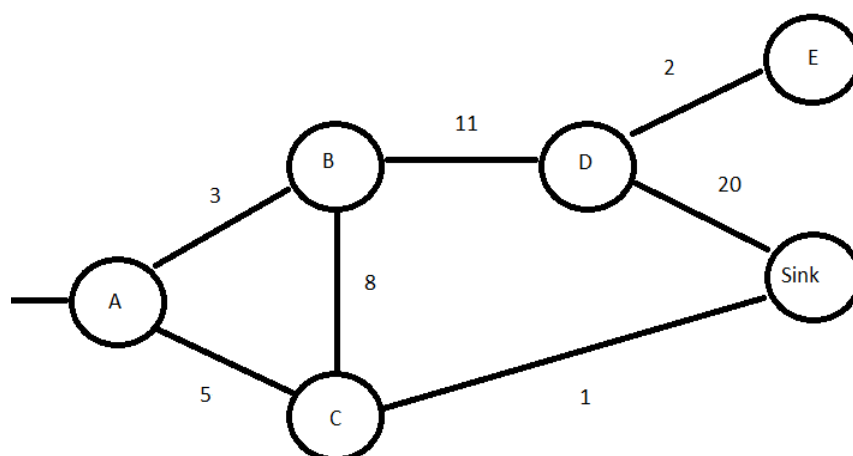
IACO is the first of two works which use reverse-pheromones for modelling traffic in urban road networks. IACO proposes a modified implementation of Dijkstra's algorithm for capturing dynamic traffic density. It also introduces the concept of variability between the total and system-compliant population. IACO's shortcomings are its reliance on constants in its pheromone update strategy, and its separation of its algorithm into two phases.

The second key citation of this project is DTPOS by Kponyo et al. DTPOS aims to improve upon existing navigation systems (such as Google Maps) by proposing that vehicles interact in a Vehicular Ad-hoc Network (VANET) as opposed to via satellite or GPS. DTPOS also gives a detailed mathematical model to describe vehicle behavior and network attributes.

The first contribution of DTPOS is its proposal of using a Vehicular ad-hoc Network (VANET) system as opposed to centralized satellites/GPS. DTPOS proposes that vehicles form clusters, where cluster heads are responsible for inter-cluster communication. The paper also proposes the installation of communication devices at road intersections to ensure the rapid delivery of real-time information. While VANET architecture is outside the scope of this project, DTPOS does show that traffic congestion reduction does not depend on the type of inter-vehicle communication system used.

The second contribution of DTPOS is its mathematical model of an ACO system for the "optimal path determination problem". The model in DTPOS also uses the classical definition of minimum travel time (d/v). The paper provides a pheromone deposit formula where the inverse of each vehicle's expected travel time is added to the edge. In contrast to the constant-deposit rate proposed by IACO, pheromone deposit in DTPOS is dependent on vehicle speed. This enables a more accurate depiction of real-time traffic flow.

While the networking and modelling insights from DTPOS are valuable, the paper itself also has a key setback. This setback is the risk of DTPOS' convergence to local optima. Upon reaching an intersection, vehicles select the edge which is i) connected to the destination and ii) has the lowest cost. Consider the following scenario, where a vehicle v must move from its source A to its destination labelled *Sink*.



In the strategy proposed by DTPOS, the vehicle will take the path of cost 3 to B. Then, the vehicle will move from B to C (8) and finally arrive at Sink from C (1). This leads to a total cost of 12. Had the vehicle considered the entire graph, it would have taken the path to C (5) and then to Sink (1) for a total of 6.

Vehicles in DTPOS evaluate the cost of edges as they are reached – vehicles embark on the lowest-cost road from their current intersection/junction. As seen in figure 1, this may result vehicles embarking on a suboptimal path.

DTPOS by Kponyo et al. and IACO by Capela et al. are the two key citations for the background of this project. Both works propose an ACO system in which vehicles (ants) are *repelled* by roads (edges) with high values of pheromone deposit. While both works deliver improvements over traffic-agnostic vehicle routing, they both suffer from certain setbacks.

Existing Approaches

As shown in the relevant literature, the current approaches for urban traffic congestion reduction revolve around graph theory algorithms. The existing literature uses the traditional terminology of vertices, edges and connectivity to define the static nature of road networks. The dynamic nature of the road network (vehicles and congestion) is described by the behavior of ants, or *agents*.

A key concept in urban traffic management is the balancing act engaged between static and dynamic attributes. Whereas graphs are known to have a static nature and networks (such as VANETs) have a dynamic nature, traffic networks have both static and dynamic components. This means that traffic networks are *pseudostatic* graphs.

The graph-theoretical nature of traffic networks permits the application of traditional graph traversal and shortest-path algorithms including Dijkstra's algorithm and Bellman-Ford-Moore. However, the analogy between graphs and road networks must be modified. The dynamic nature of urban traffic means that graphs which represent traffic networks are ones with *changing* weights.

Changes in edge-weights correspond to changes in traffic conditions. Edges also have static attributes such as their physical distance and speed limit (or their minimum travel time). When simultaneously considering the static and dynamic aspects of a road network, algorithms such as Dijkstra's search for the optimal path as opposed to the shortest path. This leads to the treatment of urban traffic congestion reduction as a multi-criteria optimization problem.

Shortest-path algorithms must take account for the fact that weights of edges are not set in stone. The weights of edges on a road network change over time. This change is modelled by the progression of a synchronized clock. Agents which embark on a route cannot evaluate the optimal path once – they must do so regularly to account for changes in traffic conditions.

Contribution

The research contribution of this project is its improvement upon the existing literature. The setbacks identified in the literature survey are to be reconciled. First, the model from IACO is modified to reflect vehicle speed. This is done by drawing inspiration from the pheromone deposit equation proposed in DTPOS. Second, Dijkstra's algorithm from IACO is selected as a pathfinding mechanism over the greedy strategy outlined in DTPOS. The pathfinding approach from IACO is then improved upon. Finally, the two-phase strategy from IACO is discarded in favor of solely using a pheromone-based strategy.

This project uses the model from IACO as a start-point, with concepts from DTPOS incorporated. The pheromone deposit equation in DTPOS relies on the inverse of the vehicle's travel-time. In this project, the pheromone deposit model is defined as the inverse of the vehicle's speed. This creates a direct relationship between vehicle speed and traffic density. Vehicles moving slowly on a road will deposit higher pheromone, and quickly moving vehicles will deposit much less. The pheromone deposit equation is therefore the following for every ij th edge:

$$Ph_{ij_t} = Ph_{ij_{t-1}} + (r) \left(\sum_{a \in A_{ij}} \frac{1}{V(a) + c} \right)$$

Where $V(a)$ is the velocity of an agent and A is the set of agents residing on an edge ij at time t . A very small constant ($c < 0.05$) is added to $V(a)$ in order to avoid a divide by zero error. Only a portion ($r < 1$) of the total sum is added to the existing pheromone of the edge to avoid a rapid increase in pheromone due to any incidental congestion.

Following the rules of ACO, the pheromone deposit equation is accompanied by a pheromone evaporation equation. This equation depends on the edge's minimum travel time under ideal conditions. To prevent rapid pheromone reduction, the edge travel time is multiplied by r . This is the same value used in the pheromone deposit equation. The pheromone of an edge may not evaporate below zero, as the minimum cost of each edge must be the cost to traverse the edge's physical distance. The evaporation function of each edge ij is defined as follows:

$$Ph_{ij_t} = \max(0, \frac{Ph_{ij_{t-1}}}{(r)D_{ij}/V_{ij}})$$

To capture both the static and dynamic nature of an edge, the weight of each edge must be described in terms of its distance and pheromone. The weight W of each edge ij at time t is defined as the following:

$$W_{ij_t} = D_{ij} + Ph_{ij_t}$$

At each discrete time-step t , the pheromone value of each edge ij is updated to i) increase as a result of its deposit function and ii) decrease as a result of its evaporation function. The weight of each edge is then assigned as the addition of the distance and current pheromone.

In addition to the new pheromone model, the pathfinding strategy for vehicles in the network is also iterated upon. As a starting point, the modified-cost implementation of Dijkstra's algorithm from IACO is used over the greedy strategy proposed in DTPOS.

In IACO, vehicles travelling to their destination will re-evaluate the cost of the network every time an intersection is reached. As a result of this re-evaluation, vehicles will change course to the optimal path with respect to the latest information regarding network traffic.

This project adjusts IACO's pathfinding system to improve the realism and performance of the vehicle routing process. This is done by establishing a constant traffic threshold T and introducing a proactive routing strategy. Instead of evaluating their shortest path at each intersection, vehicles will evaluate their shortest path at the precise moment when their current path exceeds a certain pheromone limit.

Let P denote the set of vertices which compose the path from an agent's source to destination. A threshold-based rerouting system will check the pheromone state of each edge in P against the threshold T in the following manner:

Threshold Routing

```

1: procedure CHECKROUTE(agent)
2:   if  $\exists p_1, p_2 \in P(\text{agent}), Ph_{p_1 p_2 t} \geq T$  then
3:     Reroute(agent)
4:   end if
5: end procedure

```

As a result of this adjustment, a vehicle V will reroute if and only if an edge e on the vehicle's path P is congested beyond T at the time t . An edge e is considered congested enough to trigger a reroute if its pheromone level (excluding its static cost) exceeds T . Pheromone level is exclusively considered when evaluating the congestion of a route because the static cost of an edge may exceed the threshold value.

Adjusting IACO's pathfinding system from one which always reroutes to one which only reroutes when necessary serves two purposes. These purposes are computational cost reduction and user experience improvement.

First, limiting the total number of reroutes in a path execution reduces the average case computational complexity of route discovery and route maintenance. The complexity of Dijkstra's algorithm is $O(|E| + |V| \log |V|)$, which becomes an expensive algorithm when executed each time a vehicle reaches an intersection. The cost of evaluating each edge of a route R against a threshold T is $O(|R|)$, a more scalable operation.

Introducing threshold-based route evaluation is a cost improvement since it likely leads to fewer executions of Dijkstra's algorithm. In the worst case, Dijkstra's algorithm will still be executed. However, it is expected that a tolerance for mild congestion will reduce the number of times a vehicle is rerouted.

The second benefit of reducing the number of reroutes is the improvement of user experience. In IACO, vehicles may reroute every time they reach an intersection. From the user's perspective, this may be an overwhelming and frustrating experience. In addition, waiting until an intersection is reached may leave drivers unable to reroute. If a driver is at a stoplight and is directed by an on-board system to change course, they may be powerless to do so.

The pathfinding aspect of IACO is adjusted in this project via the introduction of a proactive route evaluation procedure. This procedure is designed such that vehicles will only reroute if their existing route has reached a critical congestion level, and will do so as soon as the critical level is reached. This improves user experience and reduces overall computational cost. Adjusting the shortest-path algorithm is one of two contributions presented in this work.

The contributions of this work are an improved pheromone deposit/evaporation model, and a reduction in the number of times vehicles in the system must re-evaluate their shortest path. Both of the contributions presented in this work build upon ideas presented in the current literature. The improvements to the pheromone model and the new pathfinding mechanism deliver improvements in terms of mathematical accuracy and computational complexity. These improvements resolve the setbacks identified in both of the relevant works.

Implementation

In order to deliver upon the contributions described, a traffic-sensitive vehicle routing system was implemented. This section reviews the tools and technologies used to execute the project, and examines certain implementation strategies. These strategies include the use of multithreading in vehicle routing and the selection of vehicle routing algorithms.

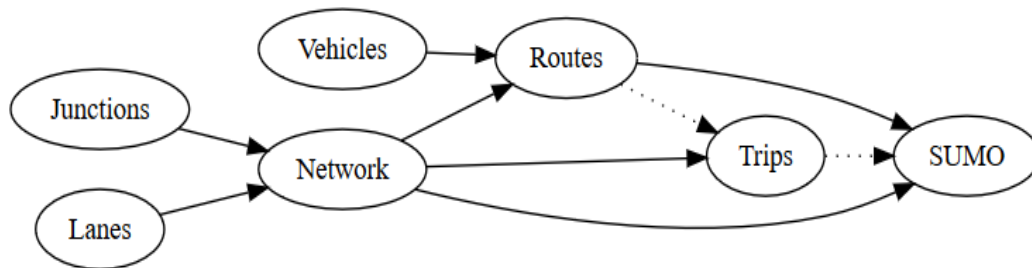
Without considering specific technology choices, the implementation of this project's simulation works as follows. The implementation uses a discrete clock to model changes in network state. Note that the pheromone level of each edge is initialized at 0. This corresponds to the fact that the network is empty when the simulation begins.

In the simulation procedure, the functions *Deposit* and *Evaporate* correspond to the deposit and evaporation equations presented earlier. *CheckRoute* is defined in the previous procedure, and *MoveForward* represents the natural progression of the vehicle *V* along its route.

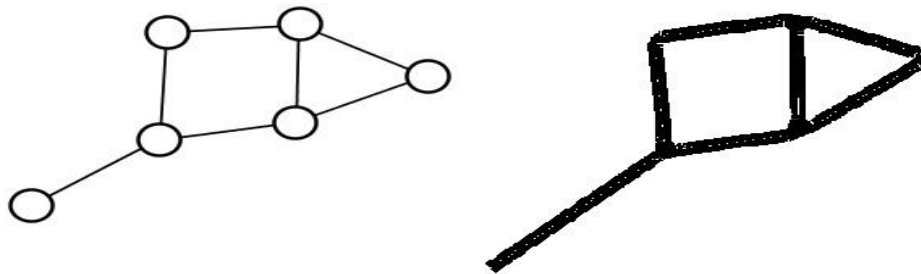
Traffic Simulation

```
1: procedure SIMULATION
2:    $t = 1$ 
3:   while Simulation not complete do
4:     for edge  $ij \in N$  do
5:        $W_{ij} \leftarrow D_{ij} + Ph_{ij}$ 
6:        $Ph_{ij} \leftarrow Deposit(Ph_{ij})$ 
7:        $Ph_{ij} \leftarrow Evaporate(Ph_{ij})$ 
8:     end for
9:     for vehicle  $V$  in simulation do
10:       $CheckRoute(V) /* Reroute if necessary */$ 
11:       $MoveForward(V)$ 
12:    end for
13:     $t \leftarrow t + 1$ 
14:  end while
15: end procedure
```

With the simulation procedure outlined, specific technologies must be selected. This project uses the Simulator of Urban Mobility (SUMO) [5] to model traffic network behavior. SUMO is a time-discrete, space-continuous microscopic simulation implemented in C++ with XML files used as assets. Assets include vehicle routes and road networks. SUMO also includes a GUI feature.



SUMO entities and their interactions.



A simple graph translated to a SUMO network.

The SUMO simulator was used to model urban traffic behavior. SUMO's python APIs were used to capture the static and dynamic nature of traffic networks. The native SUMO API (Sumolib) is

used to describe static network attributes such as edge distance and vertex degree. In contrast, SUMO ships with a tool titled Traffic Control Interface (TraCI) [6], which includes an API for dynamic network attributes. Dynamic network attributes include vehicles, their speed, and the number of vehicles on a certain edge.

The simulation algorithm was also optimized at the vehicle routing stage. Consider step 9 of the previously given simulation procedure. In this step, each agent evaluates the cost of their current path, and reroutes if necessary. At no point do vehicles *write* to the state of the road network. Therefore, this step may be parallelized without any concurrency overhead. To improve simulation performance, the routing/rerouting step of the simulation is executed in parallel for each participating vehicle.

Additionally, alternative routing algorithms were examined during the project's implementation stage. In particular, Bellman-Ford-Moore (BFM) was considered as an alternative routing algorithm over Dijkstra's. The advantage of BFM for this project is its ability to evaluate multiple paths to a destination. This would improve driver experience by offering a selection of paths which could be embarked on.

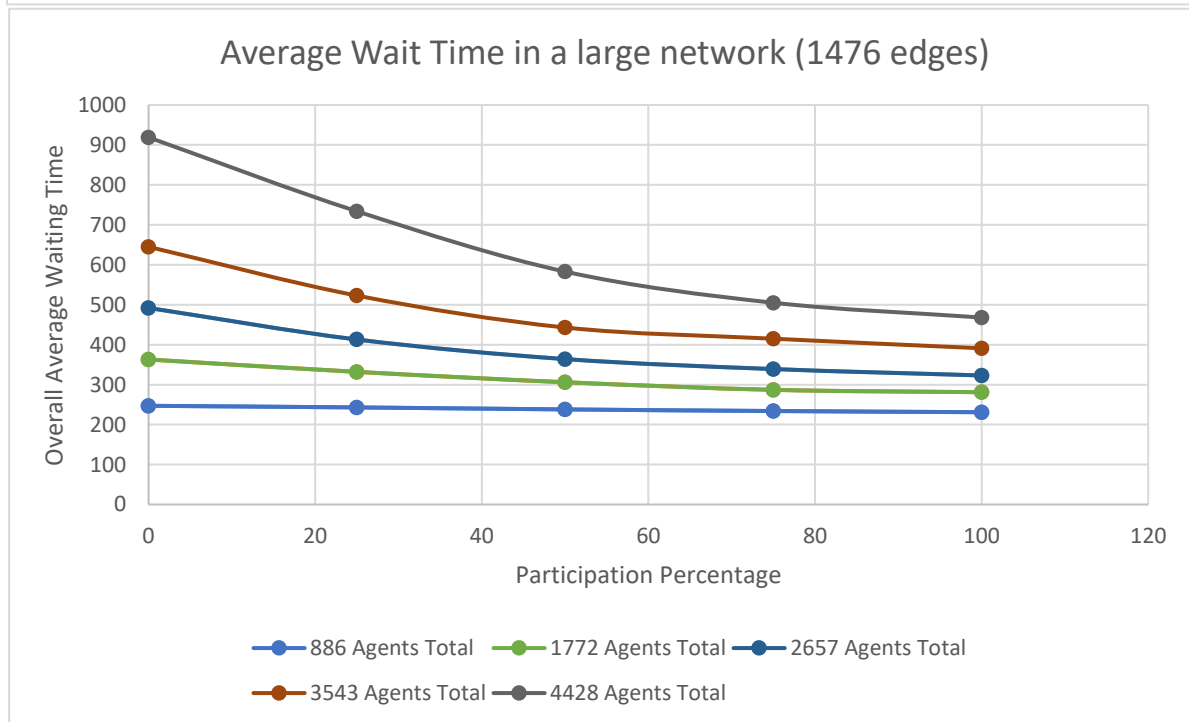
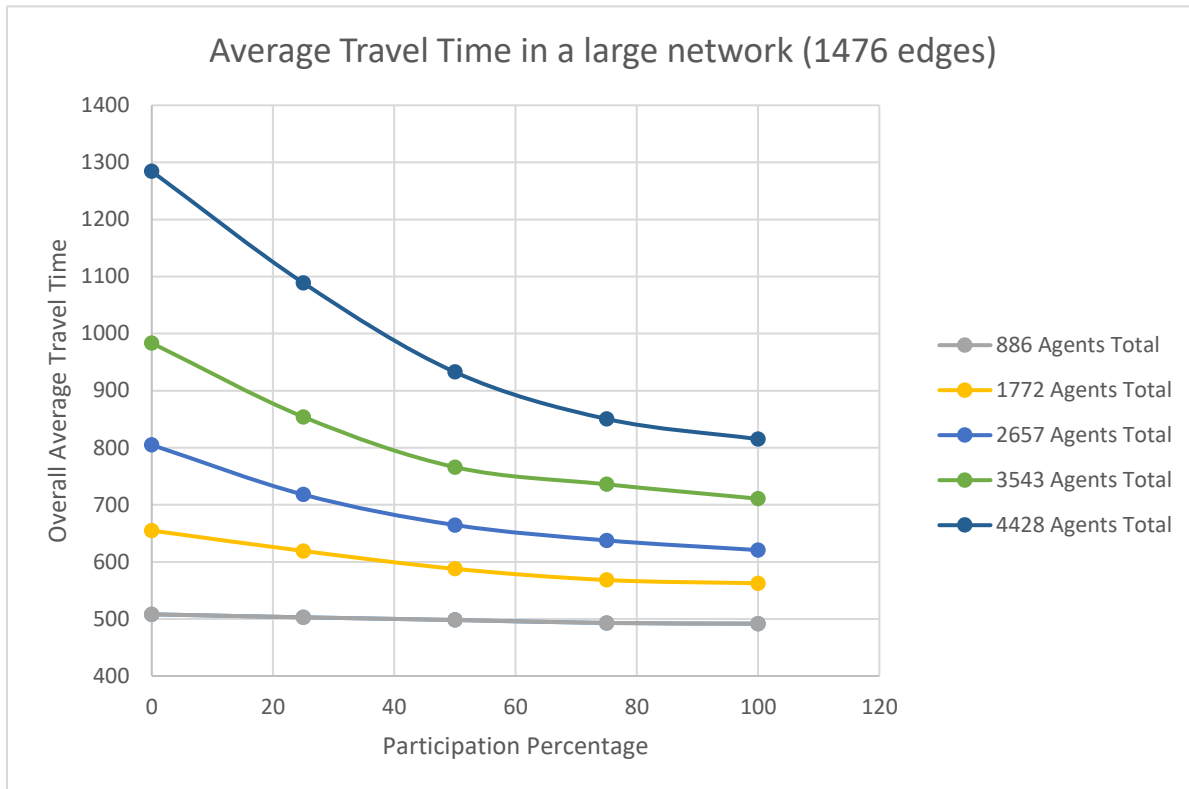
Dijkstra's algorithm was ultimately selected over BFM due to the difference in performance. The worst-case complexity of BFM is $O(|V| |E|)$, a much more costly execution than that of Dijkstra's algorithm. When scaled to very large networks, the added complexity of BFM poses a significant performance impact. The speed of Dijkstra's algorithm was therefore selected over the multi-path capabilities of BFM.

Results and Conclusion

The simulation implementation described previously was executed against a variety of input data. The output from selections of these inputs accompany this work to illustrate the results of the project's contributions. A large network consisting of 1476 edges was used with multiple population sizes to record sample results. Each of these population sizes were executed in the simulation with 0, 25, 50, 75 and 100 percent of all vehicles complying with the traffic-sensitive ACO system implemented.

It is shown in the simulation results that the model proposed in this project improves average travel times and wait times when compared with a naïve shortest-path algorithm. The simulation results also show that the proposed proactive, threshold-based routing strategy reduces the number of route executions in sufficiently small populations, but faces scaling issues when introduced to large populations.

A selection of simulation inputs were executed and recorded to analyze the results of the project's proposed pheromone model. The following two graphs illustrate the change in average wait time and average travel time for each population of vehicles. Five separate populations are executed against one large network, with increasing participation percentages.



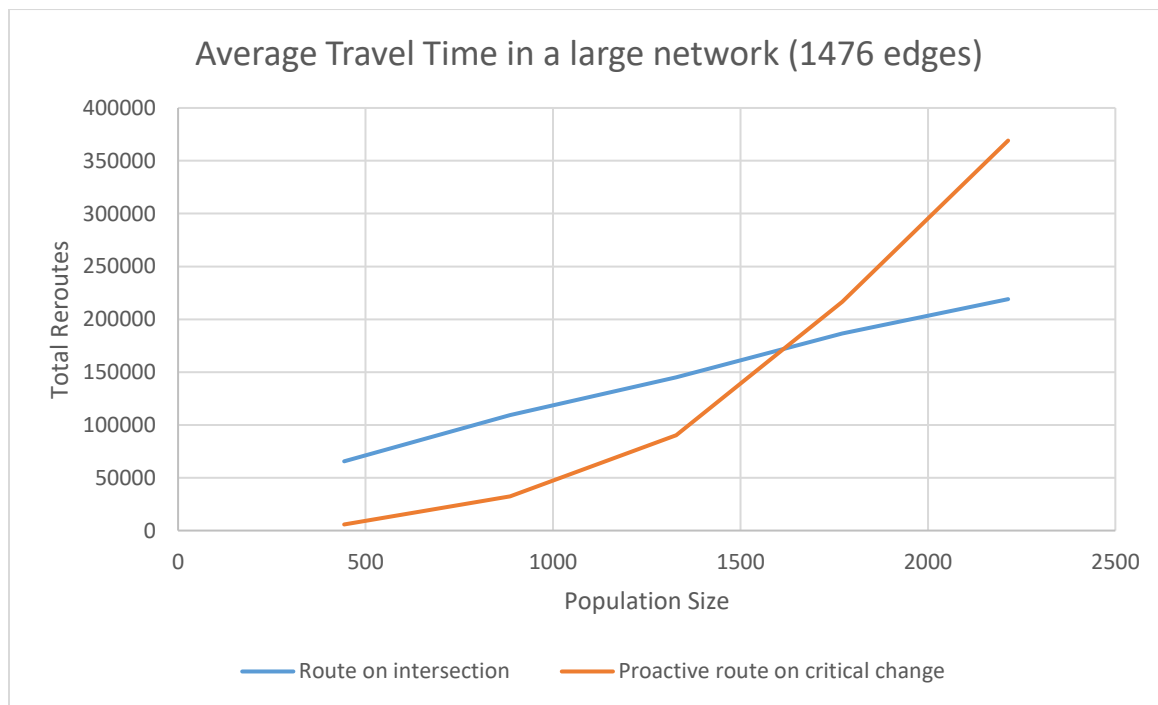
The overall travel times and overall wait times are shown to decrease as more agents participate in the traffic-sensitive ACO system. The times recorded consider all agents, including those not using the ACO model. The results show that the effectiveness of the proposed reverse-ACO strategy increases with i) the percentage of agents using the strategy and ii) the total population of agents themselves.

The values for c , r and T were tweaked based on repeated experiments to determine optimal values. These are the constant value which is added to the speed of an agent, the modifier which reduces the impact of incidental congestion or incidental congestion reduction, and the value for the maximum permissible pheromone value in a vehicle's route.

Variable	Value
c	0.01
r	0.2
T	150

Participation percentage of 0 indicates that all agents follow a static shortest-path algorithm. A vehicle's travel time is its instant of arrival subtracted from its instant of departure. Wait time corresponds to the total time a vehicle spent at a speed less than 0.1 meters per second (0.36 km/h). This is essentially the total time the vehicle spent stopped. It is shown that the introduction of an ACO strategy reduces both travel and wait times for all agents in the system.

Another set of simulation inputs were executed against the same large network to record the cost of rerouting vehicles in the system. Each population group was executed with 100% participation in order to determine the worst-case of routing a large number of vehicles. The results show that the proactive, threshold-based rerouting approach reduces the number of total reroutes in sufficiently small population sizes.



The proposed rerouting system eventually introduces a higher number of reroutes than intersection based routing due to the fact that proactive routing often leads to additional

executions of a shortest path algorithm. A vehicle may continuously search for an optimal path until the network reaches a state such that the vehicle may discover a route below the pheromone limit. While introducing congestion tolerance with a threshold system may reduce the number of optimal path searches, permitting proactive routing may introduce more executions than an intersection-based system.

Based on the recorded metrics, the IACO approach of re-routing at every intersection is less intractable than a proactive, threshold-polling system. However, the proactive-threshold system does offer improvements over IACO in terms of realism and ease of use. Forcing drivers to re-route at each intersection may be an unsatisfactory user-experience. In addition, it may not be realistic to re-route vehicles at each intersection, as such a strategy assumes that vehicles always stop once an intersection is reached.

It is worth noting that the total number of optimal path searches increases to very large numbers regardless of the approach used. This raises concerns about the scalability of this system and of similar systems. Traffic-sensitive routing introduces a high number of reroutes for participating vehicles. The number or cost of reroutes must be reduced.

This project concludes with the finding that the ACO-based traffic sensitive routing system improves the travel times and wait times of all vehicles involved. The ACO system is most effective when all vehicles in the population adhere to the system. It was also found that a proactive, threshold-based approach to routing vehicles decreases the total number of optimal path searches in certain population sizes. Further investigation is required to reduce the impact of optimal path searches in the network.

Next Steps

The next steps of this project focus on improving routing scalability. The ACO model presented in this project is strong enough to improve travel and wait times. However, the ACO system has limited scalability in terms of routing executions. Regardless of the rerouting strategy employed, a traffic-sensitive routing system will see hundreds of thousands of reroutes in the network, as shown in earlier figures.

In order to improve the scalability of this system, the impact of rerouting must be reduced. This means the number or cost of reroutes in the system must be decreased. Each vehicle reroute is necessary to maintain the integrity of the existing ACO system. Therefore, the cost of reroutes themselves must be reduced.

When vehicles re-determine their optimal path, they use Dijkstra's algorithm which inspects the entire graph. When a road network becomes very large, the scalability of repeating Dijkstra's algorithm becomes an issue. Therefore, it is proposed that the vehicle network be partitioned into *zones* in order to facilitate scalable rerouting.

Drawing from the Zone Routing Protocol (ZRP) [7], the partitioned graph will allow vehicles to reroute without inspecting every edge in the graph. The path between every pair of vertices within a zone is to be evaluated proactively, being cached in an intra-zone routing table. When vehicles need to travel between two vertices inside a zone, only a table lookup is required. Inter-zone routing will be kept as a reactive system.

Instead of evaluating each edge cost, vehicles need only find the shortest path between *zones* when attempting to travel a multi-zone distance. The introduction of zones to a road network greatly reduces the number of edges which have to be inspected by each vehicle. Instead of examining all edges individually, vehicles will only need to evaluate the path between each zone. The transfer from edge-based routing to zone-based routing is expected to greatly reduce vehicular routing cost.

Works Cited

- [1] The European Union, "Directive 2010/40/EU of the European Parliament and of the Council," *Official Journal of the European Union*, vol. 207, no. 1, pp. 1,2, 2010.
- [2] M. Barth, "Real-World CO2 Impacts of Traffic Congestion," *Transportation Research Record*, vol. 2058, pp. 163-171, 2008.
- [3] J. C. Dias, P. Machado, D. C. Silva and P. H. Abreu, "An Inverted Ant Colony Optimization Approach to Traffic," *Engineering Applications of Artificial Intelligence*, vol. 36, 2014.
- [4] J. Kponyo, Y. Kuang and E. Zhang, "Dynamic Travel Path Optimization System Using Ant Colony Optimization," in *16th International Conference on Computer Modelling and Simulation*, Cambridge, UK, 2014.
- [5] M. B. P. W. Daniel Krajzewicz, "The Open Source Traffic Simulation Package SUMO," in *RoboCup 2006 Infrastructure Simulation Competition*, Bremen, Germany, 2006.
- [6] M. P. M. R. H. H. S. F. J.-P. H. Axel Wegner, "TrACI: An Interface for Coupling Road Traffic and Network Simulators," in *Communications and Networking Simulation Symposium*, Ottawa, Canada, 2008.
- [7] Z. Haas and M. Pearlman, "Evaluation of the Ad-Hoc Connectivity with the Zone Routing Protocol," in *The Springer International Series in Engineering and Computer Science*, Ithaca, NY, Springer, 2002, pp. 202-212.