Privacy-Aware Ant Colony Optimization Algorithm for Real Time Route Planning

José António Capela Dias, Penousal Machado, Francisco Câmara Pereira
CISUC, Department of Informatics Engineering, University of Coimbra
jacdias@student.dei.uc.pt, machado@dei.uc.pt, camara@dei.uc.pt

ABSTRACT

Route planning devices have become a ubiquitous system on our lives. However, they are not normally used when travelling inside a city since the system is unaware of city's transit. Although, there are some systems that are aware of transit, many follow a reactive approach, thus only re-routing drivers after a traffic jam occurs. In this paper, we propose an algorithm that pro-actively tries to distribute possible routes for real-time city demand without the need to know in advance the origin and destination of drivers, thus eliminating the privacy concern. Our method is based on the usage of an *inverted Ant Colony Optimization* algorithm that allows a better vehicle distribution, optimizing the global efficiency of the road network. Results show that our method is able to perform better when compared to *Shortest Time* and *Shortest Distance* algorithms.

Categories and Subject Descriptors

.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence – *intelligent agents, multi-agent systems.*

General Terms

Algorithms, Measurement, Experimentation, Theory.

Keywords

Experimentation, Simulation, Traffic, MAS, Ant Colony Optimization, Agent Behavior, Artificial Intelligence.

1. INTRODUCTION

In this paper, we will study a way to influence drivers using an algorithm that pro-actively tries to distribute possible routes. The study of urban traffic is an increasingly important field given that it is necessary in order to explore new traffic-control strategies and to make important decisions in urban planning. Traffic simulation can help traffic engineers by providing a cost-effective way to model traffic flow in a realistic manner and can be used to predict these previously referred consequences and to understand certain traffic phenomena, without the overhead of costly infrastructure changes.

As previously referred, our purpose is to influence drivers and reduce traffic congestion and consequently to reduce average travel time, through traffic simulation. To accomplish this we use a multi-agent system composed of multiple interacting intelligent agents. Our strategy uses an inverted version of *Ant Colony Optimization (ACO)* algorithm in order to distribute drivers more evenly along the network. Our main contributions are:

– a distributed *Inverted Ant Colony* algorithm;

– the algorithm pro-actively distributes routes for real-time city demand;

– the algorithm does not require knowledge about the origin and destination of drivers.

The remainder of this paper is organized as follows. In section 2 we present the state of the art on driver's behavior. Section 3 describes the tool used in the development of this work. Section 4 explains the developed architecture, experiments and assumptions. Section 5 presents the results and in Section 6 the results are discussed. In Section 7, we present the final conclusions and in Section 8, the paper concludes with remarks for future work.

2. STATE OF THE ART

Dorigo and Caro [1] proposed a definition for the *Ant Colony Optimization (ACO)* meta-heuristic. *ACO* algorithms consists of using a population of ants to collectively solve an optimization problem and can be used to discover minimum cost paths through a certain graph, while respecting specific constraints. Let G = (C, L) be the graph related to a certain optimization problem. C represents the set of components that exist in the graph and L are the connections between these components. The solutions to that optimization problem can be defined as feasible paths on the graph G. While searching for the shortest path, ants deposit pheromone in the traversed route. This leaves pheromone trails that encode a long-term memory regarding the search process. The arcs of the graph might also have a heuristic value that results from a priori information or from run-time feedback. They define the properties that the ants of the colony should have such as a memory (to evaluate the current solution or to retrace the path backward), termination conditions and being able to move in its feasible neighborhood. The ants' probabilistic decision rule depends on its memory, the problem's constraints and also the ant-routing table (a local data structure with pheromone trails and heuristic values). There are two types of pheromone updates: online step-by-step update (updated by the ant when it moves through an arc) and online delayed update (updating the pheromone trails after finding a solution and retracing the path backwards). Additionally there are two other processes of updating pheromone trails: daemon actions and pheromone evaporation. Daemon actions are optional and might be used to perform actions that individual ants cannot do. Pheromone evaporation consists of decreasing the intensity of the pheromone trails over time in order to avoid convergence to sub-optimal areas and to explore new areas of the graph. Dorigo and Caro [1] applied this *ACO* meta-heuristic to two specific problems: the *traveling salesman problem* (*TSP*) and adaptive routing in communications networks. In order to solve the *TSP* problem they applied an *Ant System* (*AS*), which consists of a number m of ants positioned in parallel on m cities. When all ants have completed a tour (this happens synchronously because, during each iteration, each ant adds a new city to the path that is being constructed) they re-trace the tour backwards and increase the intensity of the pheromone trail using their memory, which is also used to avoid visiting the same city twice. In *AS*, pheromone evaporation occurs when all the ants have completed their tours and no daemon actions are performed. The amount of pheromone deposited is proportional to the quality of the produced solutions. The probability of choosing a certain connection depends on the heuristic value of that connection. Finally there are two parameters α and β that are used in the ant-routing table and that determine the importance given to distance and pheromone values. If $\alpha = 0$, the closest cities are more likely to be selected and if on the contrary, $\beta = 0$, paths with higher pheromone values will be chosen, which may lead to the rapid emergence of a stagnation. The problem of routing in communications networks is building and using routing tables to direct data traffic and maximize network performance. The ant-routing tables are bi-dimensional given that the node to which a data packet entering a generic node should be forwarded depends on the packet destination node. They developed the *AntNet* algorithm to solve this problem. In this algorithm the heuristic value of an arc is a function of the length of the queue of the link connecting two neighboring nodes. The other aspects are similar to those of the previous algorithm.

This formal description of the *Ant Colony Optimization* meta-heuristic allowed for the development of several modifications of this algorithm in order to be applied to different classes of problems.

2.1 Vehicle Routing Problem

There have been several works and different approaches addressing the *Vehicle Routing Problem (VRP),* which can be defined as the problem of minimizing time and costs of the distribution of goods performed by a fleet of vehicles, from a depot to a set of customers.

Yu et al. [2] propose a strategy to update pheromone levels, called ant-weight strategy. In the ant-weight strategy, the quantity of the global pheromone increment of each route is related to the total length of the solution. On the other hand, local pheromone increment in each link is based on the contribution of that link to the solution. They also make use of the mutation operator which performs random customer exchanges.

Bell and Mcmullen [3] suggest two modifications to the *ACO* in order to allow the search of the multiple routes of the *VRP*: use of local exchange and of a candidate list. The local exchange procedure uses the 2-opt heuristic (in which all possible pairwise exchanges of visited locations are tested) in order to improve individual routes. The other improvement strategy is the use of a candidate list for selecting the next location in a vehicle route. Each location is allocated a candidate list based on the distance to all the other elements in the location set. In terms of pheromones, they simulate natural evaporation through reducing the amount of pheromone on all visited arcs (local updating). Global updating consists on adding pheromone to all of the arcs included in the best route solution route. Negulescu et al [4] present a somewhat similar idea by introducing an *Elitist Ant System (EAS)* in which each ant that finds a better solution has the chance to deposit more pheromone.

Donati et al. [5] address the *Time Dependent Vehicle Routing Problem* by suggesting new time dependent local search procedures (such as *Customer relocation*, *Customer exchange* or *Branch exchange*) and the use of two Ant colony systems (*ACS*): *ACS-TIME* – a colony that has the objective of minimizing the total length of the solution; *ACS-VEI* – a colony which attempts to find a feasible solution with a lower number of tours than the best solution found so far. They also claim that in order to attain a faster construction of the solution it is useful to introduce a set of neighbors for each customer given that in an optimized solution there will never be trips between distant locations.

Gambardella et al. [6] makes use of two ant colonies: one colony minimizes the number of vehicles and the other minimizes the traveled distances. The cooperation between colonies is executed by the exchange of information through pheromone updating. This approach makes the vehicle routing problem closer to the traditional traveling salesman problem.

Zabala and Torres [7] propose the use of the *Ant Colony System with a Local Search* algorithm to solve the *Vehicle Routing Problem with Capacity and Time Windows*, in order to minimize the cost (traveled distance) of the delivery routes. In the beginning, ants are placed in the central depot and start moving to unvisited nodes by applying exploitation and exploration, without violating the capacity restriction and time windows. Global and local pheromone updates are performed along the tours. Choosing the next state is determined by a function that depends on pheromone levels, distance and also two random variables that determine the relative importance of pheromone vs. the distance and exploitation vs. exploration. To improve the solutions, a local search procedure is implemented using the *CROSS*-exchange operator (which interchanges consecutive customer segments between two different routes) after all the solutions were generated. The solutions that are within a percentage above the best solution found form part of the Local Search process. They also used the *Variance Analysis (ANOVA)* statistical test (which is used to study the relationship between a dependent variable and independent variables) in order to reach more reliable conclusions. In terms of parameters, they came to the conclusion that the best results are obtained when the pheromone level and the distance between nodes have the same of importance, there is a medium global pheromone evaporation level and a there is low local evaporation during the construction of the routes. Also, when finalizing the construction of the routes, an higher evaporation level should be applied in order to avoid being trapped in local optimums.

These works are focused on a problem somewhat different from the one that we tackle in this paper but their solutions use interesting modifications to the *ACO* algorithm and allowed for a better understanding of the *Ant Colony* mechanisms and how important the tuning of parameters is in order to achieve a greater performance.

2.2 Traffic Signal Timings

The ACO algorithm has also been used to optimize traffic signal timings. Bullnheimer et al.[8] makes use of *ACO* to manage signal setting parameters like number of phases, cycle length or effective green times. The results achieved by He and Hou [9] show that *ACA* is a simple and feasible method for signal timing optimization problems.

Once again, these papers focus on a different problem from the one that we will tackle but shows the wide range of usefulness of the *ACO* algorithm.

2.3 Traffic Prediction

Claes and Holvoet [10] used a combination of the *ACO* algorithm with link travel time predictions (based on current traffic situations, historic data and real-time information provided by vehicles) to find routes that reduce travel times. Also they present a new parameter (heuristic importance) to balance heuristic and pheromone values. The introduced algorithm does not require global pheromone updates and assumes that pheromones from different vehicles do not interact. Instead, they only use local updates by exploration or primer ants. To create solutions each primer ant is sent out over each route before exploration ants and will locally increase the pheromone level of all edges. This pheromone increase will guide the exploration ants to the statistically optimal solution.

Ando et al. [11] propose a traffic prediction method that employs a pheromone mechanism and in which each car is regarded as a social insect that deposits multi-semantics of pheromone on the basis of sensed traffic information. They use the idea of alarm pheromone: a leading bee informs the subsequent bees of any danger using alarm pheromone so that other bees can avoid it. They suggest the combination of three different pheromone flavors: a *Basic Traffic Pheromone* in which speed represents congestion rate; a *Braking Pheromone* in which the deposited amount depends on the number of times its brakes are applied; a *Distance Pheromone* that unlike the previous two is a pheromone of attraction that informs following vehicles about the possibility of a decrease in congestion. This pheromone flavor however assumes that vehicles are "equipped with a millimeter-wave radar". As transition functions they make use of a common evaporation mechanism (pheromone levels decreases over time) and implemented a new mechanism to represent pheromone propagation which represents the characteristic of pheromone trails spreading throughout the surrounding environment. The results of their experiments (which used real traffic data) confirm the applicability of their method to short-term traffic prediction. However, their implementations use an *Intelligent Transportation Systems (ITS)* infrastructure called the probe-car system, which only exists in Japan.

Our aim is to create a cooperative system that could be used in real-time. Claes and Holvoet [10] assume access to historic data in order to make travel time predictions. Also, their proposed use of primer and exploration ants means that for each route, the path will be traversed twice. Finally, their system is not cooperative given that pheromones from different vehicles do not interact and consequently the other vehicles in the network do not benefit from the information brought back by that particular ant. Ando et al. [11] propose a cooperative system that makes short-term predictions of traffic. There are some similarities between their concept and ours but we do not intend to actually predict traffic. Traffic prediction is a very complex process given the instability of traffic: even a short-term prediction can be rendered useless if, per example, an accident occurs. So we decided that our focus was mainly dealing with real time information.

2.4 Other approaches

Another known problem is the *Railroad Blocking Problem (RBP)*, that is addressed by Yue et al. [12], which consists in minimizing "the total operating costs of delivering the traffic on the railway network while satisfying the resource and capacity constraints at the stations and the priority constraints for shipments." In this paper they apply the *ACO* algorithm in a similar fashion to how they would solve the *TSP* problem.

Another application is to create routing algorithms for mobile ad-hoc networks (*MANET*), which are a collection of mobile nodes that communicate over radio. Gunes et al. [13] propose an new *ACO* algorithm called *Ant colony based Routing Algorithm* (*ARA*) which consists of three phases: the route discovery phase (use of a forward ant and a backward ant to create new routes), the route maintenance phase (responsible for route improvement during communication) and the route failure handling phase (routing failures are very common in mobile ad-hoc networks).

Another cooperative ACO approached is explored by Claes and Holvoet [14]. The ACO procedure allows the knowledge of previous found solutions to be embedded into the graph. Other ants then use this knowledge to guide them. This indirect means of communication and coordination is called *stigmergy* and is explored in that paper. The most desirable routes are those that lead to locations that are near their destination. To achieve this, locations are grouped based on the region they are in. However, maintaining the additional data needed for the region specific information presents an overhead compared to the original algorithm.

Garro et al. [15] propose evolving some parameters of the *ACO algorithm through a genetic algorithm* (*ACO-GA*). They introduce a new transition rule that overcomes the limitation of knowing the distance between two cities (based on the fact that real ants choose a path solely based in the levels of pheromone). So, they use a GAS to evolve the parameters α, β and γ from the transition rule. In terms of the genetic algorithm they make use of a fitness function and a crossover operator. The fitness function is based on two kinds of ants (workers and explorers). A worker ant reaches its destination and is carrying food. The best ant worker is that one that has more food. An explorer ant does not possess food and is searching for its destination. The crossover operator randomly combines the α, β and γ parameters of the best workers to generate new offspring.

Robinson et al. [16] explores Pharaoh's ant colony that comprises two types of trail pheromones: attractive and repellent. Experiments have previously shown that Pharaoh's ants use both types of pheromone. Their results show that with low traffic flow, pheromone decay overwhelms pheromone deposition indicating that small colonies of Pharaoh's ants cannot establish organized foraging trails. On the other hand, their model also predicts that above a certain threshold, increasing the ant flow will not increase foraging success.

These papers offer some interesting concepts such as worker and explorer ants, forward and backward ants and the use of attractive and repellent pheromones.

3. TOOLS

In order to simulate the traffic system logic we needed a road traffic simulator. Chen and Cheng [17] analyzed a wide range of agent-based traffic simulations systems and divided them into five different categories:

- agent-based traffic control and management system architecture and platforms;
- agent-based systems for roadway transportation;
- agent-based systems for air-traffic control and management;
- agent-based systems for railway transportation;
- multi-agent traffic modeling and simulation.

We opted for the *Simulation of Urban MObility (SUMO)* [18] which is an open source, microscopic road traffic simulation package that offers the possibility to simulate how a given traffic demand moves through a given road network. It was analyzed by [8] and is described as being multi-modal, which means that there can be various types of transportation vehicles besides passenger car. It is also purely microscopic, meaning that every vehicle has its own route, and moves individually through the network. It is a space continuous, time discrete (the default duration of each time step is one second) and collision-free system. It also offers support for traffic lights implementation. In our system we used *SUMO* version 0.12.3.

4. METHODOLOGIES

In this paper, we approach this theme by using a Multi-agent System (MAS) to simulate the effect that the usage of the proposed algorithm would have on city transit.

4.1 System Architecture

Our system is divided into five main modules:

- *Agent*: contains all the information and actions relative to each individual agent;
- *Constant*s: contains global constants that are used by the other modules;
- *Controller*: the main class. It is responsible for parsing the network file, creating the network and the population, communicating with SUMO and saving the simulation results to a file;
- *Network*: contains the relevant information of the network in question: the layout, the distances and time step occupancies. It is graph in which edges (*i, j*) connect vertices *i* and *j*. The edges represent existing roads and vertices describe existing junctions;
- *Population*: serves as an intermediate between the Controller and the Agents and is responsible for creating the agents, parsing the route files and attributing routes to the agents and for communicating.

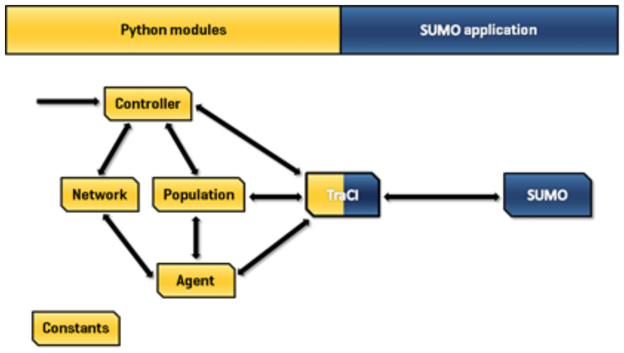

Figure 1 - Communication Process

4.2 Inverted ACO

In order to more effectively spread vehicles throughout the network the idea to implement an *Ant Colony Optimization (ACO)* algorithm emerged. This well-known algorithm is used to find optimal paths in a graph: each ant lays down pheromone trails and other ants are likely to follow the trail (instead of travelling at random) reinforcing it. Over time, pheromone trails starts to evaporate, thus reducing its attractive force.

4.2.1 Main Idea

Unlike in the traditional *ACO* algorithm, our goal is not to find the shortest distance path to a destination. The goal is distributing the load of vehicle across the city grid, preventing congestion, while proposing paths to individual drivers that are shortest time paths. So our implementation reverses the *ACO* logic and will make pheromone trails repel incoming vehicles.

Our approach is a combination of the *Dijkstra* and *ACO* algorithms in order to deal with the static (edge distances) and dynamic (car density) natures of a traffic network. The *Dijkstra* part allows us to satisfy the selfishness of the system's users and the *ACO* part allows for a cooperative mechanism that benefits subsequent vehicles.

This algorithm has the advantage of taking into account the speed at which the vehicles move through a certain edge. Given that, in each time step vehicles deposit pheromone in the current edge, if they traverse it fast they will deposit less pheromone. On the other hand, if there is congestion in that edge it will take much longer to traverse, greatly increasing the pheromone trail. The occupancy factor in itself is not a realistic indication of the existence of congestion given that an edge can have an occupancy of 50%, but if all the vehicles in it are moving at a high speed then there is little or no congestion. The *inverted ACO* combines occupancy and speed factors, giving a better portray of the real congestion status of the network.

A problem might arise from this traffic dispersion: we cannot simply send the cars on longer paths in order to keep away the congestion of the roads. Drivers are selfish and want to reach their destinations in the shortest possible time and our system provides them with the information to do just that. The "payback" to having access to such data is leaving a pheromone trail that allows other drivers to accurately evaluate traffic conditions.

4.2.2 Congestion calculation

The *SUMO* simulator offers a way to monitor congestion levels by presenting occupancy percentages for each edge of a network.

This data was used, prior to the implementation of the *ACO* algorithm, to create a *Shortest Time* algorithm that combined this occupancy information with each edge's distance in order to calculate an optimal path based on the levels of congestion. This however was a simplistic way to approach the problem given that it offered no indication regarding the speed at which vehicles move. In the *ACO* algorithm this would be the equivalent to only depositing pheromone once per edge. The developed *ACO* algorithm overcomes this fault given that, in each time step vehicles deposit pheromone in the current edge and so, if they traverse it fast they will deposit less pheromone. On the other hand, if there is congestion in that edge it will take much longer to traverse, greatly increasing the pheromone trail. The occupancy factor in itself is not a realistic indication of the existence of congestion given that an edge can have an occupancy of 50% but if all the vehicles in it are moving at a high speed then there is little or no congestion. The *inverted ACO* combines occupancy and speed factors, giving a much more realistic portray of the real congestion status of the network.

4.2.3 Initial Procedure

As previously referred, our approach is a combination of the *Dijkstra* and *ACO* algorithms. In the initial stage the developed algorithm acts solely as the *Dijkstra* algorithm given there is no pheromone data in the network yet. The following implementation of the *Dijkstra* algorithm [19] was used:

```
D = {}   # dictionary of final distances
P = {}   # dictionary of predecessors
Q = priorityDictionary()   # est.dist. of non-final vert.
Q[startEdge] = 0

for v in Q:
    D[v] = Q[v]
    if v == endEdge:
        break
    for w in self.__graph[v]:
        vwValue = 0
        vwValue = D[v] + G[v][w]
        if w in D:
            if vwValue < D[w]:
                raise ValueError, "Dijkstra: found better path
to already-final vertex"
            elif w not in Q or vwValue < Q[w]:
                #found shorter alternative
                Q[w] = vwValue
                P[w] = v
```

Figure 2 – Dijkstra algorithm

The formula that is used during the optimal path calculation is the following for each edge:

$$edgeCost\,(i,j) \;=\; distance\,(i,j) \;+\; edgePheromones\,(i,j)$$

However, although merely choosing a path based on edge distances, these first agents start depositing pheromone along their traversed route. As the agents begin to spread throughout the network, the system begins to form a clearer picture of the state of the network. Ideally, assuming that all vehicles in the network use this routing algorithm, the system would be able to offer a very realistic portrayal of congestion levels throughout the whole network.

Each time step, all vehicles receive updated information about the current pheromone levels and adjust their routes accordingly. In order to improve efficiency, route re-calculations are only allowed when vehicles are approaching an intersection. Otherwise, while following a straight line, vehicles would re-calculate their route several times without any real benefit to them and largely increasing the simulations' computational time.

4.2.4 Pheromone Deposit and Evaporation

Initially, only global pheromone evaporation was implemented. At each time step, the pheromone levels were decreased in every edge, according to the edge's length, i.e., longer edges had greater evaporation rates.

The obtained results were not satisfactory. This is due to the fact that this global evaporation does not take into account how many vehicles are in each edge. So, in a long traffic queue, pheromone levels will rise greatly and will remain high for a considerable amount of time. Consequently, in order to improve performance, a local evaporation that took into account the quantity of vehicles present in a given road was added to the algorithm.

This local evaporation consists on each vehicle withdrawing pheromone after leaving a certain edge. This withdrawal corresponds to minimum amount that a vehicle would deposit if traveling that edge at full speed:

$$minimum\ travel\ time\ = \frac{edge's\ length}{edge's\ maximum\ allowed\ speed}$$

This mechanism is more accurate and dynamic given that, if there is no congestion in a given edge, the vehicle will be able to go through it rapidly and will erase its presence after leaving the edge, therefore updating pheromones to a more realistic level. On the other hand, if there is already congestion and the vehicle is forced to slowly travel through that edge, the amount deposited will be far superior to the minimum amount deposit, leaving a trail of its presence there. This pheromone decrease encourages other ants to choose this edge, thus improving the exploitation factor of the search.

In short, this is the pheromone procedure:

```
if not willLeaveEdge(edgeId, agentId):
    pheromones[edgeId] += DEPOSIT_PHEROMONE
else:
    pheromones[edgeId] -=
               getMinimumTravelTime(edgeId)*DEPOSIT_PHEROMONE
```

4.2.5 Pheromone Variation History

Another important factor in evaluating current congestion is understanding the pheromone variation trend: if the pheromone level in one edge decreased in the last *N* time steps, then it is likely that the number of cars in that edge has diminished and consequently so has the congestion; on the other hand, an edge might currently have a low pheromone level although the tendency of the last *N* time steps might indicate an increasing demand for that edge.

Basically, this pheromone variation history mechanism adds a short history of pheromone levels in each edge, in order to better understand the variation of the traffic volume and to more efficiently evaluate traffic congestion.

The formula that is used during the optimal path calculation is altered in the following manner:

$$congestion\ (i,j) = edgePheromones\ (i,j) + C * edgePheromoneVariation\ (i,j)$$

(where C is the relative importance of the variation)

```
if $congestion\ (i,j) < 0$:
    $congestion\ (i,j) = 0$
```

(negative congestion values would imply that the physical distance was shorter than what it actually is)

$$edgeCost\ (i,j) = distance\ (i,j) + congestion\ (i,j)$$

5. Experimentation

In this section we present and analyze the experimental results. The objective of these experiments is to evaluate the performance of the proposed algorithms.

The testing process used *Shortest Time* (ST) and *inverted ACO* (IACO) algorithms and consisted in the simulation of two sets of routes, for 5.000 and 10.000 drivers. These routes were executed in two different networks: *Lattice* and *Radial and Ring*. These networks are illustrated in the following figures:

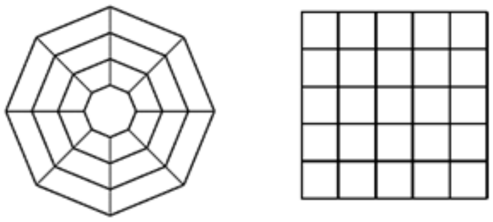

Figure 3 – Radial and Ring (left) and Lattice networks (right)

For each amount of drivers, 5.000 and 10.000, we generated 5 drivers per step/second. Also, for each of those amounts, we varied the percentage of active users of our system, testing with the following percentages: 0%, 25% and 75%. Each of these variations was simulated 30 times in order to enable averaging of the results and consequently the production of reliable data.

5.1 Experimental Results

Firstly we will compare the *Shortest Time (ST)* and *inverted ACO (IACO)* algorithms on the Radial and Ring and Lattice maps.

Figures 4 to 7 refer to experiments with 5.000 vehicles. The green line represents the behavior of vehicles when using the standard algorithm implemented in SUMO, providing a baseline for comparison.

In Figures 4 and 5 we can observe that, in *Radial and Ring* map, the *IACO* algorithm requires a higher percentage of active users than the ST algorithm. This can be explained by the lack of pheromone information that occurs with low user percentages. However, as the user percentage increases it achieves better results than the *ST* algorithm, always managing to maintain a significantly smaller route length. On the other hand, with a 100% user percentage, *IACO's* trip duration efficiency is similar to the one obtained by using the *ST* algorithm, although travelled distance is significantly smaller. The experiments conducted using the *Lattice* map indicate a similar behavior. Like previously, the *IACO* algorithm requires relatively high usage percentages to attain average trip durations that are competitive with the ST algorithm. In terms of traveled distance *IACO* clearly outperforms ST.

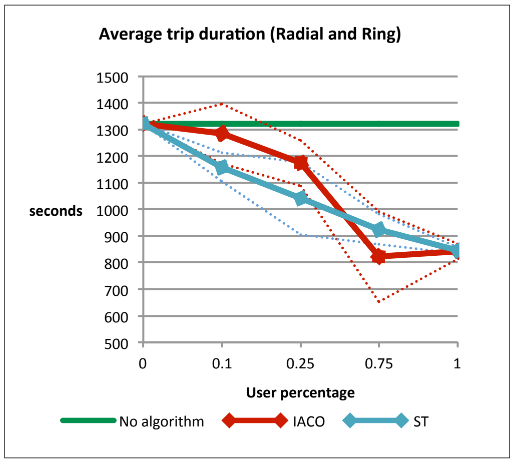

Figure 4 –Average trip duration with 5000 vehicles: Radial and Ring map

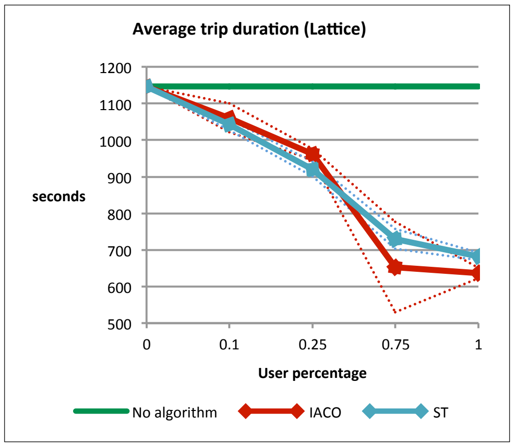

Figure 5 –Average trip duration with 5000 vehicles: Lattice map

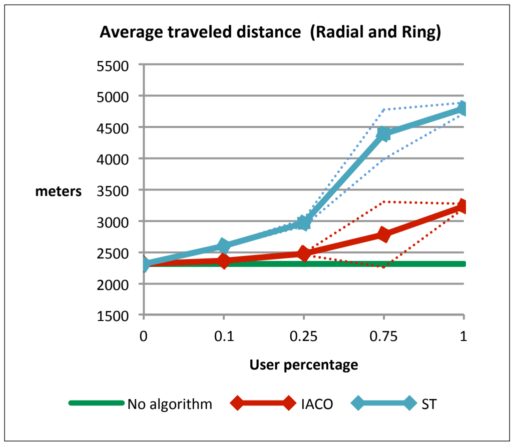

Figure 6 – Average traveled distance with 5000 vehicles: Radial and Ring map

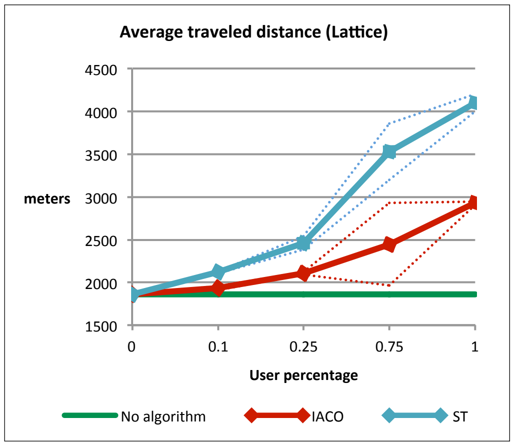

Figure 7 – Average traveled distance with 5000 vehicles: Lattice map

Figures 8 to 11 refer to experiments with a higher traffic load (10.000 vehicles):

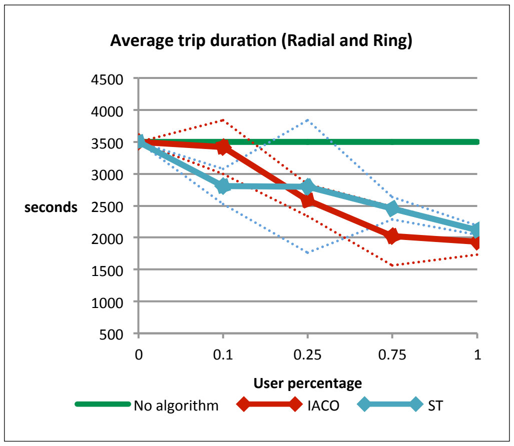

Figure 8 – Average trip duration with 10.000 vehicles: Radial and Ring map

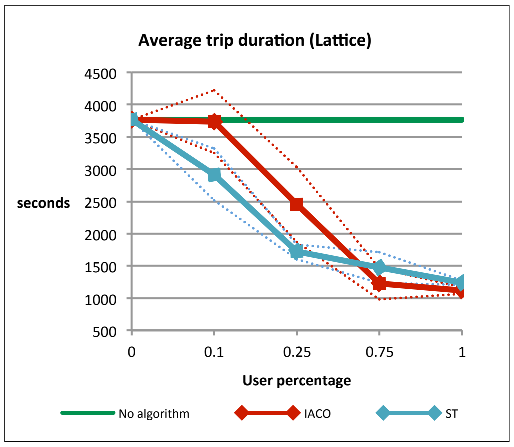

Figure 9 – Average trip duration with 10.000 vehicles: Lattice map

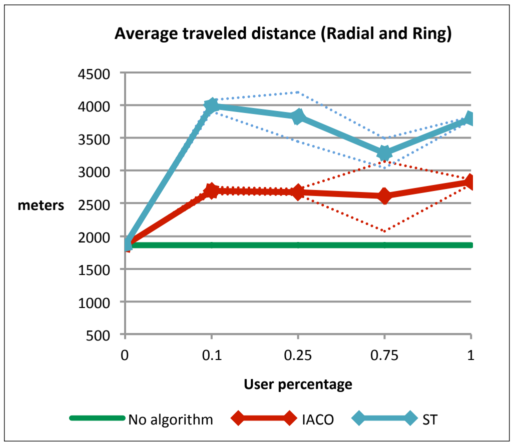

Figure 10 – Average traveled distance with 10.000 vehicles: Radial and Ring map

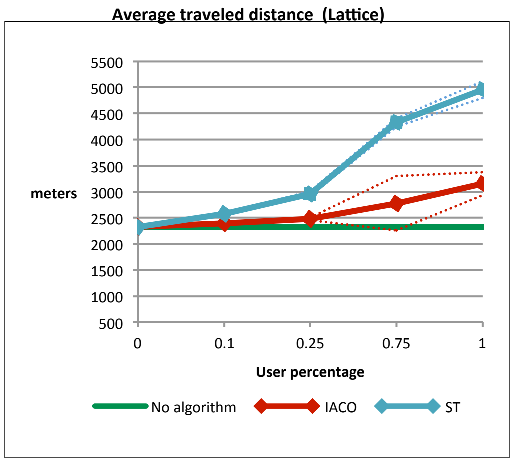

Figure 11 – Average traveled distance with 10.000 vehicles: Lattice map

As it can be observed in Figures 8 to 11, with a higher traffic load *IACO* appears to improve in the *Radial and Ring* map, achieving a better performance in terms of duration and route length, even for a 25% user percentage. This might be due to the fact that, even with a low user percentage, it detects heavy traffic in the center of the map and manages to avoid it. Additionally, the higher number of vehicles contributes to the presence of more pheromone information throughout the map. Once again, with very high user percentages, *IACO's* performance seems to stagnate in terms of trip duration. We can also observe, regarding the *Lattice* map, that with the increase in total number of vehicles the *IACO* algorithm has a slower start and only manages to surpass the *ST* algorithm when the user percentage reaches a high level. This may be due to the fact that, unlike in the *Radial and Ring* map, there is no central point in the network and so pheromone information is more disperse.

6. DISCUSSION

The *IACO* algorithm is based on a collaborative approach and tries to disperse traffic in real-time in order to enhance a network's traffic throughput. It relies heavily on the information provided by the users, i.e., the pheromone that they deposit.

Given its nature, with low user percentages, the information that is available about the state of the network in the *IACO* algorithm is somewhat fuzzy and incomplete. Therefore, its performance is worse than that of the *ST* algorithm. However, even with low user percentage it is still advantageous given that it offers a considerable reduction in trip length. This benefit should be appealing enough in order to convince people to adhere to the *IACO* system.

With the rise of the user percentage, the data regarding the network's state becomes more accurate and complete, leading to a better performance than its *ST* counterpart both in terms of average duration and length.

7. CONCLUSIONS

In this paper, we have presented an algorithm that pro-actively tries to distributed possible routes for real-time city demand without the need to know in advance the origin and destination of drivers. This way the privacy concern is eliminated. In our approach, based on the usage of an *inverted Ant Colony Optimization* algorithm, each simulated vehicle leaves a pheromone trail that permits other drivers to accurately evaluate traffic conditions, consequently allowing a better vehicle distribution, optimizing the global efficiency of the road network. Also, we developed a mechanism that, by keeping a short history of pheromone levels in each edge, allows us to better understand the variation of the traffic volume and to more efficiently evaluate traffic congestion.

To validate our approach we conducted several experiments using radial and Ring and Lattice traffic networks. The experimental results show that the *IACO* algorithm outperforms the standard SUMO routing algorithm vehicles as well as the *ST* algorithm in terms of average traveled distance. For moderate to high usage percentages the *IACO* algorithm also outperforms both in terms of trip duration.

In conclusion, this work shows that collaborative algorithms can greatly enhance a network's traffic throughput when compared to simple individual routing algorithms with general traffic information.

In terms of future work we plan to conduct experiments using real-life traffic networks and information. These experiments will focus on three main aspects.

- To study the initial viability and appeal of the system, more attention will be given to the behavior with low percentages of users: in order to be attractive and to convince people to adhere to it, the system should offer advantages even with low usage percentage.
- The adoption of the *IACO* algorithm by a percentage of users has an impact on the overall traffic flow; Thus, since it leads to lower congestion, it is fair to expect benefits even for drivers that do not adhere to the system. Measuring the impact of the algorithms on other drivers is also considered a priority.
- The data gathered in these experiments appears to indicate that IACO may lead to significant reduction of fuel consumption and CO_2 emissions, which means that this improvement in network efficiency is eco-friendly and allows drivers to save money; it is necessary to confirm it experimentally.

8. ACKNOWLEDGMENTS

The authors are grateful for the support and advice provided by João Oliveirinha. This work was partially funded by the Portuguese Foundation for Science and Technology (FCT) under project COSMO with reference PTDC/EIA-EIA/108785/2008. It is also supported by CISUC, financed by FEDER through the POFC - COMPETE and by FCT, project FCOMP-01-0124-FEDER- 022703 and by the iCIS project (CENTRO-07-ST24-FEDER-002003) which is co-financed by QREN, in the scope of the Mais Centro Program and FEDER.

9. REFERENCES

[1] Dorigo, M. & Di Caro, G. (1997). Ant colony optimization: a new meta-heuristic. *Proceedings of the Congress on Evolutionary Computation*, Vol. 2 (June-September 1999). 1470-1477. doi:10.1109/CEC.1999.782657

[2] Yu, B., Yang, Z., & Yao, B. (2009). An improved ant colony optimization for vehicle routing problem. *European Journal Of Operational Research*, *196*(1), 171-176. Elsevier B.V. Retrieved from http://linkinghub.elsevier.com/retrieve/pii/S0377221708002373

[3] Bell, J., & Mcmullen, P. (2004). Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics*, *18*(1), 41-48. Retrieved from http://linkinghub.elsevier.com/retrieve/pii/S1474034604000060

[4] Negulescu, S. C., Kifor, C. V., & Oprean, C. (2008). Ant Colony Solving Multiple Constraints Problem : Vehicle Route Allocation. *Communications*, *III*(4), 366-373.

[5] Donati, A., Montemanni, R., Casagrande, N., Rizzoli, A., & Gambardella, L. (2008). Time dependent vehicle routing problem with a multi ant colony system. *European Journal Of Operational Research*, *185*(3), 1174-1191. Elsevier. Retrieved from http://linkinghub.elsevier.com/retrieve/pii/S0377221706006345

[6] Gambardella, L. M., Taillard, É., & Agazzi, G. (1999). MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows. (D. Corne, M. Dorigo, & F. Glover, Eds.)*New Ideas in Optimization*, (IDSIA-06-99), 63-76. McGraw-Hill. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.45.5381

[7] Zabala, C., A. & Torres, J. F. (2006). Implementation of the ant colony system with local search for the vehicle routing problem with capacity and time windows (CVRPTW). *Third International Conference on Production Research – Americas' Region 2006*

[8] Bullnheimer, B., Hartl, R. F., & Strauss, C. (1999). An improved ant system algorithm for the vehicle routing problem. *Annals of Operations Research*, *89*(POM-10/97), 319-328. Springer. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.49.1415

[9] He, J., & Hou, Z. (2012). Ant colony algorithm for traffic signal timing optimization. *Advances in Engineering*

Software, *43*(1), 14-18. Elsevier Ltd. Retrieved from http://linkinghub.elsevier.com/retrieve/pii/S0965997811002419

[10] Claes, R., & Holvoet, T. (2011). Ant Colony Optimization applied to Route Planning Using Link Travel Time Predictions. *2011 IEEE International Symposium on Parallel Distributed Processing Workshops* (pp. 358-365).

[11] Ando, Y., Fukazawa, Y., Masutani, O., Iwasaki, H., & Honiden, S. (2006). Performance of pheromone model for predicting traffic congestion. *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems AAMAS 06*, 73. ACM Press. Retrieved from http://portal.acm.org/citation.cfm?doid=1160633.1160642

[12] Yue, Y., Zhou, L., Yue, Q., & Fan, Z., (2011) Multi-route railroad blocking problem by improved model and ant colony algorithm in real world, *Computers &amp; Industrial Engineering, Volume 60, Issue 1, February 2011*, 34-42. Retrieved from http://www.sciencedirect.com/science/article/pii/S0360835210002615)

[13] Gunes, M., Sorges, U., & Bouazizi, I. (2002). ARA-the ant-colony based routing algorithm for MANETs. *Proceedings International Conference on Parallel Processing Workshop*, *34*, 79-85. IEEE Comput. Soc. Retrieved from http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1039715

[14] Claes, R. & Holvoet, T., (2012). Cooperative ant colony optimization in traffic route calculations. *Practical Applications of Agents and Multiagent Systems*

[15] Garro, B., Sossa, H., & Vazquez, R. (2007) Evolving ant colony system for optimizing path planning in mobile robots. *Electronics, Robotics and Automotive Mechanics Conference (CERMA 2007),* 444-449. doi: http://doi.ieeecomputersociety.org/10.1109/CERMA.2007.60

[16] Robinson, E. J. H., Ratnieks, F. L. W., & Holcombe, M. (2008). An agent-based model to investigate the roles of attractive and repellent pheromones in ant decision making during foraging. *Journal of Theoretical Biology*, *255*(2), 250-8. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/18778716

[17] Chen, B. & Cheng, H. H. (2010). A Review of the Applications of Agent Technology in Traffic and Transportation Systems. *IEEE Transactions on Intelligent Transportation Systems*, 11(2), 485-497. doi:10.1109/TITS.2010.2048313

[18] Michael Behrisch, Laura Bieker, Jakob Erdmann and Daniel Krajzewicz. *SUMO - Simulation of Urban MObility: An Overview* In: SIMUL 2011, The Third International Conference on Advances in System Simulation, 2011.

[19] Dijkstra's algorithm for shortest paths, available at: http://code.activestate.com/recipes/119466-dijkstras-algorithm-for-shortest-paths