ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

Engineering Applications of Artificial Intelligence 00 (2014) 1–20

Eng. Applications of AI

An Inverted Ant Colony Optimization Approach to Traffic

José Capela Dias, Penousal Machado, Daniel Castro Silva and Pedro Henriques Abreu[a]

[a]*Department of Informatics Engineering, University of Coimbra / CISUC – Center for Informatics and Systems of the University of Coimbra*
e-mail: jacdias@student.dei.uc.pt, machado@dei.uc.pt, dcs@dei.uc.pt, pha@dei.uc.pt

Abstract

With an ever increasing number of vehicles traveling the roads, traffic problems such as congestions and increased travel times became a hot topic in the research community, and several approaches have been proposed to improve the performance of the traffic networks.

This paper introduces the Inverted Ant Colony Optimization (IACO) algorithm, a variation of the classic Ant Colony algorithm that inverts its logic by converting the attraction of ants towards pheromones into a repulsion effect. IACO is then used in a decentralized traffic management system, where drivers become ants that deposit pheromones on the followed paths; they are then repelled by the pheromone scent, thus avoiding congested roads, and distributing the traffic through the network.

Using SUMO (Simulation of Urban MObility), several experiments were conducted to compare the effects of using IACO with a shortest time algorithm in artificial and real world scenarios – using the map of a real city, and corresponding traffic data.

The effect of the behavior caused by this algorithm is a decrease in traffic density in widely used roads, leading to improvements on the traffic network at a local and global level, decreasing trip time for drivers that adhere to the suggestions made by IACO as well as for those who don't. Considering different degrees of adhesion to the algorithm, IACO has significant advantages over the shortest time algorithm, improving overall network performance by decreasing trip times for both IACO-compliant vehicles (up to 84%) and remaining vehicles (up to 71%). Thus, it benefits individual drivers, promoting the adoption of IACO, and also the global road network. Furthermore, fuel consumption and CO_2 emissions from both vehicle types decrease significantly when using IACO (up to 49%).

Keywords: Inverted Ant Colony Optimization, Traffic Simulation

1. Introduction

The domain of Intelligent Traffic Systems comprises several areas of application and research, including vehicle tracking, traffic load prediction and computation or real-time signal control, among others [1]. In particular, traffic management systems, which includes many sub-problems [2], is one of the most active topics nowadays. It has severe ecological implications and is considered vital to the sustainability of cities [3]. These concerns may become decisive in the approach to some classical problems such as finding the optimal route to a destination. Traditional driver behaviour tends to be a selfish one, searching for the shortest/fastest path to the destination, which ultimately causes traffic jams in widely used arteries, which in turn causes the emission of large amounts of CO_2 and other polluting chemicals [4, 5].

The main objective of this paper is to study how influencing the behavior of each individual driver, while maintaining his selfishness, would improve a network efficiency – achieving maximum throughput in the road network and minimizing road congestion. Therefore, and in order to perform the necessary simulations, we will look at the city as a complex system and each driver as an individual agent. This decentralized approach also respects the privacy concerns of drivers, since no personal data is collected, and no unique identifiers are required.

By combining GPS systems with real-time traffic information, we have a means to distribute information about traffic to virtually all drivers. However, in order to make everyone contribute to the greater good, there is a need to constantly spread updated network information to the drivers and to actually use their selfishness in order to distribute them more uniformly throughout the network. In this paper we will analyze different algorithms to try to achieve the proposed goal, with special emphasis on the developed variation of the Ant Colony Optimization (ACO) algorithm – Inverted Ant Colony Optimization (IACO). This inversion of the ACO algorithm consists in having the pheromones causing repulsion instead of attraction, trying to simulate the drivers' aversion over traffic congestions; IACO is explained in more detail in section 3. The results, provided by a road traffic simulator that reproduces the individual behavior of the agents, show that a network's efficiency can be greatly improved by the use of this algorithm, reducing CO_2 emissions by up to 49%, and also reducing average trip time by up to 84% (at the expense of increasing trip length by up to 60%), which constitutes excellent results for further experiences. This is achieved by updating network weights, which allows users to avoid congestions, thus promoting a better distribution of load throughout the network.

The remaining of this paper is structured as follows: Section 2 presents a brief description of related work; Section 3 describes the developed algorithm, IACO; Section 4 illustrates the literature review regarding traffic simulators, and describing in more detail SUMO, the simulator used in this work. Section 5 describes the experimental setup used for the simulation experiments, while section 6 presents the obtained results. Finally, section 7 presents the conclusions and some lines of future work.

2. Literature Review

Ant Colony Optimization was proposed in the early nineties by Marco Dorigo [6] [7]. The ACO algorithms consist 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. Informally, its implementation involves the collaboration between a set of ants: a small set is responsible for laying down a pheromone trail and the other set follows that trail, reinforcing it, avoiding random moves in a network. Through time, the intensity of the trail will be reduced promoting also the reduction of the ant's attraction.

Let $G = (V, E)$ be the graph related to a certain optimization problem, where V represents the vertexes and E the edges. The solutions to that optimization problem can be defined as feasible paths on the graph G. While searching for the shortest path, ants deposit pheromones 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 (deposit of pheromone 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) [8]. 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, such as centralized actions. 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.

Algorithm 1 illustrates the generic implementation of the ACO algorithm [9], where A is the set of ants, E is the set of edges in the graph, and $B \in E$ is the set of edges that form the best solution.

ACO has been applied to numerous domains, to solve optimization problems. One of the most widely known such problems is the Traveling Salesman Problem (TSP), which is tackled in [10]. Another application is in the communications area, where the AntNet variation of the algorithm allows for adaptive routing in communication networks [11]. Several approaches using ACO have also been made in the traffic area. Claes and Holvoet [12] use a combination of ACO with link travel time prediction to find routes that reduce travel times, achieving at best an improvement of 30% over the A* algorithm, used as a basis for comparison. This was accomplished using different types of ants: *primer ants* first mark the path obtained from running a static analysis using A*, and afterwards

Algorithm 1 Generic Ant Colony Optimization Algorithm

```
Initialize
while stop criteria are not met do
  for all ant a in A do
    Position a in startNode
  end for
  repeat
    for all ant a in A do
      Choose nextNode
      Pheromone_(currentNode,nextNode) += Update
    end for
  until every ant has a solution
  for all edge e in B do
    Pheromone_e += Deposit
  end for
  for all edge e in E do
    Pheromone_e -= Evaporation
  end for
end while
```

exploration ants attempt to balance exploration and exploitation of the network to determine the best path between two points in a real city road network. This solution, however, and despite working well for the pre-determined pair of origin-destination, assumes that ants and pheromones from different vehicles do not interact, while the solution presented in this paper assumes a global pheromone network, available to all vehicles. Also, another major difference is that in their work, only vehicles that control the ants use the information to optimize their route, not benefiting other vehicles, while in our approach all vehicles benefit from using IACO, even if not directly using it to plan their route.

Gambardella, Taillard and Agazzi [13] apply ACO to the vehicle routing problem with time windows, using two ant colonies: one that minimizes the number of vehicles, and one that minimizes total travel distance. These two ant colonies optimize the two different objectives of this multi-objective optimization problem, having independent pheromone trails, but collaborating by exchanging information. The focus of this work, however, is very different from the one tackled in this paper, as the authors in [13] assume all vehicles belong to the same entity and the planning is made having in mind a common goal, while in this paper the planning is made by each individual vehicle, having in mind its own goals.

He and Hou [14] use ACO to manage signal setting parameters in signal timing optimization problems; the authors compare the performance of the ACO implementation with that of other algorithms, improving performance by decreasing total time delay by approximately $16,5\%$ and number of stops by 60%. This work differs from the one presented herein as it assumes no changes are made to the route of the vehicles but only to traffic flow on intersections.

Other approaches to traffic control and management exist, such as traffic signal control systems. These use historical and real-time traffic information to fine-tune control variables for traffic signals, as to prevent traffic jams, and increase network throughput. Systems like SCOOT[1] are used to improve traffic flow in several cities throughout the world, reducing travel times and number of stops at traffic signals. Approaches to traffic signal control include Multi-Agent based approaches [15], genetic algorithms [16], BDI [17] or predictive models [18], among others. Just as with the previously mentioned approaches, these also propose changes to the network (by modifying the behavior of traffic signals), while our approach doesn't entail changes to the network.

Despite the fact that ACO has been used in several traffic management problems, the authors were unable to find any work reporting an approach similar to the one presented herein, where the goal is to maximize overall network efficiency attending to trip time, CO_2 emissions and avoidance of traffic congestions in the entire network using this new variation of ACO.

[1] Split Cycle Offset Optimization Technique. More information available online at `http://www.scoot-utc.com/`

3. Inverted Ant Colony Optimization

Having the goal of this research in mind, which is to develop an approach that avoids congestion in a city without changing the drivers individual behaviors, a hybrid algorithm is proposed. Combining two algorithms, Dijkstra and IACO, this approach deals with the static (edge distance) and dynamic (car density) natures of a traffic network. In this research, a criterium is used to calculate the cost of each edge by adding the distance of that edge as calculated by the Dijkstra algorithm to the level of pheromones in that edge, as illustrated by Equation 1.

$$edgeCost\,(i, j) = distance\,(i, j) + edgePheromones\,(i, j) \tag{1}$$

At the beginning of the network analysis (which can be translated into the early hours of the day, when the first cars start to move through the city), there are no pheromones spread in the network. If only the IACO algorithm was used in this stage, it would materialize in having the ants lost without any defined path (in this case, the cars would have no idea of the path to follow to reach their destination). To avoid this cold start problem, Dijkstra's algorithm, illustrated in Algorithm 2, was used, allowing the ants to determine an optimal path to follow to their destination, but considering a network without traffic.

Algorithm 2 Dijkstra Algorithm

```
D = {}        # dictionary of final distances
P = {}        # dictionary of predecessors
Q = priorityDictionary()     # estimated distance of non-final vertexes
Q [startVertex] = 0
for all v  in Q do
    D [v] = Q [v]
    if (v == endVertex) then
        break
    end if
    for all w in G [v] do
        vwLength = D [v] + G [v] [w]
        if (w not in Q or vwLength < Q [w]) then
            Q [w] = vwLength
            P [w] = v
        end if
    end for
end for
return(D, P)
```

After selecting the shortest path with Dijkstra's algorithm, IACO is used. In an initial phase, this algorithm consists in depositing pheromones through the network. The pheromone update process has a time complexity of $O(N)$, where N is the number of vehicles, thus becoming meaningless when compared to the computational effort of running the Dijkstra algorithm for each vehicle – $O\left(\left[(E + V)\,logV\right] * N\right)$, where E and V are the number of edges and vertexes in the network. Differing from the classical ACO approaches, pheromone levels do not produce an attraction for the other ants but rather a repulsion. This repulsion simulates the traffic congestion in the graph node and the aversion of drivers to such problem. The increase of pheromone levels is therefore proportional to the number of vehicles present in each edge, but inversely proportional to their speed. This point is crucial because if the number of vehicles in each edge is high but speed is also high – as would be the case of a highway with high traffic levels yet fluent traffic – this means that there is no traffic congestion. In the absence of significant traffic, IACO will behave similarly to ST. As traffic conditions become more severe at a particular node, the weight of this node is gradually increased, through the deposit of pheromone, promoting the distribution of traffic to other nodes of the network. When traffic conditions become better, the weight will gradually decrease due to pheromone evaporation. Therefore, the responsiveness of the system depends on the pheromone deposit and evaporation rates. For instance, if the deposit rate is too low the weight adjustment will be too slow, making IACO inefficient. On the other hand, if the evaporation rate is too low, circumstantial congestions will have a prolonged effect on the behaviour of IACO, also causing inefficiency. Additionally, there is an interaction between evaporation and deposit rates. The values for both were established empirically in the course of preliminary experiments and used throughout all experiments presented in this paper. A local evaporation function has been used, based on how many vehicles are leaving a certain edge. This evaporation corresponds to the minimum amount that a vehicle would deposit if traveling that edge at full speed based on the previously explained principle:

$$minTT_{edge} = \frac{edgeLength}{edgeMaximumAllowedSpeed} \tag{2}$$

Which means that the minimum travel time is directly proportional to the edge maximum allowed speed.

Pheromone evaporation is calculated based on the travel time and or the deposit pheromone constant:

$$pher_{edge} = \begin{cases} pher_{edge} + DEP\_PHER \\ pher_{edge} - minTT_{edge} * DEP\_PHER \end{cases} \tag{3}$$

Finally, it is important to note that, in each time step, each vehicle receives the updated pheromone levels and adjusts its route accordingly. The route recalculation is only made in the node intersections due to the lack of benefit in other scenarios (e.g. if a driver following a straight line were to recalculate its path, there would be no benefit and it would increase the computational costs). Also, the pheromone variation is stored through a historical mechanism, which allows for an analysis of traffic congestion over time. Equation 4 illustrates this concept (C controls the impact factor of recent past variations):

$$\begin{aligned} &edgeCost\,(i, j) = distance\,(i, j) + edgeCongestion\,(i, j) \\ &edgeCongestion\,(i, j) = edgePheromones\,(i, j) + \\ &\quad C * edgePheromoneVariation\,(i, j) \end{aligned} \tag{4}$$

4. Simulation Environment

In order to test different methodologies and strategies to solve the problems associated with traffic, the use of a simulator becomes necessary, as real-world testing is unfeasible. The following section presents a brief description of some simulation solutions.

4.1. Traffic Simulation

Two major traffic simulator categories can be identified: macroscopic and microscopic. Macroscopic traffic simulators often use statistical approaches to model a road network as a whole, while microscopic traffic simulators simulate individual driver behaviour. The simulation type that is the most adequate for this project is microscopic simulation, as each driver has his own origin and destination, and needs to be able to autonomously make his own decisions. This type of simulator allows for this descentralized approach to be tested, and considering different percentages of drivers following the suggestions provided by the proposed algorithm, observe the effects on global network performance.

Chen and Cheng [19] analyzed a wide range of agent-based traffic simulation 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; and multi-agent traffic modeling and simulation.

Given the characteristics of this project, the adequate category is the multi-agent traffic modeling and simulation. In the mentioned paper the authors refer two open source agent-based traffic simulators:

- Multi-Agent Transport Simulation Toolkit (MATSIM) [20], a toolbox for the implementation of large-scale agent-based transport simulations, composed of several individual modules that can be combined or used standalone. It allows demand-modeling, traffic flow simulation and running simulations iteratively.

- Simulation of Urban Mobility (SUMO) [21], a portable microscopic road traffic-simulation package that offers the possibility to simulate how a given traffic demand moves through large road networks.

In addition to these two simulators, other solutions exist, e.g. Vissim [22] or Transims [23]. Some comparison studies have been performed between two or more of these simulation platforms [24, 25, 26]. Based on the conclusions of these studies, taking into consideration the commercial nature of Vissim (which presented a good performance) and the recent developments made to each of the simulation platforms, which eliminated some of the most pressing drawbacks of SUMO that were present in earlier version, the authors have decided to adopt SUMO.

4.2. Simulation of Urban MObility (SUMO)

SUMO is an open source tool and a microscopic road traffic simulation package that supports different types of transportation vehicles. Every vehicle has its own route and moves individually through the network. This tool supports traffic lights and is space continuous and time discrete (the default duration of each time step is one second). There are three main modules in the SUMO package: SUMO (which reads the input information, processes the simulation, gathers results and produces output files, with an optional graphical interface called SUMO-GUI – see Fig. 1); NETCONVERT (a tool to simplify the creation of SUMO networks from a list of edges, and also responsible for creating traffic light phases); and DUAROUTER (a command line application that, given the departure time, origin and destination, computes the routes through the network itself using the Dijkstra routing algorithm).

Figure 1: Snapshot of the SUMO environment

As input data, SUMO needs three main files, representing routes, nodes and edges. The nodes and edges files represent the vertexes and edges in the road graph, respectively. The routes file represents the traffic demand and includes information about all the agents involved in this simulation and their characteristics (departing time, maximum acceleration, maximum deceleration, driving skill, vehicle length and color) and route (list of edges). In terms of outputs, there are different types available, such as a raw output that contains all the edges and all the lanes along with the vehicles driving on them for every time step (which results in a considerable large amount of data) or log-files created by simulated detectors.

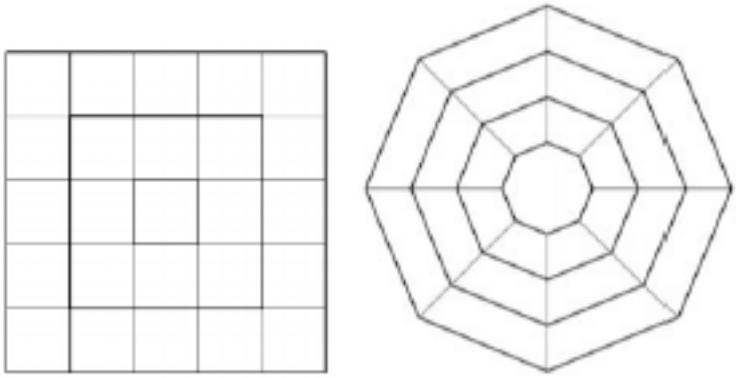
Finally, this simulator offers a way to collect metrics such as fuel consumption or pollutant emission, based on the Handbook of Emission Factors for Road Transport (HBEFA) database. According to HBEFA's website[2], it was "originally developed on behalf of the Environmental Protection Agencies of Germany, Switzerland and Austria" and is now also supported by Sweden, Norway, France and the JRC (Joint Research Centre, of the European Commission). It provides emission factors per traffic activity, i.e., it offers a way of measuring CO_2 emissions and fuel consumption, among other pollutant factors, for various vehicle categories (such as passenger cars, light duty vehicles, heavy duty vehicles, buses, coaches and motorcycles), being suitable for a wide variety of traffic situations.

[2]More information available online at `www.hbefa.net`

5. Experimental Setup

Experiments were conducted considering different percentages of vehicles adhering to the IACO algorithm: 10%, 25%, 75% and 100% of the vehicles adhering to IACO and the remaining ones following their standard shortest-path behavior. These variations were tested in three distinct scenarios: two artificial ones using Lattice and Radial and Ring networks, as illustrated in Fig. 2, and a real scenario using the city of Coimbra road network. For the artificial scenarios, two configurations were used: 5.000 vehicles and 10.000 vehicles. For the real scenario, 10.000 vehicles were used as to provide an experimental scenario simulating average traffic conditions. This paper reports the findings on the experiments performed with 10.000 vehicles for both scenarios (results for the 5.000 vehicle configuration didn't present significant differences when compared to the 10.000 configuration).

Shortest-path algorithms are usually intuitively used by drivers and several route planning tools, and then modified by the drivers daily experience to a shortest-time variation. Although most sophisticated algorithms exist, these are the most widely available and more frequently adopted by real drivers; as such, similar configurations were also used to compare the performance of IACO against a shortest-time algorithm (ST) and results were measured for both the vehicles adopting the algorithm (be it IACO or ST) and for vehicles that do not use these algorithms (these vehicles use the standard shortest-path algorithm implemented in SUMO).



(a) Lattice Network (b) Radial and Ring Network

Figure 2: Artificial Networks

Experiments were repeated 40 times for each of the eight different configuration for each scenario, as to attenuate the effect of any outliers that may appear.

Data regarding time (trip duration) and space (traveled length) were collected, as well as information regarding fuel consumption and CO_2 emissions.

5.1. Coimbra City Map as a Real Test Scenario

In order to assess the performance of the algorithm in a real world context, an actual city road network was used. In this case, the chosen city was Coimbra, a medium-sized city with a traffic network that is akin to many cities throughout the world, as can be seen in Fig. 3.

The road network was obtained from OpenStreetMap[3], and then converted into a format that could be used by SUMO, using its Netconvert tool. In order to have a realistic view on the impact of the algorithm in the city traffic, real data was used, in the form of an origin-destination (OD) matrix [27]. To create the traffic demand based on this matrix, we first needed to analyze its data. The city area was divided into several zones, as can be seen in Fig. 4, both for the inner and outer city areas (colored and white zones, respectively).

The data in this OD matrix contains real data regarding the origin and destination of about 60000 trips from the morning period (7:30 to 10:30, including the rush hour of 8:15 to 8:45), and also includes information regarding the reason for the trip (work, school, shopping, health, ...) and vehicle class. From this OD matrix, 10000 trips were selected as to provide a sample to be used for an one-hour simulation period.

[3]More information available at `http://www.openstreetmap.org/`

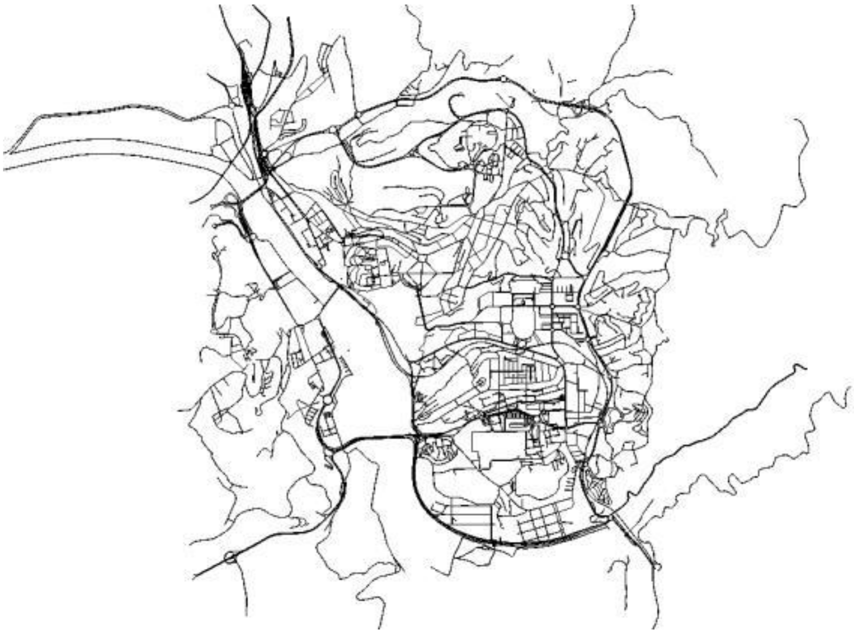


Figure 3: Map of Coimbra

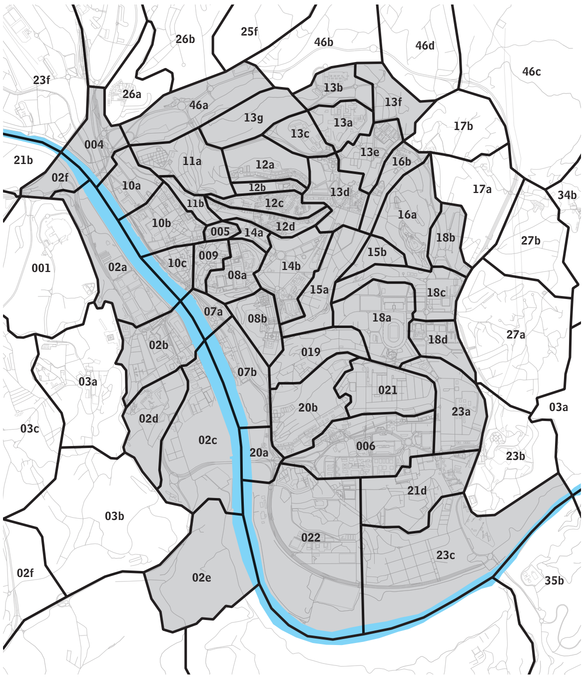


Figure 4: Coimbra OD matrix zones

5.2. *Artificial Scenarios*

The two artificial scenarios (see Fig. 2) were chosen as representative types of road networks. The Lattice Network represents classical grid pattern road networks, as can be observed in many cities, as is the case of Manhattan, New York. The Radial and Ring Network is representative of classical radial road networks, which can be found in several cities where a central focus is given to certain city locations (as is the case of some places in Paris, France).

In the artificial scenarios (both Lattice and Radial and Ring networks), the OD matrix was artificially generated, based on a random process with uniform distribution that generates origins and destinations within the networks.

6. Results

The results are divided into two parts: artificial maps and real map. In each of the two contexts, an analysis is performed regarding the impact of the use of IACO in terms of average trip time and length, when compared to a shortest-time algorithm; also, an analysis regarding the impact on fuel consumption and CO_2 emissions is given.

6.1. Artificial Maps

Experiments were first conducted using artificial maps, as to observe the behavior of the algorithms in a controlled and known environment. Two traditional network topologies were used – a radial and ring network and a lattice network.

6.1.1. Radial and Ring Network

The results of the radial and ring network experiments regarding trip length and duration are summed in Table 1. The “IACO Compliant” and “ST Compliant” columns refer to the subset of vehicles using the respective algorithm, while the “IACO Free” and “ST Free” columns refer to the subset of vehicles not using the algorithm (and still using the default SUMO routing algorithm, a shortest-path routing based on the traffic conditions found in the network at the time of departure). Finally, the “Total” columns refer to all the vehicles (both the ones adhering to the algorithm and the “Free” ones), so as to provide a global perspective of the impact of the introduction of IACO or ST.

Table 1: Trip Length (m) and Duration (s) Average and Standard Deviation for Radial and Ring Network

			IACO			ST		
			Compliant	Free	Total	Compliant	Free	Total
0%	Trip Time	Average	-	3497	3497	-	3497	3497
	Traveled Length	Average	-	1860	1860	-	1860	1860
10%	Trip Time	Average	3129	3352	3330	2544	2833	2804
		Standard Dev	358	407	402	233	282	274
	Traveled Length	Average	2714	1859	1941	3995	1861	2074
		Standard Dev	56	2	5	89	3	9
25%	Trip Time	Average	2415	2646	2583	2741	2986	2798
		Standard Dev	245	261	250	888	1095	1039
	Traveled Length	Average	2675	1857	2061	3825	1859	2350
		Standard Dev	46	4	12	78	5	15
75%	Trip Time	Average	1993	2164	2045	2456	2473	2460
		Standard Dev	433	478	450	174	200	180
	Traveled Length	Average	2593	1791	2392	3265	1855	2913
		Standard Dev	52	35	48	122	14	68
100%	Trip Time	Average	1920	-	1920	2120	-	2120
		Standard Dev	156	-	156	55	-	55
	Traveled Length	Average	2835	-	2835	3805	-	3805
		Standard Dev	25	-	25	40	-	40

The results depicted in Fig. 5 are related to trip length, presenting the results obtained in scenarios using IACO and ST. The percentage of vehicles adhering to IACO or ST varies from 0% to 100%, and their performance is depicted by the "IACO Compliant" and "ST Compliant" lines, while the remaining vehicles use the standard shortest path algorithm. Their performance is depicted by the "IACO Free" and "ST Free" lines, and is influenced by the adhesion of other vehicles to IACO or ST. Finally we also present the average trip length for all vehicles (Total Vehicles).

As can be observed IACO achieves a better performance, when compared to ST. While vehicles using the ST algorithm increase their trip length from 75,5% to 114%, vehicles using IACO increase trip length by 39% to 52%. Free vehicles have similar trip lengths in all configurations.

The increased trip length is, however, compensated by smaller trip durations, as can be seen in Fig. 6. Vehicles using IACO have a trip time reduced by 11% to 46%, depending on the percentage of vehicles using the algorithm. Vehicles using ST see their trip times reduced by 22% to 40%, depending on the percentage of vehicles using the algorithm. It is interesting to note that for a small percentage of vehicles adhering to these algorithms, ST outperforms IACO. However, considering a percentage of adhering vehicles equal or above 25%, IACO achieves a better performance. It is also important to highlight the fact that free vehicles also have reduced trip times. In the case of ST, these vehicles see their trip times reduced by 15% to 30%; for IACO, their trip times are reduced by 5% to 39%.

In spite of the fact that trip length using either ST or IACO is higher than when using shortest path, this is balanced by reduced trip time values, both for vehicles using ST and IACO as for vehicles using shortest path. In both cases, IACO seems to be the better choice, since it presents much lower trip length values than ST, for similar reduced trip times, and allows for shorter trip times when the percentage of vehicles adhering to the algorithm increases.

With higher user percentages, IACO's performance in terms of trip duration seems to tend to a limit beyond which improvement appears to be impossible.

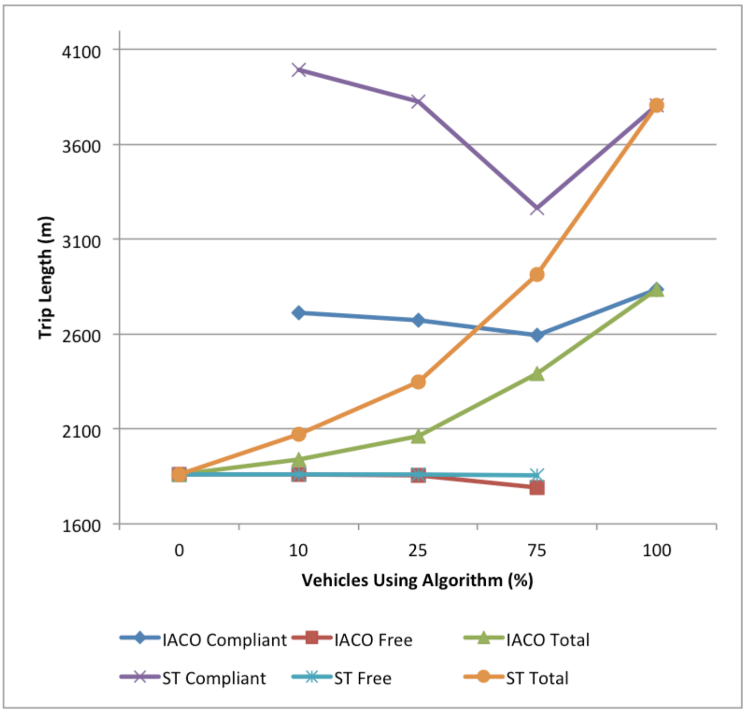


Figure 5: Effects of Percentage of Users in Trip Length by Algorithm and Type of Vehicle in a Radial and Ring Network

Regarding motor vehicle emissions (Fig. 7), we can observe that for low user percentages IACO produces a higher volume of emitted CO_2 than ST, even though both produce considerably less than when using shortest path (19% and 29% less, in the case of IACO and ST, respectively). For larger percentages of users adhering to ST or IACO, emissions become even smaller, reducing them by up to 38% in the case of ST (considering all vehicles using this algorithm) and 43% in the case of IACO (considering 75% vehicles using this algorithm).

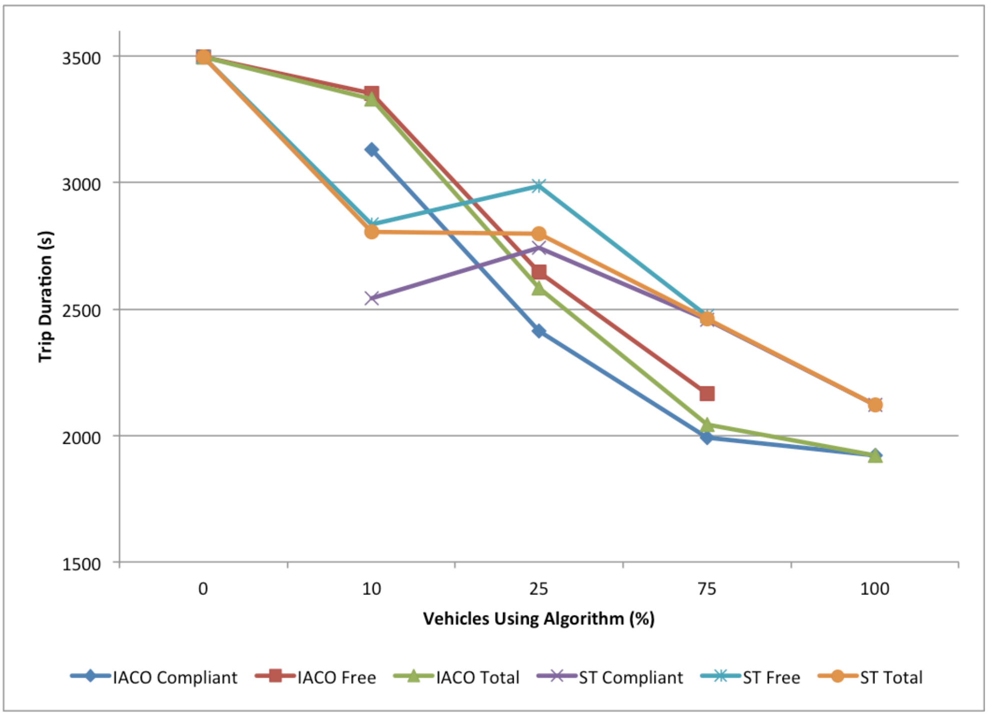


Figure 6: Effects of Percentage of Users in Trip Duration by Algorithm and Type of Vehicle in a Radial and Ring Network

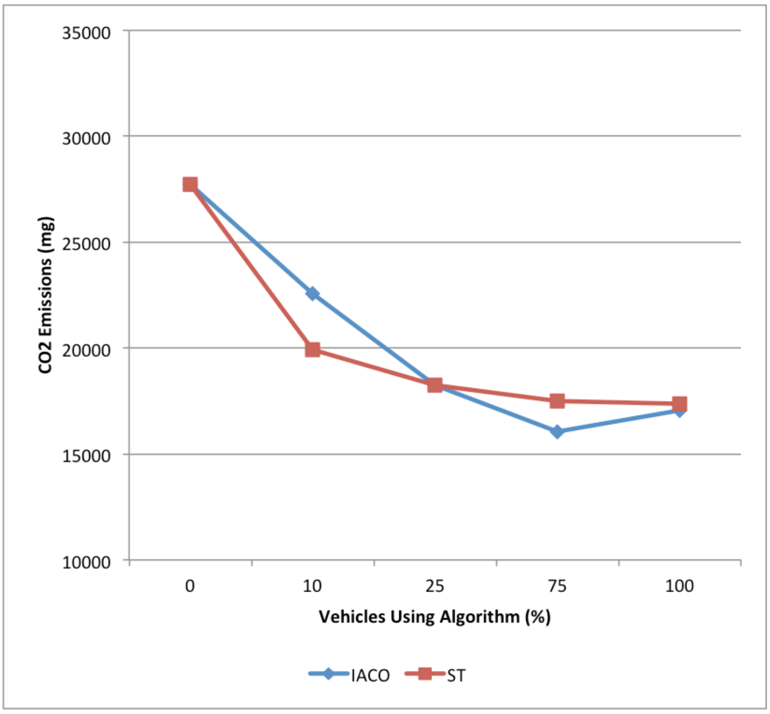


Figure 7: Effects of Percentage of Users in Average CO_2 emissions by Algorithm in a Radial and Ring Network

Similar results were obtained in terms of fuel consumption – see Fig. 8. Having the CO_2 emissions and fuel consumption results in mind, we can conclude that in this map, significant average travel time reduction can be achieved using the IACO algorithm, being an adequate alternative for this map's topology.

6.1.2. Lattice Network

The results of the lattice network experiments in terms of trip length and duration are summarized in Table 2.

Table 2: Trip Length (m) and Duration (s) Average and Standard Deviation for Lattice Network

			IACO			ST		
			Compliant	Free	Total	Compliant	Free	Total
0%	Trip Time	Average	-	3761	3761	-	3761	3761
	Traveled Length	Average	-	2325	2325	-	2325	2325
10%	Trip Time	Average	3085	3807	3735	1847	3023	2913
		Standard Dev	415	497	485	246	420	402
	Traveled Length	Average	2992	2325	2392	4810	2333	2581
		Standard Dev	77	8	7	80	3	7
25%	Trip Time	Average	2092	2573	2453	1309	1857	1720
		Standard Dev	453	623	577	123	116	112
	Traveled Length	Average	2962	2322	2482	4890	2320	2963
		Standard Dev	103	11	26	108	5	42
75%	Trip Time	Average	1121	1535	1225	1381	1754	1474
		Standard Dev	227	310	243	217	298	237
	Traveled Length	Average	2945	2271	2777	4995	2318	4326
		Standard Dev	240	358	323	159	18	82
100%	Trip Time	Average	1119	-	1119	1238	-	1238
		Standard Dev	46	-	46	34	-	34
	Traveled Length	Average	3157	-	3157	4983	-	4983
		Standard Dev	222	-	222	177	-	159

In terms of trip length, the results illustrated in Fig. 9 show that IACO achieves a better performance, when compared to ST. While vehicles using the ST algorithm increase their trip length from 106% to 114%, vehicles using IACO increase trip length by 26% to 35%. Free vehicles have similar trip lengths in all configurations.

Figure 10 shows the effect of the algorithms in trip duration. Vehicles using IACO have a trip time reduced by 18% to 70% depending on the percentage of vehicles using this algorithm. Vehicles using ST see their trip times reduced by 51% to 68%, depending on the percentage of vehicles using this algorithm. As with the radial and ring network, for small adhesion percentages, ST outperforms IACO. However, considering adhesion percentage equal or above 75%, IACO achieves a better performance. It is also important to highlight the fact that free vehicles also have reduced trip times. In the case of ST, these vehicles see their trip times reduced by 20% to 54%; for IACO, their trip times are reduced by up to 60%.

Regarding CO_2 emissions (Fig. 11) and fuel consumption (Fig. 12), the results are similar to the ones for the radial and ring network. IACO reduces emissions by 29% to 69%, depending on the percentage of vehicles using the algorithm, while ST reduces emissions by 36% to 62%

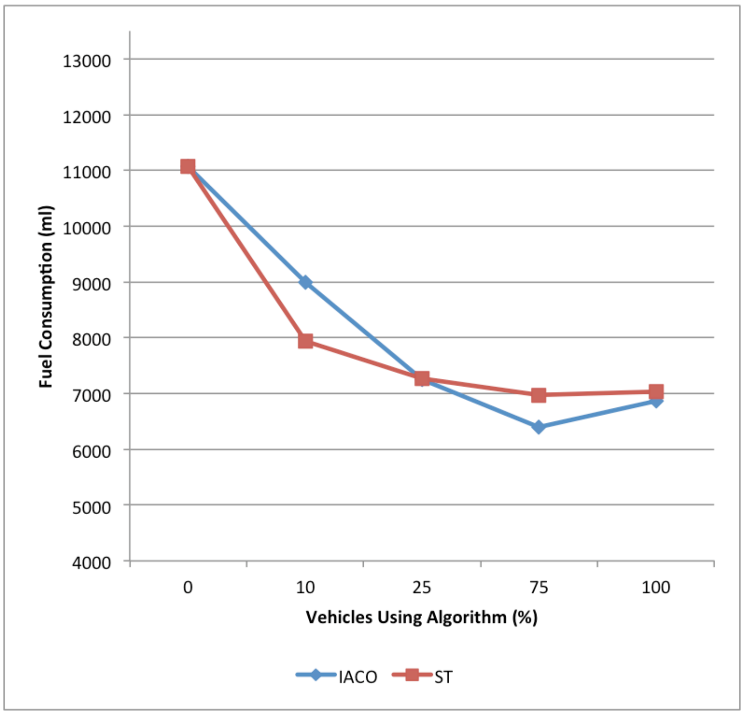


Figure 8: Effects of Percentage of Users in Average Fuel Consumption by Algorithm in a Radial and Ring Network

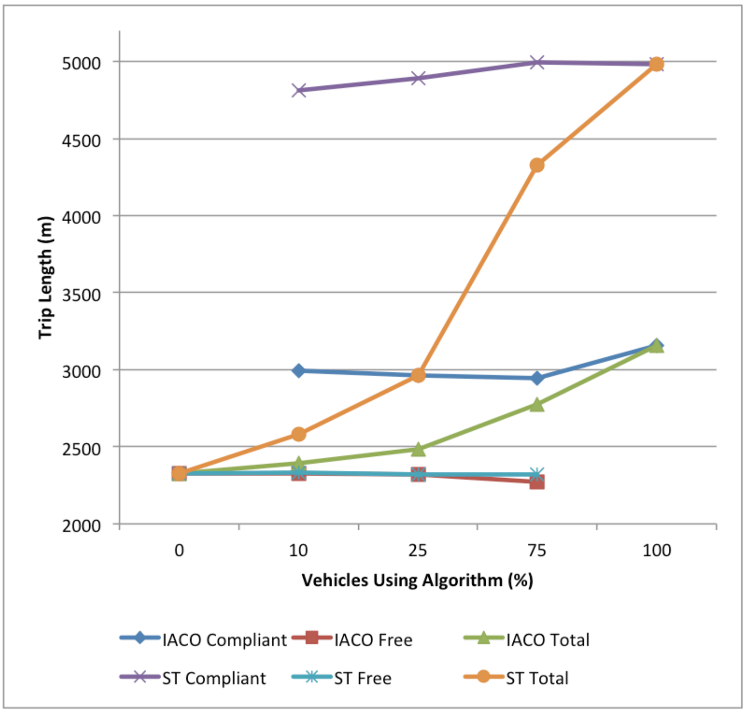


Figure 9: Effects of Percentage of Users in Trip Length by Algorithm and Type of Vehicle in a Lattice Network

6.2. *Trip Length and Time in Coimbra*

The results of the experiments in terms of trip length and duration are summarized in Table 3.

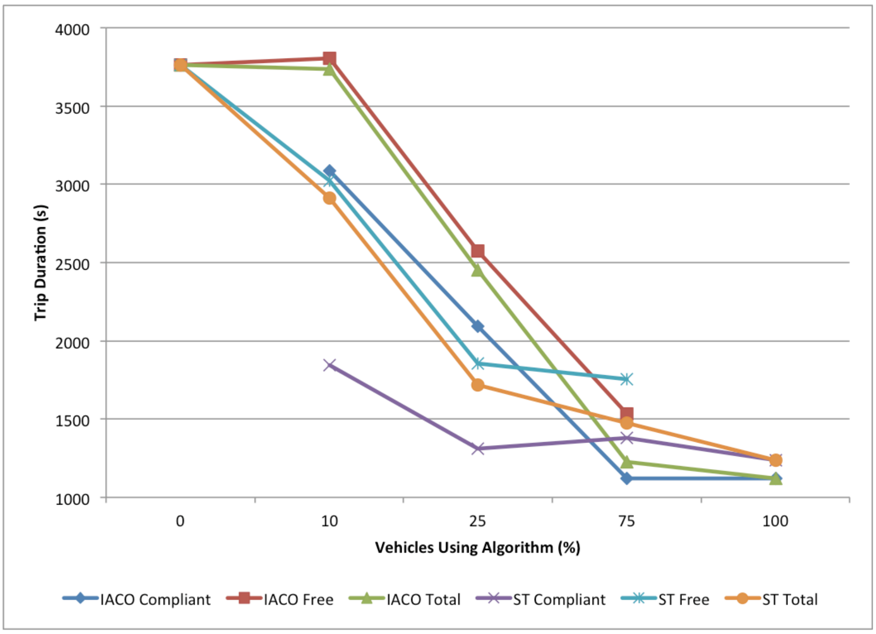


Figure 10: Effects of Percentage of Users in Trip Duration by Algorithm and Type of Vehicle in a Lattice Network

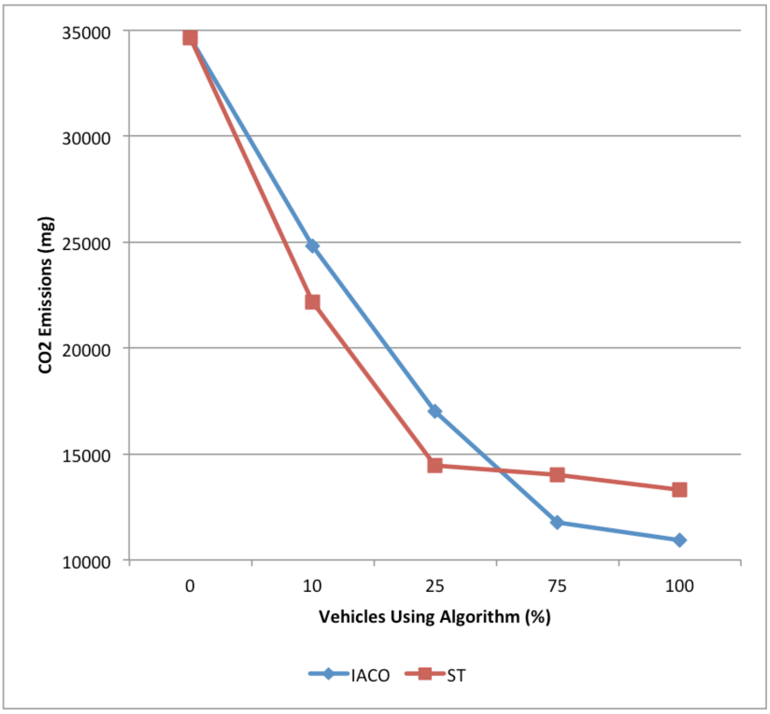


Figure 11: Effects of Percentage of Users in CO_2 emission by Algorithm in a Lattice Network

The columns show the two used algorithms (IACO and ST), and the impact in terms of different vehicle types: Compliant and Free. The rows show the impact of an increase in the percentage of vehicles using either IACO or ST to make route decisions. For each of the configurations (as mentioned before, 10%, 25%, 75% an 100% of vehicles), the average trip duration and length are shown, as well as the standard deviation for the experiments for each configuration.

Table 3: Trip Length (m) and Duration (s) Average and Standard Deviation for the Coimbra Network

			IACO			ST		
			Compliant	Free	Total	Compliant	Free	Total
0%	Trip Time	Average	-	4709	4709	-	4709	4709
	Traveled Length	Average	-	3992	3992	-	3992	3992
10%	Trip Time	Average	2554	4276	4099	2087	4283	4064
		Standard Dev	278	716	670	247	506	469
	Traveled Length	Average	6029	3990	4194	7134	3996	4309
		Standard Dev	144	7	12	210	6,8	21
25%	Trip Time	Average	1952	3156	2856	1867	3354	2982
		Standard Dev	231	384	342	115	338	279
	Traveled Length	Average	6398	3978	4583	7365	3991	4834
		Standard Dev	58	19	22	272	13	64
75%	Trip Time	Average	1005	1356	1093	1230	1979	1416
		Standard Dev	133	234	158	44	159	65
	Traveled Length	Average	6145	3983	5604	6713	3985	6031
		Standard Dev	81	38	56	197	58	145
100%	Trip Time	Average	727	-	727	1074	-	1074
		Standard Dev	34	-	34	68	-	68
	Traveled Length	Average	6066	-	6066	6568	-	6568
		Standard Dev	47	-	47	36	-	36

The results regarding trip length are depicted in Fig. 13, and show that the introduction of both ST and IACO lead to lengthier trips for the vehicles using those algorithms (free vehicles maintain the same route and therefore also maintain an almost constant trip length). IACO, however, produces shorter trips than ST, when using similar configurations – while IACO increases trip length by an average of 55%, ST increases trip length by an average of 77%, which means that ST produces trips that are, in average, 14% lengthier than those produced by IACO.

The results regarding trip duration are depicted in Fig. 14, and show that the introduction of both algorithms not only lead to shorter trip times for the vehicles using the algorithms but also for the other ones. For a small percentage of vehicles using the algorithm, ST performs better than IACO for compliant vehicles (ST decreases trip time by about 55%, while IACO reduces it by about 45%, when considering 10% compliant vehicles) and produces similar results for free vehicles (both methods result in a decrease of 9% for free vehicles); however, when the percentage of vehicles that use the algorithm increases, the use of IACO reveals itself to be advantageous for both compliant and free vehicles – for 75% compliant vehicles, ST reduces trip time of compliant vehicles by over 73%, while IACO reduces trip time by over 78%; the impact in free vehicles is also noticeable: ST decreases trip time by almost 58%, while IACO reduces trip time by over 71%. When all vehicles adhere to the algorithm being used, IACO provides a 84% reduction in trip time, while ST allows for a reduction of about 77%.

It is also noticeable that it is extremely beneficial for vehicles to comply with an algorithm (be it IACO or ST), as it drastically reduces trip time for those vehicles – even when considering an adoption ratio of 10%, IACO reduces trip times by over 45% and ST by over 55%, when comparing to the baseline values.

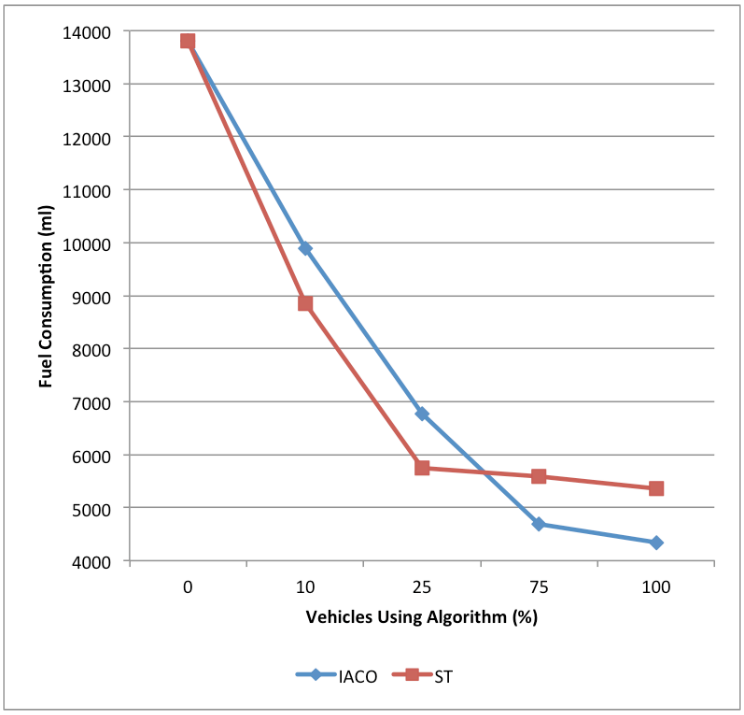


Figure 12: Effects of Percentage of Users in Fuel Consumption by Algorithm in a Lattice Network

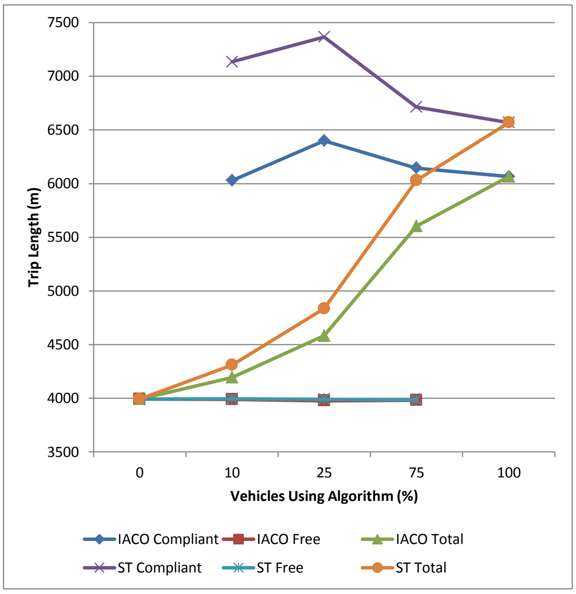


Figure 13: Effects of Percentage of Users in Trip Length by Algorithm and Type of Vehicle

6.3. Fuel Consumption and CO_2 Emissions in Coimbra

The results of the experiments in terms of CO_2 emission and fuel consumption are summarized in Table 4 below.

Table 4: Results for CO_2 emission (mg) and Fuel Consumption (ml)

			IACO	ST
0%	CO2 Emission	Average	33227	33227
	Fuel Consumption	Average	13253	13253
10%	CO2 Emission	Average	30473	30478
		Standard Dev	11092	10928
	Fuel Consumption	Average	12155	12143
		Standard Dev	1088	928
25%	CO2 Emission	Average	25252	25887
		Standard Dev	9711	9149
	Fuel Consumption	Average	10050	10382
		Standard Dev	601	490
75%	CO2 Emission	Average	18369	20550
		Standard Dev	6551	7762
	Fuel Consumption	Average	7316	8147
		Standard Dev	341	103
100%	CO2 Emission	Average	16782	19209
		Standard Dev	309	315
	Fuel Consumption	Average	6676	7796
		Standard Dev	121	114

The columns show the two used algorithms (IACO and ST), while the rows show the impact of an increase in the percentage of vehicles using either IACO or ST to make route decisions. For each of the configurations, average values for CO_2 emissions and fuel consumption are shown, as well as the standard deviation for the experiments for each configuration.

The results regarding CO_2 emissions are depicted in Fig. 15, and show that the introduction of either algorithm (IACO or ST) leads to a significative decrease of CO_2 emissions. Even for a small ratio of compliant vehicles (10%), CO_2 emissions decrease by a little over 8%. That reduction of emissions becomes even more visible with higher ratios of compliant vehicles – for a 75% compliance ratio, ST and IACO reduce emissions by about 38% and 44%, respectively. It is also noticeable that with the increase in percentage of compliant vehicles, the difference between IACO and ST also expands, with IACO presenting a performance over 12% better than ST when considering a 100% compliance ratio.

The results regarding fuel consumption are depicted in Fig. 16, and, as expected, show similar results to the ones shown above for CO_2 emissions. IACO presents a performance over 14% better than ST when considering a 100% compliance ratio, which is similar to the 12% stated above for emissions.

7. Conclusion and Future Work

In this work, the Inverted Ant Colony Optimization algorithm was introduced, inverting the logic of pheromone attraction of the standard ACO algorithm into a repulsion effect, abstracting the repulsion of drivers from traffic jams or dense traffic patterns. The effect of this behaviour, when considering drivers to be the 'ants', is a better load distribution over the traffic network, decreasing traffic density in widely used roads, and leading to improvements on the traffic network at a global level, decreasing trip time and CO_2 emissions for both vehicles adhering to the suggestions provided by the algorithm and those who still behave in the usual manner.

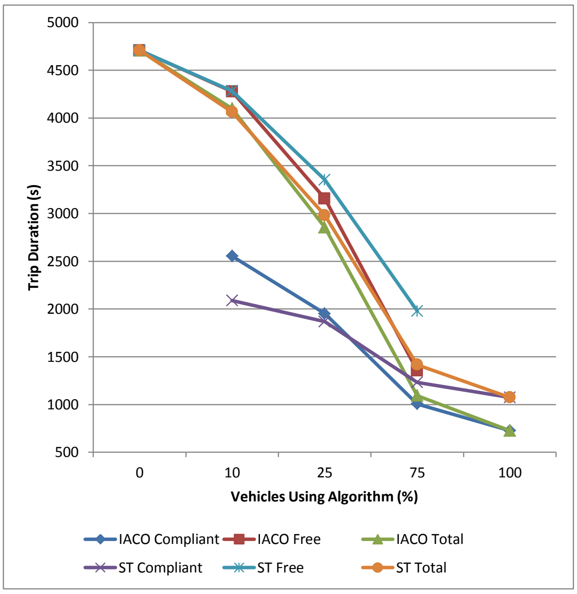


Figure 14: Effects of Percentage of Users in Trip Duration by Algorithm and Type of Vehicle

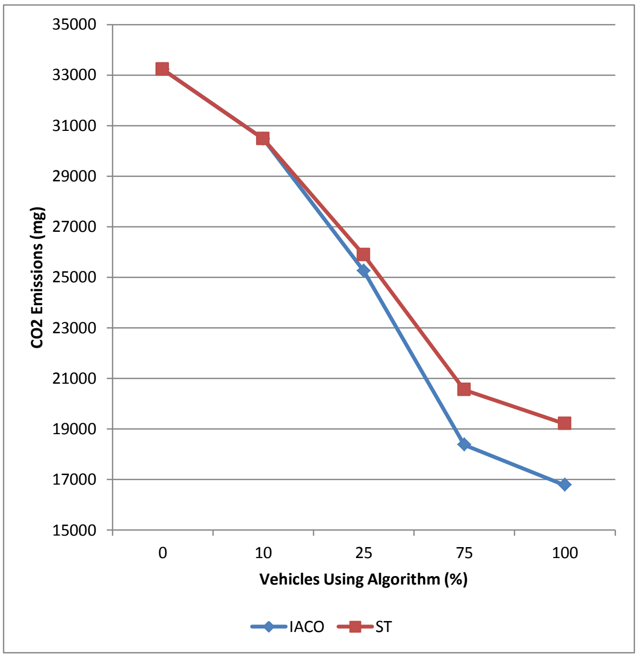


Figure 15: Effects of Percentage of Users in CO_2 emission by Algorithm

To validate the approach, two artificial scenarios (lattice and radial and ring) and a real one were used. For the real scenario, a road network model from a real city was used, along with real traffic data obtained from an origin-destination matrix. The achieved simulation results show that this kind of approach can reach the desired goals, reducing trip durations as well as CO_2 emissions. The use of IACO allows for a reduction of trip times by 45% to 84%, considering increasing percentages of vehicles using this algorithm to make decisions, when compared to a decrease in trip times of 55% to 77% for ST. These are also considered to be good results, when compared to the ones

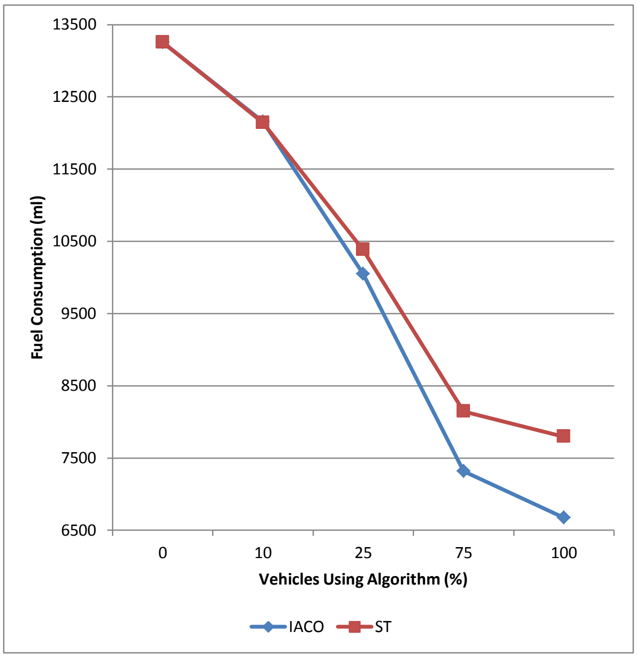


Figure 16: Effects of Percentage of Users in Fuel Consumption by Algorithm

obtained by other research works (see section 2).

Nowadays, there is a great concern regarding the ecological impact of city traffic. In any city, politicians take several measures to decrease levels of CO_2 and other polluting gas emissons, increasing the quality of the air. In this project, this issue is correlated to the avoidance of traffic congestions, and in this particular the use of IACO allows for a decrease of 8% to 49% in CO_2 emissions.

In terms of future work, there are several areas that will be explored. At the algorithmic level, there are some tweaks that can be done to the IACO algorithm, exploring different values for the C parameter present in Equation 4, as to find the best value for each city road network model, as well as different pheromone deposit and evaporation methods. Also, the adoption of this algorithm to a different context can constitute another future direction. At the experimental level, it would also be useful to compare the baseline data obtained from simulation with actual data from the real world. Another direction is the use of the algorithm with other city road networks, with different complexity and traffic levels, as to understand further strengths and weaknesses of this approach.

Acknowledgment

This work was partially funded by the Portuguese Foundation for Sciente and Technology (FCT) under project COSMO, with reference PTDC/EIA-EIA/108785/2008.

References

[1] P. Gupta, G. Purohit, A. Dadhich, Approaches for Intelligent Traffic System: A Survey, International Journal on Computer Science and Engineering 4 (9) (2012) 1570–1578.
[2] L. D. Baskar, B. D. Schutter, J. Hellendoorn, Z. Papp, Traffic Control and Intelligent Vehicle Highway Systems: A Survey, IET Intelligent Transport Systems 5 (1) (2011) 38–52.
[3] A. Kersys, Sustainable Urban Transport System Development Reducing Traffic Congestions Costs, Engineering Economics 22 (1) (2011) 5–13.
[4] F. Grimaldo, M. Lozano, F. Barber, A. Guerra-Hernández, Towards a Model for Urban Mobility Social Simulation, Progress in Artificial Intelligence 1 (2) (2012) 149–156.
[5] M. Hatzopoulou, J. Y. Hao, E. J. Miller, Simulating the Impacts of Household Travel on Greenhouse Gas Emissions, Urban Air Quality, and Population Exposure, Transportation 38 (2011) 871–887.

[6] M. Dorigo, Optimization, Learning and Natural Algorithms, Ph.D. thesis, Dipartimento di Elettronica, Politecnico di Milano (1992).
[7] M. Dorigo, T. Stützle, Ant Colony Optimization, MIT Press, 2004.
[8] M. Dorigo, T. Stützle, Handbook of Metaheuristics, Kluwer Academic, 2002, Ch. The Ant Colony Optimization Metaheuristic: Algorithms, Applications and Advances, pp. 251–285.
[9] B. K. Nanda, G. Das, Ant Colony Optimization: A Computational Intelligence Technique, International Journal of Computer & Commmunication Technology 2 (6) (2011) 105–110.
[10] M. Dorigo, G. D. Caro, Ant Colony Optimization: A New Metaheuristic, in: Proceedings of the 1999 Congress on Evolutionary Computation, 1999, p. 8 pages.
[11] G. D. Caro, M. Dorigo, Two Ant Colony Algorithms for Best-Effort Routing in Datagram Networks, in: Proceedings of the 10th IASTED International Conference on Prallel and Distributed Computing and Systems (PDCS'98), 1998, pp. 541–546.
[12] R. Claes, T. Holvoet, Ant Colony Optimization Applied to Route Planning using Link Travel Time Prediction, in: Proceedings of the IEEE International Symposium on Parallel Distributed Processing Workshops, 2011, pp. 358–365.
[13] L. M. Gambardella, É. Taillard, G. Agazzi, New Ideas in Optimization, McGraw-Hill, 1999, Ch. MACS-VRPTW: A Multiple Colony System for Vehicle Routing Problems with Time Windows, pp. 63–76.
[14] J. He, Z. Hou, Ant Colony Algorithm for Traffic Signal Timing Optimization, Advances in Engineering Software 43 (1) (2012) 14–18.
[15] M. Abdoos, N. Mozayani, A. L. Bazzan, Holonic Multi-Agent System for Traffic Signals Control, Engineering Applications of Artificial Intelligence 26 (5–6) (2013) 1575–1587.
[16] A. M. Turky, M. S. Ahmad, M. Z. M. Yusoff, The use of Genetic Algorithm for Traffic Light and Pedestrian Crossing Control, International Journal of Computer Science and Network Security 9 (2) (2009) 88–96.
[17] R. J. Rossetti, R. Liu, An Agent-Based Approach to Assess Drivers Interaction with Pre-Trip Information Systems, Journal of Intelligent Transportation Systems: Technology, Planning, and Operations 9 (1) (2005) 1–10.
[18] B. Asadi, A. Vahidi, Predictive Cruise Control: Utilizing Upcoming Traffic Signal Information for Improving Fuel Economy and Reducing Trip Time, IEEE Transactions on Control Systems Technology 19 (3) (2011) 707–714.
[19] B. Chen, H. H. Cheng, A Review of the Applications of Agent Technology in Traffic and Transportation Systems, IEEE Transactions on Intelligent Transportation Systems 11 (2) (2010) 485–497.
[20] A. Horni, A Brief Overview of the Multi-Agent Transport Simulation MATSim, in: MATSim Tutorial, May 17–20, 2011, Shanghai, China, Tongji University, 2011.
[21] M. Behrisch, L. Bieker, J. Erdmann, D. Krajzewicz, SUMO – Simulation of Urban MObility: An Overview, in: Proceedings of the Third International Conference on Advances in System Simulation (SIMUL 2011), Barcelona, Spain, 2011, pp. 63–68.
[22] M. Fellendorf, P. Vortisch, Fundamentals of Traffic Simulation, Vol. 145 of International Series in Operations Research & Management Science, Springer, 2010, Ch. Microscopic Traffic Flow Simulator VISSIM, pp. 63–93.
[23] J. D. Leonard II, S. L. Shealey, Transims and Developments of Regional Impact, Final report, School of Civil and Environmental Engineering, Georgia Institute of Technology (September 2010).
[24] M. Maciejewski, A Comparison of Microscopic Traffic Flow Simulation Systems for an Urban Area, Transport Problems 5 (4) (2010) 27–38.
[25] L. Bloomberg, J. Dale, A Comparison of the VISSIM and CORSIM Traffic Simulation Models, in: Institute of Transportation Engineers Annual Meeting, August 6-9, 2000, Nashville, Tennessee, USA, 2000.
[26] W. Gao, M. Balmer, E. J. Miller, Comparisons Between MATSim and EMME/2 on the Greater Toronto and Hamilton Area Network, Transportation Research Record: Journal of the Transportation Research Board 2197 (2010) 118–128.
[27] Á. Seco, N. Pinto, Coimbra's origin-destination matrix execution report, Tech. rep., Organization of the Trasportation System of Coimbra (2003).