An Inverted Ant Colony Optimization approach to traffic

José Capela Dias, Penousal Machado *, Daniel Castro Silva, Pedro Henriques Abreu

Department of Informatics Engineering, University of Coimbra/CISUC – Center for Informatics and Systems of the University of Coimbra, Portugal

A R T I C L E   I N F O

Article history:
Received 25 July 2013
Received in revised form 5 July 2014
Accepted 9 July 2014

Keywords:
Inverted Ant Colony Optimization
Traffic simulation

A B S T R A C T

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 do not. 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 CO2 emissions from both vehicle types decrease significantly when using IACO (up to 49%).

© 2014 Elsevier Ltd. All rights reserved.

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, (Gupta et al., 2012). In particular, traffic management system, which includes many sub-problems (Baskar et al., 2011), is one of the most active topics nowadays. It has severe ecological implications and is considered to be vital to the sustainability of cities (Kersys, 2011). These concerns may become decisive in the approach to some classical problems such as finding the optimal route to a destination. Traditional driver behavior 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 CO2 and other polluting chemicals (Grimaldo et al., 2012; Hatzopoulou et al., 2011).

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

* Corresponding author at: Universidade de Coimbra, Departamento de Engenharia Informática, Polo II, Pinhal de Marrocos 3030-290 Coimbra, Portugal.
E-mail addresses: jacdias@student.dei.uc.pt (J.C. Dias), machado@dei.uc.pt (P. Machado), dcs@dei.uc.pt (D.C. Silva), pha@dei.uc.pt (P.H. Abreu).
http://dx.doi.org/10.1016/j.engappai.2014.07.005
0952-1976/© 2014 Elsevier Ltd. All rights reserved.
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 CO2 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.

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
            \( \text{Pheromone}_{\text{currentNode,nextNode}} + = \text{Update} \)
        end for
        until every ant has a solution
        for all edge \( e \) in \( B \) do
            \( \text{Pheromone}_e + = \text{Deposit} \)
        end for
        for all edge \( e \) in \( E \) do
            \( \text{Pheromone}_e - = \text{Evaporation} \)
        end for
    end while

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 Dorigo and Caro (1999). Another application is in the communications area, where the AntNet variation of the algorithm allows for adaptive routing in communication networks (Caro and Dorigo, 1999). Several approaches using ACO have also been made in the traffic area. Claes and Holvoet (2011) 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 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 et al. (1999) apply ACO to the vehicle routing problem with time windows, using two ant colonies: one that minimizes the number of vehicles, and the other 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 Gambardella et al. (1999) assume the 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 (2012) 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 the total time delay by approximately 15.6% and the number of stops by 60%. This work differs from the one presented herein as it assumes that no changes are made to the route of the vehicles but only to traffic flow on intersections.

Another 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 (Abdoos et al., 2013), genetic algorithms (Turky et al., 2009), BDI (Rossetti and Liu, 2005) or predictive models (Asadi and Vahidi, 2011). Just as with the previously mentioned approaches, these also propose changes to the network (by modifying the behavior of traffic signals), while our approach does not 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, \(\text{CO}_2\) emissions and avoidance of traffic congestions in the entire network using this new variation of ACO.

3. Inverted Ant Colony Optimization

Having the goal of this research work in mind, which is to develop an approach that avoids congestion in a city without changing the drivers individual behaviors, a hybrid algorithm is developed.

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 criterion 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

\[
\text{edgeCost}(i,j) = \text{distance}(i,j) + \text{edgePheromones}(i,j)
\]  

(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 reach their destination, but considering a network without traffic.

**Algorithm 2. Dijkstra algorithm.**

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

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 – \(O((E + V)\log V)\mathcal{w}N\), where \(E\) and \(V\) are the number of edges and vertexes in the network respectively. 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 and 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 similar 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 behavior 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:

\[
\text{minTT}_{edge} = \frac{\text{edgeLength}}{\text{edgeMaximumAllowedSpeed}}
\]  

(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:

\[
\text{pher}_{edge} = \begin{cases} 
\text{pher}_{edge} + \text{DEP}_{\text{pher}} \\
\text{pher}_{edge} - \text{minTT}_{edge} \times \text{DEP}_{\text{pher}}
\end{cases}
\]  

(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. Eq. (4) illustrates this concept (\(C\) controls the impact factor of recent past variations):

\[
\begin{align*}
\text{edgeCongestion}(i,j) &= \text{distance}(i,j) + \text{edgeCongestion}(i,j) \\
\text{edgeCongestion}(i,j) &= \text{edgePheromones}(i,j) \\
&+ C \times \text{edgePheromoneVariation}(i,j)
\end{align*}
\]  

(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 behavior. 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 decentralized 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 (2010) 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)** (Horni, 2011), a toolbox for the implementation of large-scale agent-based transport simulations, composed of several individual modules that can be combined or used stand-alone. It allows demand-modeling, traffic flow simulation and running simulations iteratively.
- **Simulation of Urban MObility (SUMO)** (Behrisch et al., 2011), 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. Fellendorf and Vortisch (2010) or Leonard and Shealey (2010). Some comparison studies have been performed between two or more of these simulation platforms (Maciejewski, 2010; Bloomberg and Dale, 2000; Gao et al., 2010). 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).

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, 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 CO₂ 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.

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: 5000 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 5000 vehicle configuration did not 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).

Experiments were repeated 40 times for each of the eight different configurations 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 CO₂ emissions.

### Table 1

<table>
<thead>
<tr>
<th></th>
<th>IACO</th>
<th>ST</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Compliant</td>
<td>Free</td>
</tr>
<tr>
<td>0% Trip Time</td>
<td>–</td>
<td>3497</td>
</tr>
<tr>
<td>Traveled length</td>
<td>–</td>
<td>1860</td>
</tr>
<tr>
<td>10% Trip time</td>
<td>3129</td>
<td>3352</td>
</tr>
<tr>
<td>Average</td>
<td>358</td>
<td>407</td>
</tr>
<tr>
<td>Standard dev</td>
<td>56</td>
<td>2</td>
</tr>
<tr>
<td>25% Trip time</td>
<td>2415</td>
<td>2646</td>
</tr>
<tr>
<td>Average</td>
<td>245</td>
<td>261</td>
</tr>
<tr>
<td>Standard dev</td>
<td>2675</td>
<td>1857</td>
</tr>
<tr>
<td>25% Standard dev</td>
<td>46</td>
<td>4</td>
</tr>
<tr>
<td>75% Trip time</td>
<td>1993</td>
<td>2164</td>
</tr>
<tr>
<td>Average</td>
<td>433</td>
<td>478</td>
</tr>
<tr>
<td>Standard dev</td>
<td>2593</td>
<td>1791</td>
</tr>
<tr>
<td>75% Standard dev</td>
<td>52</td>
<td>35</td>
</tr>
<tr>
<td>100% Trip time</td>
<td>1920</td>
<td>–</td>
</tr>
<tr>
<td>Average</td>
<td>156</td>
<td>–</td>
</tr>
<tr>
<td>Standard dev</td>
<td>2835</td>
<td>–</td>
</tr>
<tr>
<td>100% Standard dev</td>
<td>25</td>
<td>–</td>
</tr>
</tbody>
</table>

Fig. 5. Effects of percentage of users in trip length by algorithm and type of vehicle in a radial and ring network.

Fig. 6. Effects of percentage of users in trip duration by algorithm and type of vehicle in a radial and ring network.

### 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, 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 (Seco and Pinto, 2003). 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 60,000 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, etc.) and vehicle class. From this OD matrix, 10,000 trips were selected as to provide a sample to be used for an one-hour simulation period.

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 CO₂ 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.

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 39–40%, 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 to 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 15–30%; for IACO, their trip times are reduced by 5–39%.

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

![Fig. 7. Effects of percentage of users in average CO₂ emissions by algorithm in a radial and ring network.](image)

![Fig. 8. Effects of percentage of users in average fuel consumption by algorithm in a radial and ring network.](image)
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.

Regarding motor vehicle emissions (Fig. 7), we can observe that for low user percentages IACO produces a higher volume of emitted CO₂ 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).

### 6.1.2. Lattice network

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

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 26–35%.

Free vehicles have similar trip lengths in all configurations.

Fig. 10 shows the effect of the algorithms in trip duration. Vehicles using IACO have a trip time reduced by 18–70% depending on the percentage of vehicles using this algorithm. Vehicles using ST see their trip times reduced by 51–68%, depending on the percentage of vehicles using this algorithm. As with the radial and ring

### Table 2

Trip length (m) and duration (s) average and standard deviation for lattice network.

<table>
<thead>
<tr>
<th></th>
<th>IACO</th>
<th>ST</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Compliant</td>
<td>Free</td>
</tr>
<tr>
<td>Trip time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>–</td>
<td>3761</td>
</tr>
<tr>
<td>Standard dev</td>
<td>–</td>
<td>2325</td>
</tr>
<tr>
<td>10% Trip time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>3085</td>
<td>3807</td>
</tr>
<tr>
<td>Standard dev</td>
<td>415</td>
<td>497</td>
</tr>
<tr>
<td>10% Traveled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>2992</td>
<td>2325</td>
</tr>
<tr>
<td>Standard dev</td>
<td>77</td>
<td>8</td>
</tr>
<tr>
<td>10% Trip time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>2092</td>
<td>2573</td>
</tr>
<tr>
<td>Standard dev</td>
<td>453</td>
<td>623</td>
</tr>
<tr>
<td>10% Traveled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>2962</td>
<td>2322</td>
</tr>
<tr>
<td>Standard dev</td>
<td>103</td>
<td>11</td>
</tr>
<tr>
<td>75% Trip time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>1121</td>
<td>1535</td>
</tr>
<tr>
<td>Standard dev</td>
<td>227</td>
<td>310</td>
</tr>
<tr>
<td>75% Traveled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>2945</td>
<td>2271</td>
</tr>
<tr>
<td>Standard dev</td>
<td>240</td>
<td>358</td>
</tr>
<tr>
<td>100% Trip time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>1119</td>
<td>–</td>
</tr>
<tr>
<td>Standard dev</td>
<td>46</td>
<td>–</td>
</tr>
<tr>
<td>100% Traveled</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>3157</td>
<td>–</td>
</tr>
<tr>
<td>Standard dev</td>
<td>222</td>
<td>–</td>
</tr>
</tbody>
</table>

![Fig. 9. Effects of percentage of users in trip length by algorithm and type of vehicle in a lattice network.](image1)

![Fig. 10. Effects of percentage of users in trip duration by algorithm and type of vehicle in a lattice network.](image2)

Similar results were obtained in terms of fuel consumption – see Fig. 8. Having the CO₂ 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.
network, for small adhesion percentages, ST outperforms IACO. However, considering adhesion percentage equal to 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 20–54%; for IACO, their trip times are reduced by up to 60%.

Regarding CO₂ 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 29–69%, depending on the percentage of vehicles using the algorithm, while ST reduces emissions by 36–62%.

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.

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.

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, on 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 on 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 compared to the baseline values.

6.3. Fuel consumption and CO₂ emissions in Coimbra

The results of the experiments in terms of CO₂ emission and fuel consumption are summarized in Table 4.

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 CO₂ emissions and fuel consumption are shown, as well as the standard deviation for the experiments for each configuration.

The results regarding CO₂ emissions are depicted in Fig. 15, and show that the introduction of either algorithm (IACO or ST) leads to a significant decrease of CO₂ emissions. Even for a small ratio of compliant vehicles (10%), CO₂ 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 CO₂ 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 behavior, 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 CO₂ emissions for both vehicles adhering to the suggestions provided by the algorithm and those who still behave in the usual manner.

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 CO₂ emissions. The use of IACO allows for a reduction of trip times by 45–84%, considering increasing percentages of vehicles using this algorithm to make decisions, when compared to a decrease in trip times of 55–77% for ST. These are also considered to be good results, when compared to the ones 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 emissions, 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 8–49% 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 Eq. (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 Science and Technology (FCT) under Project COSMO, with reference PTDC/EIA-EIA/108785/2008.

**References**


