Computing New Optimized Routes for GPS Navigators Using Evolutionary Algorithms

Daniel H. Stolfi
ETSI Informática - University of Malaga
Bv. Louis Pasteur, n.35 - Campus de Teatinos
Malaga, Spain 29071
dhstolfi@lcc.uma.es

Enrique Alba
ETSI Informática - University of Malaga
Bv. Louis Pasteur, n.35 - Campus de Teatinos
Malaga, Spain 29071
eat@lcc.uma.es

ABSTRACT

GPS navigators are now present in most vehicles and smartphones. The usual goal of these navigators is to take the user in less time or distance to a destination. However, the global use of navigators in a given city could lead to traffic jams as they have a highly biased preference for some streets. From a general point of view, spreading the traffic throughout the city could be a way of preventing jams and making a better use of public resources. We propose a way of calculating alternative routes to be assigned by these devices in order to foster a better use of the streets. Our experimentation involves maps from OpenStreetMap, real road traffic, and the microsimulator SUMO. We contribute to reducing travel times, greenhouse gas emissions, and fuel consumption. To analyze the sociological aspect of any innovation, we analyze the penetration (acceptance) rate which shows that our proposal is competitive even when just 10% of the drivers are using it.

CCS CONCEPTS

•Mathematics of computing → Combinatorial optimization; •Theory of computation → Evolutionary algorithms; •Applied computing → Transportation;

KEYWORDS

Application; evolutionary algorithm; road traffic; smart city; real world; smart mobility

1 INTRODUCTION

Currently, every big city has mobility issues because citizens need to travel to work, go to school, visit hospitals, shop, or make short trips to meet each other [13]. In many cases, citizens live in low-density areas where public transport is not an option or they would much rather travel in their own cars. This is happening right now, not only in the European countries [14] but also throughout the developing world [3].

Meanwhile, the infrastructure is not advancing at the same pace to support the mobility demands of users, which in turn, produces traffic congestion, longer travel times and an increment in greenhouse gas emissions and fuel consumption. This environmental impact of road traffic affects public health [6] and economy, increases medical costs, and reduces productivity through working days lost.

Global Positioning System (GPS) navigators are a part of most vehicles nowadays, as they are needed when driving through an unknown city or neighborhood. Although some of them use data representing the current state of the road traffic to calculate the route shown to the driver, this kind of service is neither updated in real time nor available everywhere in the world. As a result, routes end up being calculated by Dijkstra or A* algorithms which only use the length of the streets and their average speed to find the best way to reach a destination (shortest path).

In this article we present an alternative way of calculating routes based on the concept of dynamic user equilibrium. The alternative routes can be provided (and updated) as a complement to the cartography so they can be used by GPS navigators to improve traffic flows when assigning routes to vehicles driving through a city.

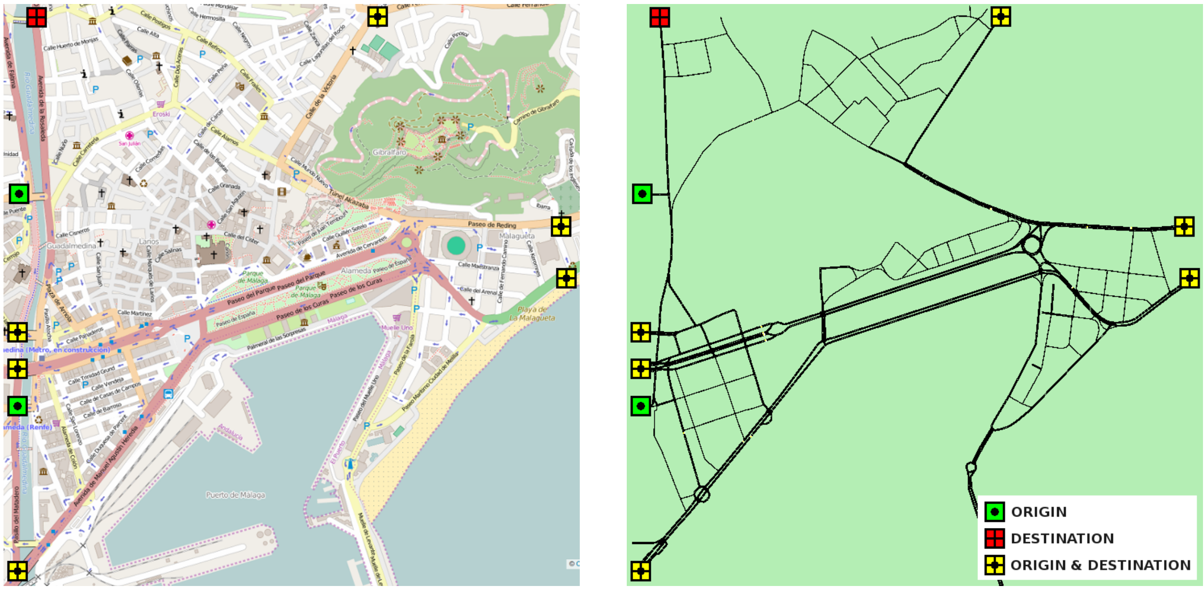
The benefits of doing this are many: a better use of the available streets, reduction of traffic jams, gas emissions and fuel consumption, and an improvement in the quality of life of citizens, etc.

The rest of this paper is organized as follows. In the next section, we review the state of the art related to our work. In Section 3 we present the characteristics of the geographical area we have analyzed. Our proposal is discussed in Section 4. Section 5 focuses on the studies done and the discussion of the results. And finally, in Section 6, conclusions and future work are given.

2 RELATED WORKS

In [1] the authors propose the ISATOPSIS (Improved Simulated Annealing Technique for Order Preference by Similarity to the Ideal Solution) algorithm to address the traffic congestion problem in smart cities. It comprises a multi-objective optimization algorithm, which combines simulated annealing (SA) with cost function based on both, multi-attribute decision making (MADM) and TOPSIS [7] to provide the driver with optimal paths. They use real-time data using V2V and V2I communications to reroute the vehicles and reduce the congestion on the roads. The experimentation presented uses maps of Sheffield and Birmingham, U.K., imported into SUMO [8] from OpenStreetMap [10]. The authors compare their results with Dijkstra and SATOPSIS, obtaining reductions in travel times, fuel consumption, and CO_2 emissions. In our article we use an evolutionary algorithm to optimize our case study, focusing on the reduction of travel times and greenhouse gas emissions by suggesting alternative routes to vehicles driving through the area under analysis. Additionally, we do not base our proposal on communications, which keeps it simple and robust.

In [4] the authors introduce a vehicle-to-vehicle congestion avoidance mechanism to minimize travel times by detecting congestion levels and rerouting vehicles in real time, based on VANETs. They create a distributed congestion avoidance scheme and consider a reactive mechanism instead of periodic broadcasts. Additionally, a dynamic route planning technique helps cars to avoid jams by choosing the route with the minimum travel time which is



(a) OpenStreetMap (b) SUMO

Figure 1: City center of Malaga. The original map (a) and how it looks after importing it into SUMO (b). Note that most of the missing streets correspond to pedestrian ways.

calculated using the congestion information available, previously collected by each car. The results show improvements in travel times when using their proposal, compared to the shortest-path routes generated by SUMO using Dijkstra. Our work is based on a different approach as we do not use vehicular networks but rather previously calculated routes and their probability of being chosen to improve the road traffic in the city.

An architecture, called Red Swarm, with the aim of reducing travel times is presented in [11]. There, the authors use a number of spots distributed around the city so that when vehicles connect to them via Wi-Fi, they receive an update of their routes toward their destination. The final route, based on probability values calculated by an evolutionary algorithm, is built by visiting just some of these spots. As a result, different routes are used, avoiding traffic jams and reducing travel times. In our present work, we do not need to build any extra infrastructure as the routes have been previously calculated and proposed by the existing GPS device in the vehicle.

In a final, relevant article a carbon-footprint/fuel-consumption-aware variable-speed limit (FC-VSL) traffic control scheme is presented in [9]. The authors minimize fuel consumption for a single vehicle under certain traffic conditions, and obtain the optimal vehicular trajectory. To do so, they designed the FC-VSL scheme based on the optimal trajectory and limiting speed, applied it to all vehicles on the road, and evaluated its performance with a detailed simulation. Their results show that the FC-VSL can reduce average fuel consumption and outperforms another VSL scheme which was designed for smoothing vehicular traffic flow. Our proposal differs from this one, especially in the strategy used to achieve a reduction of fuel consumption. We focus on the best route for each vehicle in order to prevent traffic jams and always respecting the existing maximum speed which is limited only by the restrictions of the road itself.

3 CASE STUDY

We have chosen as our case study an area of the city center of Malaga (Spain), well-known for suffering from traffic jams. The geographical area studied is delimited to the north by San Bartolomé Street and Ferrándiz Street, to the west by the Guadalmedina River, to the east by Keromnes Street, and to the south by the Mediterranean Sea, which encompasses an area of about 3 km^2.

We have imported the chosen area (shown in Figure 1) into the SUMO traffic microsimulator [8] from OpenStreetMap [10]. This allows us to work with a real scenario, e.g. streets, traffic lights, left turns, and roundabouts.

We have calculated the traffic flows for our case study by using the method presented in [12] based on the Flow Generator Algorithm (FGA). The algorithm assigns vehicles to the traffic flows generated by the program DUARUTER included in the SUMO software package. This assignation adjusts the number of vehicles in the simulation to the values measured by real sensors in the city.

Using the data published by the local council of Malaga consisting of 12 sensors, we have obtained three different scenarios corresponding to average traffic per hour during working days ($malaga_{WD}$), Saturdays ($malaga_{SAT}$), and Sundays ($malaga_{SUN}$).

The real number of vehicles in the city (Real), the value measured at each sensor when simulating the generated scenarios (FGA), and the difference percentage (Diff) are presented in Table 1.

Table 1: Real number of vehicles and the values measured at each sensor during the simulation for the three scenarios generated by the FGA.

Sensor	$malaga_{WD}$			$malaga_{SAT}$			$malaga_{SUN}$		
	Real	FGA	Diff.	Real	FGA	Diff.	Real	FGA	Diff.
5	1071	1069	-0.2%	900	900	0.0%	803	803	0.0%
6	347	348	0.3%	273	276	1.1%	227	229	0.9%
7	279	278	-0.4%	246	246	0.0%	208	208	0.0%
8	254	251	-1.2%	239	240	0.4%	212	212	0.0%
9	256	256	0.0%	248	248	0.0%	222	221	-0.5%
10	644	646	0.3%	641	640	-0.2%	584	584	0.0%
13	229	230	0.4%	214	214	0.0%	172	171	-0.6%
14	479	479	0.0%	444	443	-0.2%	348	352	1.1%
15	631	633	0.3%	566	567	0.2%	467	469	0.4%
16	518	518	0.0%	420	422	0.5%	359	358	-0.3%
17	839	854	1.8%	684	683	-0.1%	617	619	0.3%
18	600	602	0.3%	466	469	0.6%	437	440	0.7%
Avg:	512.3	513.7	**0.4%**	445.1	445.7	**0.3%**	388.0	388.8	**0.4%**

4 DYNAMIC USER EQUILIBRIUM (DUE)

The traffic assignment problem consists of assigning routes to vehicles which are moving from their origin to their destination, usually taking into account variables such as cost and benefits. It can be solved by calculating the user equilibrium route choice in which routes are assigned to vehicles so that an alternative assignation would have worsened travel times.

According to the Wardrop's first principle [15], the user equilibrium is the state in which every driver chooses a route for which the travel time is minimal. Consequently, the resulting network state is in equilibrium, since nobody can improve his travel time by choosing a different route.

We have used an approach to the assignment model which is based on an iterated simulation [5] to calculate the dynamic user equilibrium (DUE) by using tools provided by SUMO. This model uses a probability distribution for the route choice so that a route is stochastically picked for each vehicle traveling from its original location to its destination.

Our proposal consists in calculating the dynamic user equilibrium and using the new routes generated to help a GPS navigator in rerouting drivers through different streets to reach their destination, instead of using the shortest path. Concretely, we divide the city into *ad hoc* zones and use the resulting input and output streets as origin and destination of the routes throughout the zone. Then, when a vehicle enters the zone with the intention of driving through it, it will follow one of the available routes according to a previously calculated probability. Note that local trips (i.e. those whose starting and destination points are within this area) are not considered by our proposal as our intention is to favor the traffic flows that are crossing the area (and the city).

In Figure 2 we provide an example of the route assignment process when a driver intends to cross a defined zone.

The best route in terms of distance is obviously route A. Alternatively, there are two other routes, B and C, which despite being longer, may lead to a reduction of travel times for everyone, as possible congestions can be avoided by using them.

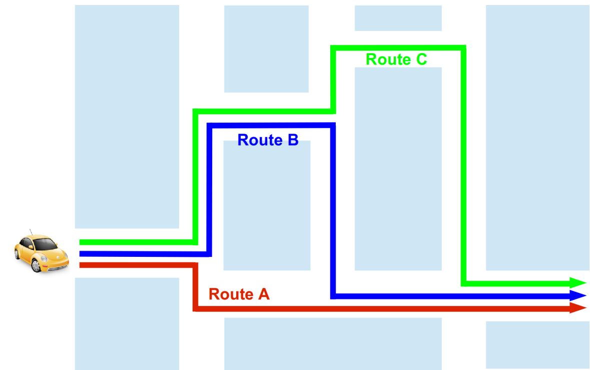


Figure 2: Possible routing example.

Our work involves not only calculating these routes but also testing three new strategies to obtain the probabilities of using them to drive through the analyzed zone, and prevent traffic jams. Of course, the probabilities not only depend on the streets' distributions (which is the reason we are using OpenStreetMap as the source for the maps), but also on the number and behavior of the vehicles involved (we use microsimulations and traffic data from the local council).

The traffic simulator SUMO (Simulation of Urban MObility) [8] includes several complementary programs. There is a tool among them, written in Python, called *DuaIterate*, which is used to calculate the dynamic user equilibrium as described in [5].

Using *DuaIterate* we have calculated the DUE for our case study and extracted the different routes, whose origins and destinations are the input streets and exits, respectively of the area under analysis, as presented in Algorithm 1.

Algorithm 1 DUE Routes

```
procedure DUERoutes
    trips ← SUMO(malaga)                                   ▷ OD Matrix
    P_d ← initializeProbabilities()                        ▷ begin DuaIterate
    while not TerminationCondition() do
        travelTimes ← SUMO(malaga, trips, P_d)
        P_d ← updateProbabilities(travelTimes)
    end while                                              ▷ end DuaIterate
    routes ← SUMO(malaga, trips, P_d)
    DUE.rp ← routes                                        ▷ DUE.rp
    DUE.r ← getUnique(routes)                              ▷ DUE.r
end procedure
```

First, the initial trips from the case study (*malaga*) are obtained in order to maintain the initial demand when calculating the new routes. Second, by using *DuaIterate* the probabilities are initialized and the first traffic simulation is carried out to assign routes to vehicles and obtain travel times. After each traffic simulation, the probabilities are updated according to the travel time values measured in the simulation so that the probability of assigning a route is higher for those with lower travel times. This process is repeated until the algorithm converges or the maximum number of steps is reached and *DuaIterate* ends.

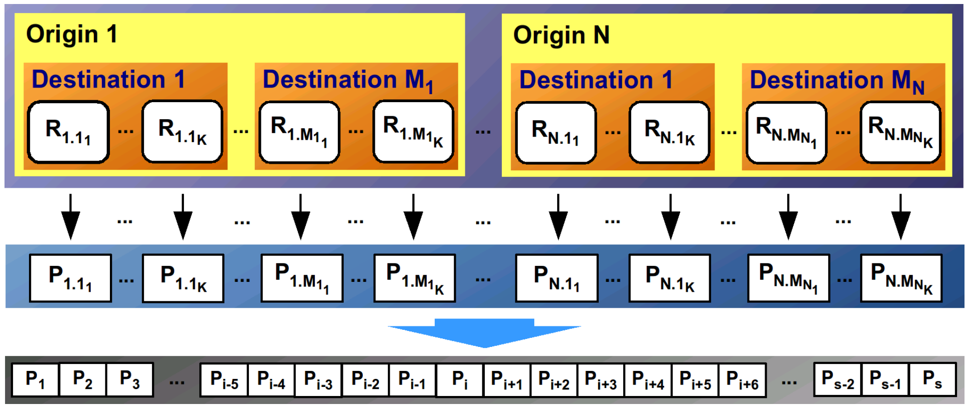


Figure 3: Solution encoding. In our case study $N = 8$, $S = 121$, and M_i depends on the origin i.

Finally, the routes resulting from the DUE process are used to build the *DUE.rp* (Dynamic User Equilibrium routes by probability) strategy, in which the probability of choosing a route from a starting point to a destination from those available, depends on how frequently it has been assigned by *DuaIterate*.

Additionally, the *DUE.r* (Dynamic User Equilibrium routes) assignation is obtained by keeping just the different routes (without repetition) so that all the routes from each origin to a destination are equiprobable.

We propose a third strategy to assign the routes included in *DUE.r*. Instead of assigning them according to how frequently they are used (*DUE.rp*) we propose an Evolutionary Algorithm (EA) to calculate the best probabilities for each route to minimize travel times. We have named our proposal *DUE.ea* and it is described in the next section.

4.1 DUE.ea

We have designed a (10+2)-EA (an elitist steady state evolutionary algorithm with a population of ten individuals generating two new individuals at each step) to calculate the probabilities of assigning one of the available routes to vehicles which are driving through the area under analysis.

4.1.1 Solution Encoding. The solution encoding consists of a numeric vector corresponding to the probabilities for the routes to be chosen. The probability values correspond to the different routes from the possible origins and their available destinations in the case study (Figure 1).

Figure 3 shows the representation of the problem where N origins are arranged into blocks containing the M_n reachable destinations from each origin. Finally, each destination could be reached by $K_{n,m}$ routes which have an associated probability value and are restricted so that the sum of them in a destination block is equal to 1.0. Note that the number of possible destinations (M) and routes (K) for each origin is not always the same as it depends on the streets' connectivity where not all destinations can be reached from each origin in the area analyzed.

Our case study contains 121 routes between their eight origins and seven destinations (Figure 1), so that the problem representation is a vector of 121 probability values.

4.1.2 Fitness function. We define the fitness function presented in Equation 1 to reduce travel times and later evaluate the rest of the metrics (gas emissions, fuel consumption, and route lengths) as a way of checking how robust our solution is and what relationships are observed.

$$F = \frac{1}{\alpha}\frac{1}{N}\sum_{i=1}^{N} travel\ time_i \tag{1}$$

The coefficient α is calculated as described in Equation 2. It is used to normalize the value returned by the fitness function so that the evaluation of each scenario (sc) of the case study is equal to 1.0. Consequently, fitness values lower that 1.0 indicate improvement in the average travel times as our aim is to minimize them, i.e. the lower, the better.

$$\begin{aligned} \alpha_{sc} &= \frac{1}{N_{sc}}\sum_{i=1}^{N_{sc}} travel\ time(sc)_i \\ sc &\in \{malaga_{WD}, malaga_{SAT}, malaga_{SUN}\} \end{aligned} \tag{2}$$

4.1.3 Operators. The selection strategy implemented in the EA is Binary Tournament. We have used a standard two point crossover as the recombination operator where the crossing points are the origin blocks as shown in Figure 4. We can see there that the entire block of probabilities belonging to the routes for each origin are exchanged between individuals, which gives the operator the ability to build new configurations at the block level.

Additionally, for the mutation operator, we have designed an operator that changes the probability values for the routes in a destination block by first selecting one of them, then incrementing its value, and finally decrementing the rest, in order to keep the sum total equal to 1.0 (Figure 5). In the example the destination j of the origin i has been selected for mutation. Then, probability $P_{i.j_3}$ for route $R_{i.j_3}$ is randomly selected to be incremented by 0.1 (probability increment). We can see in the resulting individual that not only has $P_{i.j_3}$ been incremented, but also the probabilities for the rest of routes in destination j have been decremented to keep the sum total equal to 1.0.

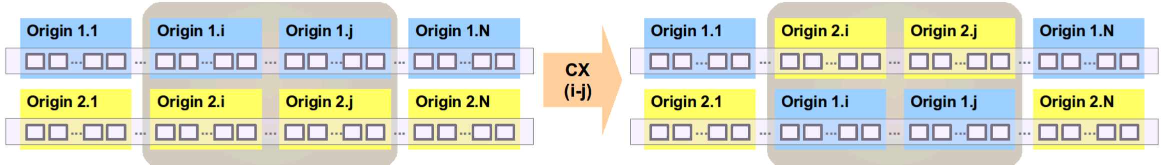


Figure 4: Crossover operator. Probabilities values for sensors i to j are selected to be exchanged between individuals.

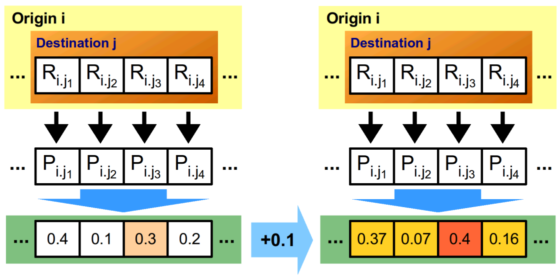


Figure 5: Mutation operator. Probability value of $P_{i.j_3}$ has been selected to be increased from 0.3 to 0.4 according to the defined increment 0.1. The rest of values are proportionally decreased to keep the sum total equal to 1.0.

We have experimentally set the crossover probability (P_c) to 0.9 and the mutation probability (P_m) to 0.1. Moreover, the probability increment performed by the mutation operator is set to 0.1.

Finally, we have performed an elitist replacement, so that the worst individuals of the population are replaced if they have a fitness value higher than the offspring produced in the current generation in order to build the next one.

5 RESULTS

We tested our proposal in our case study for one hour, to obtain not only travel times but also greenhouse gas emissions, fuel consumption, and distance traveled by vehicles.

First, we took the three scenarios of our map ($malaga_{WD}$, $malaga_{SAT}$, and $malaga_{SUN}$) calculated by using the FGA as explained in Section 3. To achieve the desired precision (greater than 99.6% in all the scenarios) we performed 90 independent runs of the FGA (30 per scenario) which lasted 5.2, 3, and 2.6 hours, respectively. Second, we obtained the Dynamic User Equilibrium routes (*DUE.r*), and the *DUE.rp* (Dynamic User Equilibrium routes by probability) as explained in Section 4. This process took about 5 minutes to converge.

Then, we have tested the *DUE.r* and *DUE.rp* routes in our scenarios by making the GPS navigators to suggest these routes. *DUE.r* routes are equiprobable, while in *DUE.rp*, the route probability depends on how much they have been assigned when calculating the user equilibrium.

Furthermore, we tested the Dijkstra shortest path algorithm [2] (*Dijkstra*) to include its results as we believe that it is the strategy most used by GPS devices nowadays. The implementation of this algorithm and the weight function used in it are provided by SUMO, which takes into account the travel time according to the street characteristics of the city.

Finally, we calculated new probabilities for the DUE routes using our EA (*DUE.ea*) and tested them in our scenarios, as well. We performed 30 independent runs of the EA on each scenario (90 runs) which lasted 3.5, 4, and 3 hours on average, respectively. Note that we used several machines to execute the 30 independent runs in parallel so that we just had to wait for the longest execution to get our results (8.3, 6, and 4.7 hours).

Table 2 shows the results obtained in terms of Travel Times (TT), Carbon Monoxide (CO), Carbon Dioxide (CO_2), Hydrocarbons (HC), Particulate Matter (PM), Nitrogen Oxides (NO), Fuel consumption ($Fuel$), and traveled distance ($Distance$). Note that we are supposing that all the drivers crossing the area have a GPS device and follow the indications given.

We can see that in spite of the reduced travel times (and emissions) produced by *DUE.r*, *DUE.rp* and even *Dijkstra*, the shortest travel times are obtained in the three scenarios when using *DUE.ea*. Our strategy also has the lowest emissions and fuel consumption as vehicles arrive at their destinations earlier, avoiding possible traffic jams. Differences in distances between the strategies are negligible (variations below 1%). The shortest ones were obtained by $Dijkstra$.

In Figure 6 the results obtained are presented as improvement percentages when vehicles are being routed according to the strategies analyzed here instead of following the flows obtained from the available real data. We can see that the greatest improvements are achieved when there are more vehicles in the area (working days) and that *DUE.r* and *DUE.rp* perform better than *Dijkstra* which was expected as they have more routes available for vehicles. However, *DUE.ea* outperforms all of them in all the scenarios and metrics, achieving improvements in travel times (up to 18%), CO (up to 14%), and fuel consumption (up to 7.5%).

By using *DUE.ea* in the GPS navigators the city's streets are exploited better by vehicles, with drivers leaving the analyzed zone, on average 63 seconds earlier. Although our initial concern was to shorten travel times, a better flow of vehicles preventing congestion has also reduced pollution levels as well as fuel consumption (9.3 liters per hour on average).

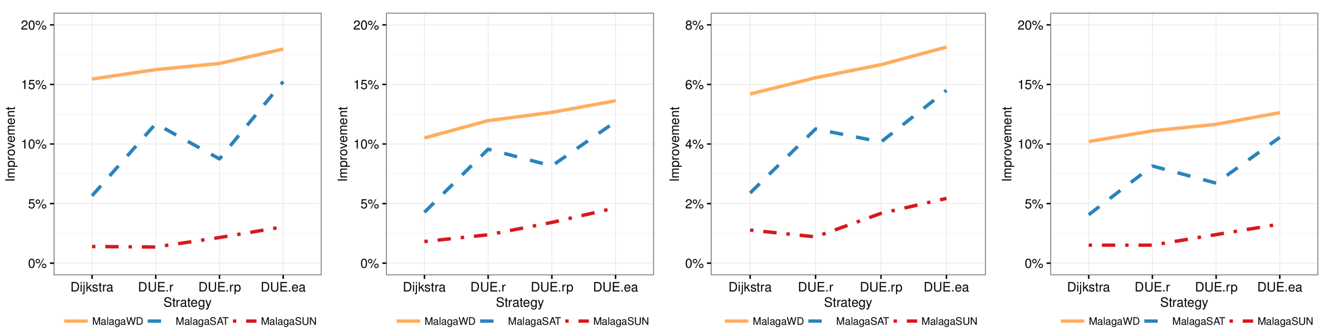
5.1 Penetration Rate

We also wished to know if our proposal would be useful when it is not being used by every single vehicle (a very real assumption). To answer this question, we tested the configuration (probabilities) achieved by *DUE.ea* when no one was using it (*Malaga* real traffic) and incrementing the penetration rate in steps of 10% until reaching a full usage (*DUE.ea* values previously reported in Table 2).

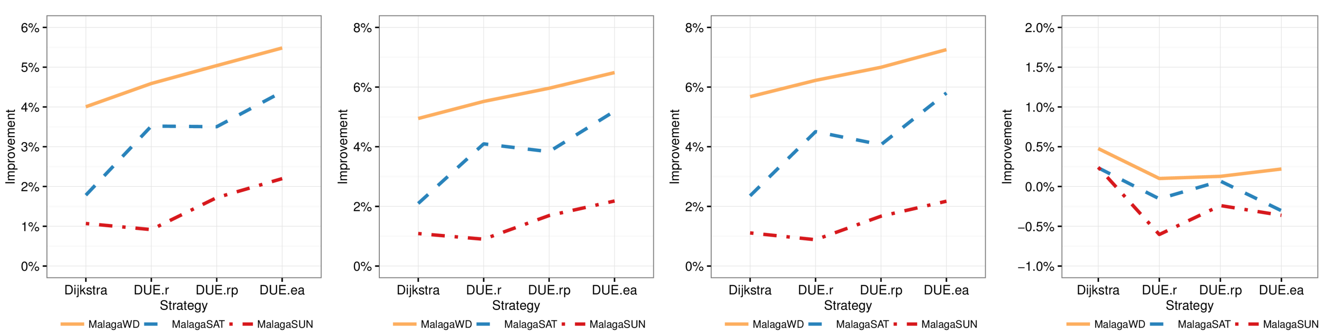
We present our penetration rate results in Figure 7. It can be seen that, despite some variations which make the increment inconsistent, all the metrics are improved with respect to *Malaga*

Table 2: Results obtained for the scenarios when vehicles are using the routes based on data publish by the Malaga local council (*Malaga*), shortest path (*Dijkstra*), Dynamic User Equilibrium (*DUE.r*), Dynamic User Equilibrium with probabilities obtained by rate of use (*DUE.rp*), and Dynamic User Equilibrium with probabilities obtained by our EA (*DUE.ea*). Additionally, a statistical analysis is provided (Friedman Ranks and Wilcoxon p-values) showing that our results are statistically significant.

Scenario	Strategy	# Veh.	*TT* (s)	*CO* (mg)	CO_2 (mg)	*HC* (mg)	*PM* (mg)	*NO* (mg)	*Fuel* (l)	Dist. (m)	Friedman Rank	Wilcoxon p-value
$malaga_{WD}$	Malaga	4883	351.6	1591.9	322840.7	88.6	20.7	554.1	128.7	1926.6	3.20	0.00
	Dijkstra	4883	297.3	1424.7	304507.5	79.6	19.9	526.7	121.4	**1917.4**	3.00	0.00
	DUE.r	4883	294.5	1401.5	302745.6	78.8	19.8	523.5	120.7	1924.6	2.98	0.01
	DUR.rp	4883	292.7	1390.5	301328.7	78.3	19.7	521.0	120.1	1924.1	2.93	0.09
	DUE.ea	4883	**288.5**	**1374.9**	**299418.3**	**77.4**	**19.6**	**518.1**	**119.4**	1922.3	**2.90**	—
$malaga_{SAT}$	Malaga	3961	344.1	1547.7	323919.4	87.1	20.9	557.0	129.1	2004.9	3.18	0.00
	Dijkstra	3961	324.7	1481.6	316290.6	83.6	20.5	545.3	126.1	**2000.2**	3.06	0.00
	DUE.r	3961	303.8	1399.7	309326.1	80.0	20.2	534.2	123.3	2008.0	2.95	0.00
	DUR.rp	3961	314.0	1421.3	310741.4	81.2	20.2	535.6	123.9	2003.5	2.97	0.00
	DUE.ea	3961	**291.7**	**1363.9**	**305130.4**	**77.9**	**20.0**	**528.1**	**121.7**	2011.0	**2.84**	—
$malaga_{SUN}$	Malaga	3679	279.6	1292.4	291131.9	74.0	19.1	503.9	116.1	1933.3	3.09	0.00
	Dijkstra	3679	275.7	1269.0	287901.5	72.9	18.9	498.4	114.8	**1928.6**	2.99	0.05
	DUE.r	3679	275.8	1261.6	288565.0	72.9	18.9	499.4	115.0	1945.0	3.04	0.02
	DUR.rp	3679	273.6	1248.3	286268.5	72.3	18.7	495.4	114.1	1937.9	2.96	0.03
	DUE.ea	3679	**271.1**	**1232.5**	**284807.0**	**71.6**	**18.6**	**492.9**	**113.5**	1940.3	2.92	—

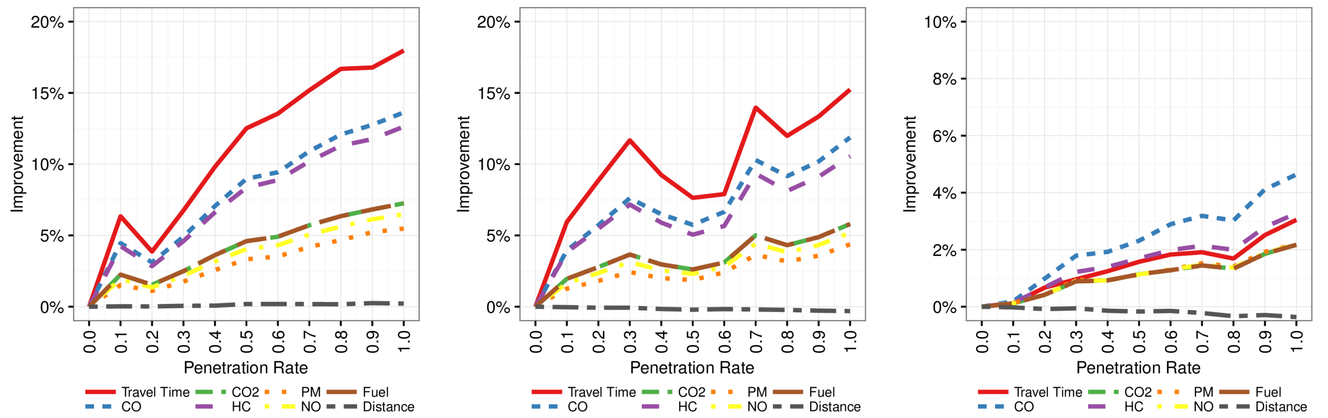


(a) *Travel Time* (b) *CO* (c) CO_2 (d) *HC*



(e) *PM* (f) *NO* (g) *Fuel* (h) *Distance*

Figure 6: Improvements in the metrics for the three scenarios of our case study when using different strategies for routing vehicles instead of the routes obtained from the data published by the local council. Note that some of the scales used are different for better visualization.



(a) $malaga_{WD}$ (b) $malaga_{SAT}$ (c) $malaga_{SUN}$

Figure 7: Penetration rate study for the three scenarios of our case study. Note that some of the scales used are different for better visualization. Even with a reduced use of our technique (10% of drivers) time and other metrics are improved.

(0% usage) as the penetration rate increased. There is at least a minimum improvement when just 10% of the drivers are using the routes and probabilities calculated by *DUE.ea* in their GPS device, which gets better as the penetration rate increases.

6 CONCLUSIONS

In this article we have proposed a way of calculating alternative routes to be used by a GPS navigator. Additionally, we have provided three different strategies to select which of these routes are presented to the drivers by the navigator, and compared them with each other and with the Dijkstra shortest path algorithm. We have worked with real maps obtained from OpenStreetMap and actual traffic distribution calculated from the data published by the local council of Malaga.

Our results show that we have improved travel times (up to 18%), greenhouse gas emissions (up to 14%), and fuel consumption (up to 7.5%) in *Malaga* when using DUE.ea. Furthermore, we have also demonstrated that our proposal is viable even when just 10% of drivers are using it.

Having tested our strategy in the case study presented we want to extend the analysis to other districts of the city with the aim of improving their road traffic. Finally, an integration of all the individually optimized areas would be the strategy to follow, to optimize the road traffic throughout the whole city. Although this would have to be done, taking into consideration the complexity that simulating a big scenario with thousands of vehicles in terms of computation time and hardware requirements entails.

7 ACKNOWLEDGEMENTS

This research is partially funded by the Spanish MINECO project TIN2014-57341-R (http://moveon.lcc.uma.es). Daniel H. Stolfi is supported by a FPU grant (FPU13/00954) from the Spanish Ministry of Education, Culture and Sports. University of Malaga. International Campus of Excellence Andalucia TECH.

REFERENCES

[1] Hayder Amer, Naveed Salman, Matthew Hawes, Moumena Chaqfeh, Lyudmila Mihaylova, and Martin Mayfield. 2016. An Improved Simulated Annealing Technique for Enhanced Mobility in Smart Cities. *Sensors* 16, 7 (jun 2016), 1013.

[2] Edsger W Dijkstra. 1959. A Note on Two Problems in Connexion with Graphs. *Numerische mathematik* (1959), 269–271.

[3] Ralph Gakenheimer. 1999. Urban mobility in the developing world. *Transportation Research Part A: Policy and Practice* 33, 7-8 (1999), 671–689.

[4] Mevlut Turker Garip, Mehmet Emre Gursoy, Peter Reiher, and Mario Gerla. 2015. Scalable reactive vehicle-to-vehicle congestion avoidance mechanism. In *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*. IEEE, 943–948.

[5] Christian Gawron. 1998. *Simulation-based traffic assignment.* Ph.D. Dissertation. University of Cologne.

[6] Ole Hertel, Steen Solvang Jensen, Martin Hvidberg, Matthias Ketzel, Ruwim Berkowicz, Finn Palmgren, Peter Wåhlin, Marianne Glasius, Steffen Loft, Peter Vinzents, Ole Raaschou-Nielsen, Mette Sørensen, and Helle Bak. 2008. Assessing the Impacts of Traffic Air Pollution on Human Exposure and Health. In *Road Pricing, the Economy and the Environment.* Springer Berlin Heidelberg, 277–299.

[7] Ching-Lai Hwang and Kwangsun Yoon. 2012. *Multiple attribute decision making: methods and applications a state-of-the-art survey.* Vol. 186. Springer Science & Business Media.

[8] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. 2012. Recent Development and Applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements* 5, 3 (2012), 128–138.

[9] Bojin Liu, D Ghosal, Chen-Nee Chuah, and H M Zhang. 2012. Reducing Greenhouse Effects via Fuel Consumption-Aware Variable Speed Limit (FC-VSL). *Vehicular Technology, IEEE Transactions on* 61, 1 (jan 2012), 111–122.

[10] OpenStreetMap-Fundation. 2017. OpenStreetMap. (jan 2017). http://www.openstreetmap.org

[11] Daniel H Stolfi and Enrique Alba. 2014. Red Swarm: Reducing travel times in smart cities by using bio-inspired algorithms. *Applied Soft Computing Journal* 24, 0 (nov 2014), 181–195.

[12] Daniel H Stolfi and Enrique Alba. 2015. An Evolutionary Algorithm to Generate Real Urban Traffic Flows. In *Advances in Artificial Intelligence.* Lecture Notes in Computer Science, Vol. 9422. Springer International Publishing, 332–343.

[13] Yuki Sugiyama, Minoru Fukui, Macoto Kikuchi, Katsuya Hasebe, Akihiro Nakayama, Katsuhiro Nishinari, Shin-ichi Tadaki, and Satoshi Yukawa. 2008. Traffic jams without bottlenecks – experimental evidence for the physical mechanism of the formation of a jam. *New J. Phys.* 10, 3 (2008), 33001.

[14] TNS Opinion & Social. 2013. *Attitudes of Europeans towards urban mobility.* Technical Report June.

[15] John Glen Wardrop. 1952. Some Theoretical Aspects of Road Traffic Research. In *Proc. Inst. Civil Eng.* 252–378.