Red Swarm: Smart Mobility in Cities with EAs

Daniel H. Stolfi and Enrique Alba

GECCO

Genetic and Evolutionary Computation Conference

2013

Amsterdam, The Netherlands
July 06-10, 2013

acm

Red Swarm: Smart Mobility in Cities with EAs

Daniel H. Stolfi
dhstolfi@lcc.uma.es
http://www.danielstolfi.com

Enrique Alba
eat@lcc.uma.es
http://www.lcc.uma.es/~eat

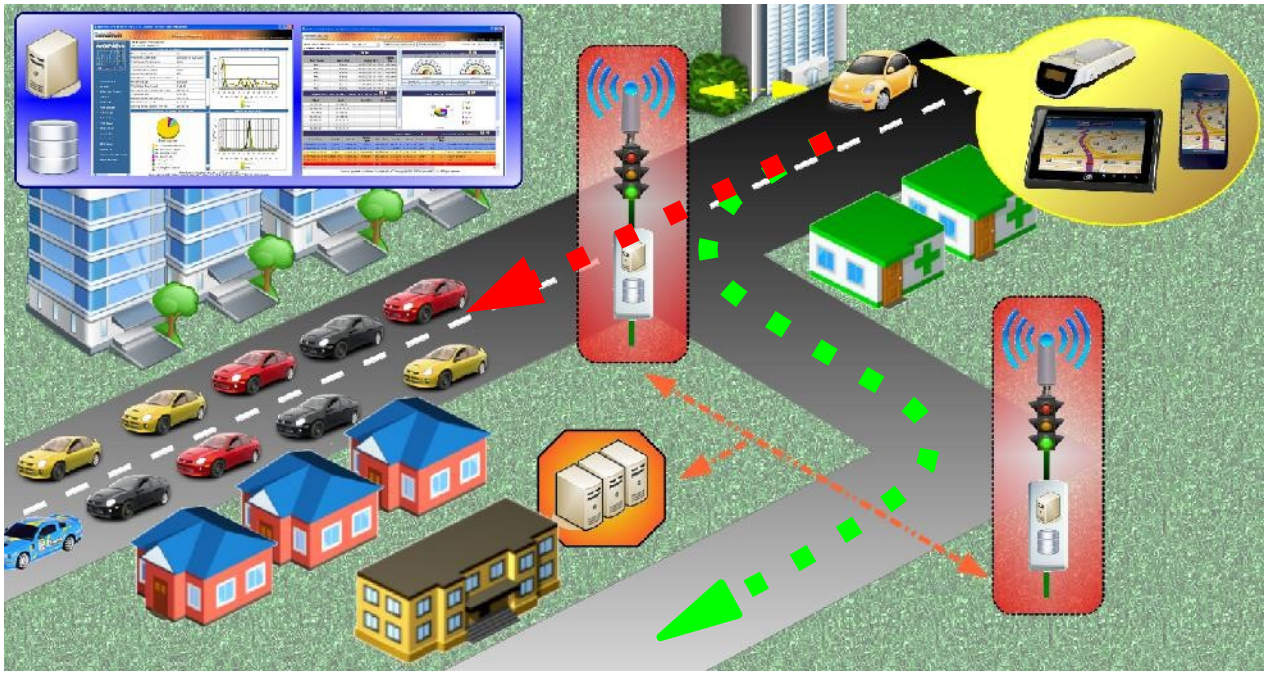
University of Malaga

July 2013

Table of Contents

Introduction

- People are living or thinking about moving to big cities
  - A higher number of vehicles are moving by the streets
  - The number of traffic Jams is rising
  - Tons of greenhouse effect gases are emitted to the atmosphere
  - The citizens' quality of life is decreasing



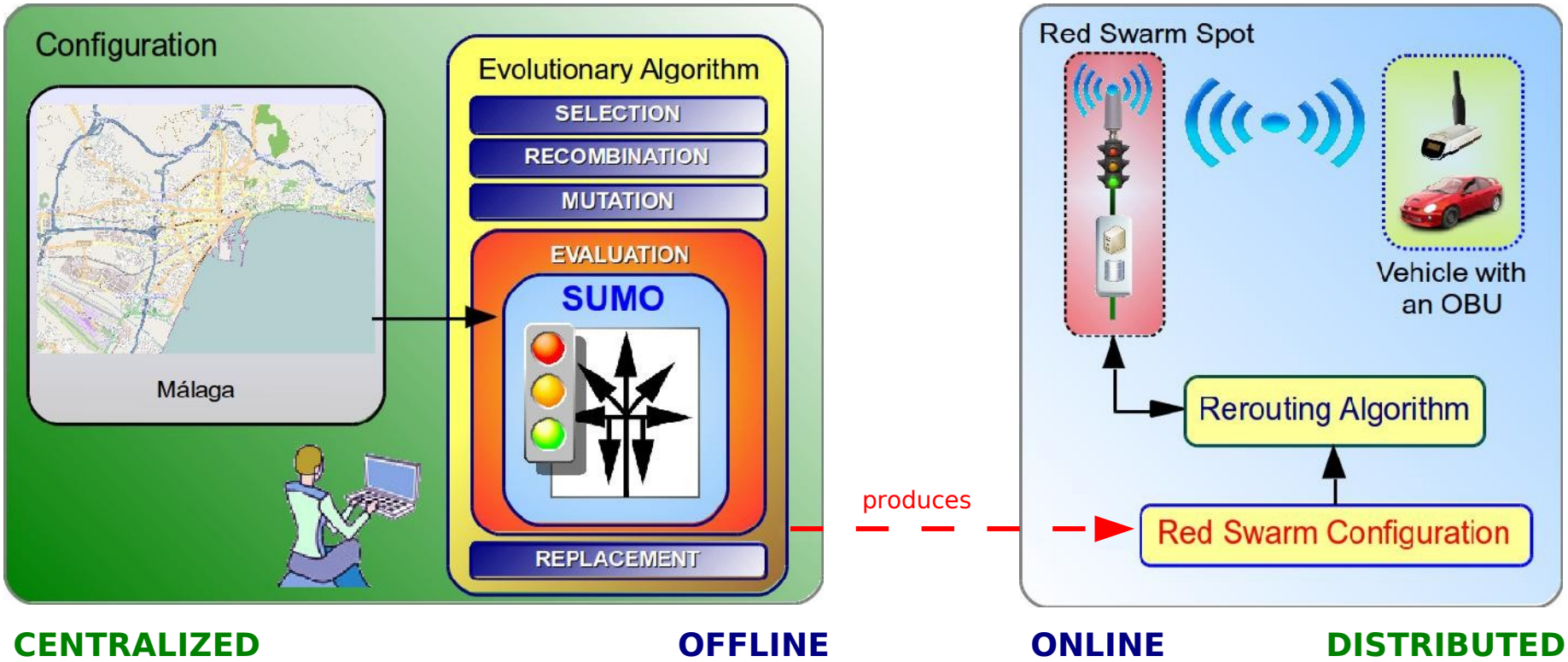
The optimization of road traffic is necessary in modern big cities

Red Swarm

Our Red Swarm consists of:

- Several spots distributed throughout the city
  - Installed at traffic lights
  - Linked to vehicles by using Wi-Fi
- Our Optimization Algorithm
- Our Rerouting Algorithm
- On Board Units (OBU)
  - Installed inside vehicles
  - Smartphones or tablets could be used instead

Architecture

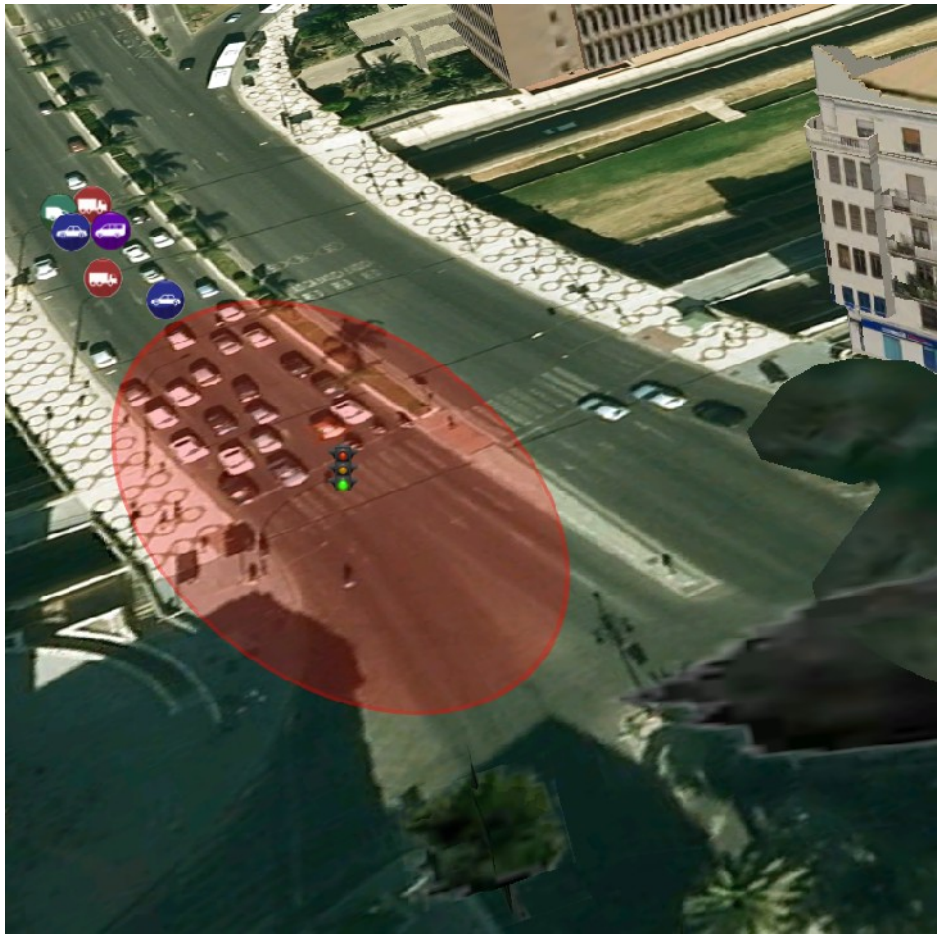


The Evolutionary Algorithm produces a configuration for the Red Swarm spots

The configured Red Swarm spots are deployed in several junctions of the city

Rerouting (t = 10 s.)

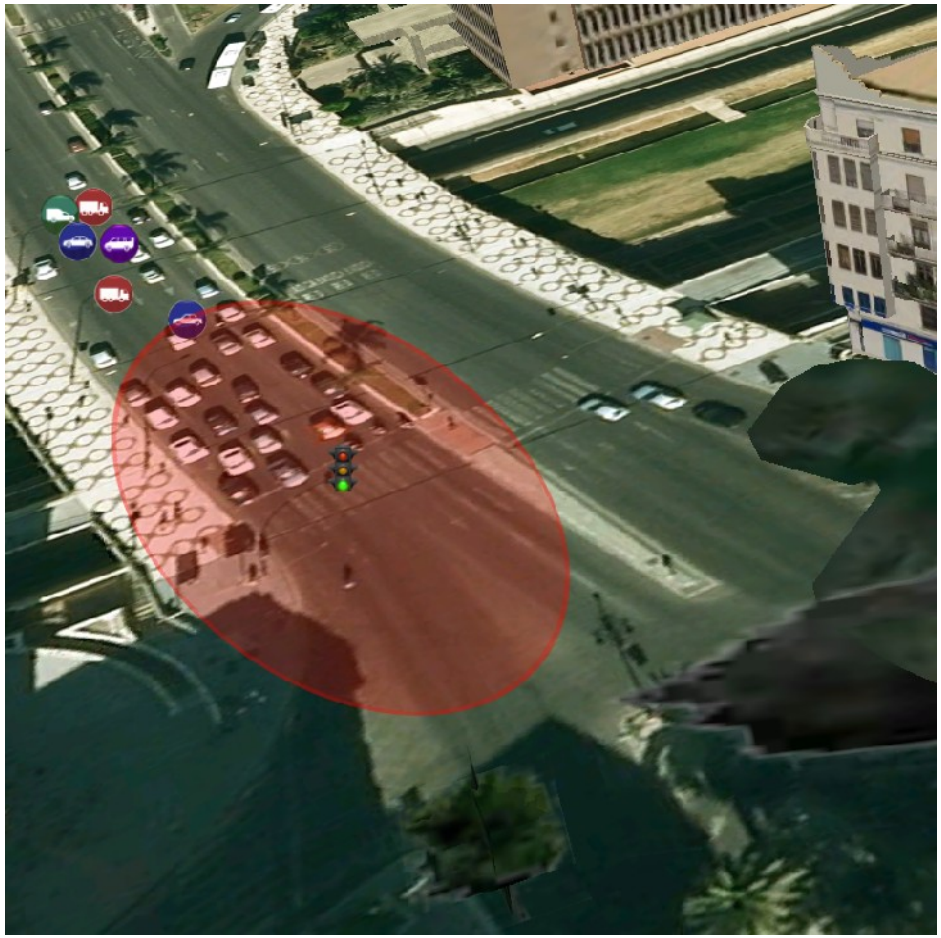
Experts' solution



Red Swam

Rerouting (t = 11 s.)

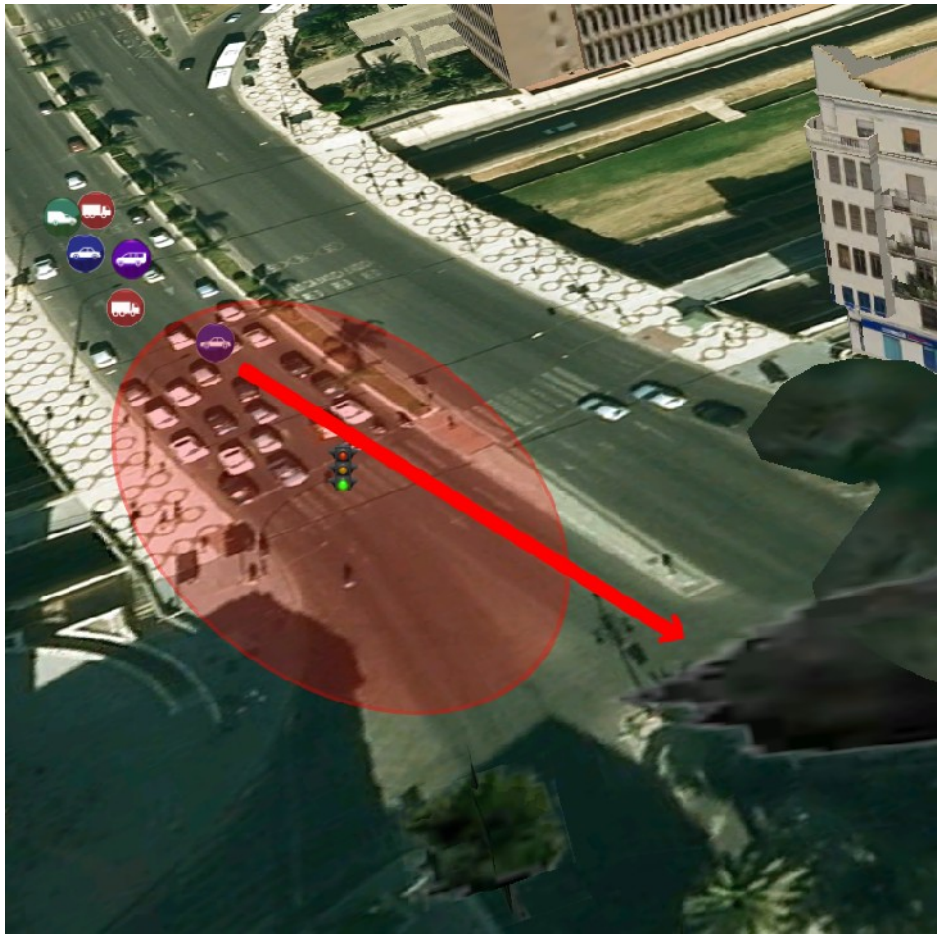
Experts' solution



Red Swam

Rerouting (t = 12 s.)

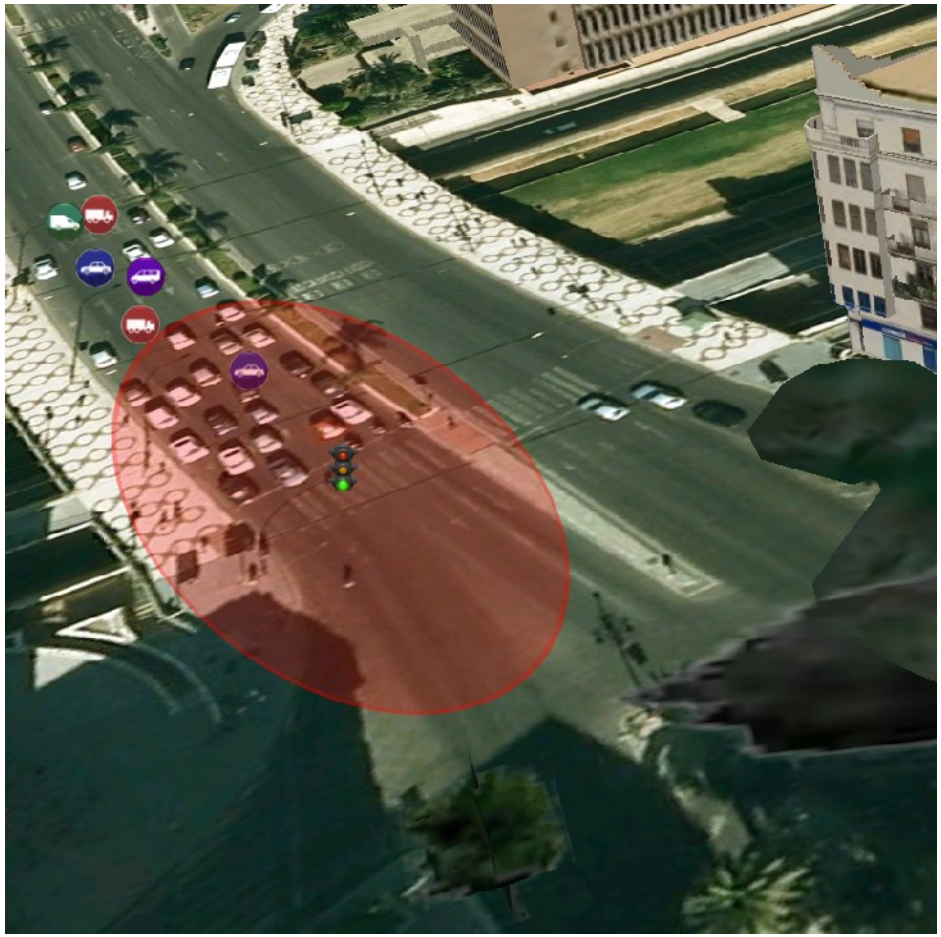
Experts' solution



Red Swam

Rerouting (t = 13 s.)

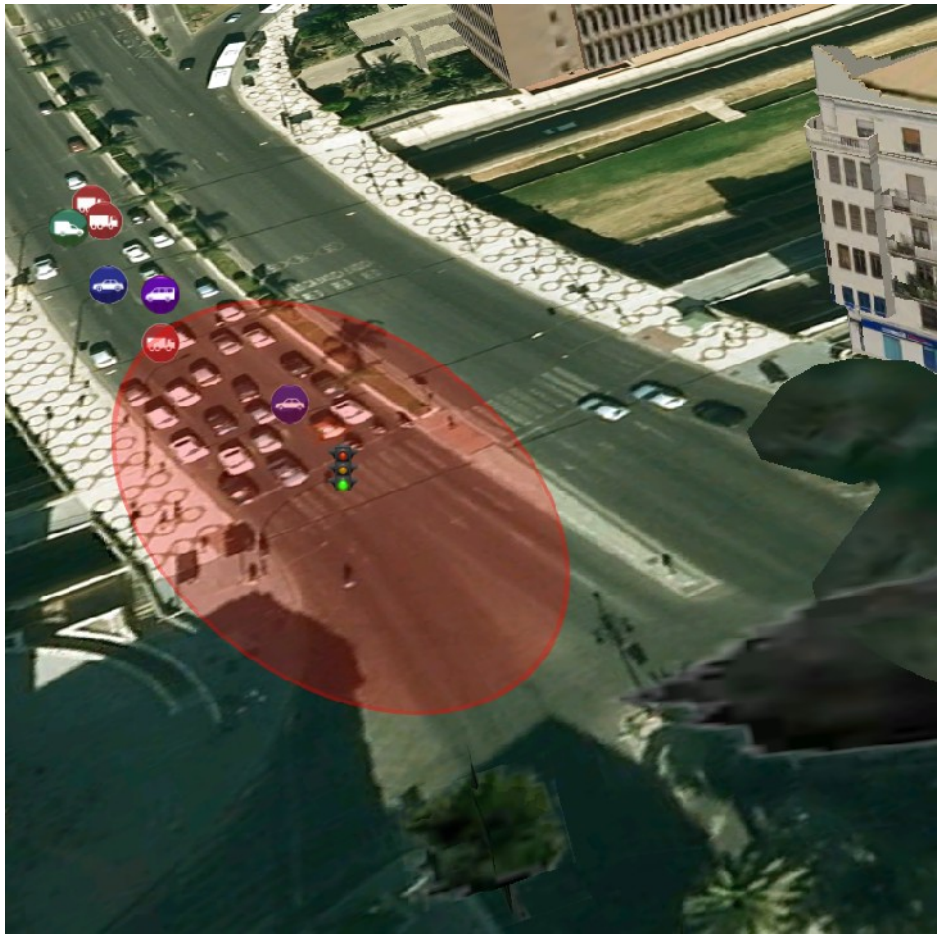
Experts' solution



Red Swam

Rerouting (t = 14 s.)

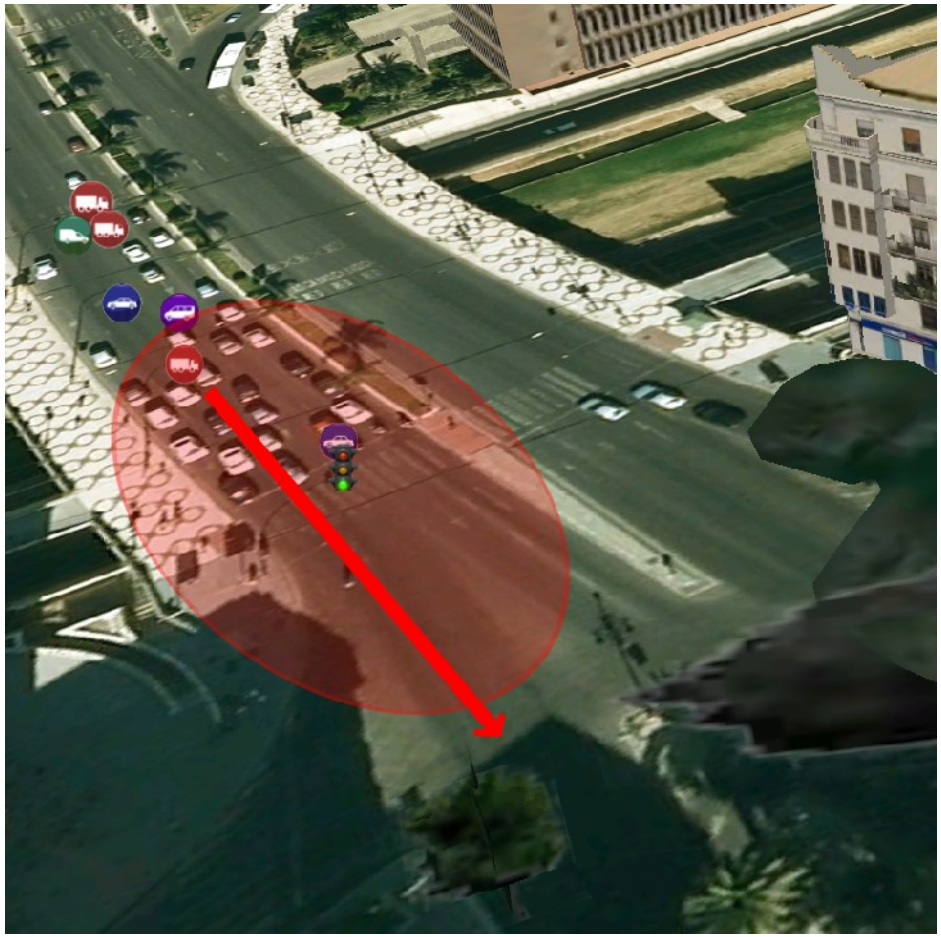
Experts' solution



Red Swam

Rerouting (t = 15 s.)

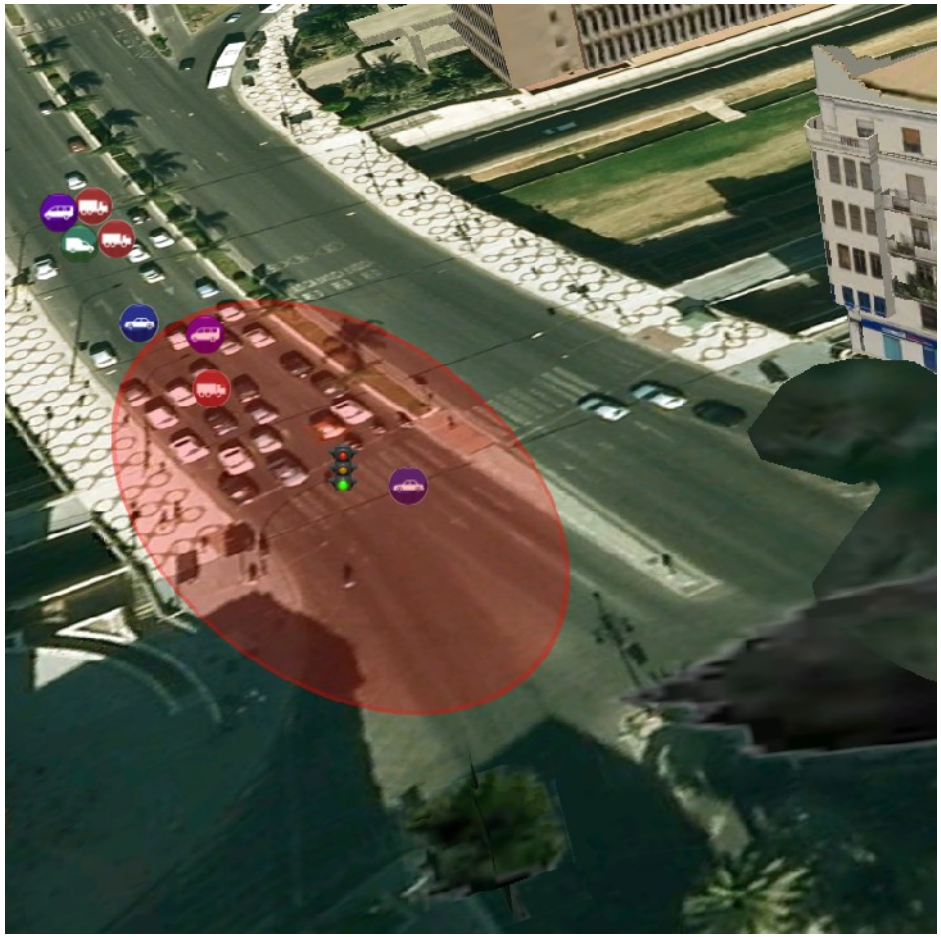
Experts' solution



Red Swam

Rerouting (t = 16 s.)

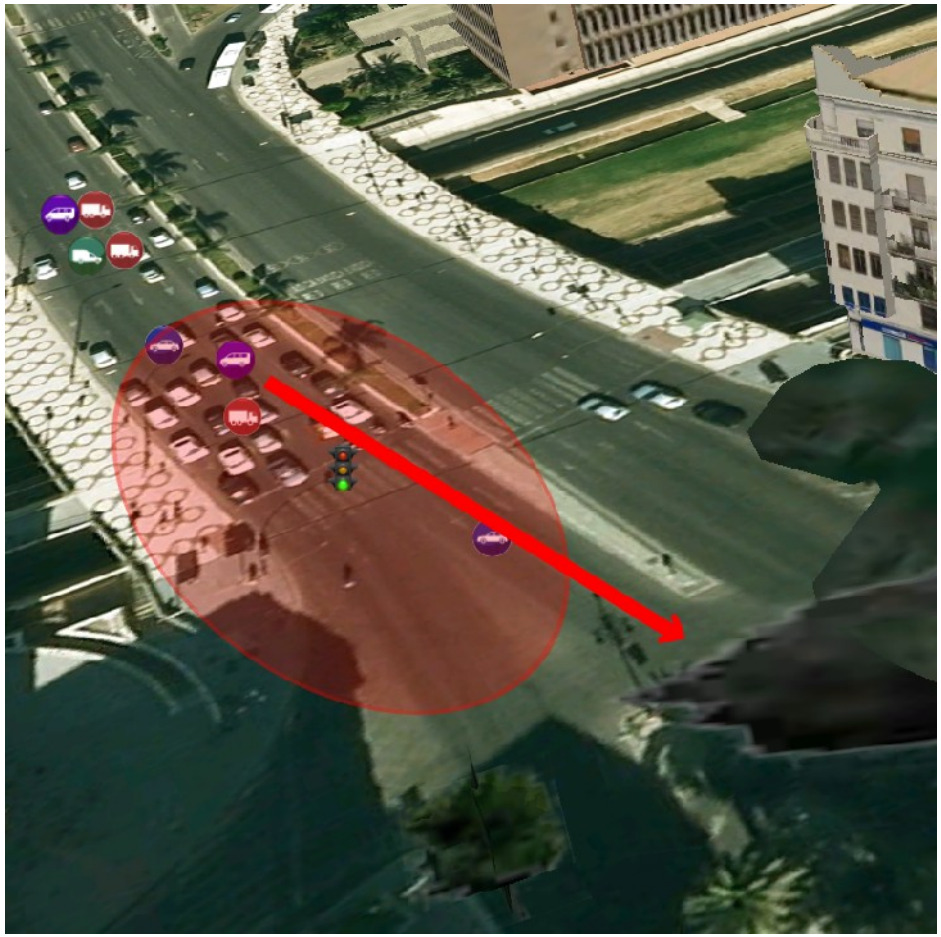
Experts' solution



Red Swam

Rerouting (t = 17 s.)

Experts' solution



Red Swam

Rerouting (t = 18 s.)

Experts' solution

Red Swam

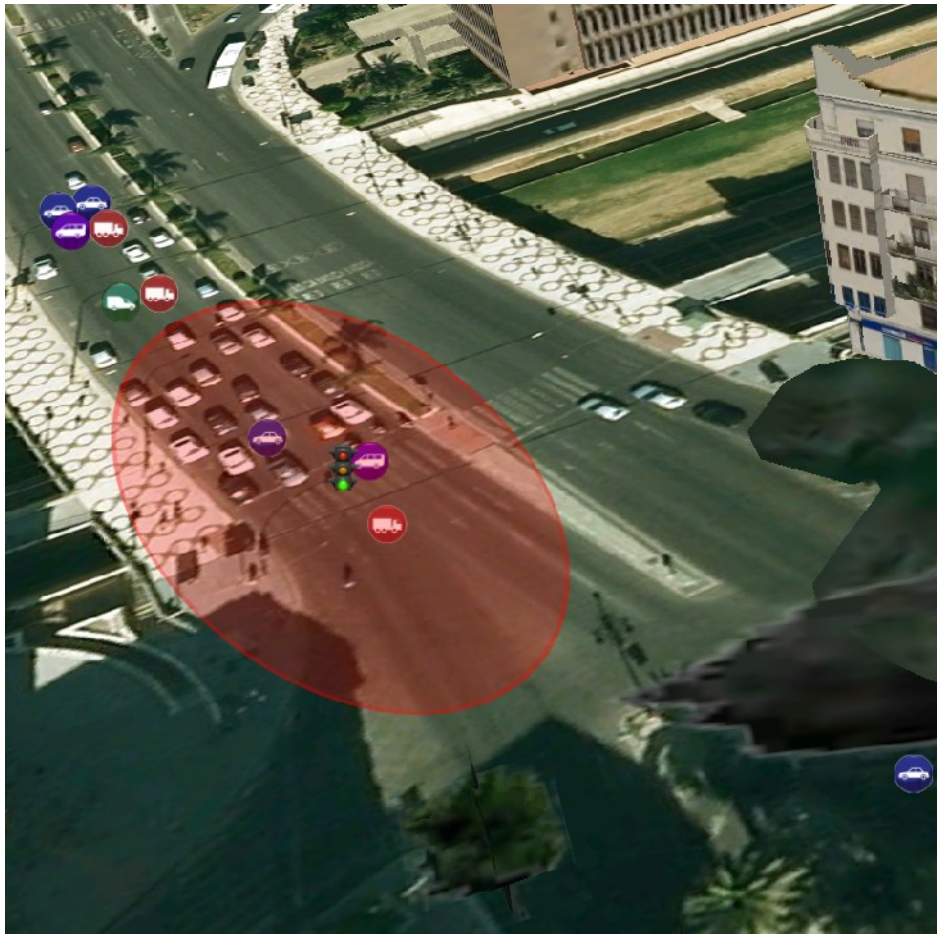
Rerouting (t = 19 s.)

Experts' solution

Red Swam

Rerouting (t = 20 s.)

Experts' solution



Red Swam

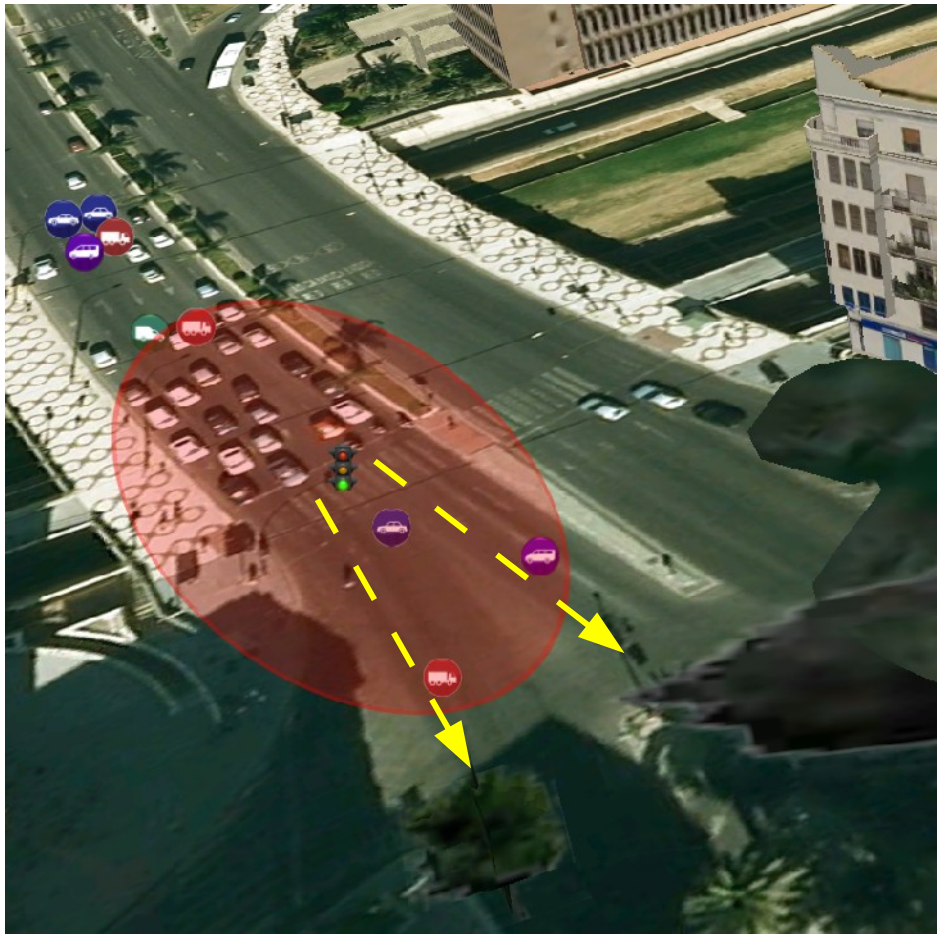
Rerouting (t = 21 s.)

Experts' solution

Red Swam

Rerouting (t = 22 s.)

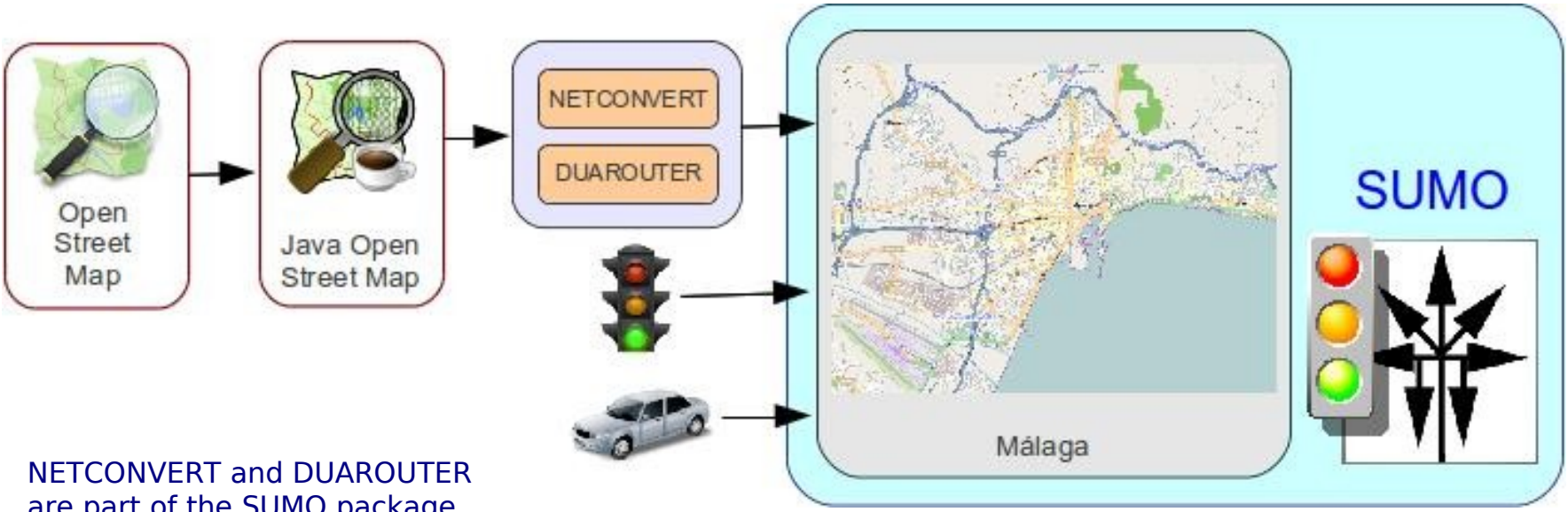
Experts' solution



Red Swam

Constructing the Scenario: Methodology

- We have worked with **real maps** imported from OpenStreetMap
- We have added 10 Red Swarm spots in junctions controlled by a traffic light
- We have imported the map into SUMO
- We have defined traffic flows for vehicles (experts' solution)
- This process can be adapted to any modern city in the world

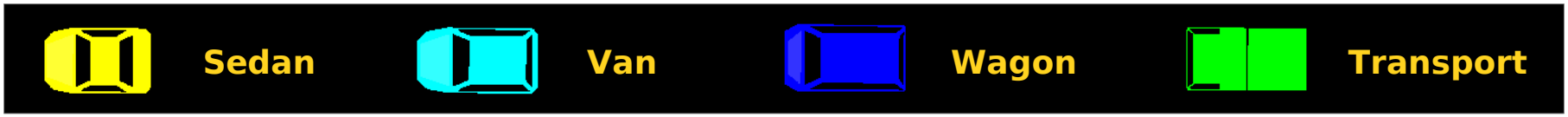


NETCONVERT and DUAROUTER are part of the SUMO package

Scenario: Malaga

MALAGA (SPAIN)

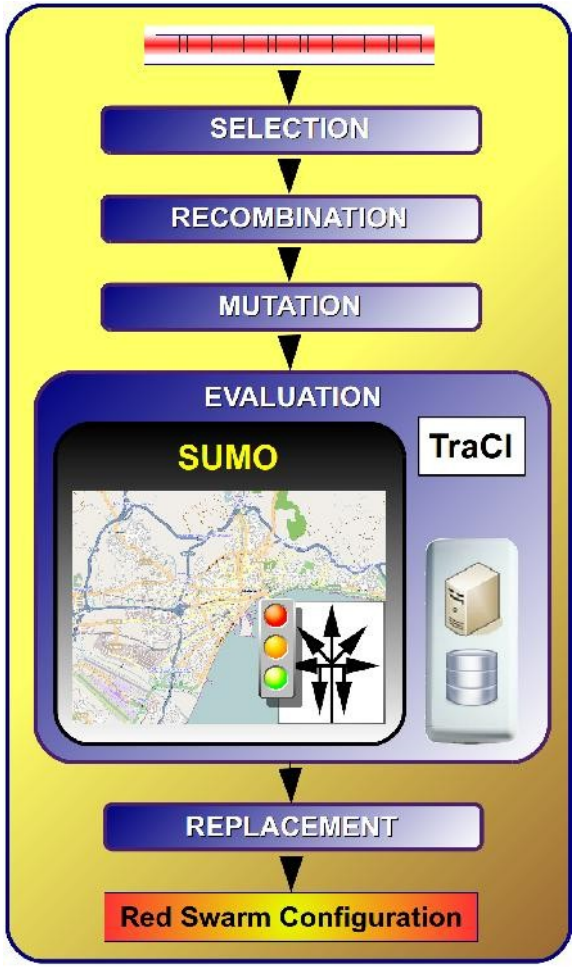
- Real Scenario
  - 261 traffic lights
  - 10 Red Swarm spots
  - 800 vehicles
  - 4 vehicle types
  - 3 different traffic patterns (*Scen1*, *Scen2* & *Scen3*)



Our goal is to reduce the travel time of the vehicles in high density conditions

Evolutionary Algorithm

- (10+2)-EA
- The fitness value is calculated by using the **SUMO** traffic simulator
- The rerouting produced by the Red Swarm spots is implemented by using **TraCI** and the **Rerouting Algorithm** which runs in each spot
- The result of the EA is a configuration for all the **Red Swarm** spots placed in the city



Evolutionary Algorithm

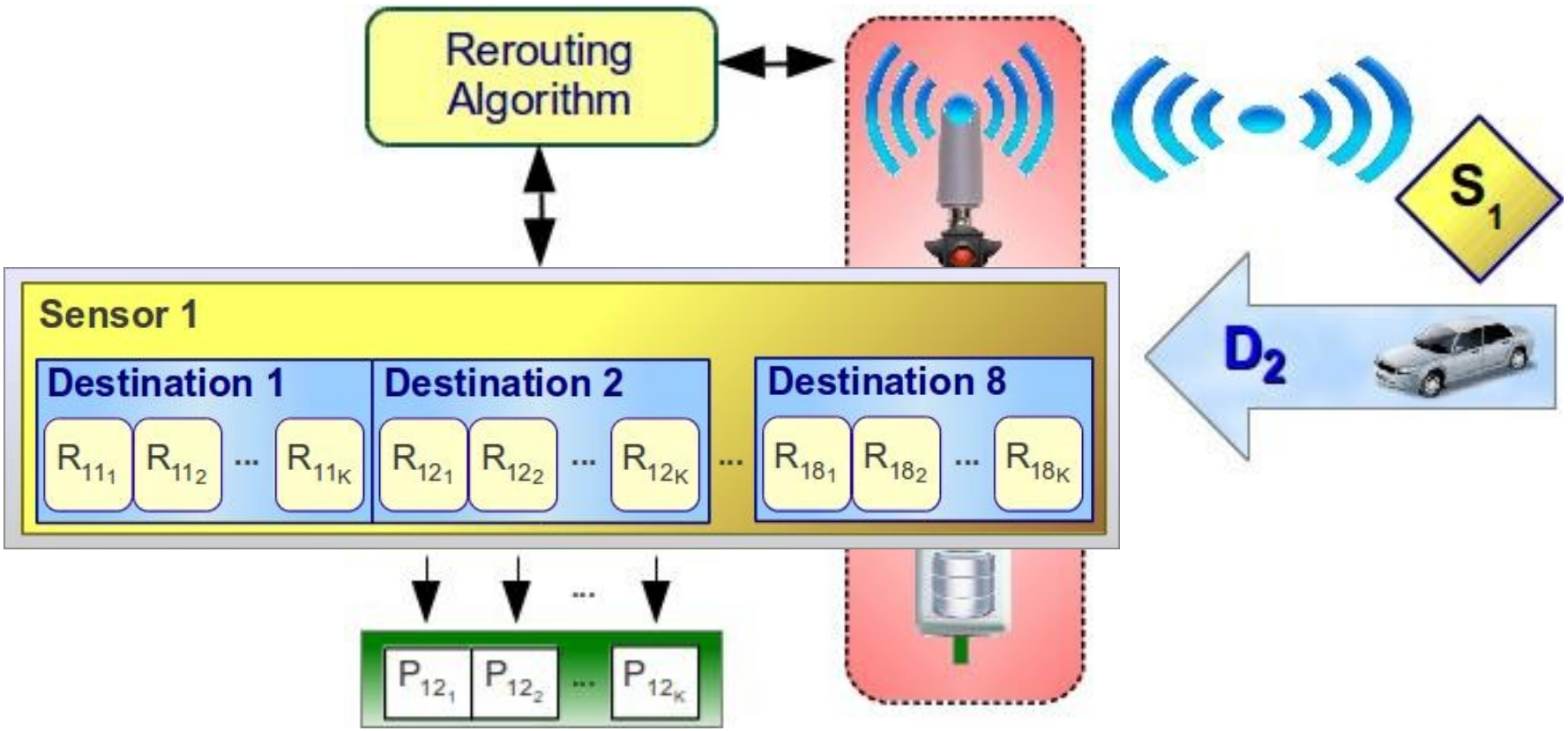
Evolutionary Algorithm: Representation (Sensors)

SENSORS

- They represent the inputs to Red Swarm spots in the simulated scenario
- Vehicles trigger the rerouting algorithm when they are detected by a sensor
- In a real city they would be radio links

Evolutionary Algorithm: Representation (Example)

For example, if a vehicle which is going to **Destination 2** is detected by **Sensor 1**, one of the routes from R_{12_1} to R_{12_K} will be chosen by the Rerouting Algorithm, depending on the values of the probabilities P_{12_1} to P_{12_K}

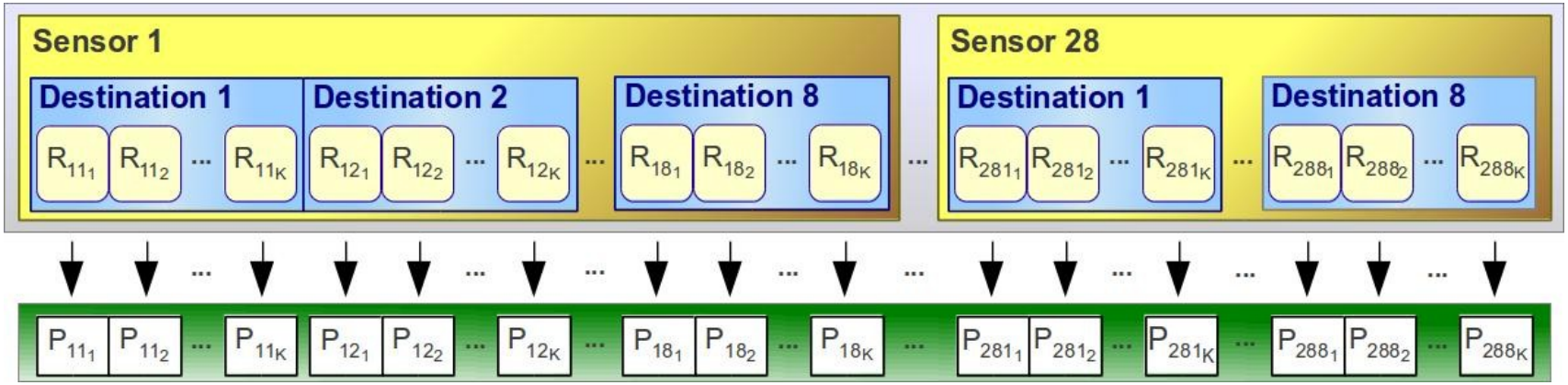


Evolutionary Algorithm: Representation

REPRESENTATION

- R_{SD_K}: Available routes from **Sensors** to **Destinations** and to other **spots**
- Each new route will be selected by the Rerouting Algorithm depending on the probabilities

Conceptual representation



Solution vector made up of 1119 floats (probabilities)

Evolutionary Algorithm: Fitness Function

FITNESS FUNCTION

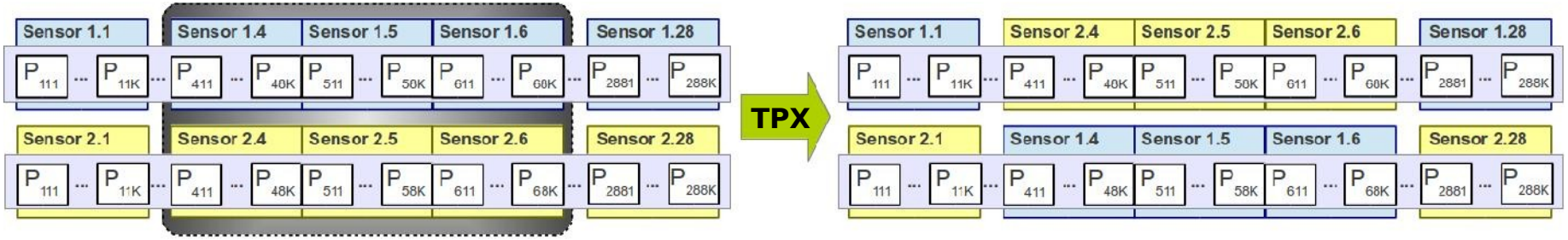
$$F = \alpha_1 (N - n_{trips}) + \alpha_2 \frac{\sum t_{trip}}{N} + \alpha_3 \frac{\sum t_{delay}}{N}$$

- N: Total number of vehicles (800 in this work)
- n_{trips}: Number of vehicles which have ended their itineraries
- t_{trip}: Trip time of each vehicle
- t_{delay}: Waiting time of each vehicle before entering the analyzed zone
- The three terms are weighted by α_1, α_2, and α_3, respectively
- We want to minimize the fitness value (the lower the better)

Evolutionary Algorithm: Recombination Operator

RECOMBINATION OPERATOR

- We use the standard two-point recombination
- The offspring are obtained by exchanging a range of sensor configurations from the selected parents



Evolutionary Algorithm: Mutation Operators

MUTATION OPERATOR

- Changes the probabilities of the routes in the sensor configurations

1) All Destinations – One Sensor

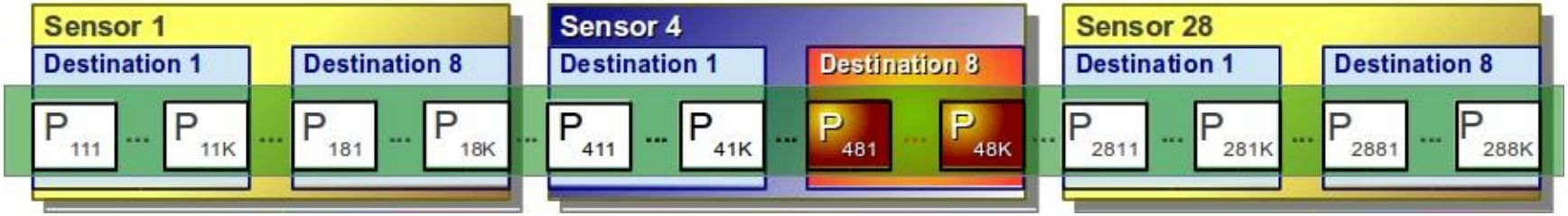
Exploration

It changes all probabilities in a Sensor block (i.e. all the probabilities in Sensor 4)

2) One Destination – One Sensor

Exploitation

It changes probabilities in a Destination block in a Sensor block
(i.e. only the probabilities of Destination 8 in Sensor 4)



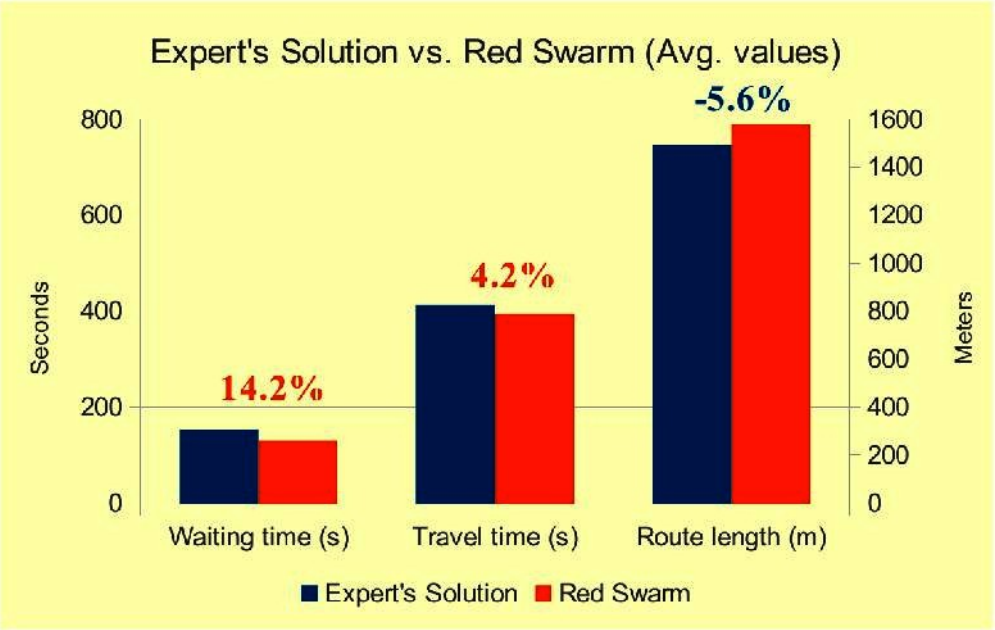
Results

The 800 vehicles leave the city in a lower time when we use Red Swarm

Metric	Expert's solution			Red Swarm		
	Sim1	*Sim2*	*Sim3*	*Sim1*	*Sim2*	*Sim3*
# Completed Trips	800	800	800	800	800	800
Total Time (s)	1283	1253	1385	**1154**	**1164**	**1206**
Minimum Fitness	532.9	530.8	549.0	**512.7**	**523.5**	**533.3**

Results show a reduction of the average waiting time and the average travel time

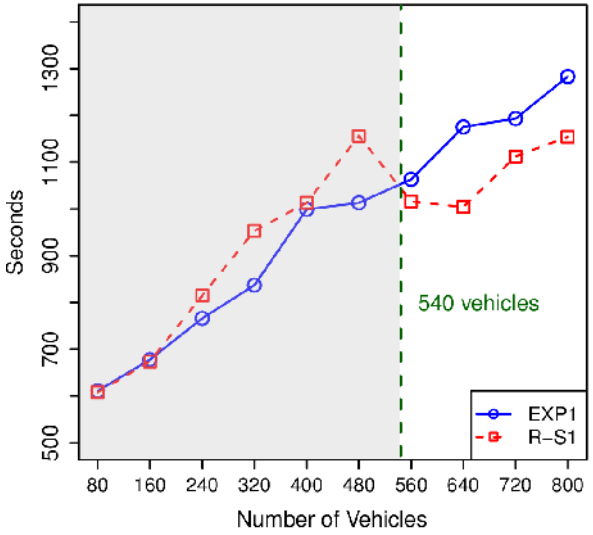
Vehicles have to travel a longer distance because they are using alternative routes



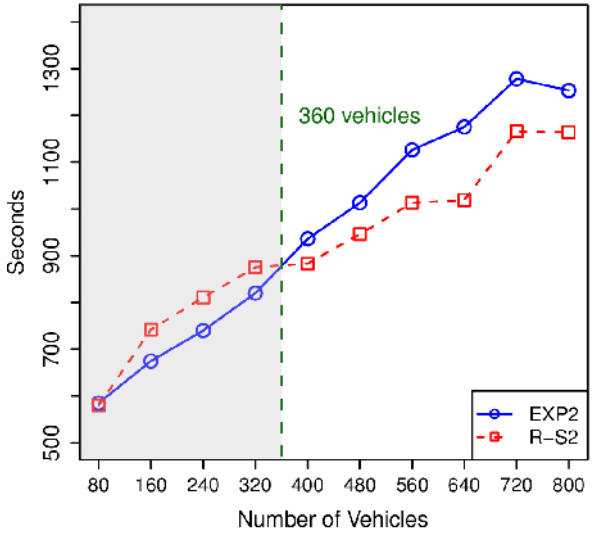
Results: Simulation Time vs. Number of Vehicles

- Figures show that the higher the number of vehicles is, the more effective Red Swarm becomes
- Under the threshold, **Red Swarm** still being useful
- Advantages are evident when the traffic density raises

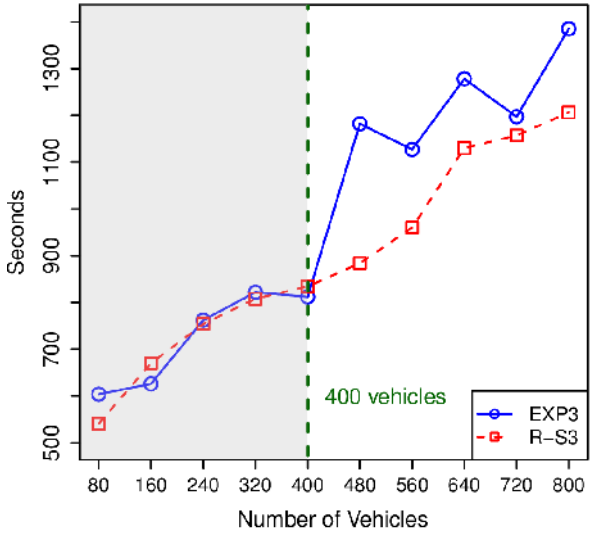
Travel Time (*Scen1*)



Travel Time (*Scen2*)

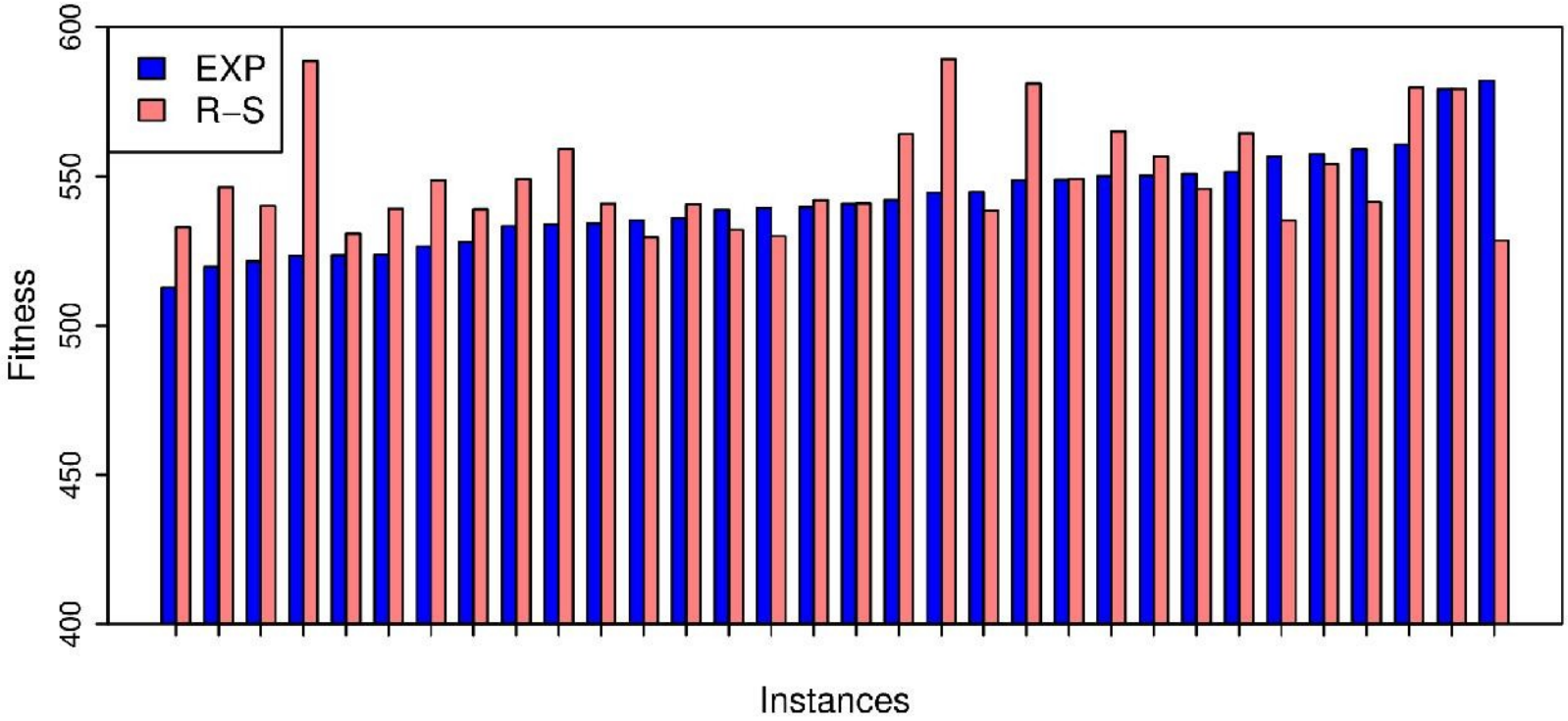


Travel Time (*Scen3*)



Generalization of Results

- This figure shows the fitness values of the execution of **Red Swarm** in 30 extra scenarios compared with the experts' solution
- Red Swarm has not only worked in all these scenarios but has also achieved lower travel times in 20 of them (66.7%)



Conclusions

- This is an innovative approach to the prevention of traffic jams
- The results confirm road traffic can be improved by using **Red Swarm**
- We have reduced the travel times and waiting times of the vehicles
- Currently, we are working on:
  - The expansion of the analyzed region
  - Collecting information from vehicles (on-line and historical data of the city)
  - The reduction of greenhouse gas emissions

http://neo.lcc.uma.es

http://en.danielstolfi.com/redswarm/

http://neo.lcc.uma.es

http://en.danielstolfi.com/redswarm/

dhstolfi@lcc.uma.es

Questions?