



**HAL**  
open science

## **Fitness landscapes analysis and adaptive algorithms design for traffic lights optimization on SIALAC benchmark**

Florian Leprêtre, Cyril Fonlupt, Sébastien Verel, Virginie Marion, Rolando Armas, Hernán Aguirre, Kiyoshi Tanaka

► **To cite this version:**

Florian Leprêtre, Cyril Fonlupt, Sébastien Verel, Virginie Marion, Rolando Armas, et al.. Fitness landscapes analysis and adaptive algorithms design for traffic lights optimization on SIALAC benchmark. Applied Soft Computing, 2019, 85. hal-02547587

**HAL Id: hal-02547587**

**<https://hal.science/hal-02547587v1>**

Submitted on 30 Jun 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Fitness landscapes analysis and adaptive algorithms design for traffic lights optimization on SIALAC benchmark

Florian Leprêtre<sup>a,\*</sup>, Cyril Fonlupt<sup>a</sup>, Sébastien Verel<sup>a</sup>, Virginie Marion<sup>a</sup>,  
Rolando Armas<sup>b</sup>, Hernán Aguirre<sup>b</sup>, Kiyoshi Tanaka<sup>b</sup>

<sup>a</sup>*Univ. Littoral Côte d'Opale, LISIC, F-62100 Calais, France*

<sup>b</sup>*Shinshu University, Faculty of Engineering, Nagano, Japan*

---

## Abstract

Finding optimal traffic light timings at road intersections is a mandatory step for urban planners wishing to achieve a sustainable mobility in modern cities. Increasing congestion situations constantly require urbanists to enhance traffic fluidity, while limiting pollutant emissions and vehicle consumption to improve inhabitants' welfare. Various mono or multi-objective optimization methods, such as evolutionary algorithms, fuzzy logic algorithms or even particle swarm optimizations, help to reach optimal traffic signal settings. However, those methods are usually designed to tackle very specific transportation configurations.

Here, we introduce an extended version of the SIALAC benchmark, bringing together several real-world-like study cases with various features related to population, working activities, or traffic light devices. We drive a fitness landscape analysis on these various benchmark instances, which helps to improve the design of optimization algorithms for this class of real-world mobility problems. Thereby, we propose a new adaptive optimization algorithm to tackle each scenario of the benchmark.

*Keywords:* real-world, benchmark, traffic lights, fitness landscapes, optimization

---

---

\*Corresponding author.

*Email address:* [florian.lepretre@univ-littoral.fr](mailto:florian.lepretre@univ-littoral.fr) (Florian Leprêtre)

## 1. Introduction

According to the United Nations, people living in urban areas will represent sixty percent of the world's population by next decade [1]. This relentlessly growing rate involves increasing mobility challenges in modern cities, such as planning efficient transportation systems or designing reliable road infrastructure. Those challenges aim to avoid critical traffic jam situations and to deliver a satisfactory level of service in the city. However, the redesign or the extent of existing road networks implies strong constraints, such as considerable costs, lack of available space, or even environmental issues. Hence, cities target the improvement of their travelers' mobility by making a better use of their existing road infrastructure. This can be achieved, in a way, by an accurate setting of traffic light systems. Searching the proper timings for each signal helps not only to streamline urban traffic flows, but also to reduce the traveler's individual carbon footprint and pollutant emissions. Furthermore, a better tuning in signal settings might induce traffic patterns to control speed in sensitive areas, resulting in improved pedestrian security.

The pursuit of effective signal controls has motivated the development of a large amount of research which can be classified into two fields. On the one hand, offline methods consider pre-timed traffic light controls, where a fixed cycle is determined for traffic lights, according to historical traffic demand. On the other hand, online methods consider vehicle-actuated traffic light controls, where the real-time traffic information is used. **Although these last methods have shown to be effective on a small scale**, they are generally hard to apply for an entire city [2, 3, 4]. Besides, the vast majority of traffic signals **are** still regulated by fixed controls (see section 2.1). This article essentially focuses on the offline methods, as we aim to determine efficient pre-timed traffic light controls, according to various known traffic demand scenarios on a city-wide scale.

Local search techniques or evolutionary algorithms provide means to tackle such offline optimization challenges. However, depending on the configuration of the con-

sidered infrastructure and on the objectives of a planner, various optimization algorithms can be employed to reach satisfying settings. From a planner expert point of view, that is not specialist in optimization techniques, selecting a relevant optimization algorithm according to a specific urban mobility scenario might represent a difficult task. Indeed, obtained results may differ depending on the considered problem instance or the chosen algorithm. In practice, the main difficulty in conceiving robust optimization algorithms mostly relies in their tunings, whose automation is a hot topic [5, 6]. Besides, these tunings are often guided by experts knowledge, which is usually dedicated to long term perspectives and to precise situations. Therefore, even for an expert in optimization, performances of algorithms may appear difficult to compare as they are applied to heterogeneous case studies. In other words, it is not straightforward that a good tuning applied to one specific problem remains good on another, in particular for mobility challenges that constantly respond to evolving cities. Though, it appears unlikely to continuously introduce experts in changing scenarios or to keep using old tunings. Moreover, simulations of synthetic mobility plans in an agent-based model constitute a very costly and computationally expensive task, which often contradicts the limited optimization budget. This is typically the case with evolutionary related techniques, which require all the more budget as there are individuals to evolve. Therefore, urban planners are usually restricted to optimize small and specific parts of the urban area, sometimes with a restrained number of objectives.

Here, we propose an approach to automate experts knowledge so as to tackle this class of real-world, continuously changing and time-expensive problems on a city-wide scale. We identify numerous challenges, both on the side of urban planners (*e.g.* understand the city and anticipate its future changes) and on the side of optimization experts (*e.g.* comprehend variable interactions and discern how to design efficient algorithms). This work introduces an extended version of the *Scenario Investigations of Agents Localizations for Algorithm Conception* (SIALAC) Benchmark, partially presented in [7].

This broader and more complete benchmark brings together a set of synthetic mobility scenarios, from the simplest to the most sophisticated one. It provides a model to tune the population size, spatial clustering of this population and its working activities, as well as the number of traffic lights involved in the simulations. We aim to show the range of properties for the SIALAC benchmark scenarios. Thus, for algorithm design purposes, this work targets two objectives:

- An analysis of fitness landscapes to understand the search space and to improve the design of optimization algorithms,
- A new adaptive algorithm for offline traffic signal setting optimization which incorporates a reinforcement learning approach as well as the knowledge provided by the fitness landscape analysis. [The reinforcement learning method is used in order to improve the search ability of the algorithm during the optimization process. The learning approach should provide means to calibrate the inner tunings of the algorithm automatically according to the problem instance. Further, the algorithm should learn where to focus optimization efforts, in order to save valuable computation time.](#)

The rest of this article is organized as follows. In section 2, we relate existing works about mobility, optimization algorithms applied to this field and fitness landscapes analyses. Section 3 gives a formulation of the underlying optimization problem and presents basic optimization algorithms. The proposed methodology is explained in section 4. The section 5 is devoted to experiments and result analysis. After discussing these results in section 6, section 7 gives concluding remarks and future works.

## **2. Related Works**

We relate relevant background concerning mobility case studies and optimization algorithms applied to them. We also give a brief introduction about fitness landscapes

which will be employed further in this article.

### *2.1. Mobility and Optimization Problems*

Understanding city mobility represents a vast field of research. To carry out convincing studies, the majority of these works relies on traffic flow simulation systems. Modeling urban flows is not an easy task and could be approached in several ways according to the required level of detail. We discern two types of simulation models in the domain: macroscopic or microscopic ones. The macroscopic models are designed to simulate traffic flows as a whole comparable to fluid streams. On the other hand, microscopic models simulate the behavior of each individual entity in the traffic flow. Such models offer very detailed and more realistic simulation of synthetic flows, although they might suffer from high computation times.

Achieving a sustainable mobility has been studied according to various criteria in the literature. Trips duration of city travelers represents one recurrent objective to optimize that is most of the time dealt with finding proper settings for traffic signal systems. This goal is often associated with environmental perspectives, as fluidizing urban flows and avoiding congestion situations tends to reduce individual travelers' car consumption and pollutant emissions. This was shown in Garcia-Nieto *et al.* works, where the environmental impact of 250 and 500 vehicles was analyzed in Seville and Malaga city centers using the SUMO simulator [8]. Studying the level of service for public and private transportation also helps to further city mobility by designing more efficient public transports. That was pointed in the Quito city case study [9] with the MATSim simulator.

Various optimization schemes can be applied to solve these different mobility objectives. We especially focus here on optimizing traffic light problems. Because of the computationally expensive aspect of microscopic models, solving such difficult combinatorial questions requires to explore a complex search space in a few number of evaluations. Several algorithms designed to overcome these limitations are listed below

with their respective case studies.

Genetic algorithms (GAs) provide adequate means to tackle this kind of optimization problem, as they evolve a population of candidate solutions that quickly covers a wide area in the search space. GAs generally help to reach more robust signal settings than those established by experts or simulators default procedures. This was established in Chester city case study, where a flow representing about 22,000-passenger car units per hour was simulated with SATURN. This was likewise shown[10] in Medina *et al.* works, where signals optimization was performed considering a typical traffic behavior of a Saragossa district made of seven intersections [11].

Fuzzy logic controllers (FLC) have also been widely investigated to regulate traffic signal systems. FLC are essentially decisions-rule systems, whose definitions are generally based on verbally formulated instructions, which might overlap each other. This way, signal system parameters evolve dynamically using only instantaneous local information and knowledge about traffic flow or traffic density. A network made of six roads is used in the simulation and each road is assigned a random vehicle arrival rate [12].

More recently, Odeh *et al.* have coupled these latter principles to recommend a hybrid fuzzy genetic algorithm [13]. In this work, GAs were employed to select the best set of decision rules along the optimization process. One rule represents an individual, thus the algorithm evolves a population of fuzzy logic rules, resulting in flexible and intelligent signal systems. Presented results were carried out using the simulation of random vehicle flows on a model of four intersecting streets.

Genetic programming has also been investigated to regulate automated traffic signals. A recent work inspired by epigenetic modifications shows satisfactory considering dynamic traffic environments simulated with a Poisson distribution results. Signal systems programming is thus able to redesign themselves for vehicle flows on a network of nine intersections [14].

Multi-objective optimization methods can also be found in the literature. Such al-

gorithms deal with several mobility constraints and provide generally a set of optimal solutions that needs to be validated against an expert decision. These methods help to clarify the nature of the conflicts between several mobility objectives. This was established in the Quito City real-world case study, using the simulation of 20,000 agents on a 40 km<sup>2</sup> transport network area including 70 traffic lights [15].

Recently, artificial bee colony algorithms showed convincing studies to find efficient traffic light settings. The latter are based on the collective behavior of self-organized systems. These works were carried on a synthetic network made of nine intersections, undergoing 16 different signals configurations [16].

Overall, this brief outline of previous research shows that each of these works is carried out on specific problem instances, where a specific optimization algorithm is applied. Most of the time, performances of proposed algorithms are validated against expert settings and are often compared to the algorithm itself with various parameter tunings. However, this brief survey points out the lack of comparison to existing algorithms in the literature. Thus, it seems very complicated to test a new algorithm and to tell how well it performs to solve such real-world optimization problems, regardless of the study case. In the following subsection, we present a method to understand the spatial features of specific study cases in order to design appropriate optimization algorithms.

## *2.2. Fitness Landscape*

In optimization field, fitness landscape (FL) is a useful tool to describe and to analyze the geometry of the search space from the point of view of local search algorithms, such as evolutionary algorithms (EAs) or single solution based local search algorithms (simulated annealing, tabu search, etc.). FL provides a metaphorical picture of the search space (peaks, valley, plateaus, etc.) which helps to design optimization algorithms. It brings as well a portrait of the problem structure with a set of metric features,



in order to quantify and compare the search difficulty of different possible representations, local search operators, or objective functions.

Following the work in combinatorial optimization [17], a FL is formally defined by a triplet  $(\mathcal{X}, \mathcal{N}, f)$  where  $\mathcal{X}$  is the set of potential solutions,  $f : \mathcal{X} \rightarrow \mathbb{R}$  is the objective function of the optimization problem, also called fitness function in EAs and  $\mathcal{N} : \mathcal{X} \rightarrow 2^{\mathcal{X}}$  is the neighborhood relation between solutions. The neighborhood relation gives the set of neighboring solutions to each solution. To be relevant, the neighborhood relation must be related to the local search operator used in the class of the studied optimization algorithms.

A large amount of research exists on the FL analysis from the early days of EAs to recent developments. In the late eighties, the goal of researchers was to understand and characterize a problem difficulty (mainly combinatorial optimization problems) with a single metric, such as the well-known Fitness Distance Correlation [18]. More recently, with the progress of machine learning techniques, the fitness analysis is used to bring a set of numerical features in order to predict the running time of search algorithms [19, 20]. Mainly, FL analysis has been performed on artificial combinatorial benchmarks such as Traveling Salesperson Problem, Quadratic Assignment Problem, NK-landscapes [21], or more recently on artificial numerical problems [22]. Whereas FL is well suited for black-box optimizations where an analytic definition of the fitness function is not required, only few works [23] have used it on real-world problems.

In optimization, we discern two recurring shapes in fitness landscape. Flat geometries, dominated by plateaus and hilly ones, dominated by local optima. The first geometries require random walks to compute features related to the neutral networks, as graphs induced by the neighborhood relation. In the case of multimodal geometries, basins of attraction can be detected and other metrics can be defined, although they could be expensive to estimate as they are often computed by multiple hill-climbing walks [24]. Another set of features for multimodality is based on the ruggedness. The

autocorrelation of fitness introduced by Weinberger [25] defines the correlation of fitness values during a random walk. The correlation value and the autocorrelation length give a measure of the ruggedness of the landscape [26]. Besides, the accuracy of the autocorrelation coefficients is defined by  $\frac{1}{2\sqrt{\ell}}$ , where  $\ell$  is the length of the random walk [27].

Another way to quantify the ruggedness is offered by the entropy approach [28]. In this approach, the sequence of fitness values collected during a random walk is converted in a sequence of  $-1$ ,  $0$  and  $1$ :  $0$  when the difference of two consecutive fitness values is below a threshold  $\epsilon$ ,  $1$  when the difference is positive and greater than  $\epsilon$  and  $-1$  otherwise. Then, the metric is defined by the entropy of the set composed by the possible subsequences of size 2 in the sequence:

$$H(\epsilon) = - \sum_{p \neq q} P_{[pq]} \log_6 P_{[pq]}, \quad (1)$$

where  $pq$  are the possible subsequences from the set  $\{-1, 0, 1\}$  and  $P$  is their respective frequencies in the original sequence.

This metric allows estimating the ruggedness of a landscape with respect to the landscape neutrality, according to a threshold  $\epsilon$ . For a broader overview, see the review of Malan [29] on the existing fitness landscape analyzing methods.

### 3. Traffic Light Optimization Problem

Regarding the context of low-budget and time-expensive optimizations, we mainly target algorithms that converge quickly towards good solutions – the lower simulation need, the better. In last section, we noted that the sophistication of the majority of algorithms in the literature might not suit these needs. Among others, evolutionary algorithms, particle swarm optimization or even genetic programming are robust, but require to evolve a set of solutions that necessarily induces higher simulation need, thus higher budget consumption. Further, the combinatorial aspect and the extensive

neighborhood of this class of problems do not fit well with known techniques, such as differential evolution or tabu search.

In the rest of this article, we consider the context of Armas *et al.* works [15] in order to formally define the traffic lights optimization problem. We introduce the fundamentals of basic optimization algorithms and identify features that would lead to the design of fast-convergence algorithms.

### 3.1. Problem Definition

From the point of view of city mobility experts, travel duration is one of the first concrete information to interpret. During a day, the average trip duration is one of the main measures of congestion in the city, which is strongly linked to pollution issues. Most of the time, experts aim to reduce this duration by finding proper traffic lights settings.

In a traffic light optimization context, a signal system is modeled according to three properties: cycle, offset and phases. Phases represent the set of movements that can take place simultaneously in a signal system and are set independently for each road axis of the intersection (typically North-South and West-East). They basically define how long signals show green and inter-green lights along their respective axes. The sum of green and inter-green times among all phases defines the cycle time. Finally, the offset time specifies the time to wait before traffic lights start their cycle. The latter might be useful to synchronize neighboring traffic lights. Finally, a signal system  $S$  is encoded as follows:

$$S_h = (C_h, \Theta_h, \Phi_{h,1}, \dots, \Phi_{h,r}), \quad (2)$$

with  $h$  the system number,  $C_h$  the cycle time,  $\Theta_h$  the offset time and  $\Phi_{h,1}, \dots, \Phi_{h,r}$  the green times for the  $r$  road axes of the intersection (see Figure 1). These components are constrained by several bounds and equations [15].

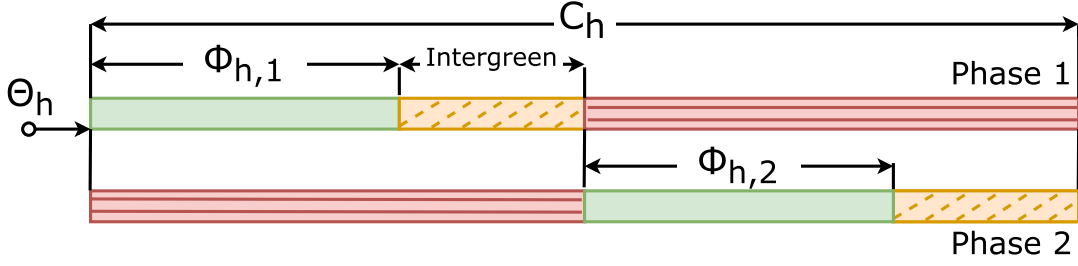


Figure 1: Representation of a signal  $h$  with two phases (two road axes).

Hence the mean trip duration of city travelers is defined as the fitness function  $f$  to minimize (see section 4.2.2). A candidate solution (*i.e.* a potential configuration for all signal systems of the city) is encoded as a vector of settings (from Equation 2):

$$x = (S_1, S_2, \dots, S_n), \quad (3)$$

with  $n$  the number of signal systems in the road network. Thus, experts must search for the optimal settings  $x^*$  that result in the lowest average trip duration:

$$x^* = \operatorname{argmin} f. \quad (4)$$

### 3.2. Mutation Operators

Mutation operators are defined in order to explore the search space of all feasible solutions  $\mathcal{X}$ . They create neighboring solutions from a candidate solution by altering its inner properties. For example, the neighbors of a solution in a bit string search space could be the set of solutions at Hamming distance 1, which is clearly related to the one bit flip mutation operator. In the case of traffic light optimization, three operators are identified: cycle time, offset time and green times mutators – each adds or withdraws a fixed number of seconds to these respective timings. Thus, mutating a solution  $x$  consists in altering one of these three properties (*i.e.* cycle, offset, or green times) of one signal system  $S$  in the solution. The choice of the used mutation operator depends on

specific mutation rates following Armas *et al.* works [30]. The rates are defined as  $P_{\text{cycle}}$ ,  $P_{\text{green}}$  and  $P_{\text{offset}}$  for cycle time, green times and offset time mutators, respectively. Generally speaking, finding good solutions in a black-box optimization context depends on the ability of the employed algorithm to explore and to exploit the search space efficiently. Exploration favors the search in the whole search space, while exploitation focuses the search around a given solution. The following subsections describe algorithm categories which emphasize exploration (random walks), exploitation (hill-climbers) or both (bandits).

### 3.3. Random walks

Random walks are typically maximum exploration algorithms. Therefore, these algorithms should not be considered on their own to find a problem's optima. Nevertheless, they provide an informative outlook on the problem's search space. A first solution is randomly generated as the starting point of the walk. At each step, *i.e.* at each iteration of the algorithm, a neighboring solution  $x$  is randomly sampled and its fitness value  $f(x)$  is evaluated (see Algorithm 1). Then, this neighboring solution is selected as the new starting point for the next step.

---

#### Algorithm 1: Random Walk

---

```

1  $x \leftarrow$  initialize random solution
2 while stopping criterion not met do
3   | compute  $f(x)$ 
4   |  $x \leftarrow$  mutate  $x$  (see section 3.2)
5 end

```

---

### 3.4. Stochastic Hill-Climbing Algorithms

Hill-climbing algorithms are typically known as maximum exploitation algorithms. In the case of stochastic hill-climbing, neighboring solutions  $x'$  are randomly sampled

and evaluated from the set of all possible neighbors, until one of them improves the currently best known solution. Then this solution is selected as the new starting point for the next iteration (see Algorithm 2). Such algorithms may get stuck in local optima, nevertheless the stochastic mutation operator processed in line 3 is likely to escape them. From an evolution strategy perspective, this algorithm is related to the  $(1 + 1)$ -ES [31, 32].

---

**Algorithm 2:** Stochastic Hill Climber

---

```

1  $x \leftarrow$  initialize random solution
2 while stopping criterion not met do
3    $x' \leftarrow$  mutate  $x$ 
4   if  $f(x') < f(x)$  then
5      $x \leftarrow x'$ 
6   end
7 end

```

---

### 3.5. Bandit Algorithms

Balancing the trade-off between exploration and exploitation is not an obvious task. Therefore, optimization algorithms are sometimes coupled with an online learning process which helps to decide whether the algorithm has to explore or to exploit the search space, depending on its previous actions.

One historic paradigm for online learning is defined as the multi-armed bandit problem [33] that involves  $k$  bandit arms with unknown reward probabilities. At each step, a player or a program selects an arm and receives a new reward. The goal is to maximize the cumulated reward gathered over all time steps  $N$ . This is equivalent to minimize the regret, defined as the loss incurred compared to the best arm. One recent way for dealing with this problem is the Upper Confidence Bound (UCB) algorithm [34]. At each step, the algorithm selects an arm  $i$  which maximizes the quantity:

$$\hat{r}_i + R \cdot \sqrt{\frac{2 \ln N}{n_i}}, \quad (5)$$

where  $\hat{r}_i$  is the average reward of the arm  $i$  and  $n_i$  is the number of times the arm  $i$  has been selected amongst  $N$  steps. This formula is based on the trade-off between exploitation and exploration, which is balanced by the  $R$  constant. Exploitation tends to select the arm with the optimal average reward, whereas exploration tends to select the arm which has been selected the most rarely.

Using such an algorithm is a well-known practice in optimization fields, especially when the optimization process can be performed thanks to various operators with unknown interest, in terms of fitness improvements. Thus we let the algorithm learn itself the operator to use amongst the portfolio and expect it will tend to select the most rewarding one along the optimization [35, 36]. Recently, Liu *et al.* use ucb algorithms to optimize a solution part by part [37]. Each part is considered as an arm of a multi-armed bandit. At each iteration of the optimization process, *i.e.* at each time a part of the solution is mutated, the incurred difference of fitness is computed as follows:

$$r = f(x) - f(x'), \quad (6)$$

where  $f$  is the fitness function,  $x$  the actual solution and  $x'$  the mutated solution. The empirical mean of these fitness differences represents the exploitation term for each arm in the ucb formula. Thereby, the algorithm tends to focus the search on the part of the solution that offers the best expected fitness improvement.

#### 4. Bandit Descent on SIALAC Benchmark

We describe a method to efficiently optimize traffic light settings. Figure 2 illustrates the main ideas of the approach with a flowchart. We introduce the SIALAC benchmark and we design the Bandit Descent algorithm.

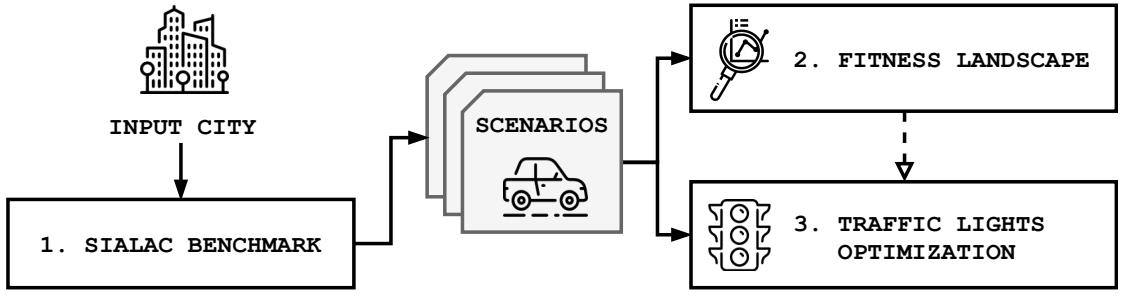


Figure 2: Methodology flowchart.

#### 4.1. Overview of the methodology

We use an extension of the SIALAC benchmark for any given city whose road network is known (step 1 in Figure 2). This benchmark provides a set of scenarios that mimics plausible traffic schemes in the city (working days, weekends, attractive poles, etc.). In other words, the benchmark grants a portfolio of scenarios which can be used as inputs in a traffic simulator. The simulator and the design of these scenarios are presented in section 4.2. Consequently, we expect to overcome the problematics established in sections 1 and 2.1 (*i.e.* understand the properties of real-world mobility problems and compare performances of diverse optimization algorithms applied to various study cases).

We analyze fitness landscapes of these scenarios (step 2) or run optimization algorithms to find adequate signal system settings (step 3) so as to tackle the diverse traffic schemes. Notice that the study of fitness landscapes is not mandatory in an optimization context. One should obviously skip this step if just on-the-go optimization is needed on a specific scenario. However, we will show in the following of this article how to use such a knowledge, which can replace an urban expert knowledge, to design an efficient optimization algorithm. This is presented in section 4.3.

#### 4.2. SIALAC Benchmark

Without a loss of generality, the following considers the road network in the city of Calais, France [38]. We briefly introduce the employed traffic simulator, we present the



design process of the proposed benchmark, and we list the various case studies involved in this work.

#### 4.2.1. Simulator and equilibrium state

The simulation system considered in this work is the Multi-agent Transport Simulation, MATSim [39] (see Appendix E for a visual example). MATSim requires as inputs a road network model and the initial mobility plans for a set of agents (*i.e.* a set of travelers' time schedules). Thus traffic flows are microscopically simulated according to agents' initial plans. These agents individually try various routes to accomplish their schedule at each iteration of the simulation, until the simulator reaches the Wardrop equilibrium state. Equilibrium state is defined as a stable condition when no traveler can improve his travel time by unilaterally changing routes. This state requires high computational needs to be achieved and often takes a duration of about an hour to be reached. As a consequence, optimization experts generally pre-calculate this equilibrium state in order to use it as a starting point for optimization purposes and thus avoid overly expensive computation times [9, 30, 40, 41].

#### 4.2.2. Road network and trip duration

Road network is the infrastructure on which agents can move around. In MATSim, the network representation consists of nodes and links. In graph theory, such representation is a static, connected and directed graph consisting of vertices and edges. Nodes can be interpreted as decision points in space and are generally located at link junctions. Links depict road segments and handle such features as length, flow capacity, number of lanes or maximum permitted speed.

In this work, we consider a mobility plan as a round trip between home and an activity (work, study, leisure). Therefore, the average trip duration  $f$  introduced in section 3.1 is defined as:

$$f(x) = \frac{1}{\#A} \sum_{a \in A} (t[M(x), a, \text{home} \rightarrow \text{activity}] + t[M(x), a, \text{activity} \rightarrow \text{home}]), \quad (7)$$

where  $x$  is a configuration for all signal systems of the city, (*i.e.* a solution, see Equation 2),  $A$  is the set of simulated agents,  $M(x)$  is the mobility simulation of all agents using the configuration  $x$ , and  $t$  is the time duration in seconds for the agent  $a$  in the simulation  $M(x)$  to accomplish a given trip. These time durations are computed from MATSim inner simulations  $M(x)$  and are mainly estimated according to the aforementioned link features. One should note agents travel at maximum permitted speed respectively to the link they drive, as long as they are not slowed by other agents or stopped by traffic signals.

#### 4.2.3. Mobility plans definition

We define home and activity clusters within the road network. Mobility plans are then synthesized according to these clusters whether we define agents' home or activity locations. A more detailed list of these plans is given in the section 4.2.4.

To populate such clusters, we first collect every node of the road network model. We assume these nodes represent our synthetic population's departure or arrival points. First, we split the network into rectangular grid cells; the finer the grid, the more precise the repartition of agents. Then we can choose a grid cell as a cluster center. Thereby, to each neighboring cell is assigned a probability to receive an agent, which decreases according to the distance from the center cell in an exponential way, according to normal distribution from the center distance. The variance of the normal distribution is the parameter that tunes the spatial spreading of the cluster on the city map. The global probability density matrix is next built considering every cluster center of the plan. Consequently, agents are assigned to a cell, and more particularly to a network node

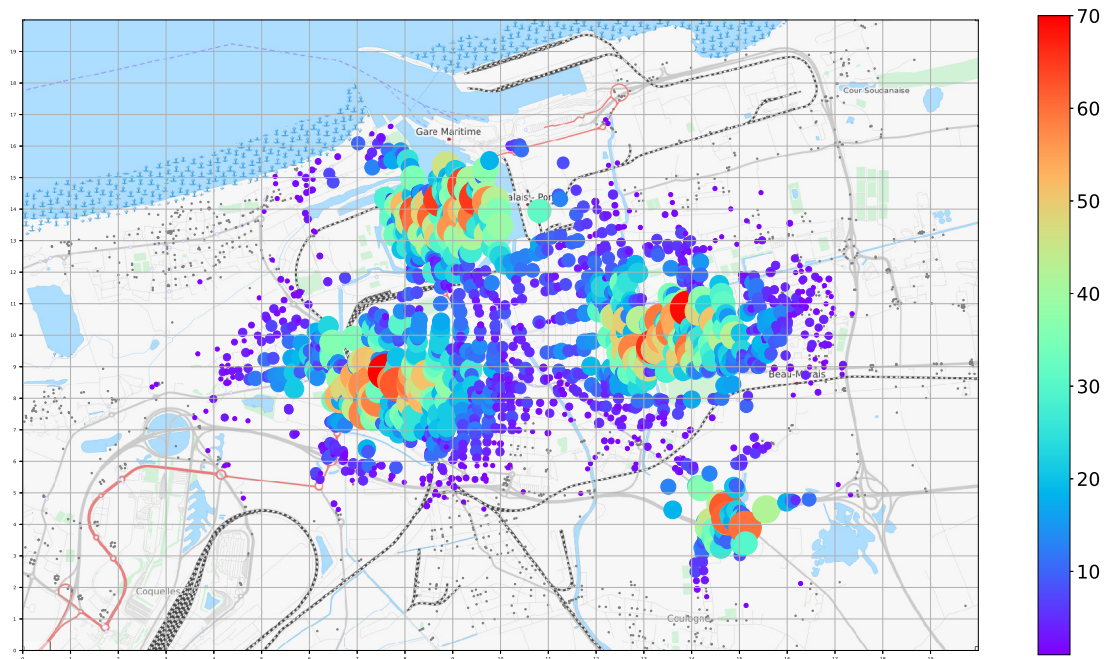


Figure 3: Four home clusters inside Calais road network. Colors indicate the number of agents departing from a node. *Colors should be used in print.*

contained in the cell, depending on the likelihood of the cell to receive an agent. To complete the mobility plan definition, agents are finally assigned a random departure time inside a time slot, as well as an activity duration. Departure times are randomly set between 7 AM and 9 AM, corresponding to typical morning rush hours. Activity duration is set to four hours. An example of four home clusters is given in Figure 3. Each cluster is centered on the main residential districts and the variance parameter is set according to the area sizes, following the expertise of the urban planners.

#### 4.2.4. Scenarios

We investigate census data provided by city hall experts of Calais [42] in order to integrate mobility plans [43, 44]. These data include such information as living quarters, main entries and exit points of the city, attractive activity sectors and exact locations of traffic light devices. Thus, we apply the agents' generation process described in the

section 4.2.3 to synthesize plausible mobility schedules and we increasingly activate signal systems involved in the simulations. These diverse case studies constitute the SIALAC benchmark scenarios involved in this work. They are tuned with four parameters, listed hereafter.

The *number of agents* synthesizes the scale of the scenarios. This parameter is critical in congestion problems and will be varied between 5000 and 20 000 agents to detect the threshold from the congestion apparition. These numbers represent respectively 6% and 24% of Calais’s population, being close to the Calais population real mobility, respectively weekends and working days.

*Home locations* are distributed following two ways: inside specific clusters, or uniformly in the city. Clusters choice allows to simulate different setups, such as mono-centric or polycentric cities. Uniform distribution will be the basic reference when no or few data are known about population distribution.

*Activity locations* are only distributed inside clusters and follow home locations scheme. The case of uniform distribution for activities in the whole city space is not relevant, as the attractive locations are usually located in specific spots.

*Signal systems* can be activated or deactivated during the simulations. The ratio of activated systems will be varied between 100%, 75% and 50% of the real count of systems in the road network. When this ratio is decreased, the most peripheral systems are deactivated, leaving only those in the very center of the city when only 50% traffic lights are involved.

Finally, 72 scenarios are created from the combination of these four parameters, summarized in Table 1.

### 4.3. Algorithm Design

We design the Bandit Descent optimization algorithm using a preliminary study of the fitness landscapes induced by the benchmark scenarios.

Table 1: Parameters involved in SIALAC scenarios.

Number of agents	$\{5, 10, 15, 20\} \times 10^3$
Home	{1 cluster, 4 clusters, uniform}
Activity	{1 cluster, 4 clusters}
Signal systems	{50%, 75%, 100%}

#### 4.3.1. Importance of a Variable

Intuitively, the importance of a variable quantifies its ability to modify the fitness value of a solution. In the case of a multi-linear regression with a normalized predictor, such importance can be defined by the coefficient value of the regression. For random forest classifiers, the importance of variables is defined by the mean decrease impurity [45]. In the context of expensive black-box optimization problems, we assume a variable is important when its mutation implies a large modification in the fitness value of the solution.

More formally, let  $\text{op}_i(x)$  be the local search operator that modifies the variable  $i$  of a solution  $x$ . At a given threshold  $\tau > 0$ , the variable is important when:

$$|f(\text{op}_i(x)) - f(x)| > \tau. \quad (8)$$

Importance of the variables can be estimated during a random walk that modifies one variable each step. In that case, a random walk is a sequence of solutions  $(x_1, x_2, \dots, x_\ell)$  where for all  $t \in [1, \ell - 1]$ , it exists  $i$  such as  $x_{t+1} = \text{op}_i(x_t)$ . The importance of a variable  $i$  is defined by the number of times that the difference of fitness between two successive solutions in the walk is important:

$$I_\tau(i) = \#\{x_t : t \in [1, \ell - 1] \text{ and } |f(\text{op}_i(x_t)) - f(x_t)| > \tau\}. \quad (9)$$

In order to measure the importance of variables across the walks, we normalize the fitness value for each random walk so as to have similar means and variances, approaching

0 and 1, respectively. For each random walk, the normalized fitness function is defined by:

$$\hat{f}(x) = \frac{f(x) - E[f(x_t)]}{\sqrt{\text{Var}[f(x_t)]}}, \quad (10)$$

where  $E[f(x_t)]$  and  $\text{Var}[f(x_t)]$  are mean and variance of the fitness values along the walk.

#### 4.3.2. *Bandit Descent*

Some traffic signals may have more impact on the urban traffic time than other ones. As a consequence, tuning the mutation operator which modifies the appropriate systems (variables) according to the importance of a signal represents a difficult task. Consequently, we propose an adaptive approach in Algorithm 3 that automatically selects the most promising signal systems to be mutated. Once they are selected, mutation operators are applied according to fixed mutation rates (see 3.2). This algorithm is inspired by Liu's works which were discussed in section 3.5.

When using a bandit strategy, two main issues have to be defined: the number of arms and the reward function. The convergence of the bandit strategy slows down with the number of arms. Then, in order to reduce the number of arms (that potentially equals the number of signal systems in the context of Liu's method), we first split all systems of the network into several groups. For example, these groups could be defined regarding the geographical positions of the systems (urban expert point of view) or regarding the importance of those systems (optimization expert perspective, see 4.3.1). Then, each of these traffic light groups is seen as an arm. Therefore, at each step of the algorithm, we use the bandit selection strategy to select a group. Finally, we mutate the settings of a number of signal systems belonging to the selected group in line 4 of Algorithm 3. The mutation operator used is stochastic, as explained in section 3.4. Unlike Liu's works, we make the hypothesis that the mutation of a group is relevant when the group

of traffic lights has a high impact on the fitness function. As a consequence, the reward of a group is defined by the absolute fitness difference incurred after each mutation:

$$r = |f(x) - f(x')|, \quad (11)$$

where  $f$  is the fitness function,  $x$  the actual solution and  $x'$  the solution after the mutation of a group. The empirical mean of this quantity defines the exploitation part for each group in the UCB formula. This way, the algorithm is encouraged to mutate the group of signals that disturbs the most the fitness function, for good or for bad. We suppose a group that alters mean trip duration the most must be critical for optimization. Thus we tend to focus mutations on this group and expect to quickly converge to a satisfying global signal system settings.

---

**Algorithm 3:** Bandit Descent

---

**Input:**  $G$  (systems groups)

```

1  $x \leftarrow$  initialize random solution
2 while stopping criterion not met do
3    $g \leftarrow$  select best group in  $G$ 
4    $x' \leftarrow$  mutate  $g$ 
5   if  $f(x') < f(x)$  then
6      $x \leftarrow x'$ 
7   end
8   update rewards
9 end

```

---

## 5. Experiments

We present the experimental protocol we employ. We investigate the fitness landscape structures in various ways and finally study performances of the previously intro-

duced optimization methods.

### 5.1. Experimental Setup

We run whole 72 SIALAC scenarios on the Calais city road network model, made of 2720 nodes and 6049 links, resulting in a simulated area of 35 km<sup>2</sup>. The Wardrop equilibrium state is reached for each scenario without prior consideration of traffic lights (see section 4.2.1). Then we specify the positions of 33 intersections with traffic lights (see Appendix D) and the pre-calculated equilibrium states are used as starting points for further simulations. Mutation rates are fixed to 50%, 30% and 20% for  $P_{\text{cycle}}$ ,  $P_{\text{green}}$  and  $P_{\text{offset}}$ , respectively.

### 5.2. Trip Duration

In order to study fitness landscape structures, we run 20 random walks (see Algorithm 1) of 50 steps each on every case study of the benchmark, so as to achieve an accuracy of 10 percent for the calculation of statistical metrics in the following section. At each step of these walks, we randomly mutate a unique signal system and observe the evolution of the global fitness along the walk. Figure 4 shows median travel durations computed from the sample of solutions provided by the random walks, for each benchmark scenario.

Let us explain these charts. As expected, travel times thus congestion phenomena increase with the number of agents and the number of activated signal systems. Nevertheless, there is no significant difference between median travel times when 50% or 75% traffic lights are activated, which suggests signal systems in the center of the city have a strong hold on trip durations.

Furthermore, population spreading as well as activity locations have an obvious impact on the journey time. To ease the interpretation of these charts, we can divide the six scenarios of each agent number values into two groups: one called  $G_1$  containing observations for one activity area and one or four home clusters and the other one called



$G_2$  containing the four other scenarios. Whatever the number of agents is, the lowest travel time is observed for  $G_1$  scenarios. This may be explained since these corresponding situations are less conducive to crossovers, as each agent heads towards the same direction.

The greater the number of agents, the more we note important differences among  $G_2$  scenarios. The highest median travel time generally happens when all agents are assigned to one home cluster and head towards four activity zones; this could be explained by a potentially increased congestion for this specific configuration. On the other hand, uniformly distributed homes and one activity cluster appear to generate more traffic jam when few signal systems are involved, even though this configuration generally shows the lowest median times in  $G_2$  observations. Overall we note that four activity areas mostly increase the travel duration, while four home clusters against uniform distribution tend to decrease it.

These observations can lead city experts to various interpretations. Obviously, the number of travelers in the city impacts the average trip duration. Yet, some congestion phenomena do not appear when the number of travelers is small (*e.g.* with 5000 agents, the most conducive scenario to congestion is unclear). Then, more activity clusters seem to increase the average journey time, as more congestion situations might occur in different parts of the city. Finally, home clusters have less impact on trip duration than activity ones, but spreading the population into a few clusters (but not only one) might reduce travel time.

Thus, from an urban planner point of view, these results seem different in terms of trip duration from one scenario to another and give useful information for future possible plan of the city. However, these different trip durations (*i.e.* different responses of the fitness function) on the various scenarios may not imply different difficulties in solving the underlying optimization problems. That will be discussed in the next section.

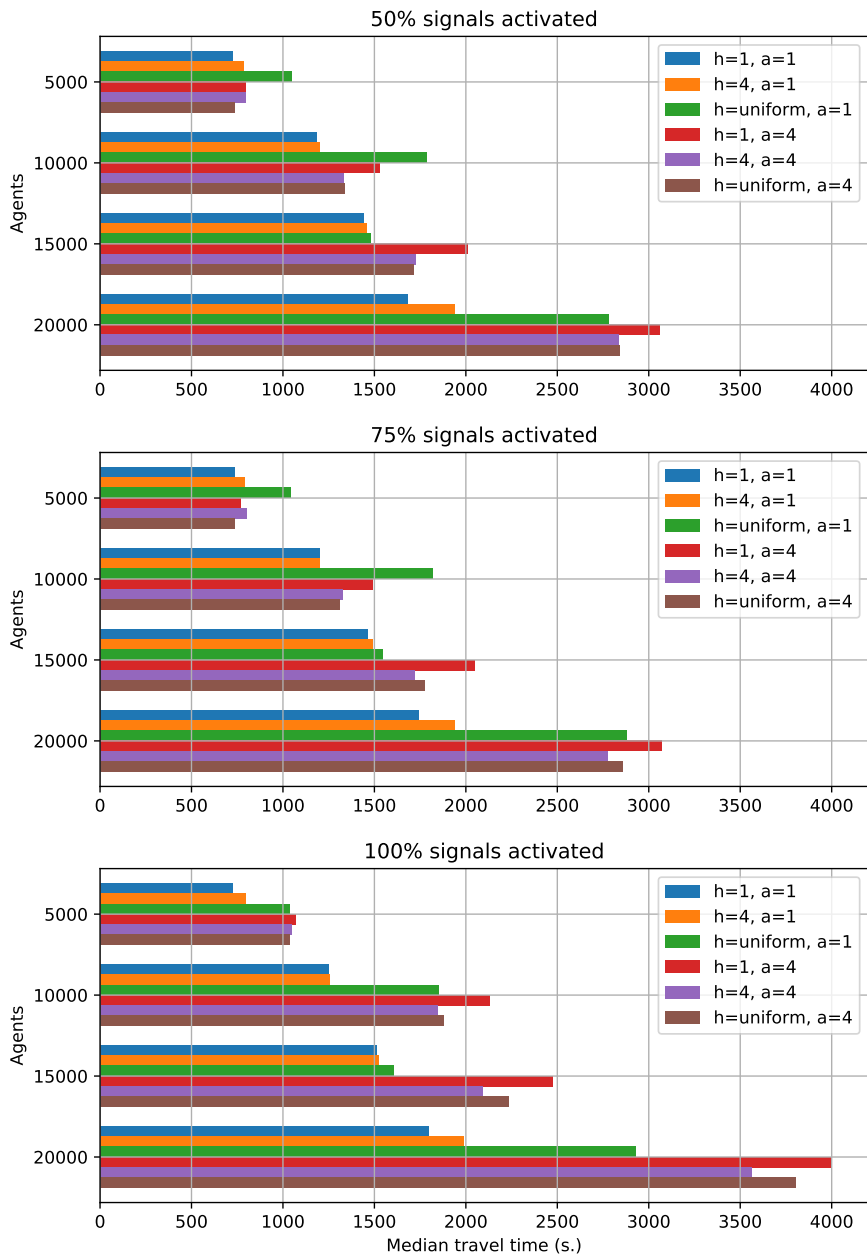


Figure 4: Median travel duration (in seconds) for SIALAC benchmark scenarios computed on the sample of solutions given by the random walks ( $h$ : home clusters,  $a$ : activity clusters). Colors should be used in print. Otherwise, bars are ordered as the legend.

### 5.3. Ruggedness

From the random walks computed on the different scenarios, ruggedness of landscapes can be estimated using the autocorrelation of fitness (ACF), or by evaluating the maximum entropy of *Vassilev*'s sequences ( $H_{\max}$ ) according to several  $\epsilon$  values, sampled in  $[0, 10]$  (see section 2.2). Results of these metrics are reported in Table 2.

Surprisingly, mean autocorrelation lengths and maximum entropies are very similar: for the 72 scenarios, the lengths vary between 7 and 10 and there is no significant difference according to a Mann-Whitney test at level 5%. Although the lengths are short and indicate a rugged landscape, the difficulty of underlying optimization problems is analogous according to ruggedness. This means that favoring a class of optimization algorithms over another to tackle such a problem is not an obvious task. This observation is reinforced by the maximum entropy measures that are capped to 0.84 for each scenario.

Indeed, Figure 5 (left) shows examples of random walks on two different scenarios. The walks look similar for every scenario. The fitness displays sudden rapid *jumps* upwards or downwards. This suggests some variables seem more important than others, as their mutations lead to abrupt changes in the fitness function. Then, we analyze the importance of variables across the scenarios.

### 5.4. Importance of Variables

Figure 5 (right) shows the absolute value of the difference  $\delta_t = \hat{f}(x_{t+1}) - \hat{f}(x_t)$  across all walks over all scenarios. In logarithmic scale, the distribution is unimodal. All the differences  $\delta$  are similar across the scenarios, but the tail of the distribution for high value of  $|\delta_t|$  is large. This allows us to define a meaningful threshold to detect the importance across the scenarios. The threshold  $\tau$  is defined so that 10% of the  $|\delta_t|$  are considered to be a significant difference.

Table 3 shows the average and the standard deviation of importance for each variable, *i.e.* for each road intersections, across all scenarios. We deduce signal systems

Table 2: Mean autocorrelation lengths (ACF) and maximum entropies of *Vassilev's* sequences ( $H_{\max}$ ) for each SIALAC benchmark scenarios, computed on the sample of solutions given by the random walks, with 100% signal systems activated ( $h$ : home clusters,  $a$ : activity clusters).

Agents	$h$	$a$	ACF	$H_{\max}$	Agents	$h$	$a$	ACF	$H_{\max}$
5000	1	1	8.4	0.81	15 000	1	1	8.7	0.84
		4	7.3	0.82			4	8.8	0.81
	4	1	7.4	0.81		4	1	8.4	0.83
		4	7.2	0.82			4	7.9	0.81
	uniform	1	8.9	0.82		uniform	1	7.7	0.83
		4	8.1	0.79			4	8.3	0.82
10 000	1	1	8.2	0.83	20 000	1	1	7.3	0.84
		4	8.0	0.82			4	8.1	0.85
	4	1	9.2	0.82		4	1	8.8	0.83
		4	8.5	0.82			4	7.9	0.82
	uniform	1	9.2	0.84		uniform	1	8.6	0.84
		4	8.5	0.84			4	7.6	0.83

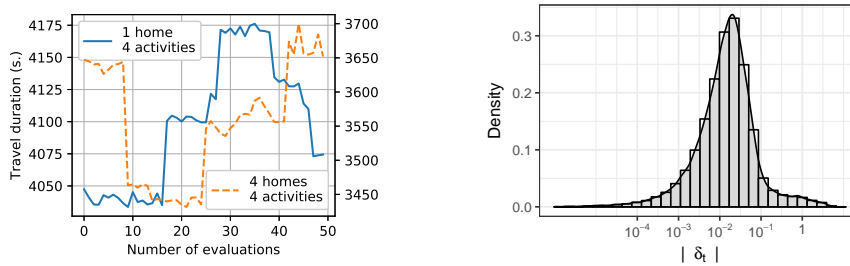


Figure 5: Examples of random walks for 20.000 agents and two study cases (left). Distribution of  $|\delta_t|$  where  $\delta_t = \hat{f}(x_{t+1}) - \hat{f}(x_t)$  is the difference of normalized travel duration (right). Notice  $x$ -log scale.

have high importance when the average is greater than 2 (signal systems with id. 2, 8, 11, 13 and 29 among others), or little importance regardless the scenario when it is lower than 1 (signal systems 15, 16, 30, 31 and so on). However, the standard deviation values are high and often larger than the average values, which suggests that the important signal systems are not the same regarding the scenarios.

Besides, we stated in section 5.2 there is no noticeable difference in median travel durations between scenarios involving 50% and 75% of traffic lights in the city. Actually, this observation can be explained by the addition of low importance traffic lights to move from 50% to 75% scenarios. On the other hand, moving to scenarios involving 100% of traffic lights in the city requires to add peripheral signal systems with high importance (such as those with id. 22, 29), thus implying a noticeable increase in travel durations.

Figure 6 shows the mean importance according to the main parameters of the benchmark. For example, on top, the mean importance is computed for all scenarios from 5,000 to 20,000 agents. Without going into the many details, importance of an intersection can change from one scenario to another one. For example, importance of intersection with id. 10 is almost 4 for 5,000 agents and less than 2 for 15,000 agents. The benchmark parameter which impacts the most the importance of signals is the number of activity clusters. For example, for the intersection with id. 29, importance drops from 6.2 with four clusters of activity, to nearly 0 with one cluster. The ruggedness and the difficulty tend to be similar across the benchmark scenarios, but the variables (signal systems) that impact the most the fitness function change according to the benchmark parameters. Nevertheless, signal systems with high average importance across all scenarios (deduced from Table 3) are clearly noticeable on these charts (*i.e.* similar peaks for signals with id. 11, 13, 14, or 29).

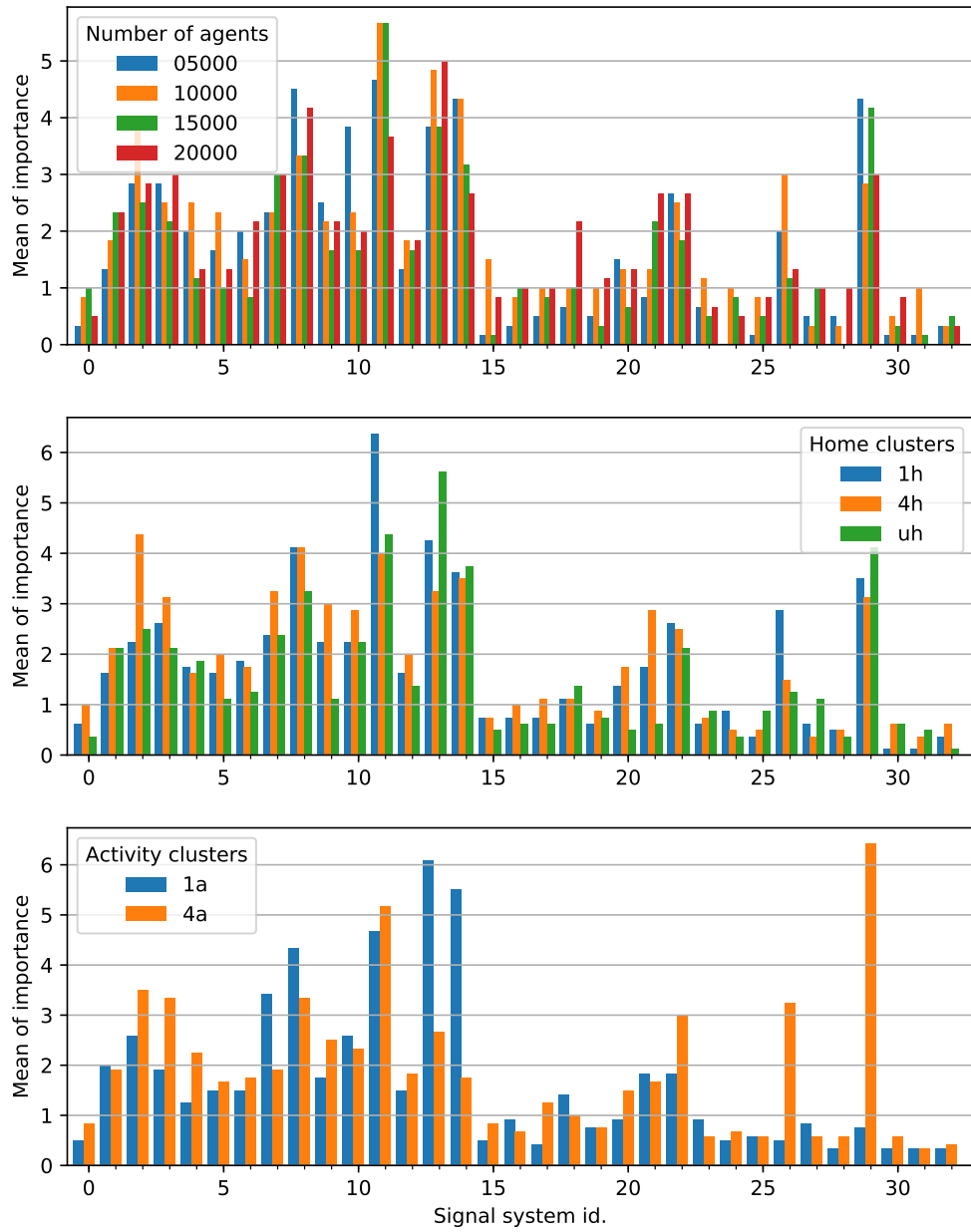


Figure 6: Mean signal system importance according to the benchmark parameters, with 100% signal systems activated. *Colors should be used in print. Otherwise, bars are ordered as the legend.*

Table 3: Average ( $\mu$ ) and standard deviation ( $\sigma$ ) of traffic signal importance over all benchmark scenarios with 100% signal systems activated.

id.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\mu$	0.67	1.96	3.04	2.62	1.75	1.58	1.62	2.67	3.83	2.12	2.46	4.92	1.67	4.38	3.62	0.67	0.79
$\sigma$	0.75	1.31	1.84	1.65	1.33	1.04	1.15	1.7	1.4	1.72	1.53	2.77	0.9	2.43	2.53	0.99	0.96
id.	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
$\mu$	0.83	1.21	0.75	1.21	1.75	2.42	0.75	0.58	0.58	1.88	0.71	0.46	3.58	0.46	0.33	0.38	
$\sigma$	1.07	0.96	0.83	1.	1.56	1.41	0.83	0.91	0.64	2.15	0.84	0.76	3.3	0.64	0.75	0.56	

### 5.5. Performances of Optimization Algorithms

We now search for high-quality traffic light settings. In the context of time-consuming simulations, the less the simulator is requested, the better the optimization algorithm. In this regard, an efficient optimization algorithm must converge to satisfying signal settings in a minimum of optimization budget. Therefore, we assume the settings found will be local most of the time. However, in a black-box and computationally expensive context, we are not primarily seeking the globally optimum settings; at least we target to reach optima in the dedicated optimization budget.

Therefore, we compare the performances of five optimization algorithms: a stochastic hill-climbing algorithm (see Algorithm 2), a simple elitist evolutionary algorithm from [30] and three bandit-based algorithms which differ on the way their arms and rewards are defined (from equation 5). The first bandit algorithm considers the context of Liu’s works, which has been outlined in section 3.5; in that respect, there are as many bandit arms as there are signal systems in the road network. However, we aim to reduce this potentially high number of arms in a low-budget and fast-convergence perspective. Moreover, some signal systems appear to be more critical than others in terms of optimization according to the previous analyses. Thus we rely on this information to design signal system groupings (*i.e.* bandit arms) for the proposed Bandit Descent (see Algorithm 3). Two ways to establish these groups are envisaged. As a planner would probably do, the first way considers three geographically-based groups which gather

neighboring signal systems in the city space (*e.g.* signal systems belonging to the same road axis). The second way collects signals in three groups according to their importance, deduced from the previous analyses: critical (importance larger than 2), moderate (between 1 and 2) and quasi-neutral signal systems (below 1).

Each of these five optimization algorithms are run 40 times on every benchmark scenarios involving 5000 agents. One single run considers a budget of 2000 simulator evaluations and requires about 15 hours of computation on a single-threaded machine (see Appendix C). We select four random signal systems for the mutation phases, following Armas *et al.* works [15, 30] in a way to ease comparisons and discussion in section 6. For bandit algorithms, the selection strategy determines a group to be mutated, then four signal systems belonging to the latter are mutated. The exploration parameter  $R$  is arbitrarily set to 2 (see Equation 5).

Figure 7 presents the obtained mean fitness evolution for each of the scenarios according to the evaluation budget. Notice that in this case, one evaluation corresponds to one request to the simulator. [Results reported for the evolutionary algorithm consider the best-of-generation's fitness.](#) At a first look, every optimization algorithm basically converges towards the same setting quality (respectively to the scenario they are applied on, see Appendix B), though with different speed. In a low-budget context, the evolutionary algorithm appears as the less competitive one: such algorithm requires to evolve a set of solutions, therefore it uses even more simulator calls before reaching satisfying settings as there are more individuals in the population. A stochastic hill-climbing performs generally well, but our results show that grouping signal systems according to their importance, in terms of fitness importance, mostly leads to superior quality settings since the very first evaluations. On the other hand, geographical groupings are surprisingly often as effective as a stochastic hill-climbing, or even worse regarding the *uh-1a* scenario. Defining an arm for each signal system slows down the convergence speed, as illustrated by *BD-full* in the charts. This is because the algorithm requires to



explore too many groups before it can determine where to exploit efficiently.

The fast convergence of the Bandit Descent based on the importance of variables is highlighted in Table 4. Here, we assume that the simulator response is accurate to 10 percent. Then, we compute the average evaluations needed for the algorithms to converge to 10 percent of the best settings found all runs combined (accordingly to each scenario). The gain of the adaptive algorithm over a stochastic hill-climbing or an elitist evolutionary algorithm is confirmed by Kruskal-Wallis tests, showing the statistical difference of the settings found in the early evaluations. One should note that the algorithms work in an equivalent computation time, which is about 10 000 times smaller than the time allocated to simulation (see Appendix C). This reinforces our arguments for developing efficient low-evaluation-consumption algorithms in this low-budget context. Nevertheless, the efficiency of the Bandit Descent is noticeable to a certain extent, depending on the case study. For example, the gain in using the proposed method on the *4h-1a* scenario is less significant. Another example can be noted on the *uh-1a* scenario, where the gain of the online-learning method is clearly noticeable during the very first evaluations. These observations may be explained by the choice of the exploration parameter  $R$ . This will be discussed in the next section.

## 6. Discussion

The pertinence of the presented benchmark and algorithms is discussed. We return to the parameterization of the Bandit Descent and to the choice of traffic light groupings. We deepen the analysis of the traffic light settings found in the previous section. Finally, we conduct the whole methodology of this article on a different real-world mobility case study.

### 6.1. Selection Strategy and Traffic Light Groupings

As stated in the last section, some performances may appear similar given a mobility scenario and an algorithm setting. This similarity can be explained by the choice of the

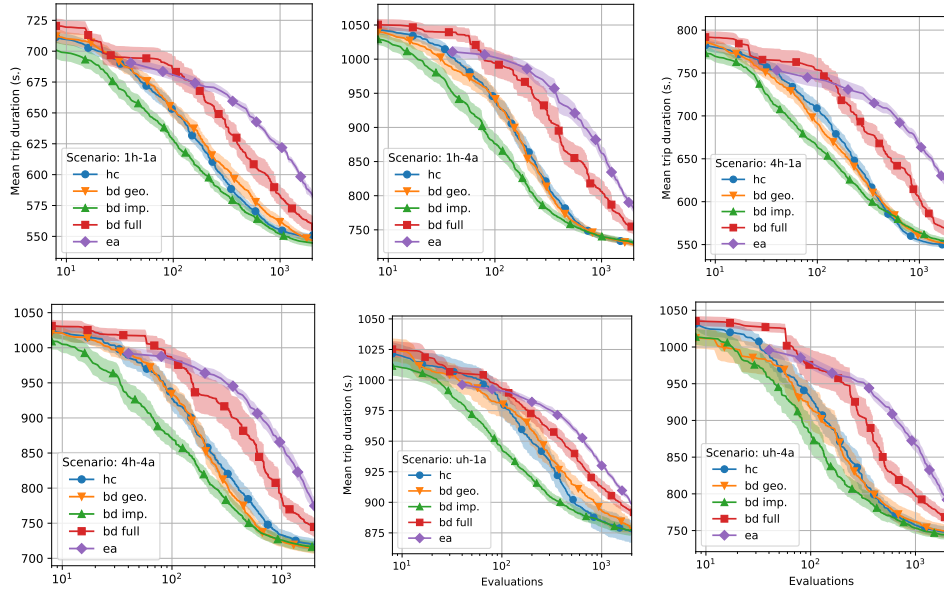


Figure 7: Minimization of mean trip duration for 5000 agents and 100% signal systems activated ( $h$ : home clusters,  $a$ : activity clusters,  $u$ : uniform repartition). Five optimization algorithms are compared : stochastic hill-climbing (hc), **best-of-generation of an evolutionary algorithm (ea)** and Bandit Descents ( $bd$ ) with geographical ( $geo.$ ), importance-based ( $imp.$ ) or *full* groupings. The exploration parameter is set to  $R = 2$  for ucb algorithms. Notice  $x$ -log scale.

Table 4: Average evaluations (simulator calls) to reach good quality settings at precision 10%, for stochastic hill-climbing (hc), elitist evolutionary algorithm (EA) and importance-based Bandit Descent (BD) algorithms. Values in bold are statistically different according to Kruskal-Wallis H-test at level 5%.

Scenario	Evaluations		
	HC	EA	BD
1h-1a	246	1400	<b>165</b>
1h-4a	362	1680	<b>236</b>
4h-1a	400	2000	<b>315</b>
4h-4a	372	1600	<b>218</b>
uh-1a	146	280	<b>63</b>
uh-4a	308	1240	<b>196</b>

exploration parameter  $R$  in the `ucb` strategies, which is probably maladjusted compared to some scenarios. In Figure 8, we study further the *uh-1a* scenario to show that the  $R$  value can be refined so as to drive to different algorithm behaviors. A small value certainly leads to a fast minimization during the very first iterations, but risks to get stuck in local optima (*i.e.* too much exploitation). However, the higher the  $R$  value, the greater the similarity with a stochastic hill-climbing (*i.e.* too much exploration). Thus, the benefit of the proposed adaptive algorithm becomes more noticeable when it is adequately set up. Moreover, we use a very basic `ucb` strategy without windows management nor restart strategy, which might improve the efficiency for long running times.

Furthermore, at equivalent settings for the selection strategies, we show that geographical groupings lead most of the time to poorer performances than importance-based groupings and sometimes than a stochastic hill-climbing. Although it may seem unnatural to a planner *prima facie*, our methodology to detect and to cluster critical traffic lights in the city seems to be more effective to reach good quality settings. Actually, it is not so much the selection strategy that leads to good performances, but rather the way in which groups are constituted in advance.

## 6.2. Relevance of Settings

Due to the expensive simulation times, the high number of settings to be evaluated for each optimization algorithm and the numerous mobility scenarios to study, one might wisely think to take advantage of the various benchmark scenarios to speed up the optimization algorithms. Assuming that some scenarios share the same features (*e.g.* number of home or activity clusters), solutions found on a specific benchmark scenario might be envisaged to tackle similar scenarios. In other words, we should not start the optimization algorithm from totally random settings, but rather from superior solutions discovered in similar benchmark scenarios.

Table 5 lists the average number of iterations needed to reach a good quality solution

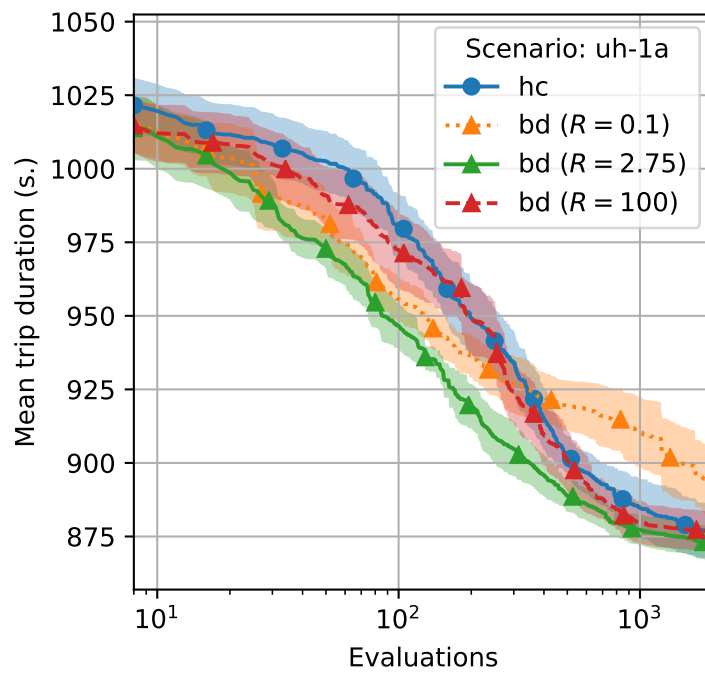


Figure 8: Varying exploration parameter  $R$  and associated importance-based Bandit Descent performances (BD), on a specific benchmark scenario. Stochastic hill-climbing optimization (HC) stands as a baseline reference.

Table 5: Mean iterations needed to reach good solutions when the search is initialized with the best settings from another scenario. The  $\infty$  symbol stands when the search could not reach a good quality solution.

From \ To	<i>1h-1a</i>	<i>1h-4a</i>	<i>4h-1a</i>	<i>4h-4a</i>	<i>uh-1a</i>	<i>uh-4a</i>
<i>1h-1a</i>	–	$386.4 \pm 32.5$	$355.1 \pm 44.8$	$349.8 \pm 35.3$	$45.3 \pm 14.1$	$379.3 \pm 26.1$
<i>1h-4a</i>	$\infty$	–	$\infty$	$0 \pm 0$	$\infty$	$0 \pm 0$
<i>4h-1a</i>	$25.3 \pm 8.0$	$\infty$	–	$\infty$	$161.1 \pm 51.1$	$\infty$
<i>4h-4a</i>	$157.7 \pm 24.8$	$40.0 \pm 16.9$	$209.3 \pm 28.3$	–	$292.0 \pm 69.4$	$0 \pm 0$
<i>uh-1a</i>	$268.5 \pm 76.0$	$400.6 \pm 20.3$	$404.6 \pm 52.6$	$325.4 \pm 19.7$	–	$338.7 \pm 27.9$
<i>uh-4a</i>	$323.8 \pm 66.5$	$44.3 \pm 19.7$	$326.2 \pm 62.9$	$0 \pm 0$	$401.0 \pm 50.0$	–

over 40 runs of a stochastic hill-climbing optimization (see Figure 7 and Appendix B), when the algorithm is initialized with the best solution found from another scenario. As a reminder, it takes about a thousand evaluations to reach such a good solution when the algorithm starts from random settings in section 5.5. Instead, one should clearly benefit from superior settings found in prior optimizations to tackle other scenarios, especially if these last share same features. Figure 9 illustrates an example of this behavior: initializing the search from optimized settings mostly leads to a good fitness value from the very first iterations. Further, it often takes some iterations more to converge towards a high quality solution set. Yet, some settings might mislead the optimization algorithm for long running times and slow down its convergence, especially when benchmark scenarios are dissimilar.

To deepen the discussion, we inject the best signal settings found from scenarios involving 5000 agents into the same scenarios with the number of agents increased to 20000 – that take more than twice as long to simulate. As stated in Table 6, we show once again that it is worthwhile in most cases to start the minimization from the best solutions found when a lower number of agents is involved. These solutions still perform well when more agents are simulated, thus saving precious computation time

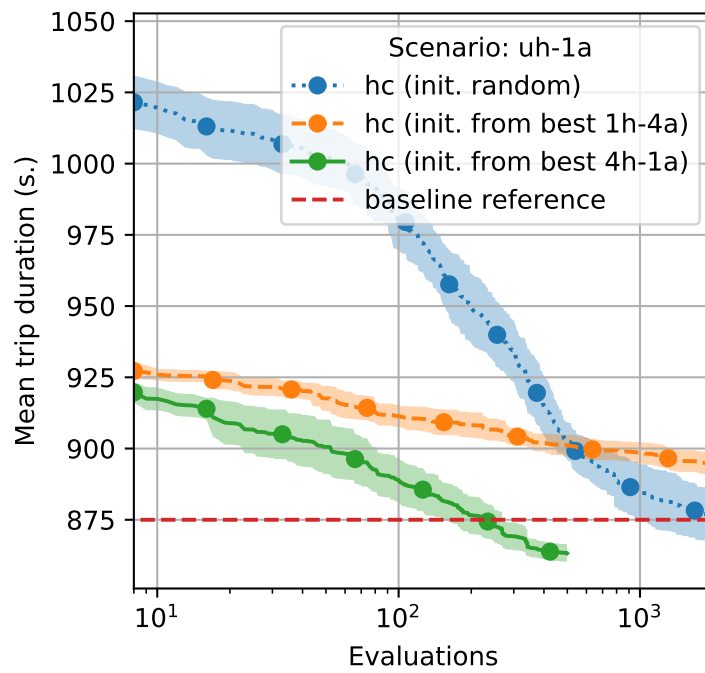


Figure 9: Minimization of mean trip duration for the *uh-1a* scenario and 5000 agents. Stochastic hill-climbers (hc) are initialized randomly or from best known settings.

Table 6: Mean iterations needed to reach a good solution for scenarios involving 20 000 agents. The search is initialized with the best settings found from the same scenario and 5000 agents.

<i>1h-1a</i>	<i>1h-4a</i>	<i>4h-1a</i>	<i>4h-4a</i>	<i>uh-1a</i>	<i>uh-4a</i>
$407.2 \pm 110.8$	$\infty$	$332.1 \pm 83.9$	$287.7 \pm 54.3$	$291.4 \pm 51.5$	$303.1 \pm 51.2$

in reaching superior settings.

As concluding remarks, this brief study shows that the best signal system settings found from our experiments in section 5.5 appear to be relevant when they are confronted with the various SIALAC scenarios. Therefore, one should benefit from his prior optimizations when looking for good quality traffic light settings in new – albeit similar – mobility scenarios. Moreover, this might suggest to consider multi-objective heuristics to tackle several benchmark scenarios in the same optimization process, so as to reach very robust settings.

Finally, we would like to point out that the quality of optimized settings found in this work – regarding real ones actually applied in Calais city – is not so obvious to estimate, and is left for future works. Actually, the proposed methodology does not stand for this purpose, but instead provides planners an unusual outlook on their city. We aim to open reflexion thanks to both optimization experts and urban experts’ points of view, so as to research for the most efficient traffic light settings.

### 6.3. Validation of the Benchmark

This first research step opens many perspectives from real-world mobility applications to robust optimization techniques. The SIALAC benchmark is an open benchmark that can be extended with various city maps, or more precise travel plans, among others. It can be reused by a larger research community to draw a more consistent comparison between several optimization algorithms and potentially improve them.

As a starting point for future works, we experience our methodology on Quito city, Ecuador, from [30]. Utilizing the various benchmark scenarios and their analyses (see

Appendix A), the importance-based Bandit Descent manages once again to find high-quality traffic light settings (see Figure 10). These brief results, computed according to the same conditions as in section 5.5, are only taken from two specific benchmark scenarios, but show encouraging results in generalizing our method to various cities around the world.

Ideally, given a city, a planner can simulate various mobility scenarios using the SIALAC benchmark, then analyze these last to extract critical crossroads, so as to ease the search for good quality traffic light settings. If, at some future, traffic conditions are to change (*e.g.* opening of new commercial areas, relocation of companies, etc.), then city experts should reiterate the proposed methodology to understand the new traffic dynamics and hence recalibrate the traffic light controls accordingly. The offline optimization process appears to be a relevant choice when it is necessary to improve traffic conditions with existing road infrastructures, awaiting the transition to more state-of-the-art technologies, such as intelligent and traffic-actuated traffic lights. Moreover, the proposed optimization methodology could be used with online intelligent system for which some meta-parameters need to be set. Indeed, the meta-parameters could be tuned at the city scale with the same methodology by replacing the direct parameters of the traffic lights by the potential meta-parameters of the controller (thresholds, states representation, reward functions, etc.).

## 7. Conclusion

In this work, we present an extension to the SIALAC benchmark for real-world mobility problems; 72 scenarios regrouping various mobility features are proposed and studied for Calais city, France. With a fitness landscape analysis, we show that the global structure of the search space is similar and rugged for all scenarios. We propose a methodology based on random walks to detect the important variables in the optimization problems.



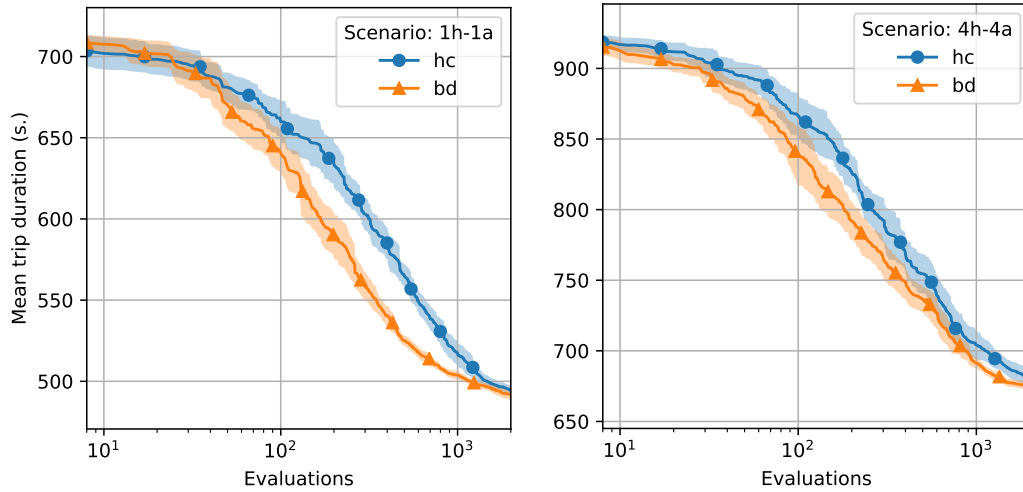


Figure 10: Minimization of mean trip duration for 20 000 agents and two benchmark scenarios on Quito city. Stochastic hill-climbing (hc) and importance-based Bandit Descent (bd) are compared. Exploration parameter is set to  $R = 1$ .

This information helps us to design an adaptive optimization algorithm based on an reinforcement learning strategy, the Bandit Descent, in order to tackle scenarios with various features. Our experiments indicate that reinforcement learning coupled with the detection of important variables seems to potentially reach faster high quality solutions across a number of scenarios, and require less urban expert knowledge.

Beside the analysis of fitness landscapes, the adaptive algorithm based on ucb selection strategy can be improved with existing restart strategies, time windows and a more robust scaling of the exploration constant. Moreover, highlighting the importance of variables during a random walk opens a path to incorporate an adaptive detection of important variables within the search process. The new Bandit Descent pushes to design an efficient mimetic algorithm that combines the proposed local search with a smart crossover, that could also be based on the variable importance, in parallel environment of computation. [Further, a complexity analysis of the Bandit Descent is envisaged on simplified problem instances in order to show the theoretical improvement of the technique.](#)

Finally, developing a surrogate model to speed up traffic flow simulations is intended thanks to the plenty of data induced by the diversity of the benchmark scenarios.

### **Acknowledgment**

We are grateful to Interdisciplinary Graduate School of Science and Technology, Shinshu University (Nagano, Japan) for the support and informed advices.

We thank Calais city (France) for the data and its support.

Experiments presented in this paper were carried out using the CALCULCO computing platform, supported by SCOSI / ULCO (Service COMMun du Système d'Information de l'Université du Littoral Côte d'Opale).

### **References**

- [1] The world's cities in 2016, United Nations, Department of Economic and Social Affairs (2016).
- [2] H. Wei, G. Zheng, H. Yao, Z. Li, Intellilight: A reinforcement learning approach for intelligent traffic light control, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM, 2018, pp. 2496–2505.
- [3] S. S. Mousavi, M. Schukat, E. Howley, Traffic light control using deep policy-gradient and value-function-based reinforcement learning, IET Intelligent Transport Systems 11 (7) (2017) 417–423.
- [4] J. Gao, Y. Shen, J. Liu, M. Ito, N. Shiratori, Adaptive traffic signal control: Deep reinforcement learning algorithm with experience replay and target network, arXiv preprint arXiv:1705.02755 (2017).
- [5] H. H. Hoos, Automated Algorithm Configuration and Parameter Tuning, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 37–71.

- [6] M. López-Ibáñez, J. Dubois-Lacoste, L. P. Cáceres, M. Birattari, T. Stützle, The irace package: Iterated racing for automatic algorithm configuration, *Operations Research Perspectives* 3 (2016) 43 – 58.
- [7] F. Leprêtre, C. Fonlupt, V. Marion, S. Verel, Sialac benchmark: On the design of adaptive algorithms for traffic lights problems, in: *The Genetic and Evolutionary Computation Conference*, Kyoto, Japan, 2018.
- [8] J. Garcia-Nieto, J. Ferrer, E. Alba, Optimising traffic lights with metaheuristics: Reduction of car emissions and consumption, in: *2014 International Joint Conference on Neural Networks (IJCNN)*, 2014, pp. 48–54.
- [9] R. Armas, H. Aguirre, K. Tanaka, Multi-objective optimization of level of service in urban transportation, in: *Proceedings of the Genetic and Evolutionary Computation Conference*, ACM, 2017, pp. 1057–1064.
- [10] F. Teklu, A. Sumalee, D. Watling, A genetic algorithm approach for optimizing traffic control signals considering routing, *Computer-Aided Civil and Infrastructure Engineering* 22 (1) (2007) 31–43.
- [11] J. J. Sanchez-Medina, M. J. Galan-Moreno, E. Rubio-Royo, Traffic signal optimization in La Almozara district in Saragossa under congestion conditions, using genetic algorithms, traffic microsimulation and cluster computing, *IEEE Transactions on Intelligent Transportation Systems* 11 (1) (2010) 132–141.
- [12] S. Chiu, S. Chand, Adaptive traffic signal control using fuzzy logic, in: *The First IEEE Regional Conference on Aerospace Control Systems*,, 1993, pp. 122–126.
- [13] S. M. Odeh, A. M. Mora, M. N. Moreno, J. J. Merelo, A hybrid fuzzy genetic algorithm for an adaptive traffic signal system, *Adv. Fuzzy Sys.* 2015 (2015).

- [14] E. Ricalde, W. Banzhaf, A genetic programming approach for the traffic signal control problem with epigenetic modifications, in: *Genetic Programming*, Springer, 2016, pp. 133–148.
- [15] R. Armas, H. Aguirre, S. Zapotecas-Martínez, K. Tanaka, Traffic signal optimization: Minimizing travel time and fuel consumption, in: *Artificial Evolution*, Springer, 2016, pp. 29–43.
- [16] K. Gao, Y. Zhang, A. Sadollah, R. Su, Improved artificial bee colony algorithm for solving urban traffic light scheduling problem, in: *2017 IEEE Congress on Evolutionary Computation (Congress on Evolutionary Computation)*, 2017, pp. 395–402.
- [17] P. F. Stadler, Fitness landscapes, in: *Biological Evolution and Statistical Physics*, Vol. 585 of LNP, Springer, 2002, pp. 187–207.
- [18] T. Jones, Evolutionary algorithms, fitness landscapes and search, Ph.D. thesis, University of New Mexico, Albuquerque (1995).
- [19] M. A. Muñoz, Y. Sun, M. Kirley, S. K. Halgamuge, Algorithm selection for black-box continuous optimization problems: A survey on methods and challenges, *Information Sciences* 317 (2015) 224–245.
- [20] F. Daolio, A. Liefooghe, S. Verel, H. Aguirre, K. Tanaka, Problem features versus algorithm performance on rugged multiobjective combinatorial fitness landscapes, *Evolutionary Computation* 25 (4) (2017) 555–585.
- [21] H. Richter, A. Engelbrecht, *Recent Advances in the Theory and Application of Fitness Landscapes*, Springer, 2014.
- [22] P. Kerschke, M. Preuss, S. Wessing, H. Trautmann, Low-budget exploratory land-

- scape analysis on multiple peaks models, in: Proceedings of the Genetic and Evolutionary Computation Conference 2016, ACM, 2016, pp. 229–236.
- [23] M. Muniglia, S. Verel, J.-C. L. Pallec, J.-M. Do, Massive asynchronous master-worker EA for nuclear reactor optimization: a fitness landscape perspective, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, ACM, 2017, pp. 295–296.
- [24] K. Alyahya, J. E. Rowe, Simple random sampling estimation of the number of local optima, in: Parallel Problem Solving from Nature, Springer, 2016, pp. 932–941.
- [25] E. D. Weinberger, Correlated and uncorrelated fitness landscapes and how to tell the difference, in: Biological Cybernetics, 1990, pp. 63:325–336.
- [26] S. Tari, M. Basseur, A. Goëffon, Sampled walk and binary fitness landscapes exploration, in: Artificial Evolution, Springer International Publishing, Cham, 2018, pp. 47–57.
- [27] G. E. Box, G. M. Jenkins, G. Reinsel, Time series analysis: forecasting and control holden-day san francisco, BoxTime Series Analysis: Forecasting and Control Holden Day1970 (1970).
- [28] V. K. Vassilev, T. C. Fogarty, J. F. Miller, Smoothness, ruggedness and neutrality of fitness landscapes: from theory to application, in: Advances in evolutionary computing, Springer, 2003, pp. 3–44.
- [29] K. M. Malan, A. P. Engelbrecht, A survey of techniques for characterising fitness landscapes and some possible ways forward, Information Sciences 241 (2013) 148 – 163.

- [30] R. Armas, H. Aguirre, F. Daolio, K. Tanaka, Traffic signal optimization and coordination using neighborhood mutation, in: 2016 IEEE Congress on Evolutionary Computation (CEC), 2016, pp. 395–402.
- [31] J.-M. Montanier, N. Bredeche, Embedded evolutionary robotics: The (1+ 1)-restart-online adaptation algorithm, in: New horizons in evolutionary robotics, Springer, 2011, pp. 155–169.
- [32] H.-K. Hwang, A. Panholzer, N. Rolin, T.-H. Tsai, W.-M. Chen, Probabilistic analysis of the (1+ 1)-evolutionary algorithm, *Evolutionary computation* 26 (2) (2018) 299–345.
- [33] T. L. Lai, H. Robbins, Asymptotically efficient adaptive allocation rules, *Advances in applied mathematics* 6 (1) (1985) 4–22.
- [34] P. Auer, N. Cesa-Bianchi, P. Fischer, Finite-time analysis of the multiarmed bandit problem, *Mach. Learn.* 47 (2-3) (2002) 235–256.
- [35] Á. Fialho, L. Da Costa, M. Schoenauer, M. Sebag, Dynamic Multi-Armed Bandits and Extreme Value-based Rewards for Adaptive Operator Selection in Evolutionary Algorithms, in: Learning and Intelligent Optimization (LION 3), Vol. 5851/2009 of LNCS, 2009, pp. 176–190.
- [36] C. Jankee, S. Verel, B. Derbel, C. Fonlupt, Distributed Adaptive Metaheuristic Selection: Comparisons of Selection Strategies, in: 13th International Conference on Artificial Evolution (EA 2015), LNCS, Springer, 2015, pp. 83–96.
- [37] J. Liu, D. P. Liebana, S. M. Lucas, Bandit-based random mutation hill-climbing, in: IEEE Congress on Evolutionary Computation, 2017, pp. 2145–2151.
- [38] F. Ramm, C. Karch, J. Topf, Geofabrik, <https://www.geofabrik.de>, accessed: 2018-01-24.

- [39] A. Horni, K. Nagel, K. Axhausen (Eds.), *Multi-Agent Transport Simulation MAT-Sim*, Ubiquity Press, London, 2016.
- [40] D. Grether, J. Bischoff, K. Nagel, Traffic-actuated signal control: Simulation of the user benefits in a big event real-world scenario, in: *2nd International Conference on Models and Technologies for ITS*, Leuven, Belgium, 2011, pp. 11–12.
- [41] D. Grether, *Extension of a multi-agent transport simulation for traffic signal control and air transport systems*, Ph.D. thesis (06 2014).
- [42] C. Vilet, L. Pezin, *Schéma de développement urbain*, Tech. rep., Ville de Calais (July 2010).
- [43] K. Müller, K. W. Axhausen, *Population Synthesis for Microsimulation: State of the Art*, ETH Zürich, Institut für Verkehrsplanung, Transporttechnik, Strassen-und Eisenbahnbau (IVT), 2010.
- [44] B. Aboutaib, C. Fonlupt, V. Marion, S. Verel, Urban traffic modeling and simulation: Moroccan capital case study, in: *The 2nd International Conference on Smart Applications and Data Analysis for Smart Cities SADASC'18*, Casablanca, Morocco, 2018.
- [45] K. J. Archer, R. V. Kimes, Empirical characterization of random forest variable importance measures, *Computational Statistics and Data Analysis* 52 (4) (2008) 2249 – 2260.

## **Appendix A. Experiments on Quito, Ecuador**

Results presented in section 6.3 rely on the real-world mobility case study of Quito city, Ecuador [9, 15, 30]. The transport network covers a 40 km<sup>2</sup> area and includes 8192 links and 70 signal systems.

24 real-world-like scenarios are generated with SIALAC benchmark, involving the same parameters as those presented in Table 1. The ratio of activated signal systems is fixed to 100%. Home and activity clusters are defined according to census data and attractive destinations in the city (university campuses, working districts, shopping malls, ...).

The same experimental protocol as described in section 5 is conducted on these new case studies. Fitness landscapes and their ruggedness are computed from a set of 20 random walks on each benchmark instance. Importance of variables are estimated with equation 9. As for the Calais city study case, some signal systems appear to be more or less important according to the studied instance (see Figure A.11 and Table A.7). Critical traffic lights are deduced when their average importance is greater than 3, otherwise they are not important when it is lower than 2. Once again, we observe high standard-deviation values, often close to average values, suggesting important signal systems are not the same according to the considered instances. Signal system groupings for the online-learning algorithm are based on top of this analysis.

Original works simulate the movement of 20 000 vehicles in the transport network. Therefore, the performances depicted in Figure 10 are computed from two benchmark instances involving 20 000 agents.

## **Appendix B. Convergence of Algorithms**

We extend the optimization runs for one SIALAC scenario involving 5000 agents. The experimental setups follow the specifications of sections 5.1 and 5.5, except the optimization budgets are doubled to 4000 simulator evaluations in a way to observe the



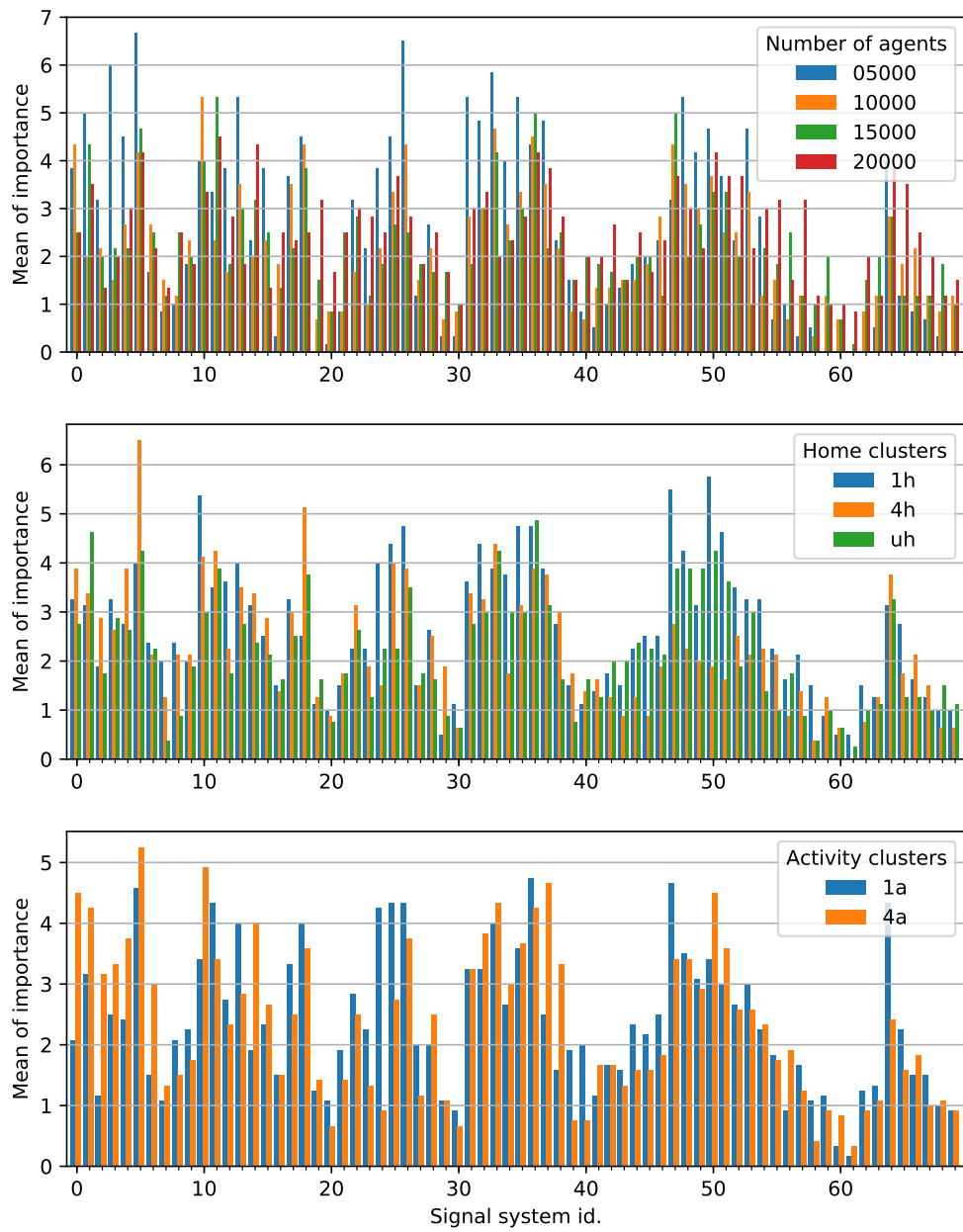


Figure A.11: Mean signal system importance according to the benchmark parameters, with 100% signal systems activated on Quito scenario. *Colors should be used in print. Otherwise, bars are ordered as the legend.*

Table A.7: Average ( $\mu$ ) and standard deviation ( $\sigma$ ) of traffic signal importance over all benchmark instances with 100% signal systems activated on Quito scenario.

id.	0	1	2	3	4	5	6	7	8	9	10	11	12	13
$\mu$	3.29	3.71	2.17	2.92	3.08	4.92	2.25	1.21	1.79	2.	4.17	3.88	2.54	3.42
$\sigma$	2.34	2.39	1.97	2.48	2.5	2.52	1.88	1.15	1.53	0.82	2.15	2.24	1.76	2.16
id.	14	15	16	17	18	19	20	21	22	23	24	25	26	27
$\mu$	2.96	2.5	1.5	2.92	3.79	1.33	0.88	1.67	2.67	1.79	2.58	3.54	4.04	1.58
$\sigma$	2.05	1.78	1.55	1.58	2.5	1.37	1.17	1.49	1.37	1.32	2.58	2.36	2.23	1.29
id.	28	29	30	31	32	33	34	35	36	37	38	39	40	41
$\mu$	2.25	1.08	0.79	3.25	3.54	4.17	2.83	3.62	4.5	3.58	2.46	1.33	1.38	1.42
$\sigma$	1.53	1.26	1.15	2.18	2.	2.09	1.7	2.63	1.94	2.33	1.53	1.46	1.52	1.47
id.	42	43	44	45	46	47	48	49	50	51	52	53	54	55
$\mu$	1.67	1.46	1.96	1.88	2.17	4.04	3.46	3.	3.96	3.29	2.62	2.79	2.29	1.79
$\sigma$	1.43	1.38	1.49	1.36	1.21	2.42	2.08	2.52	2.35	1.84	1.63	1.98	1.54	1.61
id.	56	57	58	59	60	61	62	63	64	65	66	67	68	69
$\mu$	1.42	1.46	0.75	1.04	0.58	0.25	1.08	1.21	3.38	1.92	1.67	1.25	1.04	0.92
$\sigma$	1.41	1.71	1.23	0.98	1.	0.6	1.29	0.91	1.98	1.8	1.95	1.23	1.02	1.08

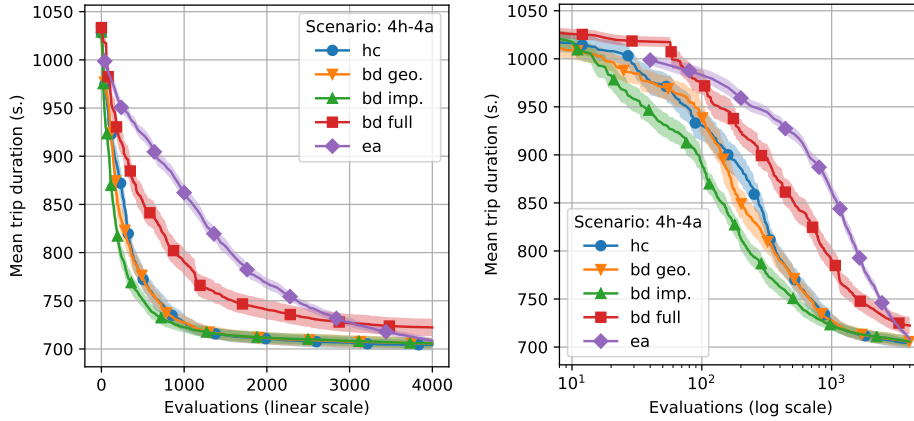


Figure B.12: Minimization of mean trip duration for 5000 agents and 100% signal systems activated ( $h$ : home clusters,  $a$ : activity clusters). Five optimization algorithms are compared : stochastic hill-climbing ( $hc$ ), an evolutionary algorithm ( $ea$ ) and Bandit Descents with geographical ( $geo.$ ), importance-based ( $imp.$ ) or  $full$  groupings. The exploration parameter is set to  $R = 2$  for UCB algorithms. Studied scenario is the same for the two plots. Notice the linear (left) and logarithmic (right) scales.

convergence of the compared algorithms. These experiments represent a computational effort of about 30 hours for one single run on a single-threaded machine. Figure B.12 shows the minimization of the mean trip duration according to the number of evaluations (*i.e.* simulator calls), which is displayed on two scales to ease observations.

The linear scale clearly shows that each algorithm converges to an equivalent set of optimal traffic light settings at the end of the 4000 evaluations. This statement is confirmed by Kruskal-Wallis statistical tests at level 5%. Nevertheless, the logarithmic scale highlights the performance of importance-based Bandit Descent, which finds the best quality traffic light settings when considering a few evaluations (this was discussed in section 5.5).

## Appendix C. Running Times of Algorithms

In short, one iteration consists of two main phases: first, the optimization algorithm evolves one or a set of solutions (*e.g.* crossover and mutation); then the fitness function

of the latter is evaluated in the simulator. Table C.8 and Table C.9 list the average running times for one iteration on all benchmark scenarios. Stochastic hill-climbing, elitist evolutionary algorithm and the proposed importance-based Bandit Descent are compared. Overall, most if not all of the consumed time for one iteration is spent in simulation, while the execution time of the evolution phase is negligible (in the order of milliseconds). As a consequence, the evolutionary algorithm definitely requires much more time to complete an iteration, as it requests even more simulator calls as there are more individuals to evaluate in the population. Indeed, the elitist evolutionary algorithm follows Armas *et al.* setup [30]: population size, offspring size and the number of elite individuals are set to 20, 20 and 10, respectively. As a result, one iteration of the elitist evolutionary algorithm represents  $20 \times 2$  simulator evaluations, whereas one iteration of stochastic hill-climbing or Bandit Descent requires only one simulator evaluation. This explains the highly contrasting computation time per iteration in Table C.8. However, notice that all algorithms are compared with the same number of evaluations using the simulator (*e.g.* same running time budget) in section 5.5.

Finally, every algorithm reaches a quality-equivalent set of solutions when the optimization budget is extended (see Appendix B). These observations reinforce the arguments of developing algorithms able to converge towards acceptable solutions in a minimum of simulation requests.

#### **Appendix D. Location of Traffic Light Systems**

Figure D.13 depicts a map of the Calais city road network, and locates the 33 traffic lights.

#### **Appendix E. Simulation Visualization**

Figure E.14 visualizes a simulation of traffic flows in the Calais city road network, utilizing MATSim.

Table C.8: Average running times on Calais, France scenarios.

Agents	$h$	$a$	Simulation (in s)	Iteration (cumulated time in s)		
				HC	EA	BD
5000	1	1	31.0	31.0	1243.0	31.0
		4	31.7	31.7	1271.2	31.7
	4	1	32.0	32.0	1283.2	32.0
		4	30.9	30.9	1239.1	30.9
	uniform	1	33.6	33.6	1347.4	33.6
		4	34.0	34.0	1363.4	34.0
10000	1	1	42.4	42.4	1700.2	42.4
		4	46.8	46.8	1876.7	46.8
	4	1	42.5	42.5	1704.6	42.5
		4	45.6	45.6	1828.6	45.6
	uniform	1	44.7	44.7	1792.5	44.7
		4	48.9	48.9	1960.9	48.9
15000	1	1	50.6	50.6	2029.0	50.6
		4	55.7	55.7	2233.6	55.7
	4	1	50.6	50.6	2029.1	50.6
		4	56.9	56.9	2281.7	56.9
	uniform	1	50.3	50.3	2017.0	50.3
		4	52.3	52.3	2097.2	52.3
20000	1	1	58.0	58.0	2325.8	58.0
		4	63.8	63.8	2558.4	63.8
	4	1	59.7	59.7	2394.0	59.7
		4	63.0	63.0	2526.3	63.0
	uniform	1	62.0	62.0	2486.2	62.0
		4	67.8	67.8	2718.8	67.8

Table C.9: Average running times on Quito, Ecuador scenarios.

Agents	$h$	$a$	Simulation (in s)	Iteration (cumulated time in s)		
				HC	EA	BD
5000	1	1	28.0	28.0	1122.8	28.0
		4	29.7	29.7	1191.0	29.7
	4	1	30.1	30.1	1207.0	30.1
		4	28.9	28.9	1158.9	28.9
	uniform	1	29.6	29.6	1187.0	29.6
		4	29.8	29.8	1195.0	29.8
10000	1	1	33.5	33.5	1343.4	33.5
		4	33.8	33.8	1355.4	33.8
	4	1	34.6	34.6	1387.5	34.6
		4	36.0	36.0	1443.6	36.0
	uniform	1	34.5	34.5	1383.5	34.5
		4	34.0	34.0	1363.4	34.0
15000	1	1	39.0	39.0	1563.9	39.0
		4	38.6	38.6	1547.9	38.6
	4	1	40.5	40.5	1624.0	40.5
		4	46.1	46.1	1848.6	46.1
	uniform	1	42.5	42.5	1704.3	42.5
		4	46.2	46.2	1852.6	46.2
20000	1	1	46.6	46.6	1868.7	46.6
		4	48.4	48.4	1940.8	48.4
	4	1	51.4	51.4	2061.1	51.4
		4	50.4	50.4	2021.0	50.4
	uniform	1	48.4	48.4	1940.8	48.4
		4	49.5	49.5	1985.0	49.5

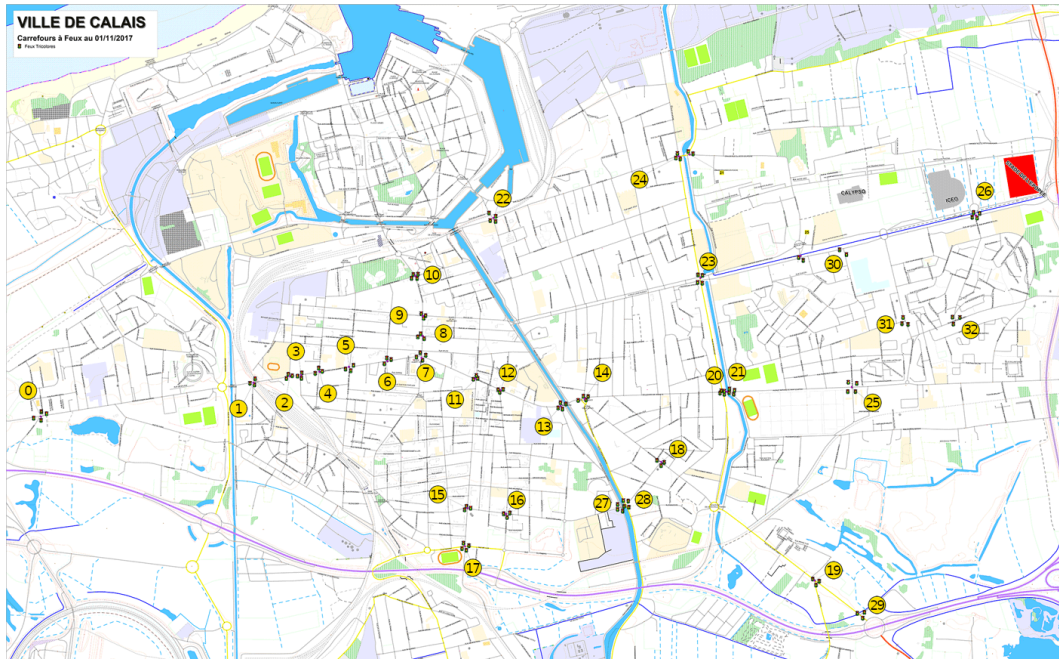


Figure D.13: Location of the 33 systems of traffic lights involved in Calais city road network.



Figure E.14: Simulation of traffic flows in the Calais road network.