DEGREE PROJECT, IN COMPUTER SCIENCE , SECOND LEVEL
STOCKHOLM, SWEDEN 2015

Multiple-objective optimization of traffic lights using a genetic algorithm and a microscopic traffic simulator

NICOLAS DAMAY

KTH ROYAL INSTITUTE OF TECHNOLOGY

SCHOOL OF COMPUTER SCIENCE AND COMMUNICATION (CSC)

Nicolas Damay, damay@kth.se

Multiple-objective optimization of traffic lights
using a genetic algorithm and a microscopic traffic simulator

KTH Royal Institute of Technology,
School of Computer Science and Communication (CSC)
Supervisor: Orjan Ekeberg
Examiner: Anders Lansner

June 3, 2015

Abstract

Given the demand for mobility in our society, the cost of building additional infrastructures and the increasing concerns about the sustainability of the traffic system, traffic managers have to come up with new tools to optimize the traffic conditions within the existing infrastructure. This study considered to optimize the durations of the green light phases in order to improve several criteria such as the ability of the network to deal with important demands[1] or the total pollutant emissions.

Because the modeling of the problem is difficult and computationally demanding, a stochastic micro-simulator called 'Simulation of Urban MObility' (SUMO) has been used with a stochastic optimization process, namely a Genetic Algorithm (GA).

The research objective of the study was to create a computational framework based on the integration of SUMO and a Multi-Objective Genetic-Algorithm (MOGA). The proposed framework was demonstrated on a medium-size network corresponding to a part of the town of Rouen, France. This network is composed of 11 intersections, 168 traffic lights and 40 possible turning movements. The network is monitored with 20 sensors, spread over the network. The MOGA considered in this study is based on NSGA-II. Several aspects have been investigated during the course of this thesis.

An initial study shows that the proposed MOGA is successful in optimizing the signal control strategies for a medium-sized network within a reasonable amount of time.

A second study has been conducted to optimize the demand-related model of SUMO in order to ensure that the behavior in the simulated environment is close to the real one. The study shows that a hybrid algorithm composed of a gradient search algorithm combined with a GA achieved a satisfactory behavior[2] for a medium-size network within a reasonable time.

[1]The demand is defined as the number of cars willing to enter the network in a given amount of time

[2]This satisfactory behavior is described in 3.2.6

Acknowledgment

I would like to thank Yann SEMET, Research Engineer at Thales Research and Technology, for his valuable guidance during the course of this project, for all the work he achieved before my arrival especially for the creation of the benchmark and for the time he spent to collect/analyze the data. I would also like to thank him for his trust and for the great freedom he granted me during my work. All those aspects helped me to turn this research project into a very stimulating and interesting experience. Finally, I would also like to thank him for his always meaningful quotes about the optimization science and D. Knuth.

Simon FOSSIER, research Engineer at Thales Research and Technology, and Loic MONGELLAZ, intern at Thales Research and Technology, for their help in gathering and processing the field data for the case study.

Wilco BURGHOUT and Xiaolang MA, searchers at the Centre for Traffic Research, KTH, for their guidance and advices before this work took place.

Benedicte GOUJON and Hellia POUYLLAU, who i shared the office with, for all those discussions and entertaining moments.

I thank Jean-Louis ROUQUIE and Thierry GLAIS for their supervision during my work, their expertise and their contributions in the field-related topics which turned this project in both a research and applied optimization project.

I would like to thank all the staff of the LDO laboratory at Thales Research and Technology for their warm welcome and for their advices and help.

I would also like to thank my relatives who supported me during the course of this work.

Acronyms

APM . . . . . Assignment Proportion Matrix
CO . . . . . . Carbon Monoxide
CO_2 . . . . . . Carbon Dioxide
GA . . . . . . Genetic Algorithm
GSM . . . . . Gradient Search Method
HC . . . . . . Hydrocarbon
ITS . . . . . . Intelligent Transportation System
MO . . . . . . Multi-Objective
$MOEA$ . . . . Multi-Objective Evolutionary Algorithm
$MOGA$ . . . . Multi-Objective Genetic Algorithm
$MOOP$ . . . . Multi-Objective Optimization Problem
MSA . . . . . Memetic Search Algorithm
NO_x . . . . . . Nitrogen Oxides
$NSGA$ . . . . Non-dominated Sorting Genetic Algorithm
ODM . . . . . Origin-Destination Matrix
PM_X . . . . . Particles
SSA . . . . . . Stochastic Search Method
$SPEA$ . . . . Strength Pareto Evolutionary Algorithm
$SUMO$ . . . . Simulation of Urban MObility

Contents

Chapter 1

Introduction

1.1 Context

It appears that congestion problems have been a serious issue in a large number of cities. Traffic congestion in big cities might have important economic and social impacts. Congestion leads to high time-delay and reduce the productivity as described by C.S-W (2014) [1]. Simultaneously, congestion favors speed fluctuations and high fuel-consumption. On the other hand, a decrease in the time spent in the traffic and in the average fuel-consumption results in enhanced labor costs and public health.

Yet, it is possible for traffic managers to have a positive influence on a given traffic situation by acting on the network. For instance, influencing the durations of the green-light phases of the different intersections can have an impact on the ability of the network to cope with the demand[1].

The dynamic, complex and unstable behavior of the urban traffic process makes the optimization of the green light phases difficult and computationally demanding for two main reasons:

- First, the estimation process is time-consuming. Indeed, the process of turning a given network, a traffic light setting and the description of the demand into estimated global indicators, such as the total waiting-time or the pollutants emissions, is complex.
- Second, the optimization of the traffic light setting is also a complex problem for big networks. Indeed, for a small town, a medium quarter or several intersections, there are tens or hundreds of green-light phases. The main difficulty thus lies in the dimension of the search space (The space of the traffic light settings). Hence, the use of deterministic methods may require a prohibitory amount of time.

[1] The demand is defined as the number of car willing to enter the network in a given amount of time

In order to achieve an acceptable accuracy for the evaluation process this paper focuses on micro-simulators based on second by second estimates for each vehicle. More precisely, this thesis used the traffic simulator called SUMO.

A broad range of stochastic-based methods have been studied in recent years to optimize the traffic light setting. Several criteria have been successfully optimized in recent studies, e.g: Waiting-time, fuel-consumption or noise emissions. Yet, those studies usually consider a single objective whereas traffic managers have to deal with several conflicting objectives simultaneously.

Among all the available techniques we will focus on Multi-Objective Genetic Algorithms (MOGAs). The aim of those algorithms is to optimize a problem with respect to several objectives simultaneously (eg: time-delay and pollution) using stochastic-based methods.

One of the main reason to use GAs to solve the MOOP is their population-based approach. Those algorithms manipulate several solutions for the problem simultaneously. In the best case, those solutions will eventually represent all the possible trade-offs between the different objectives.

Moreover, MOGAs are efficient in high dimension spaces and they require little knowledge ‘a priori‘ on the problem they solve, this knowledge mainly appears in a ’fitness function ’ or ’cost function’ which guides the evolutionary process. In our case this fitness function is derived from the traffic-simulator. MOGAs are thus very flexible and can be applied wherever the user is able to define a cost function.

Yet, for a given MOGA, the performances are highly correlated to the instance of the optimization problem. More precisely, under some assumptions, a general-purpose universal optimization strategy is theoretically impossible, Wolpert (1997) [2]. The first part of this work has thus been devoted to the calibration of the GA for the traffic optimization problem.

For this part, the research question is: ***Is it possible to use a GA to achieve important gains within a reasonable time for a medium-sized network?***

One of the main drawback of the GAs is that they usually do not take any feedback into account. The search is stochastic. Thus time required to find the optimal solution might be important. In order to address this issue several strategies have been studied in this part:

- Objective Selection. This section aims at providing supporting tools for traffic managers to decide which objectives should be optimized. The point here is to reduce the number of objectives that were to be optimized in order to reduce the size of the search-space.

- Manual optimization of some parameters through an experimental campaign based on empirical rules. The goal here is to identify an estimate of the optimum static values for each parameters of the GA.

- Dynamic optimization of key parameters through an experimental campaign based on different strategies. The purpose is to tune the key-parameters of the GA in an automatic way.

- Study of the inoculation strategy in order to reuse previously-computed solutions. This last part is a little different from the other parts. The question is: Are the solutions found by the algorithm resilient to a small change in the problem definition? If so, re-using previously-computed solutions obtained in different conditions to seed the initial population of the algorithm will save time.

This first study has been conducted with a given demand description. Another independent yet connected study has been conducted to study the demand-related modeling of SUMO. The aim of this part is to ensure that the simulated environment behaves as close as the real one as possible. The study adapted and extended an already existing algorithm used for this purpose, the Gradient Search Method (GSM). The research question of this part is: ***Is it possible to calibrate the demand-related model of SUMO in order to reach a satisfactory behavior within a reasonable time for a medium-sized network using the GSM?***

A subsidiary research question has been investigated: ***Is it possible to improve the performance of the GSM using a GA for a so-called Stochastic Search Algorithm (SSA)?***

This study focuses on the network presented in figure 1.1. This network has been built using the traffic simulator 'SUMO' and field data, provided by the transport unit of Thales.

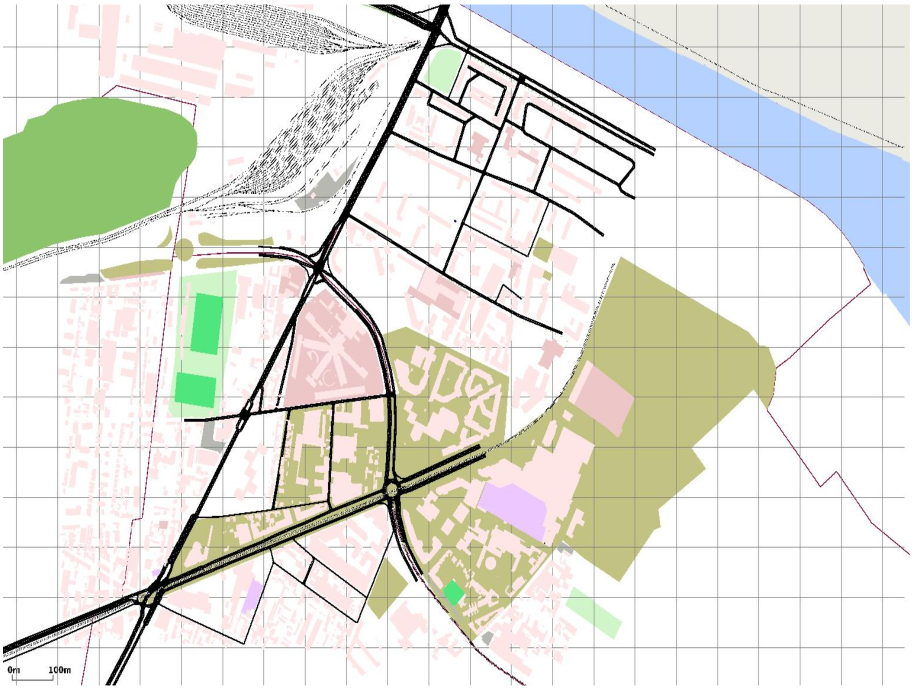


Figure 1.1: Case study: Town of Rouen, France.

1.2 Major Contributions

The main contributions of the study are the following:

- Create a computational framework based on the integration of SUMO and a GA.
- Design and implement a computational framework to calibrate the demand-related model of SUMO to ensure a reliable behavior of the simulated environment.
- Study the correlations among the different objectives for the MOOP in order to propose a strategy for the objective-selection.
- Find an estimate of the optimum set of parameters for the considered GA for the traffic optimization problem.
- Evaluate and compare different signal control strategies for the traffic optimization problem by conducting case studies for both single and multi-objective evolutionary algorithms.
- Analyze and discuss the resilience of the solutions found by the algorithm with respect to small changes in the problem definition.

1.3 Structure

The paper is made out of 4 chapters, starting with an introduction to the study.

Chapter 2 starts with an introduction to the GA. A state of the art review of traffic optimization using MOEAs and some previous studies and literature published on the topic are then described. The methodology adopted in this study is then described, the different contributions are presented, and the results are discussed.

Chapter 3 starts with an overview of the main traffic modeling component and presents some previous studies and literature published on the topic. The second part presents the proposed methodology to calibrate the demand-related model. The algorithms and the different contributions of this thesis are thus described and the results obtained on the case study are discussed.

Finally, the chapter 4 gathers and summarizes all the conclusions of this study.

Chapter 2

Traffic Optimization Using Genetic Algorithm

This chapter investigates the question: ***Is it possible to use a GA to achieve important gains within a reasonable time for a medium-sized network?***

2.1 Introduction

There has been considerable research into traffic optimization using Evolutionary Algorithms (EA). Especially the Mono-Objective GAs, a sub family of EAs, have been successfully applied to traffic optimization, yet we would like to investigate the behavior of GAs for the MOOP. Indeed, traffic managers have to consider several objectives simultaneously, e.g: average time-delay, pollution, average speed.

Some of those objectives might be conflicting, i.e: a good solution with respect to one objective might have poor performances with respect to the others e.g: The drivers spend little time in the network but release important amount of particles NOx. We are interested in finding several solutions to the problem at once, more accurately we want to find all the range of possible trade-offs with respect to the different objectives.

Yet, the process of measuring traffic-related indicators on a real-world network is long and inconvenient, a very popular approach in the research community has been to use traffic simulators.

2.1.1 Traffic Simulators

The following sections describe the three main families of traffic simulator described by Pursula (1999) [3].

Macroscopic Simulators

A Macroscopic traffic flow model is a model that formulates the relationships among traffic flow characteristics like density, flow and mean speed of a traffic stream. The underlying assumption is that traffic flow is comparable to fluid streams. Those models have been mostly investigated between 1960 and 1980. Two macroscopic models are TRANSYT-7F [4] and SYNCHRO [5].

Boxill, Adams Yu (2000) [6] studied those simulators. They argue that macroscopic simulators cannot differentiate between individual vehicles, and usually do not cater for different vehicle types. They lack the ability to model complex roadways. Thus, this study did not consider any macroscopic traffic simulator.

Microscopic Simulators

In contrast to macroscopic models, microscopic traffic flow models simulate single vehicle-driver units. The dynamic variables of the models represent microscopic properties like the position and the velocity of every single vehicles. Microscopic models usually require more detailed geometric and traffic information than macroscopic models.

Whereas the model validation for a macroscopic simulation model is usually undertaken at a macroscopic scale only as described by Skabardonis, el (1989) [7], microscopic models should be validated at both a microscopic and a macroscopic level according to Brackstone et al (1993) [8]. It is thus more time-consuming to manually create a benchmark using this approach.

Although microscopic traffic flow models are computationally more demanding. (It took over a month of computing power to run an optimization process using VISSIM and a MOEA for a small network presented by Stevanovic and Kergaye (2011) [9]), microscopic-simulators achieve the desired features since they aggregate second-by-second indicators for every single vehicle. It is thus possible to estimate very accurately the traffic indicators we are interested in.

Mesoscopic Simulators

Mesoscopic models try to combine macroscopic and microscopic methods. Mesoscopic models simulate individual behaviors but describe their interactions based on aggregated relationships. Those models produce less consistent result than microsimulators but they are computationally less demanding.

2.1.2 Evolutionary Algorithms

Different EAs have been used for traffic optimization. We will try to investigate which are the main techniques and their advantages/drawbacks.

Genetic Algorithms

GAs form a class of algorithms inspired by the evolutionary process. Those algorithms are stochastic-based algorithms that implement the principle of the 'survival of the fittest'. Those algorithms are mainly used when little knowledge about the problem is available or when the 'search space' is too big for a deterministic algorithm or when it is difficult to find an analytic solution of the problem.

At the beginning of the algorithm, a population of individual is created. Those individuals, also known as 'chromosomes' represent possible solutions for the optimization problem. Thus, they are also referred as 'solutions'. They are composed of a number of bits called 'genes'. Those genes encode properties 'alleles'.

The evolution is conducted through an iterative process. Given a population 'parents', a succession of stochastic evolutionary operators are applied to create a new population 'offspring'. Each individual of this new population is evaluated according to the 'fitness function' or 'cost function'. This function measures how well an individual solves the optimization problem. See figure 2.1. The probability for an individual to be selected during the evolutionary process depends on its cost.

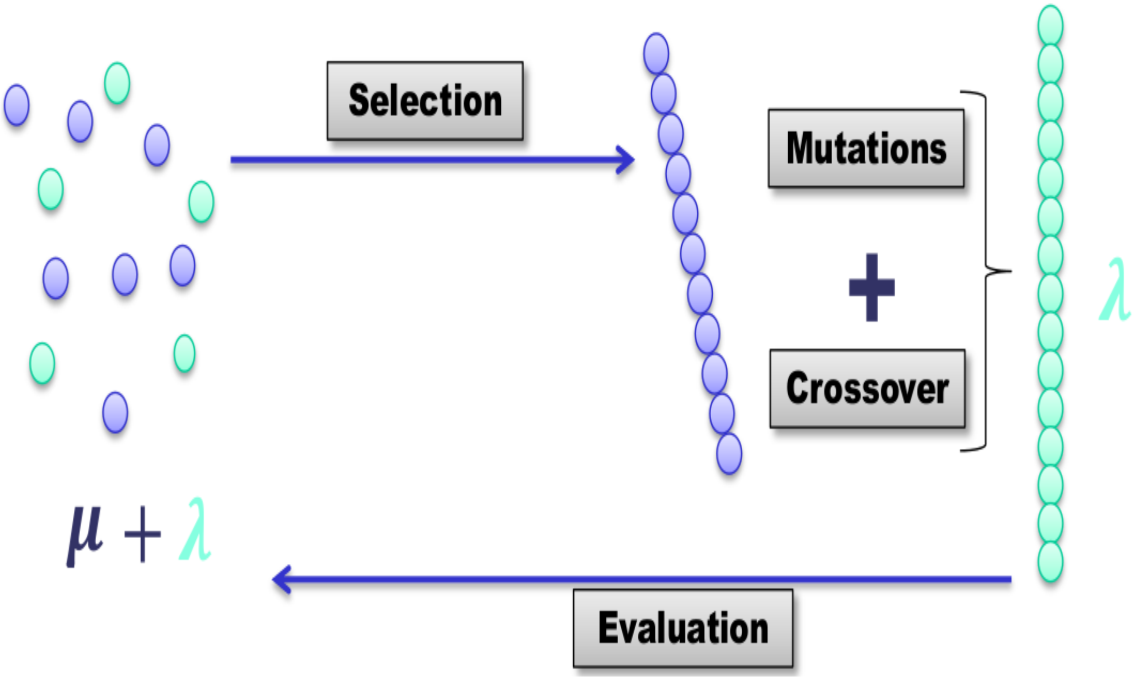


Figure 2.1: Example of a $(\lambda + \mu)$ GA

Among a population of λ parents, a selection operator selects and duplicates some individual to create a mating pool. The evolutionary operators are applied on this mating pool to create μ offsprings which are evaluated using the 'fitness function'. Among the $\lambda + \mu$ individuals available at that stage, a second selection operator (possibly the same) is applied to keep the λ best ones.

The succession of stochastic evolutionary operators is defined by the user and is generally designed based on empirical rules. Yet, most of the evolutionary algorithms are composed of three main steps:

- The selection operator: This operator selects the individuals that are to evolve.
- The crossover operator: This operator mixes the genetic material of the selected individuals.
- The mutation operator: This operator randomly changes the genetic material of the selected individuals.

The selection operator is mainly used to discard individuals with poor finesses and to ensure a variety in the population. The crossover and mutation operators are used to explore the search space. The relative importance of those two last operators is being extensively discussed. Indeed, a crossover can always be achieved through a succession of mutations and many mutations can be obtained through a succession of crossovers if there is an important diversity in the population. A more complete description of GAs has been done by Whitley (1994) [10].

The role of those operators depends on the instance of the optimization problem and there are no consensus on the relative importance of those operators. More precisely, under some assumptions, a general-purpose universal optimization strategy is theoretically impossible, Wolpert (1997) [2]. Thus a study of those operators must be led for each instance of optimization problems.

Historically the genes have been encoded with binary bits. We will not consider any distinction in the way the genes are encoded. Different approaches might be used provided that the operators are consistent with the representation of the variables. We will not investigate this topic any further, the details of the implementation being left aside.

Multi-Objective Genetic Algorithms

Traffic managers are not interested in finding the best solution with respect to one objective at all cost. A trade-off between antagonistic objectives must be found. Thus, the optimization must be done with respect to several objectives simultaneously.

MOGAs are an extension of the traditional GAs. Instead of measuring the fitness function with a single objective, several kind of mechanisms are used to aggregate several 'objectives' or 'fitness function'. The main difference appears in the selection operator. See figure 2.1.

One of the main reason to use GAs to solve the MOOP is their population-based approach. The algorithm manipulates a whole population. Thus, one run of the algorithm provides the user with several solutions of the problem. In the best case, those solutions represent all the possible trade-offs between the different objectives.

Moreover, MOGAs are efficient in high dimension space and they require little knowledge 'a priori' on the problems they solve, this knowledge mainly appears in a 'fitness function' which guides the evolutionary process. In our case this fitness function is derived from the traffic-simulator. MOGAs are thus very flexible and can be applied whenever the user is able to define a cost function.

All the knowledge and the parameters used to model the traffic are defined in the microscopic simulator. This provides a traffic control center with a large modularity, a new function can be implemented at small extra cost. For instance optimizing the travel time and the noise pollution simultaneously only requires to define a 'cost function' for the travel time and the noise pollution based on the output of the simulator. In our case the simulator outputs the aggregated travel time per edge in a file.

The mathematical formulation of the MOOP can be stated as:

$$\min\left(f_1(x), f_2(x), ..., f_n(x)\right)$$
$$\text{subj to: } x \in \chi$$

Where $f : \chi \longrightarrow \Re^n$ is an objective function that maps the feasible search space χ to the objective space.

A common difficulty with multi-objective optimization processes is the appearance of an objective conflict Hans et al (1988). None of the feasible solution allows simultaneous optimal solution for all the objectives. We are thus interested in the Pareto-optimum front which offers least objective conflicts.

Pareto Dominance An important notion for MOGAs is the dominance relationship. This dominance is used in the selection operator. See figure 2.1.

Let $A, B \in \chi$ be two solutions, A is preferred over B if all the values of $f(A)$ are better (smaller in the case of a minimization problem) than those of $f(B)$ and one at least being strictly better. A is said to Pareto-dominate B (denoted $A \prec B$). Figures 2.2, illustrates the dominance in the two dimensional space.

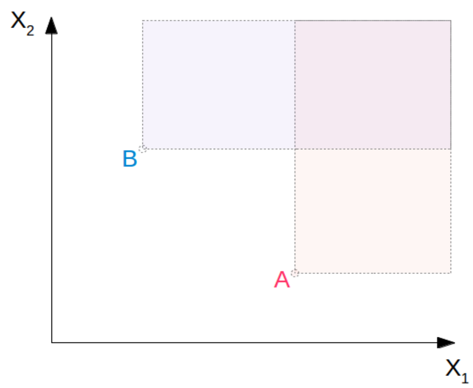


Figure 2.2: Dominance relationship in the case of a min-min problem.

The point A dominates the space described by the red area, the point B dominates the point described by the blue area. There is no dominance relationship between A and B.

The Pareto-dominance relation is not a total order since two solutions might not be in a dominance relationship. Therefore, the set of interest for the MOOP is the Pareto set of all the solutions of the decision space that are not dominated by any other. Those solutions represent the different possible trade-offs between the antagonistic objectives.

Our objective is to find the non-dominated solutions in the objective space with a large diversity and to find an optimal front as close as possible of the true Pareto-front.

2.1.3 State of the art

Mono-Objective

GAs and their applications to traffic optimization have been considerably studied.

Singh et al (2009) [11] used a GA connected with a micro-simulator developed in JAVA to optimize the throughput of a four-way intersection. Some other studies focused on different goals such as the noise, Rahmani et al (2011) [12].

Qian et al (2013) [13] optimized the pollutant emissions for isolated intersections. Other variant of GAs such as particle swarm have been used with the traffic simulator SUMO by Abushehab et al (2014) [14]. In particle swarm optimization each individual, 'particle', explores the research space stochastic-ally but the evolution is guided by both the best solution found so far for this particle and the global direction of the other particles.

Yet, one of the most serious drawbacks of GAs applied to online traffic light optimization is the computational power required, especially during the call of the 'fitness function' which usually requires to run a traffic simulator. It took over a month of computing power to run an optimization process using VISSIM and a MOEA for a small network presented by Stevanovic and Kergaye (2011) [9]

To deal with this drawback some techniques have been considered. Sanchez et al (2008) [15] suggested to use a faster microscopic simulator. A Cellular Automata (CA) has thus been developed. In CA models, the space is sampled into a grid and the time is sampled through the iterations of the process. With each new iteration, a cell decides on it's next state depending on it's previous state and it's neighbors state. Two extra levels of complexity have been added in this study to represent 'path' and 'vehicles'. The main advantage of this technique is to simplify as much as possible the simulation process to diminish the computational power required. In addition, the author used a cluster to distribute the computations over several computers. Although the results are very interesting we will not consider this approach as it is difficult to add other features to this structure such as pedestrians behavior, accident handling or weather calibration without considerably slowing down the simulation.

Another approach has been used by Iscaro et al (2013) [16]. Here the optimization was still performed with a GA and a microscopic simulator SUMO but the optimization ran only when a set of fuzzy rules detected a problem on an intersection. To speed up the search process a case-based reasoning was led and the search was inoculated with the traffic light plans of other controller which had faced the same conditions. This approach is very interesting and we re-use some of the methods presented in this paper.

Multi-Objective Optimization

Several classical methods can be applied to aggregate several objectives.

Weighted sum One of the simplest method to aggregate several objective is the weighted sum of all those objectives into a global 'cost function' Z:

$$Z = \sum_i w_i f_i(x), \text{ with } \sum_k w_k = 1$$
$$\text{Where } x \in \chi$$

In this method, the optimal solution is controlled by the weight vector $\mathbf{w}$. Yet, this method requires to explicitly express the priority of each objectives. Thus to find different tradeoffs between objectives, several optimizations are required with different weight vectors $\mathbf{w}$.

Distance function Another classical method that can be applied to the MOOP is the method of distance functions. In this method, the user defines a demand level vector $\tilde{y}$ the objective function is then expressed as:

$$Z = (\sum_i (|f_i(x) - \tilde{y}_i|)^r)^{1/r}, \text{ with } r \in \Re^{+*}$$

Here again the solution to the optimization problem depends on the demand level vector which might be undesirable. Moreover, the user must have a prior knowledge of the location of this demand level vector.

Min Max formulation Finally the Min-Max formulation can be applied. This method tries to minimize the relative deviation of the single objective functions from individual optimum. It attempts to minimize the objective conflict.

$$\text{minimize} F(x) = \max[Z_j(x)], \ j \in [1..N]$$
$$\text{Where } x \in \chi \text{ and } Z_j(x) = (f_j - \tilde{f}_j)/(f_j)$$

This method can be used when the objectives to be optimized have equal priority. Yet, here again the formulation of the optimization problem requires some knowledge on the problem being solved.

In all those classical methods, some knowledge about the problem is required. Moreover, the priority is always set explicitly or implicitly among all the objectives and the solution will depend on those priorities. Yet, decision makers generally need different alternatives for a multi-objective optimization process.

Other approaches One of the first strategy used to address those two aspects is the Vector Evaluated Genetic Algorithm (VEGA) presented by Schaffer (1984). Schaffer's approach was to use an extension of the Simple Genetic Algorithm (SGA). At each generation, a number of sub populations are created by performing proportional selection according to each objective in turn. For k different objectives, k sub-populations of N/k individuals are created. Those populations are then shuffled and the evolution takes place as for the SGA.

The main advantage of VEGA is its simplicity of implementation. One of the weakness of VEGA, as stated by Schaffer himself, is its bias towards some Pareto-Optimal solution. This problem, also referred to as 'speciation', arises because this technique selects individuals who excel in one dimension. Thus, individuals which achieve an interesting trade off for every objectives but do not excel in a single one are likely to be discarded during the selection process. Moreover, Richardson et al (1989) noted that the shuffling corresponds to averaging the fitness values associated with each of the objectives. Since Schaffer used proportional fitness assignment, these fitness components were in turn proportional to the objectives themselves. This corresponds to a linear combination of the objectives where the weights depend on the current distribution of the population as shown by Richardon et al. Thus, it is difficult for VEGA to produce Pareto-Optimal solutions for a non-convex search space.

In order to overcome this 'speciation', several other MOGAs have integrated a ranking procedure using the notion of dominance described previously. Among those stand the algorithms presented by Fonseca and Fleming (1993) [17] but also the Strenght Pareto Evolutionary Algorithm (SPEA) presented by Zitzle et al (1999) [18] and its extension SPEA-II Zitzler et al (2001) [19] and the Non-dominated Sorting Genetic Algorithm (NSGA) Srinivas et al (1994) [20] and its extension NSGA-II Deb et al(2000) [21].

Some studies have used those MOGAs to solve the multi-objective traffic optimization problem.

Branke et al (2007) [22] optimized the number of stops and the travel times using NSGA-II and the traffic simulator called VISSIM. Robles (2012) [23] extended this study. He included the emissions in the objectives to be optimized. In both those studies NSGA-II succeeded in finding optimized solutions for the traffic signal settings. Those studies were yet limited to small intersections. Li et al (2013) [24] focused on the over saturated network and used the throughput maximum and average queue ratio minimum as the optimization objectives. Finally, Ohazulike Brands (2013) [25] also used NSGA-II to optimize the traffic.

Yet, all of those studies described a small network composed of very few intersections.

Other Approaches

Other kinds of optimization techniques have been applied to the traffic optimization problem. This section presents a state-of-the-art review of some of those techniques.

Models based approaches Traffic controllers based on very robust mathematical models such as Petri nets. Petri nets are directed bipartite graph. One of the first application of Petri-nets to traffic controller has been conducted by DiCesare et al (1994) [26]. A more recent study is shown by dos Santos Soares et al (2012) [27]. Although Petri-nets allow mathematical demonstration of physical properties they tend to be hard to calibrate in order to match the actual traffic configuration. Those models are thus not flexible enough to be used as on-line controllers since they fail to capture the changes in the traffic configuration e.g: rain, accident, construction work.

Reinforcement Learning (RL) Reinforcement Learning is a stochastic search process. One or several agents can perform a given set of actions in a set of environment states. Each action performed by an agent leads to another state and grants the agent with a reward. Credit for successful behavior is granted whereas poor behaviors are punished. The knowledge of the agent(s) is collected in a look-up table built with past observations. The application of RL in the context of traffic signal control has been pioneered by Thorpe and Anderson (1996) [28].

Brys et al (2014) [29] have used the SARSA method. It is a model free method that estimates an action-value function, $Q(s, a)$, measuring the expected return of taking action a in state s from experience. After each state transition, The Q table is updates. Here the Q table is shared by all the agents. SARSA is thus able to learn at each iteration. Yet this approach requires exponential number of iteration to accurately update its estimates. (increasing number of states/actions). Simultaneously the Q matrix being shared by several agents having different objectives the convergence of the Q matrix is not guaranteed.

Bazzan et al (2007) [30] tried to organize agents in groups of limited size to reduce the number of joint actions. Those groups being controlled by other agents a RL technique has been used. Several levels of decision have been implemented. A first layer collects information and store the knowledge in a database while the second layer uses this knowledge to suggest actions to the agents. Those agents will decide whether they want to follow the suggested action. They also decide whether they update their Q table or not.

Khamis et al (2014) [31] used a vehicle-based approach for the multi-agent system. The number of states will grow linearly in the number of lanes and vehicles. More discussion on Collaborative Reinforcement Learning are held by Dusparic et al (2007) [32].

Reinforcement learning has also been combined with Fuzzy rules. A fuzzy controller uses simple rules designed by traffic managers with If - Then conditions. A good controller strategy can be achieved by adequately combining such rules. Example: If a car approaches the intersection and all the other lanes are empty then switch to the green light. The process of manually combining rules is time-consuming and complex. It cannot be used to achieved adaptive controllers. A Reinforcement Learning approach allows us to automatically combine the fuzzy rules without human intervention. Bingham (2001) [33] presented an RL algorithm which stochastic search. Although the results of this technique are interesting we will not go further in the subject as this method does not allows us to define clearly which objectives have to be optimized from a macroscopic point of view. For example the operator cannot implement a strategy to reduce specifically the pollution. The cost of labeling each fuzzy-rule with the corresponding objectives being too important.

2.1.4 Background Material

An important relevant work has been done before this thesis took place. Indeed, a first single-objective optimization framework had already been implemented using a GA by Y. SEMET, research engineer at Thales Research and Technology who supervised the work done in this study. This algorithm was tuned and thus an estimate of the best set of parameters for the GA was available.

Yet, a whole new optimization framework has been implemented during this thesis. However, some parts of the pre-existing framework have been used. The optimization algorithm is thus different. The mutation operator especially are different and another experimental campaign had to be conducted. Yet, the same representation for the variables, the same benchmark and the same traffic simulator have been used.

2.2 Methodology

2.2.1 Variables

An important part of a GA is the representation of the variables. Figure 2.3 presents the representation of the variables. It is important to observe that one variable impacts a whole intersection.

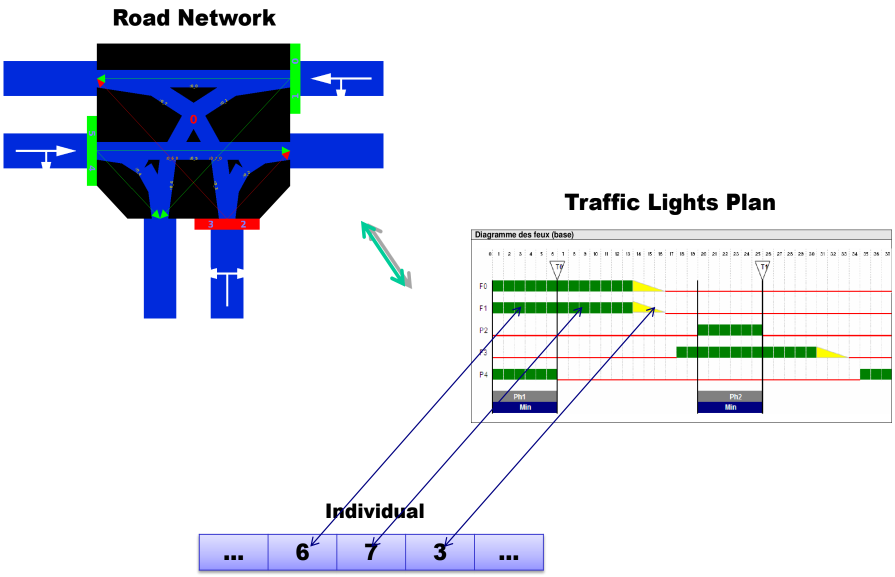


Figure 2.3: Decision variables.

Example of the description of an interactions in the simulator. The traffic lights are ruled by a TLP. This plan is composed of six rows, one per traffic light. Each column describes the state of the traffic lights at a given time. We can thus split the cycle in several phases described by a duration and a set of states for the traffic lights. For this intersection, there are 10 phases. The durations of those phases are the variables of the algorithm.

2.2.2 Simulator

The objective of this thesis is to optimize some traffic-related indicators. It is thus important to ensure that those indicators realistically reflect the reality. This thesis thus focuses on the microscopic traffic simulator among the traffic simulators described previously. This choice has been made because the microscopic traffic simulators are based on second by second estimates for each vehicles. They are thus more accurate than macroscopic simulators.

More precisely the simulator called SUMO has been used for it is an open source traffic simulation package including the simulation application itself as well as supporting tools, mainly for network import and demand modeling. SUMO helps to investigate a large variety of research topics, mainly in the context of traffic management and vehicular communications. Krajzewicz et al (2012) [34]. Other simulators are presented in Boxill et al (2000) [35].

One should note that the simulator is stochastic, i.e: two simulations with the same network, the same demand and the same traffic light setting might give two slightly different results. This is mainly due to uncertainty in drivers behavior and injection times. To address this uncertainty, every 'evaluation process' refers to the aggregation of several simulations led with different 'random seeds' for the simulator.

For a given TLP (set of variables described above), the simulator SUMO outputs several indicators.

- Accumulated Waiting-Time.
- CO : Carbon Monoxide
- CO_2 : Carbon Dioxide
- HC : Hydrocarbon
- NOx : Nitrogen Oxide
- PMx : Particles
- Fuel.

Those indicators define the available objectives for the optimization process.

2.2.3 Algorithms

For all the following discussions, an 'experiment' refers to a set of independent 'runs' of the optimization process led with several 'random seeds'. This behavior has been designed to address the variability induced by the evolutionary algorithm.

Mono-Objective optimization process

The algorithm used in this part is described as :

Algorithm 1: GA ($\mu + \lambda$)

Population size - Parents:	μ ;
Population size - Offsprings:	λ ;
Selection:	Pairwise tournament. ;
Crossover:	Two-points crossover. ;
Mutation:	Uniform mutation. ;

The two-points crossover has been chosen in order to take benefit of swapping intersections. A strategy well suited for one intersection might have good performances for another one. The mutation operator applies variation in a small range around the previous value of a gene. This behavior has been chosen in order to avoid destroying an existing good solution by applying too strong mutations.

Multi-Objective optimization process

Below is a list of MOEAs described by Coello (2007):

- Vector Evaluated Genetic Algorithm (VEGA)
- Vector Optimized Evolution Strategy (VOES)
- Weight Based Genetic Algorithm (WBGA)
- Indicator Based Evolutionary Algorithm (IBEA, IBEA II)
- Multiple Objective Genetic Algorithm (MOGA)
- Niched Pareto Genetic Algorithm (NPGA, NPGA2)
- Non-dominated Sorting Genetic Algorithm (NSGA,NSGA II)
- Distance-Based Pareto Genetic Algorithm (DPGA)
- Thermodynamical Genetic Algorithm (TDGA)
- Strength Pareto Evolutionary Algorithm (SPEA, SPEA2)
- Multi-Objective Messy Genetic Algorithm (MOMGA-I, II, III)
- Pareto Archived Evolution Strategy (PAES)
- Pareto Enveloped Based Selection Algorithm (PESA, PESA II)

Among all those MOEAs, this thesis focuses on two algorithms : Nsga-II, Deb et al(2000) [21] and Spea-II, Zitzler et al (2001) [19]. Those algorithms have been chosen because they are fast and elitist multi-objective genetic algorithms capable of finding multiple Pareto-optimal solutions during a single run. They have the three following features:

- An elitist principle
- An explicit diversity preserving mechanism
- Non-dominated solutions are emphasized

Both of them rely on the notion of dominance defined in the introduction. As the dominance relationship is not a total order, those algorithms use the notion of distance to discriminate two individuals mutually non-dominant. The distance is a way to both:

- Maintain a good genetic variability in the population.
- Discriminate two individuals having the same rank.

Those notions are illustrated in figure 2.4

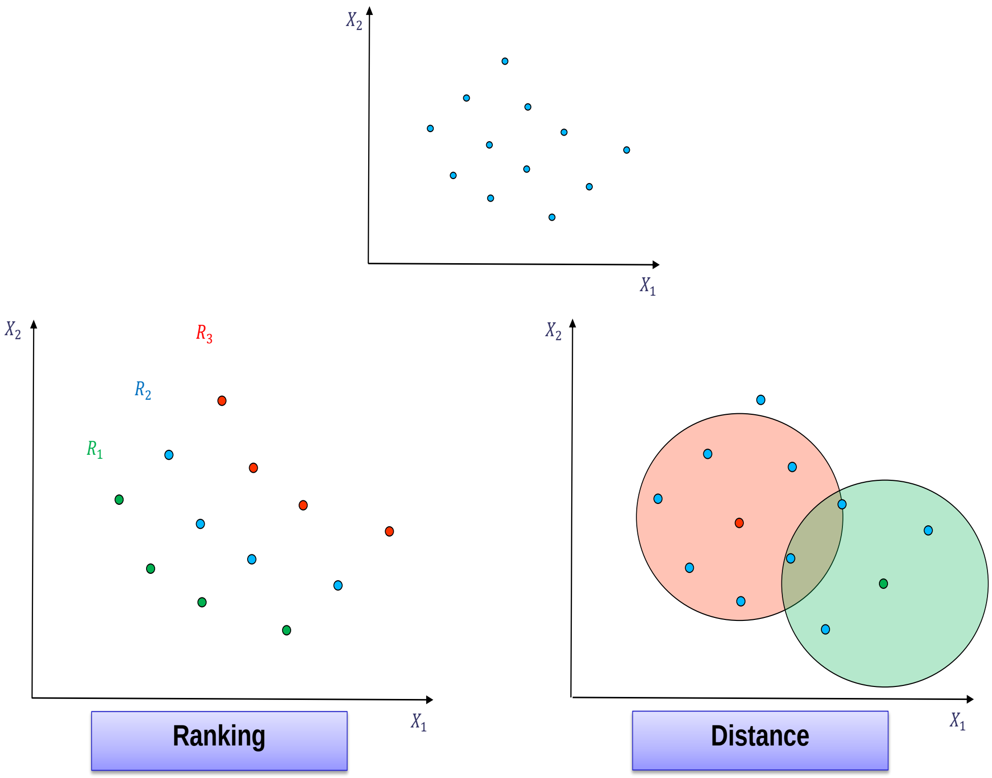


Figure 2.4: Dominance and Crowding Distance

a) Ranking: For the considered population on the two-dimensional space, this graph presents the ranking strategy. The green points form the non-dominated set, they are given the rank 1*. The blue points are non-dominated if the rank* 1 *is discarded, they form the rank* 2*. The red points are non-dominated if the ranks* 1 *and* 2 *are discarded, they form the rank* 3*.*

b) Crowding Distance: Two points with identical rank cannot be discriminated. We thus have to use an additional feature: The crowding distance. This distance measures the density of the solutions close to the considered solution. Solutions in high populated area will be penalized whereas solutions in empty areas will be promoted. Here the green point will be promoted whereas the red point will be penalized

Nsga-II In the algorithm described by Deb et al(2000) [21], the first step of the algorithm is to create an initial population P_i. The population is then sorted using the pre-defined notion of dominance. The non-dominated individuals form the first rank, the individuals only dominated by the first front form the second rank etc. See figure 2.4.

The evolution operators are then applied to the ranked population. The probability for an individual to take part in the evolution depends on both its rank and the distance to it's nearest neighbors. The new population is evaluated and a selection operator, also based on dominance and distance, is applied on both the parents and the offspring population to ensure the elitism.

See figure 2.5

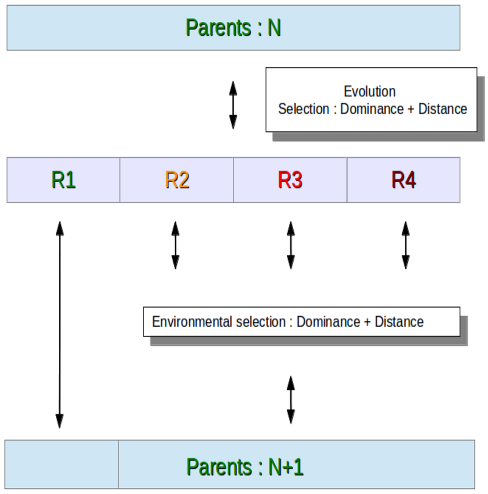


Figure 2.5: Evolutionary strategy

The algorithm can thus be seen as:

Algorithm 2: MOGA $(\mu + \lambda)$

Population size - Parents:	μ ;
Population size - Offsprings:	λ ;
Mating Selection	Tournament (DCD). ;
Crossover:	Two-points crossover. ;
Mutation:	Uniform mutation. ;
Selection:	DCD (Dominance + Crowding Distance). ;

Figure 2.6 presents the flowchart of Nsga.

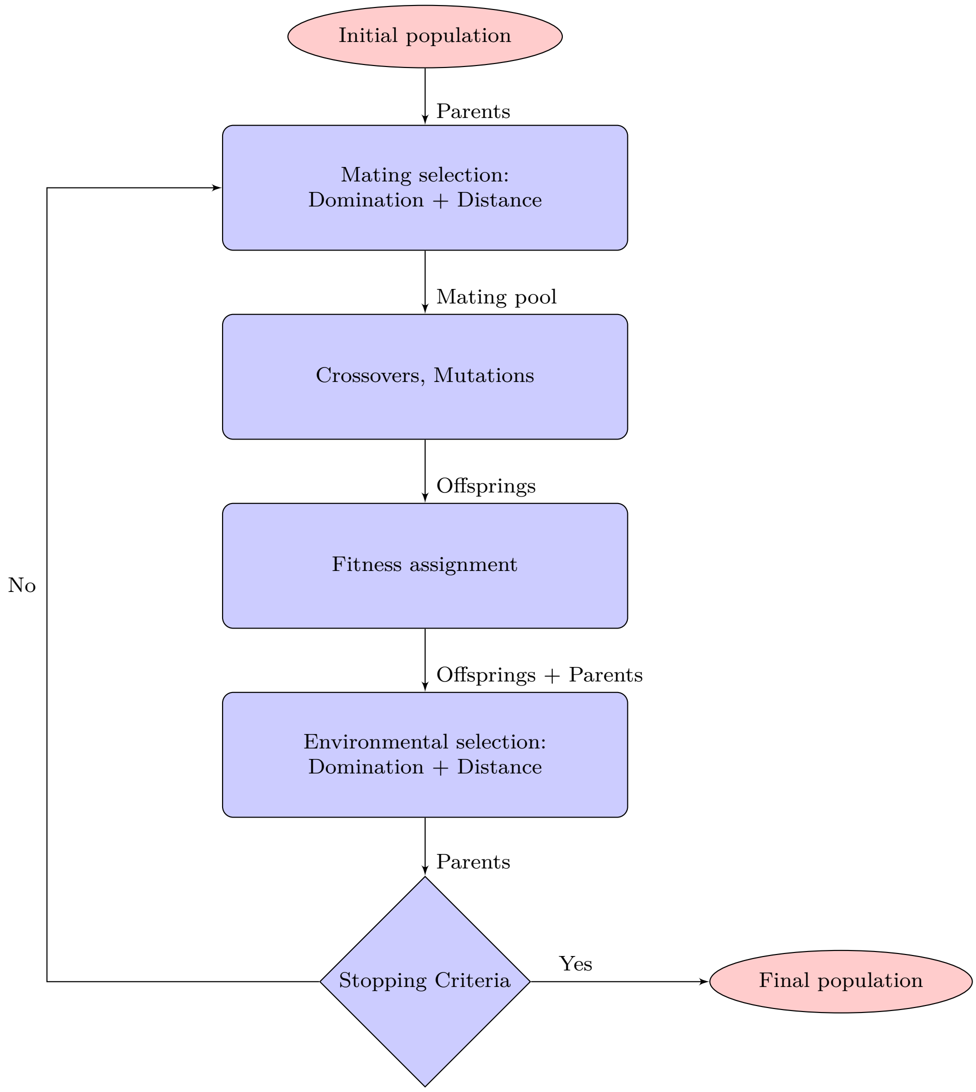
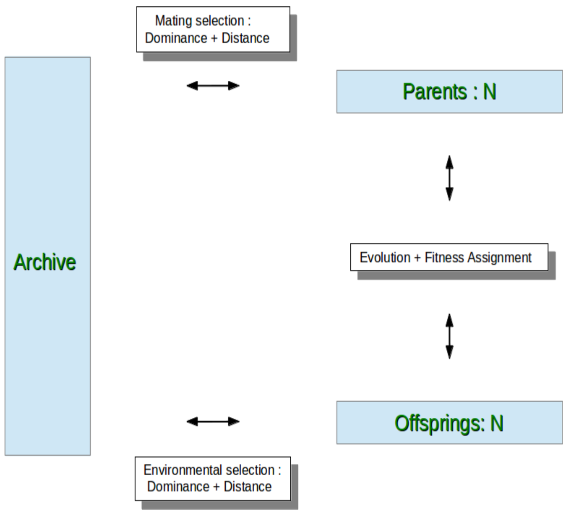


Figure 2.6: Flowchart: NSGA

SPEA-II The Strength Pareto Evolutionary Algorithm (SPEA) proposed by Zitzler et al (2001) [19] is very similar to NSGA. The main difference lies in the mating selection operator's behavior. Instead of gradually improving a population, SPEA gradually improves an archive. The mating selection operator is applied on the archive which plays the role of a memory.



(a) Evolutionary strategy

Figure 2.7: Evolutionary strategy

The algorithm can thus be seen as:

Algorithm 3: MOGA $(\mu + \lambda)$

Population size - Parents:	μ ;
Population size - Offsprings:	λ ;
Mating Selection	Binary tournament (DCD). ;
Crossover:	Two-points crossover. ;
Mutation:	Uniform mutation. ;
Selection:	DCD (Dominance + Crowding Distance). ;

Figure 2.8 presents the flowchart of Spea.

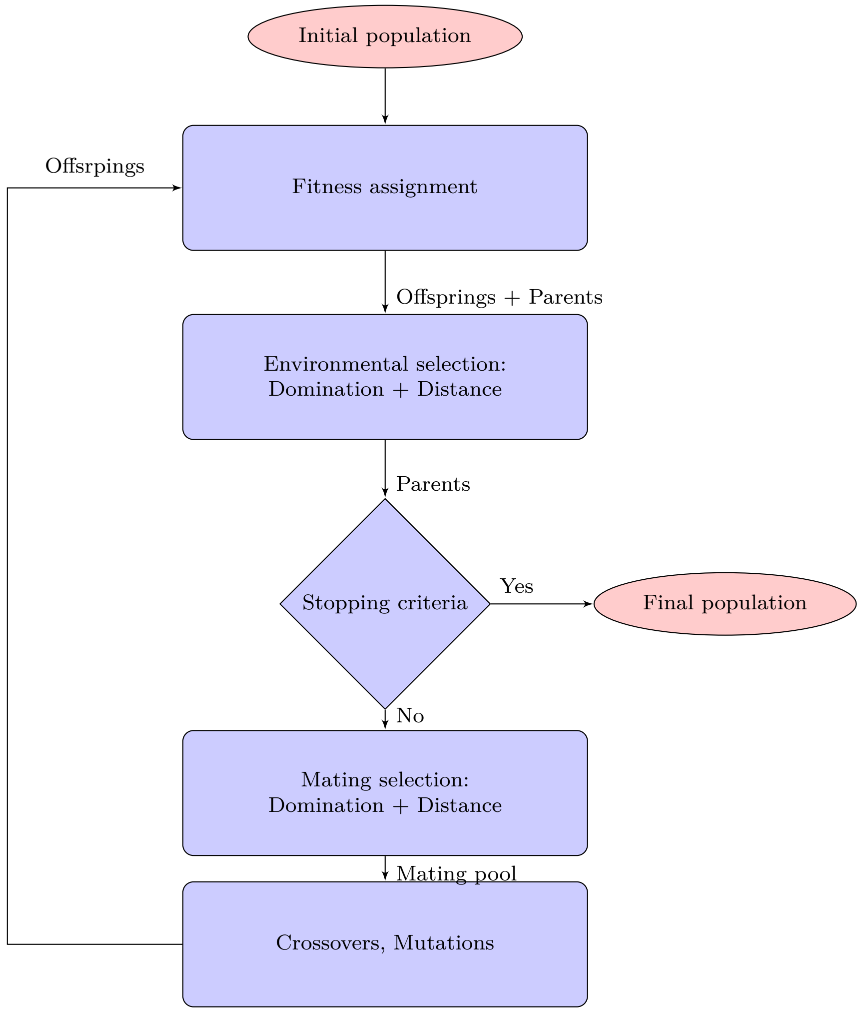


Figure 2.8: Flowchart: SPEA

2.2.4 Contributions

In order to address the research question, ***Is it possible to use a GA to achieve important gains within a reasonable time for a medium-sized network.***, four main techniques have been studied, each of which speeds up the optimization process by improving the algorithm or pre/post-processing some previously-computed results.

Those choices will be justified in the corresponding sections and the methodology for each of those approaches will be described.

2.3 Objective Selection

One of the main problem of the multi-objective optimization lies in the degradation of the performances in high dimensions i.e: With a large number of objectives.

Hughes (2005) [36], Purhouse et al (2003) [37], Khare (2003) [38]. Yet, some studies tend to indicate that this problem is reduced when the objectives are correlated Ishibuchi, (2013).

The traffic simulator is considered as a black box. Yet, one might think that several of the objective described above are correlated. It would come as no surprise to observe that the CO_2 and the CO emissions are connected.

The goal of the study is to provide supporting tools to help the traffic managers to choose which objective should be optimized among all the available objectives. The idea investigated here might be stated as: Is it possible to compensate the loss of information induced by discarding an objective by highest performances in the remaining sub-space? If so, optimizing the traffic in this sub-space will save time.

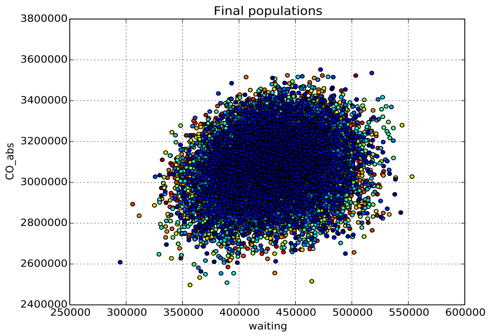
This question arises since the time required to optimize the traffic grows with the number of objectives. In the considered algorithms, no preferences are defined for the different objectives. Thus the algorithms will try to improve every objective and devote some time for it.

2.3.1 Methodology

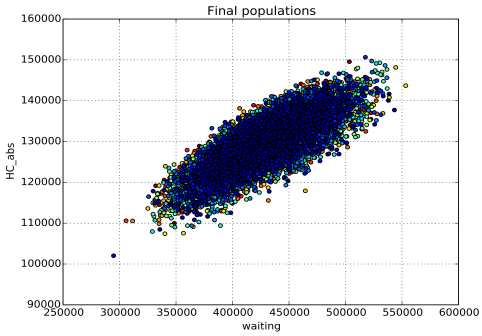
The measure of the correlation between two objectives is a complex, ill-defined and dynamical problem. Two objectives might be correlated in a region of the search space and anti-correlated in another.

The objective selection proposed here is based on the pair-wised correlation between different objectives for randomly-chosen points. As the initial point for the optimization, which has obtained using field data, is close to the area covered by randomly chosen points, this approach has been chosen to study the possible correlation of the different objectives in a close region around the initial point.

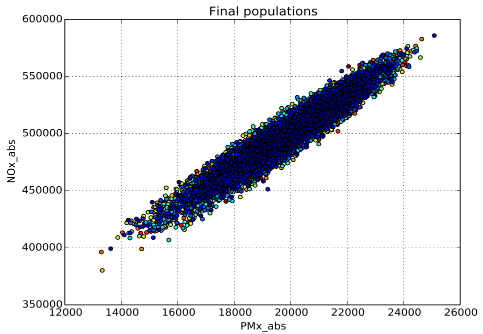
22000 randomly-chosen TLPs have been evaluated with respect to all the objectives. In order to measure the pair-wised correlation of the objectives those results have been plotted in a two dimensional space. Figure 2.9 shows the four different kinds of shapes observed for the so-called 'random-clouds'. In figure 2.9a, nothing indicates that the two objectives are likely to evolve in the same way whereas in figure 2.9b and 2.9c an evolution in one objective is likely to come with the same evolution with respect to the other. The objectives are said to be correlated. Figure 2.9d shows an example of antagonistic objectives. A decrease of an objective is likely to come at the cost of an increase for the other one.



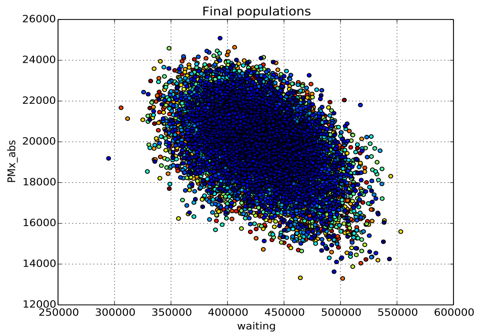
(a) Objectives: Waiting - CO



(b) Objectives: Waiting - HC



(c) Objectives: PMx - NOx



(d) Objectives: Waiting - PMx

Figure 2.9: Random clouds composed of 22000 *randomly chosen points for different pairs of objectives.*

Using this approach, a so-called 'correlation matrix' has been built. This matrix reflects the correlation of the objectives in an close area around the initial point.

In order to go further and assess the reliability of this 'correlation matrix', a dynamical study has been conducted to ensure that the correlation described in this matrix is confirmed by an experiment. In this experiment, the evolution process has been guided by a single objective, every N generations, the best individual has been re-evaluated for the second objective.

2.3.2 Results

Correlation matrix

Based on this pair-wised comparison, we have built the qualitative correlation matrix presented in figure 2.10.

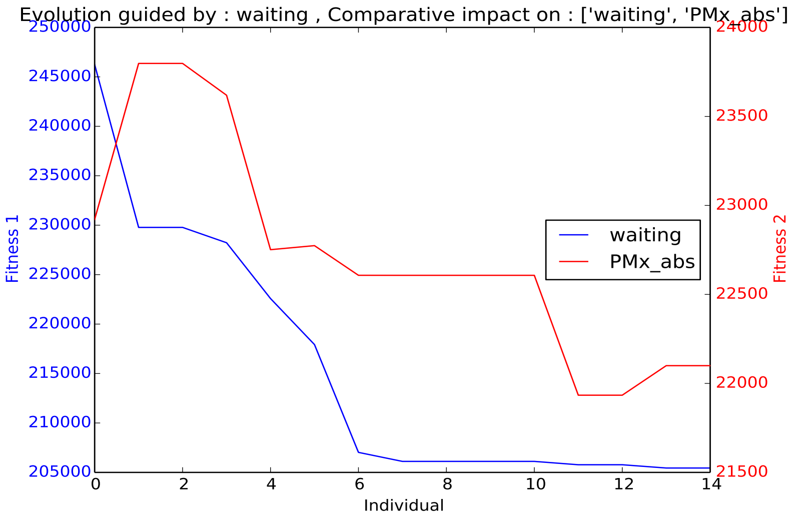
	Waiting	HC_abs	PMx_abs	NOx_abs	CO2_abs	Fuel	CO_abs
Waiting		+	-	X	X	X	X
HC_abs			X	X	+	+	+
PMx_abs				++	+	+	+
NOx_abs					++	++	++
CO2_abs						+++	+++
Fuel							++
CO_abs							

Figure 2.10: Matrix representing the correlation between the different objectives. Those correlations have been decided based on 22000 *randomly chosen points. A* $+$ *represents a correlation, a* $-$ *represents an anti-correlation and a* x *represents the absence of correlation-relationship*

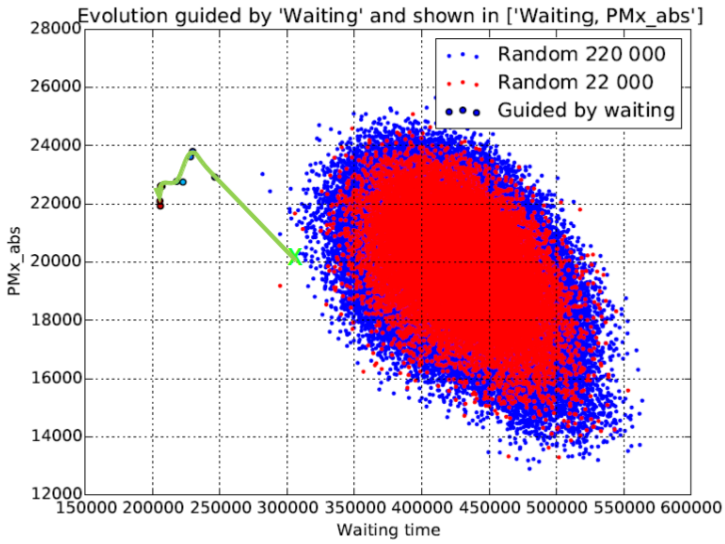
Dynamical study

The correlation matrix (figure 2.10) presents an anti-correlation between the objectives: Waiting-time and PMx emissions for the 22000 randomly-chosen points.

Figure 2.11 illustrates the dynamical behavior study for those two objectives.



(a) Comparative evolution



(b) Trajectory in the Objective-Space

Figure 2.11: Comparative evolution for two objectives.

a) Waiting-time and PMx emissions, function of the time for the mono-objective optimization process guided by the metric: waiting-time

b) Trajectory in the objective space. The green cross represents the initial point, the green arrow represents the trajectory described by the best individuals. The blue and red points represent 242000 *randomly chosen points.*

We observe that in the beginning of the optimization, the two objectives are strongly anti-correlated. This correspond to the result established in figure 2.10. A decrease of the waiting-time comes with an important increase for the PMx emissions. From a probabilistic point of view, this is coherent with the distribution of the randomly chosen points in the objective space. Indeed, the initial point (the green cross in figure 2.11b) is located in an area where it is likely to achieve better for the waiting-time at the cost of an increase for the PMx emissions.

Yet, this behavior changes during the optimization process. Starting from the second measured point, the two objectives are more correlated. In the objective space, the trajectory describes a shift almost orthogonal to the edge of the distribution of the randomly chosen points.

As a conclusion for this section, the matrix presented in figure 2.10 can be used as a basis to choose among all the available objectives which one should be used for the MOOP. Yet, this matrix describes a local behavior. Evidences show that during the course of the optimization the correlation between different objectives changes significantly.

We do not consider any further work in this direction for the topic is too vast to study in a extensive way and raises field-related questions. Yet, here are some suggestions to extent the work presented above:

- Measure the gains/loss achieved by discarding some objectives by conducting experiments in both cases and comparing the final gain achieved.
- Study the dynamical correlation of the different objectives during the course of the optimization.

For the following, the multi-objective optimization process was limited to two objectives.

2.4 Parameter tuning

As stated in the introduction, there is no consensus on the best evolutionary operators for a GA applied to a specific optimization problem. Hence, for every optimization problem an experimental campaign has to be held in order to assess the relative importance of those operators. More precisely, under some assumptions, a general-purpose universal optimization strategy is theoretically impossible, Wolpert (1997), [2]. Thus, one of the first step toward the optimization of the algorithm is the tuning of its parameters.

2.4.1 Methodology

Every parameters of the GA has been studied through an experimental campaign based on empirical rules for both the objectives: Waiting-Time and CO_2 emissions for both the mono and multi-objective algorithms.

Mono-objective

In this section different values have been used for each parameter. The base case was obtained thanks to an estimate of the optimal static values for a comparable algorithm implemented by Y. SEMET, research engineer at Thales Research and Technology. Empirical variations have been applied for each parameters, all other things remaining equal.

The parameters that have been studied are:

- The population size λ, μ
- The mutation parameters $[p_m, p_c]$

Multi-objective

In this section different values have been used for each parameters. The base case was obtained with empirical rules based on the optimal values obtained for the mono-objective optimization process. For every considered parameters, empirical variations have been applied all other things remaining equal.

The parameters that have been studied are:

- The algorithm $NSGA-II$, $SPEA-II$
- The population size λ, μ
- The mutation parameters $[p_m, p_c]$

In order to ensure that the comparison between the two algorithms is as fair as possible different archive sizes have been tested for $SPEA-II$.

2.4.2 Results

Mono-Objective Optimization

The experimental results are available in the appendix. The conclusions presented in this section have been based on the dynamical behavior and the final gain obtained by the different versions.

- An average offspring population size achieves better than a small or a high one. Indeed $\lambda = \mu$ achieved better than $\lambda = 0.5\mu$ and $\lambda = 2\mu$ for both the objectives: Waiting-Time and CO_2 emissions.
- An important mutation probability achieved better than an important crossover probability for both the objectives: Waiting-Time and CO_2 emissions. Yet, the crossover operators still improves the results. $[p_m = 0.75, p_c = 0.25]$ achieved better than $[p_m = 1, p_c = 0]$ and $[p_c = 0.5, p_m = 0.5]$ in every cases.

Multi-Objective Optimization Problem

The experimental results are available in the appendix. The conclusions presented in this section have been based on the dynamical behavior and the final gain obtained by the different versions but also on the final Pareto-front obtained.

- NSGA-II achieves better than SPEA-II in every cases whichever the archive size for SPEA.
- An important offspring population size achieves better than a small or a medium one. Indeed $\lambda = 2\mu$ achieved better than $\lambda = 0.5\mu$ and $\lambda = \mu$ for both the objectives Waiting-Time and CO_2 emissions. This suggests that the selective pressure...
- An important mutation probability achieved better than an important crossover probability for both the objective: Waiting-Time and CO_2 emissions. Yet, the crossover operators still improves the algorithm. $[p_m = 0.75, p_c = 0.25]$ achieved better than $[p_m = 1, p_c = 0]$ and $[p_c = 0.5, p_m = 0.5]$ in every cases.

2.4.3 Discussion

The experimental campaigns led for both the mono and the multi-objective optimization process allow the algorithm to perform well. More precisely, we have an estimate of the optimal setting for the algorithm. Another step toward the optimization of the algorithm lies in its ability to adapt its parameters to the current state of the optimization in an automatic fashion.

2.5 Control Strategy

An important part for GAs lies is the way the exploration/exploitation problem is addressed. For a given number of 'evaluation processes' (or for a given time) the results of the optimization will differ whether the algorithms tries to explore new solutions or improves the existing ones.

This issues is partially addressed in the trade-off problem between the crossover and the mutation operators. The crossover operator tends to exploit existing solutions whereas the mutation operator tends to explore by creating new solutions. Yet, the exploration/exploitation trade-off is a dynamical problem.

This section investigates different kind of control strategy for the mutation strength parameter to address this issue for the traffic optimization.

2.5.1 Methodology

So far the mutation strength P_g has been kept constant. In order to study the impact of the mutation strength on the optimization process several control strategies have been tested.

In this part the mutation strength is not kept constant throughout the optimization but evolves according to a time-dependent function. The underlying hypothesis is that as times goes by, the solutions found by the optimization process are more and more structured and thus the mutation strength should decrease in order to avoid destroying good solutions by applying too strong mutations. The different control strategies used in this thesis are presented in figure 2.12.

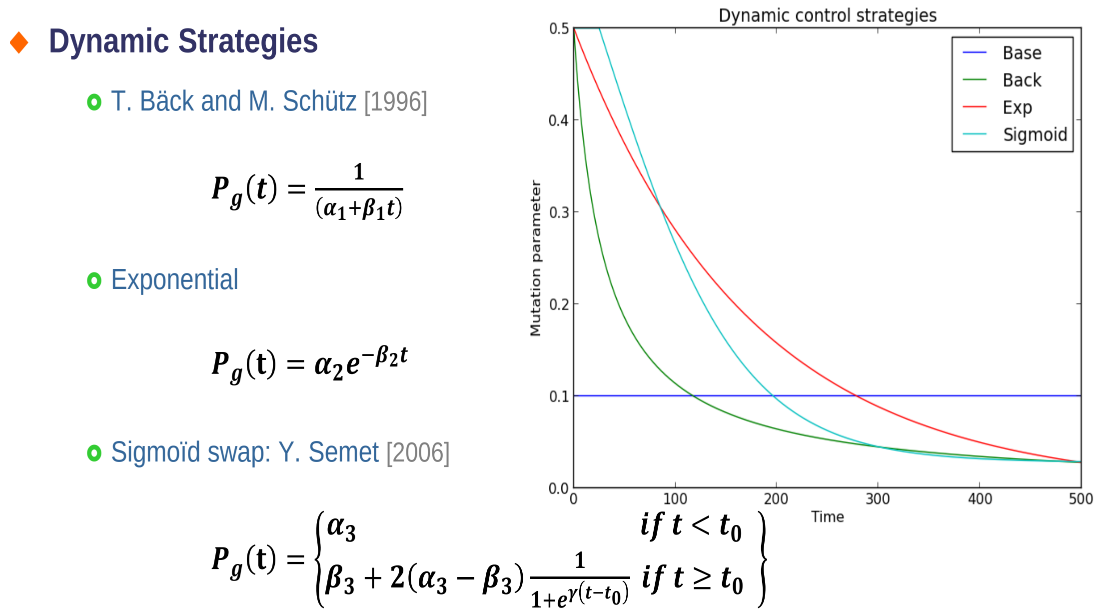


Figure 2.12: Different time-dependent strategies used in thus thesis. The graph presents the evolution of the mutation parameter for all the different time-dependent strategies used in this chapter.

For the first strategy coined by T.Back and M. Schutz [39], two parameters can be used. They have been used to set the upper and lower bounds for the mutation parameter. The same goes for the exponential strategy proposed in this thesis.

The sigmoid swap proposed by Y. SEMET and M. Schoenauer, (2006) [40] has two additional degrees of freedom. Those parameters can be used to tune the offset and the dynamic of the evolution of the mutation parameter.

Although the use of time-dependent control strategies allow us to improve the algorithm, it requires an additional effort to calibrate the control law. Indeed some parameters need to be tuned/optimized. This makes those techniques difficult to use since traffic managers cannot always afford spending time finding the proper set of parameters for every traffic situation.

Moreover, this paradigm supposes that the optimal parameter setting depends only on the time. Yet, the optimal strategy might also depend on the location of the current solution in the search space, its closest neighbors in both search and solution spaces, the network itself, etc.

Another approach lies in the auto-adaptive control strategy. Back, (1992) [41]. The idea here is to define a mutation parameter for each individual. This/those parameter(s) is/are included within the genotype of the individuals and will thus evolve with the individuals throughout the evolutionary process. The underlying assumption is that individuals with optimal mutation parameters will, in the long-run, have better probabilities of achieving good performances than the other individuals. They will thus be more likely to be selected for the following of the optimization. With the biologic analogy those individuals are supposed to have a 'selective advantage'.

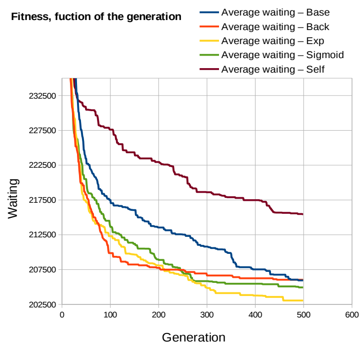
2.5.2 Results

All of the strategies described above have been tested on the case study for both the mono and multi-objective algorithms. The results are available in the appendix.

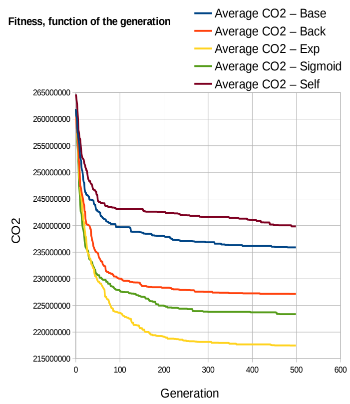
Mono-objective

We observe in figure 2.13 that the exponential and the sigmoid operators achieve the best performances. As we can see in figure 2.12, those two time-dependent laws always have, for the considered setting, a higher value for the mutation parameter than the strategy described by back. This tends to indicate than the back operator do not explore enough the search-space during the optimization. On the other hand the sigmoid swap operator do not seem to benefit from the higher mutation rate in the beginning of the optimization compared with the exponential operator. The exponential law seems to provide the best trade-off for the exploration/exploitation.

On the other hand, the auto-adaptive strategy achieves the worst performances both in terms of final gain and of dynamic behavior. Yet the comparison should be done with a number of precautions. Indeed the experiment have been led all other things remaining equal. The proposed analysis of the observed behavior lies in the insufficient selective pressure with respect to the mutation parameters. An efficient self-adaptive control might require a more important selective pressure.



(a) Mono-objective - Waiting



(b) Mono-objective - CO_2

Figure 2.13: Mono-objectives: Different control strategies.

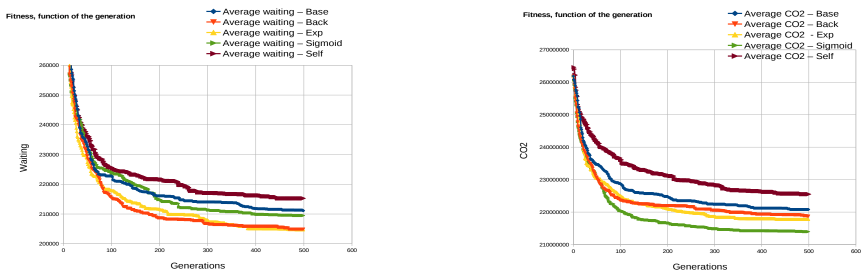
Evolution of the indicator function of the generation for different control strategies. The brown curve represents the self-adaptive control strategy, in blue the base strategy with constant value for the mutation parameter, in green the sigmoid operator, in yellow the exponential strategy and in red the back strategy.

Multi-objective

We observe in figure **??** that the exponential and the back operators achieve the best performances for the objective: waiting time, whereas the sigmoid operator achieves the best performances for the objective: CO_2. Those unexpected and unclear results show the complexity of the analysis for the MOOP.

In both case the exponential law achieves an interesting trade-off.

On the other hand, the auto-adaptive strategy still achieves the worst performances both in terms of final gain and of dynamic behavior, all other things remaining equal.



(a) Multi-objective Waiting *(b) Multi-objective CO_2*

Figure 2.14: Multi-objectives: Different control strategies.

Evolution of the indicator function of the generation for different control strategies. The brown curve represents the self-adaptive control strategy, in blue the base strategy with constant value for the mutation parameter, in green the sigmoid operator, in yellow the exponential strategy and in red the back strategy.

2.6 Inoculation Strategy

The term 'Inoculation' usually refers to the injection of a substance (usually a vaccine) for a patient. In order to keep the genetic analogy, 'Inoculation' will refer to the injection of a carefully-chosen initial point within the initial population of an EA.

This sections investigates whether the GA used in the optimization process benefits from the inoculation of a carefully chosen point. This question is not trivial since a GA is likely to converge too fast if the inoculant is located in, or close to, a local minimum. If the algorithms indeed benefit from the inoculation the time required to reach a given gain will decrease.

2.6.1 Methodology

In order to investigate this topic, several experiments have been led to study the resilience of the solutions with respect to variations in:

- The demand.
- The network.
- The objectives.

This aim of those experiments is to study if the solutions are resilient to changes in the problem definition.

Variation in the demand In this study, an experiment has been done with a demand described by an Origin Destination Matrix (ODM)[1] M_0, the best individual has been recorded and used as an inoculant for another experiment led with a modified ODM $M_0{}'$. This modified ODM has been obtained by applying variations on M_0 in a range $+/-30\%$ (corresponding to the variability observed on the traffic counts [2]. This variation in the demand is important and likely to change the relative importance of the traffic flows.

This experiment has been designed to study the resilience of good solutions with respect to variations in the demand. The underlying assumption is that the best solution implicitly provides some traffic flows with a high priority and that a variation in the demand will result in a small variation in the relative priority of the different traffic flows.

This study corresponds to the following use case: Traffic managers want to optimize the traffic knowing the optimized solution of the previous hour/day.

Variation in the network In this study, an experiment has been led for a normal traffic situation. The best individual has been recorded and used as an inoculant for another experiment led with an abnormal traffic situation. The abnormal traffic situation has been obtained by creating an accident involving two lanes within the network.

This experiment has been designed to study the resilience of good solutions with respect to variations in the network itself. The underlying assumption is that a small modification in the network can be addressed by slightly modifying an existing good solution. This experiment corresponds to the following use case: An accident occurred. A lane is stuck.

Variation in the objective This experiment has been led for the MOOP and measured the impact of the initial point on the performances of the algorithm when the objective changed.

In this study, an experiment has been led for a normal traffic situation with respect to some objective(s). The best individual(s) has/have been recorded and used as inoculants for other experiments, led with other objectives.

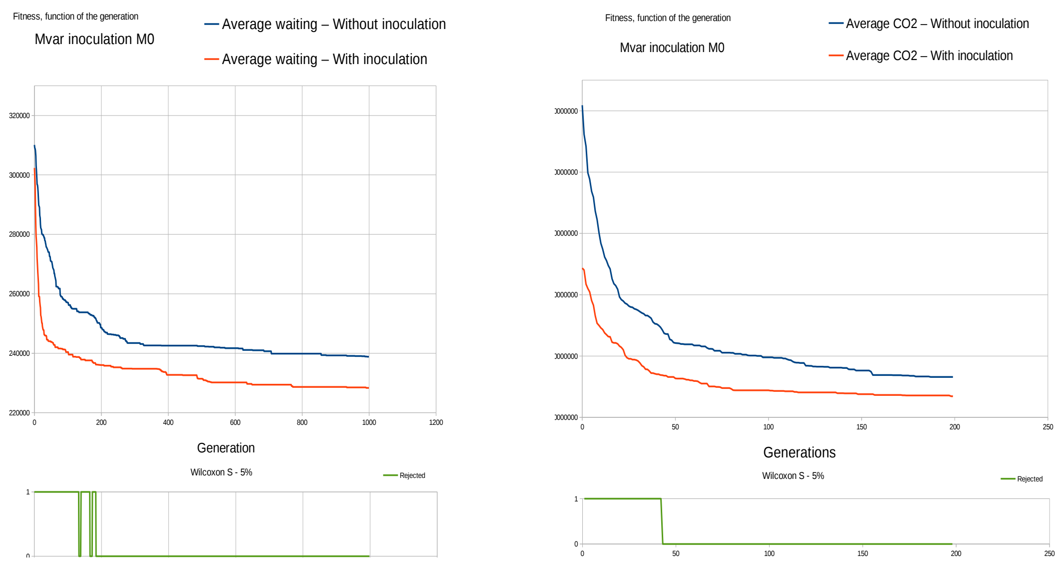
[1] ODM are more extensively described in chapter 3

[2] More information about the variability observed on the traffic counts will be presented in chapter 3

This study has been designed to investigate the resilience of the algorithm with respect to the desired objectives. This experiment corresponds to the following use case: A solution has been found for one objective, is it possible to add another objective?

2.6.2 Results

Variation in the demand Figure 2.15 presents the results of the inoculation for two different experiments, one with the objective: waiting-time (Figure 2.15a) and one for the objective: CO_2 emissions (Figure 2.15b).



(a) Waiting *(b) CO_2*

Figure 2.15: Inoculation - Different Demands.

Comparison of two experiments. In both the experiments, the demand is a modified version of the real demand. $(+/-30\%)$ on each ODM pair of M_0.

The first experiment (blue) is inoculated using the field Traffic Light Plan as an initial point

The second experiments (red) is inoculated using a previously-computed Traffic Light Plan good with respect to the considered objective.

This experiment shed light on two interesting phenomenons.

- For both objectives - Waiting-time and CO_2 - the inoculated version achieved better in terms of final gain and dynamical behavior.
- An interesting result lies in the shape of the slope for the objective: waiting time in the inoculated version. The very important slope indicates that the inoculated version achieved important improvements in a small amount of

time. This tends to indicate that the inoculant was located in an interesting region for the optimization.

- The difference between the two experiments (red and blue) is statistically significant in the beginning of the optimization process. It is thus profitable to inoculate the optimization. In the end of the optimization process the statistical test cannot ensure us that the difference is statistically significant.

The solutions are, to some extent, resilient to variations in the demand. This study consider only a single instance of the problem. To improve the reliability of those conclusions, additional test should be conducted for other benchmarks or scenarios.

Variation in the network Figure 2.16 shows the result of the experiment.

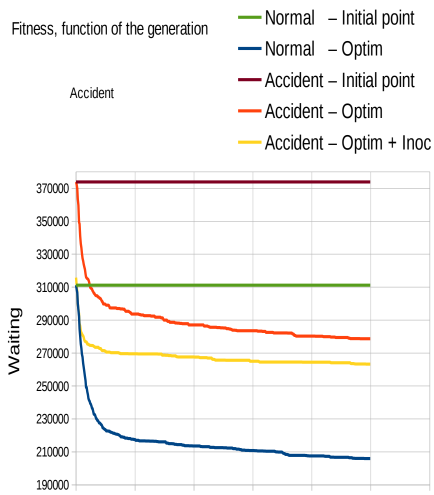


Figure 2.16: Result of the inoculation for the variation in the network.

The green curve presents the value of the initial point used for the normal traffic situation. The blue curve presents the result of the optimization starting from this initial point.

The brown curve presents the value of the initial point (the same one) used for the abnormal traffic situation. The red curve presents the result of the optimization in this configuration.

Finally, the yellow curve presents the result of the optimization in the abnormal situation starting from the point obtained with the optimization in a normal traffic situation (at the end of the 'bleu' curve).

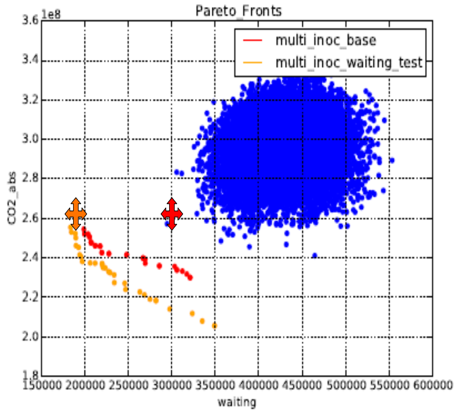
A comparative study between the red curve and the yellow curve shows that the optimization benefits from the inoculation.

- The final gain is more important for the inoculated version.
- An interesting result lies in the slope of the yellow curve at the beginning of the optimization. This important slope indicates that important gain are achieved in a small amount of time. This tends to indicate that the inoculant is located in an interesting region for the optimization

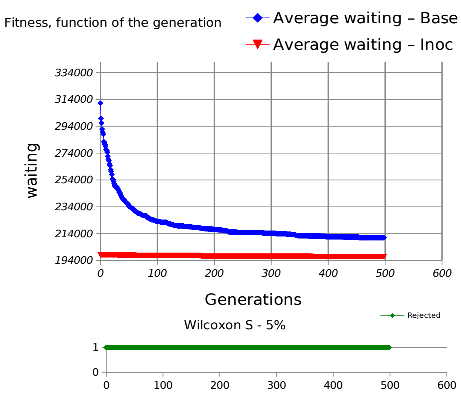
It is also possible to observe that the optimized solutions are more sensitive to the variation in the network than the non-optimized one. The optimized solutions are structured.

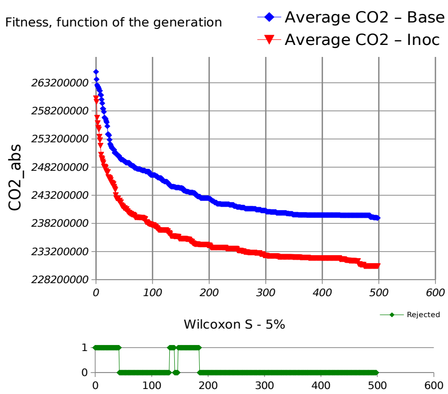
The solutions are, to some extent, resilient to variations in the network. Unfortunately, this study consider only a single instance of the problem. To improve the reliability of this conclusions, additional test should be conducted for other benchmarks or scenarios.

Variation in the objective Figures 2.17, 2.18, 2.19 shows the result of the experiment for different initial points.



(a) Pareto-Front

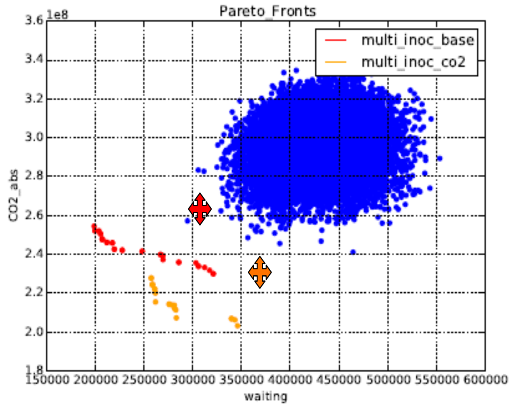




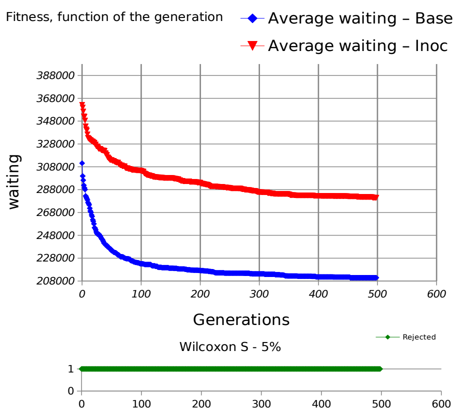
(b) Waiting *(c) CO_2*

Figure 2.17: Inoculation with a good solution with respect to the objective: Waiting-time.

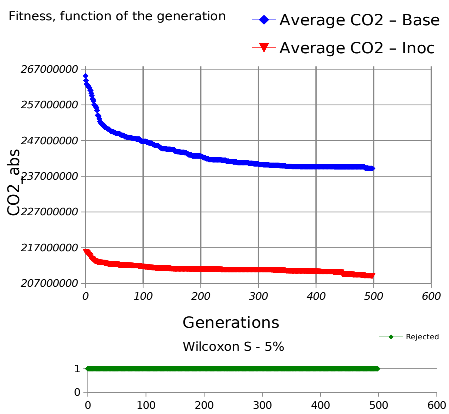
Comparison of two experiments. The first experiment (blue) is inoculated using the field TLP as an initial point. The second experiments (red) is inoculated using a previously-computed TLP good with respect to the waiting time.



(a) Pareto-Front



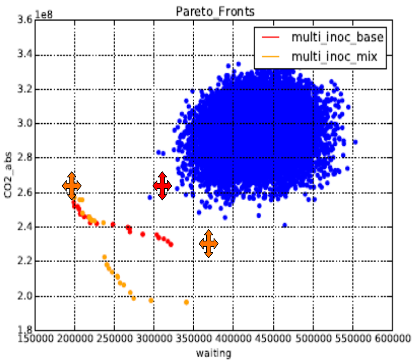
(b) Waiting



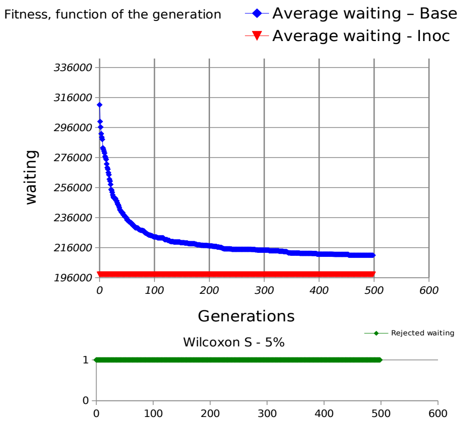
(c) CO_2

Figure 2.18: Inoculation with a good solution with respect to the objective: CO_2.

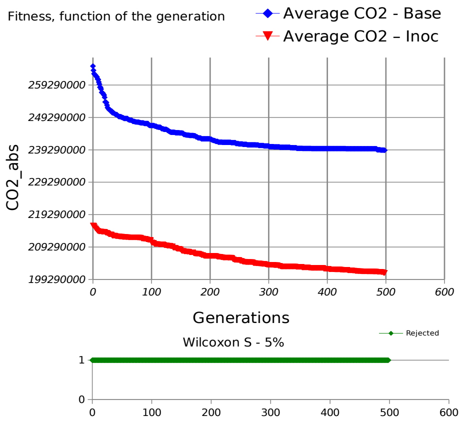
Comparison of two experiments. The first experiment (blue) is inoculated using the field TLP as an initial point. The second experiments (red) is inoculated using a previously-computed TLP good with respect to the CO_2.



(a) Pareto-Front



(b) Waiting



(c) CO_2

Figure 2.19: Inoculation with a good solution with respect to the objective: Waiting-time, and one with respect to to: CO_2:

Comparison of two experiments. The first experiment (blue) has been inoculated using the field TLP as an initial point. The second experiments (red) has been inoculated using two previously-computed TLP good with respect to the waiting time and the CO_2 emissions.

We observe different kind of results. In the first case, the inoculation results in enhanced performances for both of the objectives. This is a case of successful inoculation for the inoculant did not lead the process into a local minimum. Yet we observe that very little additional gains are achieved with respect to the objective: Waiting-time.

In the second case however, the inoculation results in enhanced performance for one objective and a lower performance for the other. This is a case of unsuccessful inoculation for the algorithm could not reach the same region of the search space. The algorithm does not take a complete advantage of re-using previously-computed results.

In the last case, with two inoculants, the algorithms take full profit of re-using previously-computed results. Even though the performance in term of Waiting-time did not improve much, a larger credit was devoted to the optimization of the CO_2 resulting in better performances.

The algorithm is resilient, to some extent, to a variation in the objectives. Unfortunately, this study consider only a single instance of the problem. To improve the reliability of those conclusions, additional test should be conducted for other benchmarks or scenarios.

2.7 Conclusions

Several strategies have been presented to improve the performances of the GA in order to achieve important gain in a limited amount of time.

This can be done by carefully choosing the objectives that are to be optimized, by finding the optimal set of parameters for the GA either with a static or dynamic strategy and by re-using previously computed solutions to seed the initial population of the algorithm.

Those results still have to be confirmed on other benchmarks.

Chapter 3

Demand Modeling Optimization

This chapter addresses the research question: ***Is it possible to calibrate the demand-related model of SUMO in order to reach a satisfactory behavior within a reasonable time for a medium-sized network using the GSM.***

3.1 Introduction

In the previous study, the demand has been kept constant and supposed to accurately represent the real demand. Yet, we would like to deepen the study of the demand-related model of SUMO in order to ensure that the simulator behaves as the real network. Indeed, the confidence in the results of the optimization is strongly dependent on the way the simulator behaves.

In this paper, the desired behavior of the simulator was stated as follow: ***The simulator is said to be reliable if, for a given traffic configuration, the observed flow counts on the real network equals the simulated flow counts in the simulator.***

The modeling of the demand for such a network is a challenging task. Several different modeling issues make it difficult to achieve this ideal behavior. Among the modeling issues listed by May (1990) [42], this following modeling errors have been observed in the study:

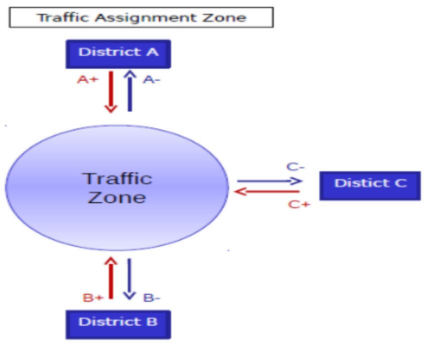
- The networks itself: The width of a lane, the maximum allowed speed or the position of the 'simulated' sensors might be inaccurate.
- The route choice model: Real drivers might use short-cuts.
- The demand-related model: The timing of the incoming cars is computed based on the demand and a distribution law (Uniform, Poisson, Time-dependent, etc...). Thus the model cannot capture fast or unpredictable changes. Plus, in the real network some cars 'depart' and 'park' within the network.

- The description of the features of the cars: Width, length, acceleration/breaking parameters, oil consumption etc..
- The drivers behavior: Reaction-time, impatience, infractions etc...

Considering all of those sources of modeling errors, this thesis investigates the way the demand is modeled in SUMO in order to reduce the spread between the simulated behavior and the real one.

3.1.1 Problem Definition

A common way of describing the demand is to use an ODM. Cascetta, Ennio and Nguyen, Sang (1988) [43] Yang et al (1992) [44], Cascetta et al (1993) [45]. An ODM is a matrix where each cell represents the number of vehicle going from a specific origin to a given destination. See figure 3.1 for more detailed description.



(a) Origin - Destinations

« Origin/Destination » matrix

	A	B	C
A	-	γ(B, A)	γ(C, A)
B	γ(A, B)	-	γ(C, B)
C	γ(A, C)	γ(B, C)	-

(b) Corresponding ODM

Figure 3.1: Origin Destination Matrix description.

In the ODM, figure 3.1b, the term $\gamma(B, A)$ describes the number of cars willing to go from A to B. The variable B and A being defined as in figure 3.1a

Creating this ODM is a challenging task in itself. With n origins/destinations there are n^2 unknowns. Collecting ODM information directly by conducting surveys or interviews is a very long, uncertain, and costly process. Other methods are model-based estimations, models are used to relate social, geographic and economics features to the ODM. Yet, those models usually require advanced data. Furthermore, those models need to be calibrated for the considered network.

Recently, a large number of studies have been devoted to indirect estimation of the ODM parameters using traffic counts. Those studies have mostly been motivated by the the relative ease of obtaining traffic counts compared with other more advanced data.

The notations used in this paper are described below:

The network is modeled by a directed graph $G(C, L)$ where C is a set of nodes and L is a set of links. $L' \subseteq L$ is the subset of monitored links. We are interested in finding the ODM $X = \{x_n\}$ which describes the travel demand for each OD pair $n \in N$, with N the number of OD pairs.

We use field traffic-counts observed on the real network $\tilde{Y} = \{\tilde{y}_{l,t}\}$ where $\tilde{y}_{l,t}$ is the number of car going through link l at time t. Finally we also use an evaluation function Y that computes the simulated flow counts for a corresponding ODM: $Y : X \longrightarrow Y(X) = \{y_{l,t}\}$. Where $y_{l,t}$ is the number of car traveling on link l at time t for the current estimation of the ODM X.

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$$

(a) Origin Destination Matrix

$$\tilde{Y} = \begin{bmatrix} \tilde{y}_{1,1} \\ \vdots \\ \tilde{y}_{L',1} \\ \vdots \\ \tilde{y}_{l,t} \\ \vdots \\ \tilde{y}_{1,T} \\ \vdots \\ \tilde{y}_{L',T} \end{bmatrix}$$

(b) Traffic Counts

The optimization problem can be stated as:

$$X^* = \underset{X}{\operatorname{argmin}}\, Z(X) = \underset{X}{\operatorname{argmin}}\, F(Y(X) - \tilde{Y})$$

Where F measures the distance between the desired and observed flow counts Y and $\tilde{Y}$.

The ODM estimation problem is composed of two sub-problems:

- Traffic assignment. The purpose is to map OD flows described in X to link counts $Y(X)$.
- Computation of a new ODM X^* using available traffic counts in the network, $Y(X)$ in order to reduce the distance between the simulated counts and the real one.

An iterative process between those two steps is required to optimize the ODM.

The traffic assignment is usually obtain through traffic simulation. It is possible to map each drivers to a link if the drivers route choice behavior, the travel times through each link of the network and the demand are available.

Unlike the non-congested case, there is no linear relation between the ODM and the traffic counts for the congested case. This is described by Cascetta et al (2001) [46]. The travel times to go through a link depend on the current state of the traffic and are thus time-dependent. In addition, some highly unpredictable and non-linear phenomenon might occur.

Finally, a last important notion is the user equilibrium. The user equilibrium is defined as ***the point where drivers maximize their perceived utilities across all feasible routes. No driver can reduce his travel time by unilaterally taking another route.***

As stated above, the traffic simulator used in this paper is supposed to be a black box. The model used in SUMO is based on the Dynamic User Equilibrium (DUE) designed by Gawron (1998) [47] which has been proven to be relevant by Behrish et al (2008) [48].

The problem addressed in this thesis is the off-line estimation of static ODM with no previous information on the ODM for a congested medium-size network using the Dynamic User-Equilibrium.

3.1.2 State of the art

Several techniques have been studied to solve this problem.

A first family is composed of the traffic modeling based approaches. This approach has historically been the first proposed. Among those models figures the Gravity (GR) model based approach and the Gravity-Opportunity (GO) among others. In those approaches the ODM estimation problems is reduced to the estimation of some modeling parameters and the target ODM is generally assumed to be close to an old ODM. The distance between the historic and the target ODM is minimized subject to the flow constraints. Such a work has been presented by Van Zuylen and Willumsem (1980) or Fisk (1988).

The bi-level approach has also been investigated because it ensures the interdependency between the ODM and the traffic assignment. A general formulation of bi-level optimization can be written as follow:

$$\min_{x \in X, y \in Y} F(x, y)$$

$$s.t. G_i(x, y) \leq 0, \forall i \in 1, 2, ..., I$$

$$y \in \operatorname*{argmin}_{z \in Y} f(x, z)$$

$$s.t. g_j(x, z) \leq 0, \forall j \in 1, 2, ..., J$$

Where F represents the upper-level objective function, and f represents the lower-level objective function. Similarly, x represents the upper level decision vector and y represents the lower level decision vector.

Because the number of unknown in ODM is usually more important than the number of equations obtained on the monitored streets, the ODM estimation problem is under-defined. Several ODM could produce the desired traffic counts. Thus an additional constraint should be taken into account. This constraint is usually addressed in the upper-level problem using one of the statistical techniques such as Maximum Likelihood (ML), Generalized Least Square (GLS) or Bayesian Inference (BI) approaches. Those techniques aims at minimizing the sum of error measurement in traffic counts and ODM with the field measurements and the historical ODM.

The lower level problem represents the stochastic user equilibrium constraint.

Fisk (1988) [49] used entropy maximum models with equilibrium condition to construct the bi-level programming problem. Heuristic iterative algorithms for the bi-level estimation problem solution were proposed by Spiess (1990), Yang et al. (1992), Yang (1995), Yang and Yagar (1995) and Maher and Zhang (1999).

Yang Sasaki and Lida (1991) [50] explicitly take the uncertainties in both the target ODM and traffic counts into account to solve the bi-level programming problem with the GLS approach. The GLS estimator, is under some assumptions, the best linear unbiased estimator (BLUE) of the ODM, i.e. the estimator of minimum variance in the class of all unbiased estimators linear in the ODM and the traffic counts.

Bi-level programming problems are usually difficult to solve because the upper-level objective requires the solution of the lower-level optimization problem. Furthermore, because the lower-level problem is in general non-convex. Thus, the algorithm is likely to find local optimums and finding the global optimum is in general difficult. Friesz et al (1990), Yang et al (1994).

Kim et al. (2001) and Al-Bataineh and Kaysi (2006) used GAs as an alternative approach for solution of the bi-level ODM estimation problem.

An other approach is the GSM. The problem formulation might be stated as follow:

$$\min Z(x) = \frac{1}{2} \sum_{l \in L} (y_l - \tilde{y}_l)$$

$$s.t y = assign(x)$$

where the pseudo-function $assign(x)$ is used to indicate the link traffic flows y resulting from an assignment of the ODM x, L is a set of the network links, $\tilde{y}$ is a vector of observed link traffic flows. Spiess (1990) adopted this approach and proposed to multiply the negative gradient by the ODM in order to ensure that the zero elements of the ODM remain unchanged.

Kolechkina and Toledo (2010) [51] used this technique to solve the ODM estimation problem. This thesis will adapt and extend the work presented in Kolechkina and Toledo (2010) [51].

3.1.3 Background Material

An important preliminary work has been done by Y. SEMET, research engineer in Thales Research and Technology who supervised this study. Thus, the network itself was already created in a SUMO-compliant format, in addition, the field data have already been collected. They were thus usable with little additional efforts.

Yet, the data processing and the definition of the success criteria have been done during this study.

Although the work presented by Kolechkina and Toledo (2010) [51] has been adapted and extended for the GSM presented in this paper, several important adaptations have been made.

The SSA and the MSA are part of the contributions of this thesis.

3.2 Methodology

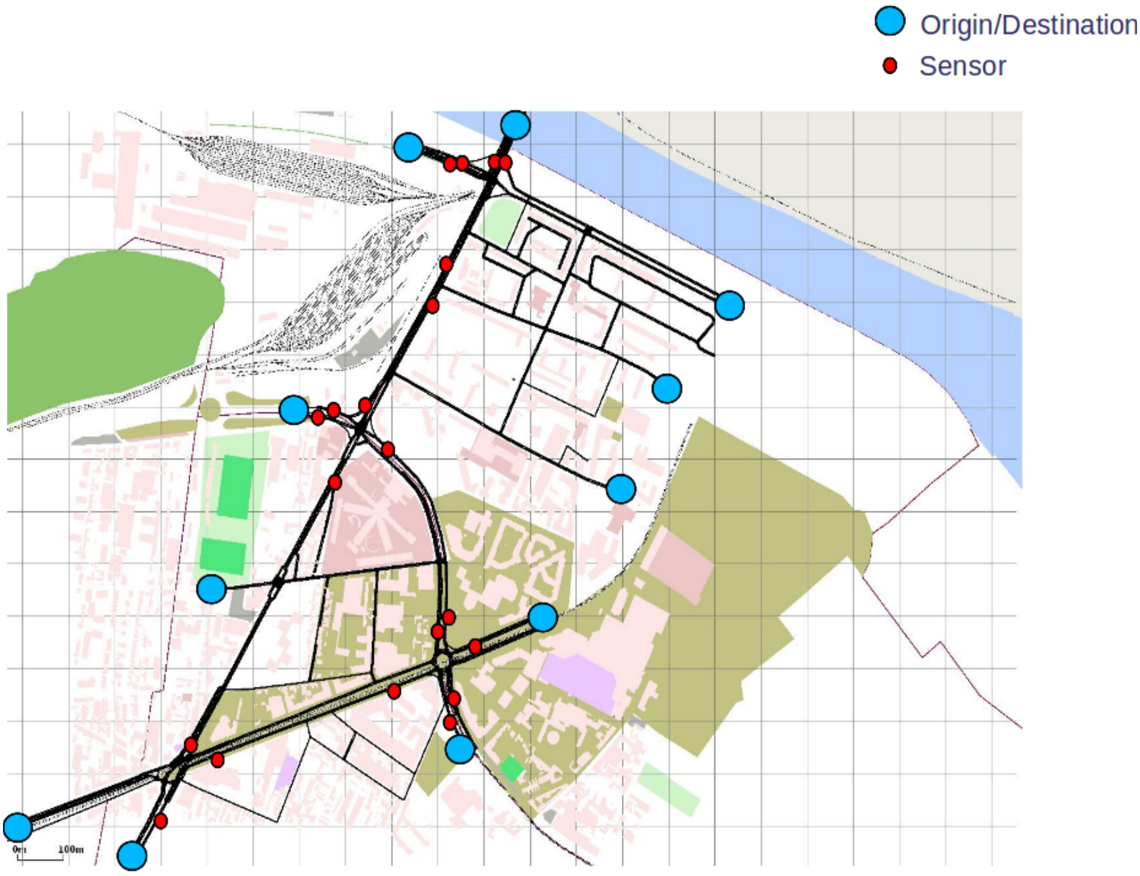
3.2.1 Simulator

The thesis used the microscopic simulator called SUMO for it is an open source traffic simulation package including the simulation application itself as well as supporting tools, mainly for network import and demand modeling. SUMO helps to investigate a large variety of research topics, mainly in the context of traffic management and vehicular communications. Krajzewicz et al (2012) [34]. Other simulators are presented in Boxill et al (2000) [35].

One should note that the simulator is stochastic, i.e: two simulations with the same network, the same demand and the same traffic light setting might give two slightly different results. This is mainly due to uncertainty in drivers behavior and injection times. To address this uncertainty, every 'evaluation process' refers to the aggregation of several simulations led with different 'random seeds' for the simulator.

3.2.2 Network

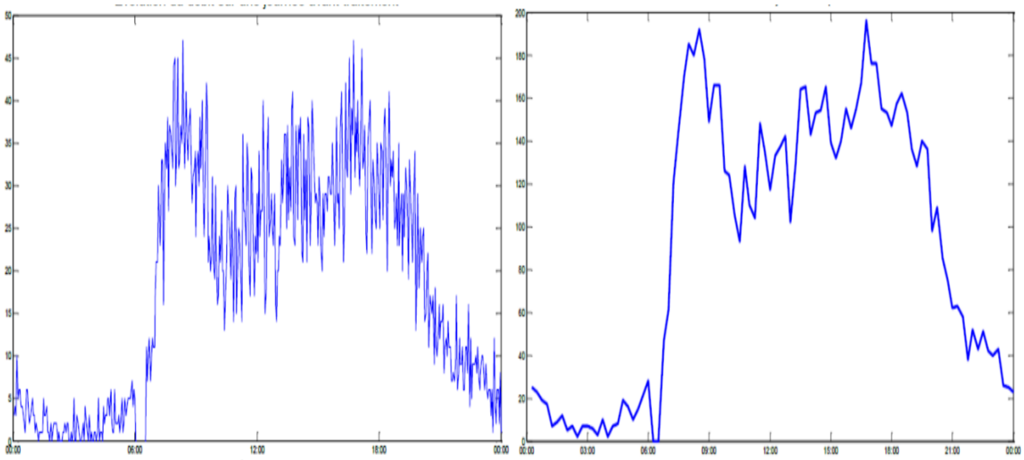
The case study represents the center of the town of Rouen, France. This network is composed of 11 intersections controlling 168 traffic lights (see figure 1.1). I define 11 origins. Without loss of generality, every origin point is supposed to be an available destination point. The network is covered with 20 sensors. Those origins/sensors are described in figure 3.3a. Some important areas of the network are not covered with any sensor. The relatively small number of sensors used in the benchmark is due to both the coverage of the real network and the reliability of the data produced by those sensors.



(a) Origins/Destinations

Figure 3.3: Positions of the sensors and the origins/destinations of the network.

The data used for all the experiments shown below have been collected during one year. A measure has been recorded every 3 minutes for every sensor. The data have been aggregated over a period of 15 minutes and outliers have been removed. Figure 3.4 shows the result of this processing.



(a) Data before processing (b) Data after processing

Figure 3.4: Data processing

3.2.3 Demand Modeling

Among all the sources of error identified in the introduction, this paper focuses on the optimization of the demand-related model to calibrate the simulator.

This choice has been made under the following assumptions:

- The network was supposed to represent the real network in a satisfactory way.
- The simulator was considered as a black box. We thus did not consider to optimize the route choice model algorithm of SUMO.
- All the cars and their drivers shared the same features.

In this study we did not consider to optimize the drivers-related parameters. Some work in that direction were presented by Aji and Taufiq (2012) [52] where they tuned for SUMO and AIMSUM.

The demand modeling has been done using the ODM approach because this method has several advantages:

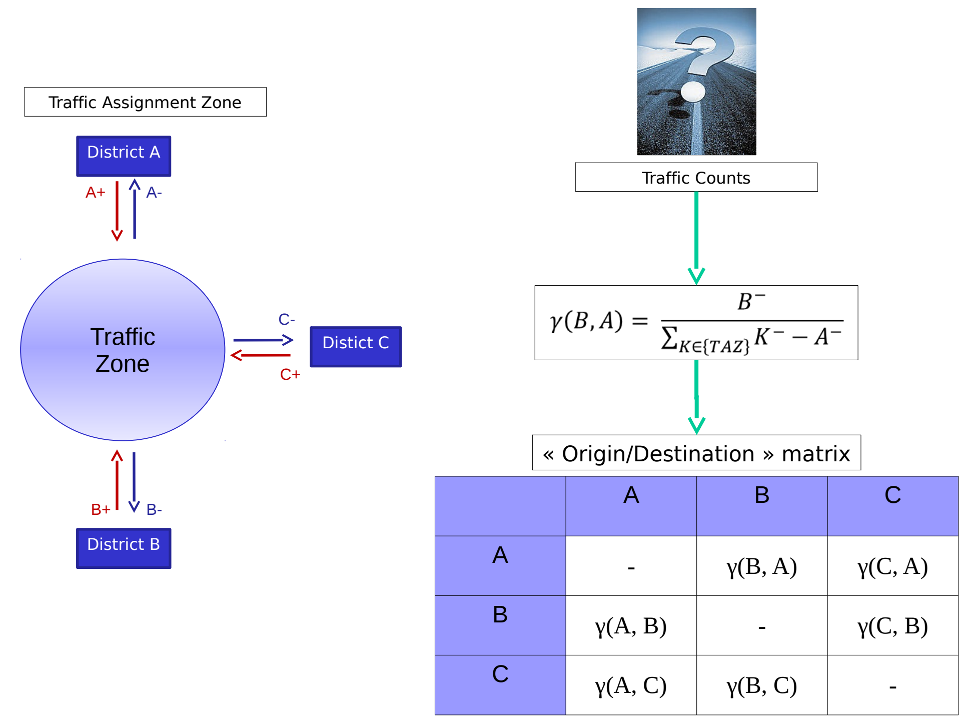
- First, the ODM model is robust to generalization. ODM are more likely to generalize better than other techniques based on-line data processing since they represent the data with two layers of aggregation. Indeed the cars going from A to B can depart at different times and use different routes. The data are thus aggregated over time and space.

- Another advantage of ODM is that they do not require advanced data such as instantaneous speed or car density. From a theoretical point of view, the ODM model only requires traffic counts of every OD pair of the network. For any network, those counts can be easily obtained using a simple camera or police records. Yet, for big networks it is hard to cover every OD pair.

3.2.4 Proposed Heuristic for the initialization.

As stated in the introduction, it is still scarce and expensive to cover every origin-destination pair with sensors. Only a fraction of those pairs are covered. The ODM estimation should thus be based on a sub-set of monitored link.

The proposed heuristic presented in figure 3.5a allowed us to build an initial ODM using only the traffic counts on the inputs of the network.



(a) Heuristic used to create the initial ODM

$$\gamma(B,A) = \frac{B^-}{\sum_{K\in Districts} K^- - A^-}$$

The number of cars going from A to B was estimated by the ratio between the number of cars leaving on B and the total number of cars leaving the network but in A. This initialization strategy theoretically ensured that the level of the demand and the direction of the main flows were correct. This techniques had the advantage of requiring few data. In order to assess the relevance of this technique, this initial point has been compared with:

- A completely random initial point. In thus case two parameters 'min' and 'max' have been used. Those parameter defined reasonable limitations for the maximum/minimum flows. Example: No less than 1 vehicle per minute, no more than 3 vehicles per second.

$$\gamma(B, A) = rand(min, max)$$

- An uniform initial point. In this case, we only used an indicator D representing the total number of cars willing to cross the network in order to ensure the good order of magnitude for the demand.

$$\gamma(B, A) = D/N$$

This thesis investigated the relevance of the heuristic-based initialization technique by comparing it with other initialization techniques. Yet this initialization strategy has originally been proposed by Y. SEMET, research engineer in Thales Research and Technology, who supervised this study.

3.2.5 Algoritms

To address the research question several algorithms have been tested.

- The GSM has been used as suggested by Kolechkina and Toledo (2010) [51].
- A GA (the SSA) has been used to compare the results with the GSM
- An hybrid algorithm called the Memetic Search Algorithm (MSA) has been implemented to take profit of the advantages of the two other techniques.

The algorithms, the results and the conclusions will be presented in the next sections of this chapter.

3.2.6 Success Criteria

In order to address the research question, one must define what a satisfactory behavior for the simulator is. To define this behavior, the notion of intrinsic data variability has been used. The algorithms will be compared using this success criteria.

Data Variability, Seasonality

Urban traffic presents several layers of variability. The traffic increases or decreases depending on the year, the season, the day in the week and more importantly the hour. For example, two peaks of traffic are generally observed in the morning and in the evening.

The weather also plays a role with respect to the demand and the behavior of the drivers. This study did not consider the variability in the demand due to the weather or due to the local, yet frequent, abnormal situations such as accidents, breakdowns or construction work.

All the simulations shown in this section have been conducted over a period of one hour during the peak hour of the morning (congested case) and the demand has been assumed to be constant during this period.

We proposed to measure the intra-day variations with the following indicator: ***The mean over the monitored streets of the standard deviation over the considered periods, of the accumulated counts for one hour***. This definition has been adopted to answer the question: From one day to another, on average, what is the amplitude of the change in the observed flows? Since the worst case scenario for the demand-modeling calibration is the congested network, this indicator has been computed for the morning peak. Besides, weekends have been discarded.

Figure 3.6 shows the evolution of this indicator for the considered case, as a function of the aggregation period. The intra-day variations in the field data has been assumed to be 30 percent. See figure 3.6

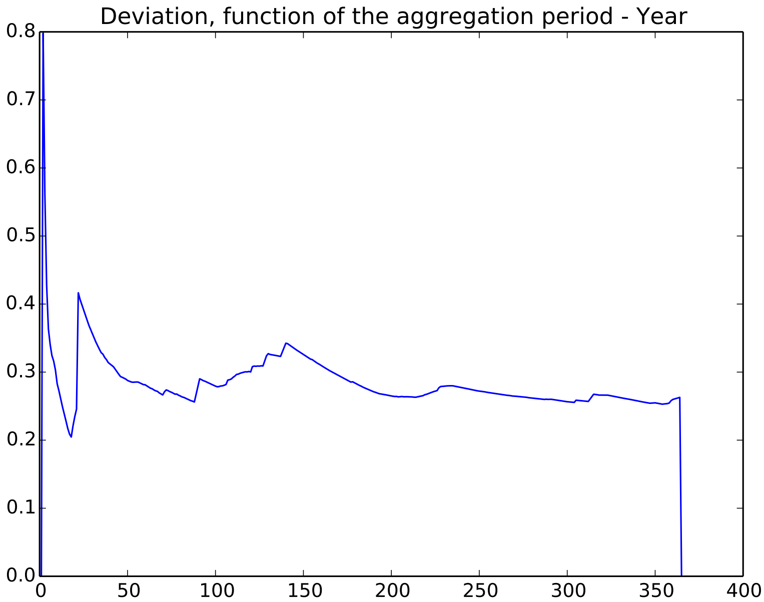


Figure 3.6: Deviation in the field data as a function of the aggregation period. The deviation is computed as the average over the streets of the relative changes over the days.

The result of this analysis might be stated as: During the morning peak from one day to another the observed counts vary, on average, in a range of {+/-30%}

Proposed Success Function

With those notions of intrinsic data variability and modeling errors, the following success function has been used:

$$\text{Success} = \begin{cases} 1 & \text{If } I(X) =< \frac{\sum_{t \in T} |y_{l,t}(X) - \tilde{y}_{l,t}|}{\sum_{t \in T} \tilde{y}_{l,t}} >\leq \text{Inter-day variability} = 0.3 \\ 0 & \text{Else.} \end{cases}$$

Where $\tilde{Y}$ has been computed as the average over several days of the traffic counts in order to avoid over-fitting.

The optimization was considered to be successful if the error was less than the Inter-day variations. That is, the errors are smaller than the variability in the data. For all the reasons stated above this objective is challenging.

Figure 3.7 presents the demand-related model optimization framework used.

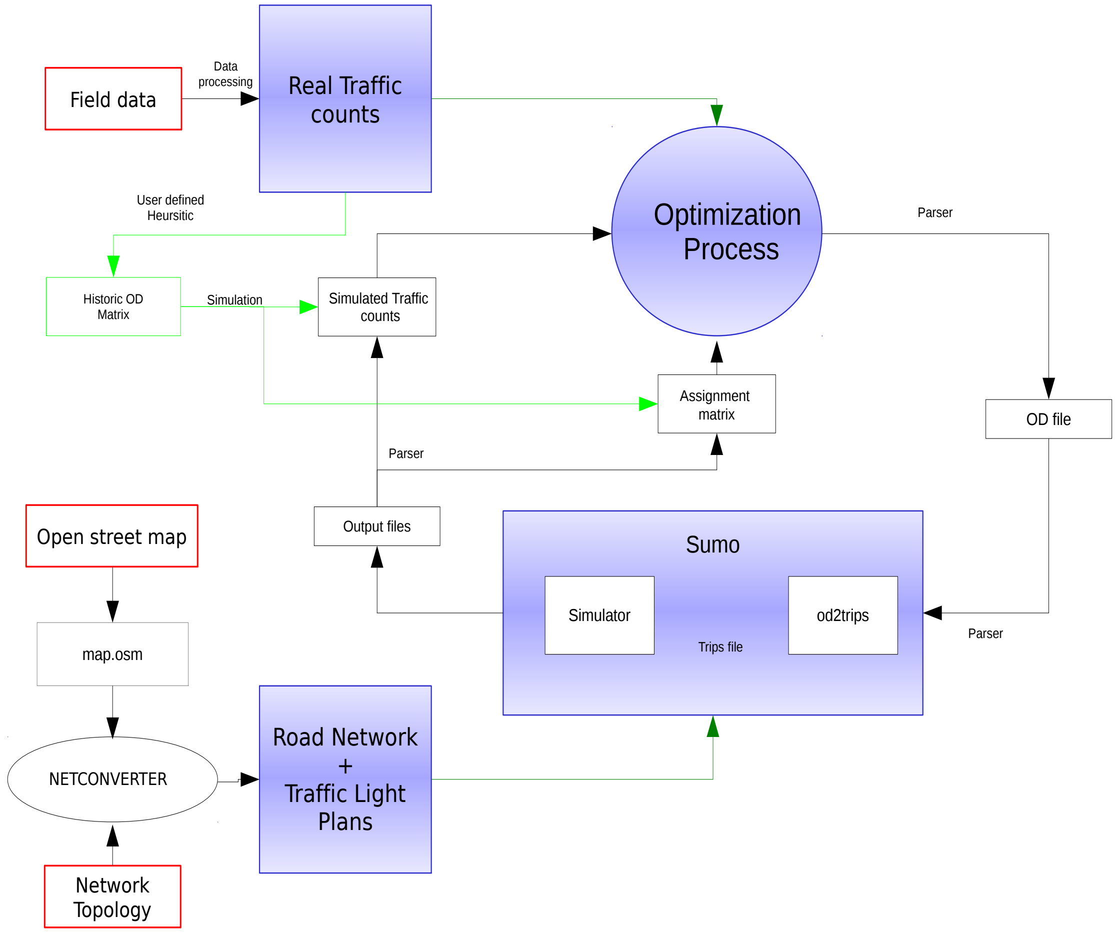


Figure 3.7: Demand-Related Model optimization framework

3.2.7 Main contributions

The main contribution in the ODM estimation problem are:

- Propose a heuristic to create an initial ODM using a subset of traffic counts.
- Implement and extend the work presented by Kolechkina and Toledo (2010) [51] for the GSM using a relative step length.
- Compare the GSM with a GA (the SSA).
- Implement a hybrid algorithm, called the MSA, using both the GSM and the SSA.

3.3 Optimization: Gradient Search Method

3.3.1 Presentation

The method investigated in this section implements and extends the technique proposed by Kolechkina et al (2010) [51]. Yet some important modifications have been done.

The GSM proposed by Kolechkina et al (2010) [51] has the advantage of being a deterministic method which tries to locally improve a given initial point.

Given an ODM, SUMO produces an output which can be turned into simulated flow counts. Yet the input, the initial ODM, and the output, the error observed in the flow counts, lie in different spaces. The main difficulty of the GSM is thus to find in the ODM space the direction corresponding to the steepest descent in the error space. See figure 3.8

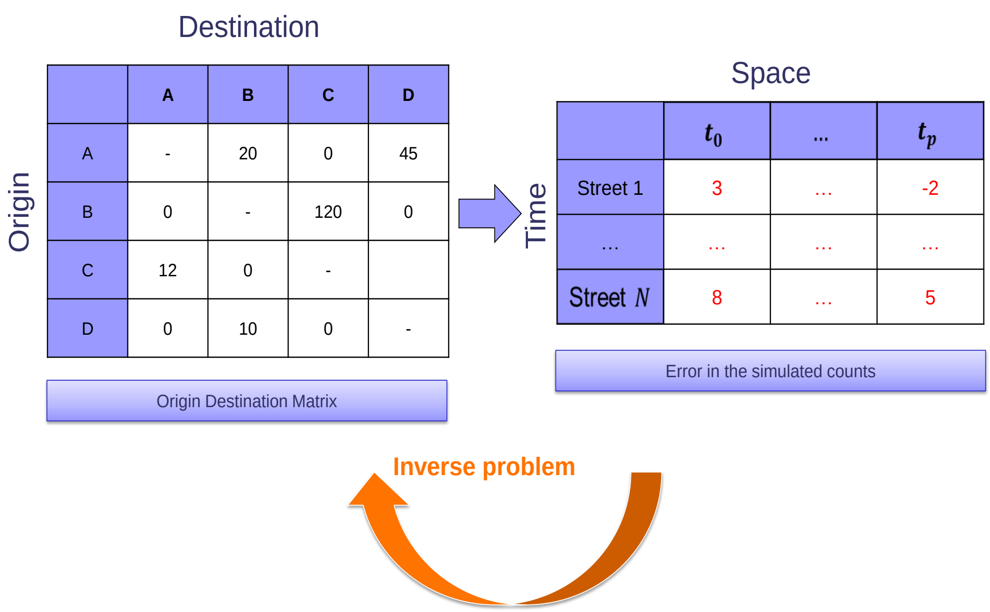


Figure 3.8: Inverse Problem

Once this direction is known, the GSM algorithm will move the current estimate of the ODM in the direction of the gradient.

An important element is the Assignment Proportion Matrix (APM): $A = \{a_n{}^{lt}\}$ where each element of the matrix is defined as the proportion of x_n going through link l at observation time t. This matrix is used to find the correspondence between the two different spaces described above.

$$A = \begin{bmatrix} {a_1}^{1,1} & \cdots & {a_N}^{1,1} \\ \vdots & \ddots & \vdots \\ {a_1}^{L,1} & \cdots & {a_N}^{L,1} \\ \vdots & \ddots & \vdots \\ {a_1}^{1,T} & \cdots & {a_N}^{1,T} \\ \vdots & \ddots & \vdots \\ {a_1}^{L,T} & \cdots & {a_N}^{L,T} \end{bmatrix}$$

(a) Assignment Proportion Matrix

One specificity in the problem investigated here is that SUMO uses its own traffic assignment model based on the Dynamic User Equilibrium. Thus we were not interested in computing this APM but only in retrieving it.

Using this APM, the relationship between the flow counts and the ODM is:

$$y_{l,t} = \sum_n {a_n}^{lt} x_n \qquad \forall (l,t) \in [L' \times T]$$

In matrix form:

$$Y(X) = AX$$

Yet, this APM has been built 'given an ODM' and thus depends on this ODM. Therefore, the APM is defined as a function of the ODM

$$A = A(X)$$

hence, the optimization problem can be restated as:

$$\min_X F(A(X)X - \tilde{Y})$$

The flowchart of the GSM is presented below:

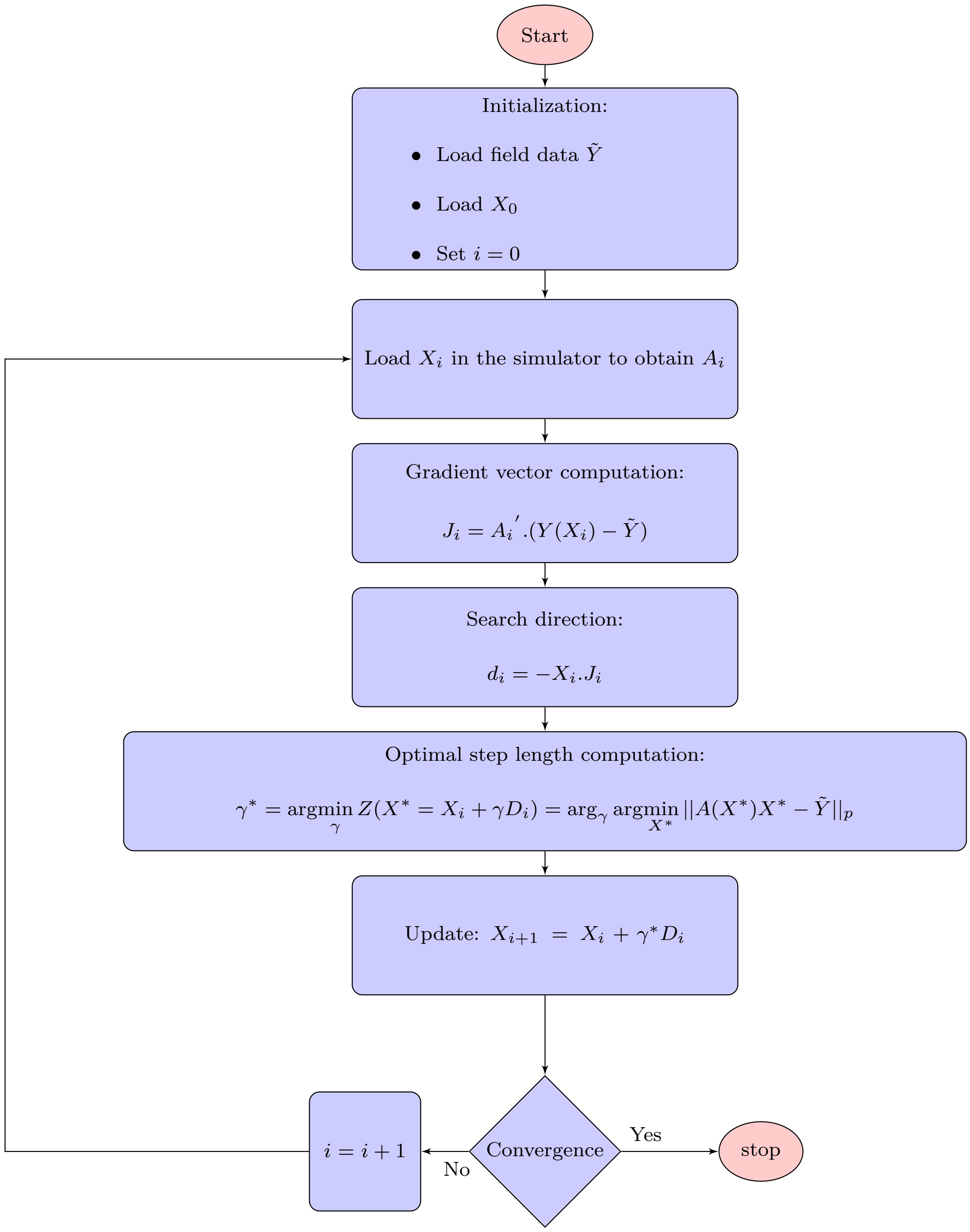


Figure 3.10: Flowchart: Gradient Search

Assignment Proportion Matrix Computation.

To retrieve the APM that has been used during the simulation, the output file, describing the trajectory of every vehicle through the network, has been parsed. See figure 3.7.

Conversely to what has been done by Kolechkina et al (2010) [51] this study did not use a linear interpolation of the APM. The dimensionality of the problem made this technique inefficient. We chose instead to retrieve the APM from a given ODM for every desired point. This techniques resulted in a higher computational demand but provided us with an improved accuracy.

Gradient Vector Computation.

Inspired by Spiess et al (1990) [53], the current ODM is changed in the direction corresponding to the steepest descent. Given a current ODM estimate X_i and the corresponding APM $A(X_i)$, the gradient J_i is computed using the following formula:

$$J_i = \bigtriangledown_{Z(X_i)} = A_i^{'}.(A_i X_i - \tilde{Y})$$

For robustness, the following formula has been used instead:

$$J_i = \bigtriangledown_{Z(X_i)} = A_i^{'}.(Y(X_i) - \tilde{Y})$$

Where $Y(X_i)$ has been computed directly with the output of the simulator. This modification makes the algorithm slower since the simulated traffic counts have to be computed but the estimation is more accurate.

Search Direction.

In the traditional GSM, the descent direction is obtained as the negative of the gradient. Spiess (1990) [53] proposed to use the relative change of the demand. This method prevents drastic change in the ODM values and ensure that a zero demand remains unchanged. The search direction is thus computed as:

$$D_{n,i} = -x_{n,i} J_{n,i}, \qquad \forall (n,i) \in [N \times I]$$

Optimal Step Length Computation.

Given a search direction, the step length still have to be determined. This step length is computed by minimization of the objective function.

$$\gamma^* = \underset{\gamma}{\operatorname{argmin}}\, Z(X^* = X_i + \gamma D_i) = \arg_\gamma \underset{X^*}{\operatorname{argmin}}\, ||A(X^*)X^* - \tilde{Y}||_p$$

3.3.2 Results

The results of the GSM is strongly correlated to the initial point. Moreover, the GSM is deterministic, the search will progress until no further improvements are found. Experimentally we observed a strong relationship between the level of the demand and the results of the optimization.

Several experiments have been conducted to study the impact of those parameters.

Figures C.1 and C.2 show the result of the GSM on simple cases for small demand levels and for different initialization strategies. We observed that for small demands the GSM achieved good performances whichever the initial point.

Figures 3.11 presents the result of the GSM on the case study with different initialization strategies. We observed that for GSM failed to reached the goal of 30% of error whichever the initial point. Yet, the initial point with the proposed heuristic achieved better than the random and the uniform initial point.

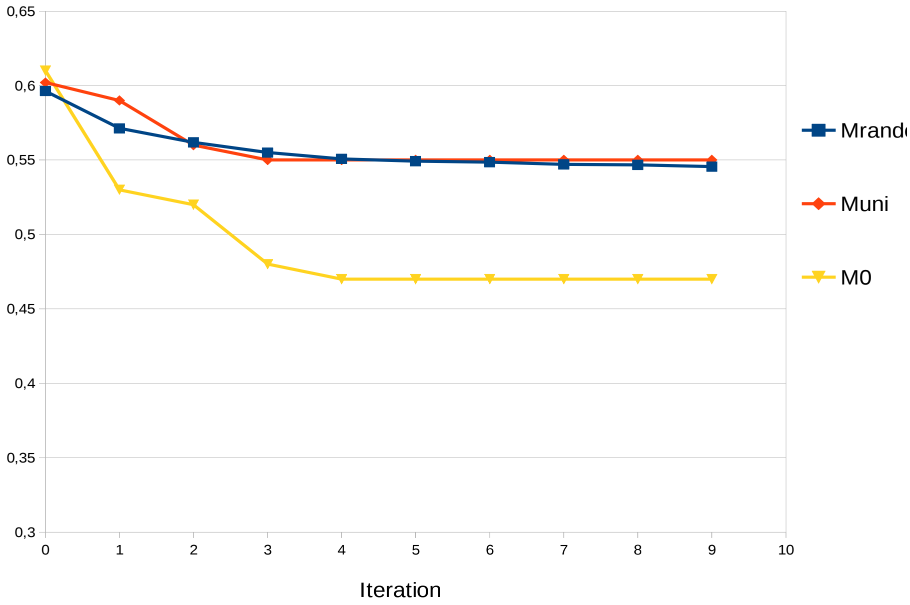
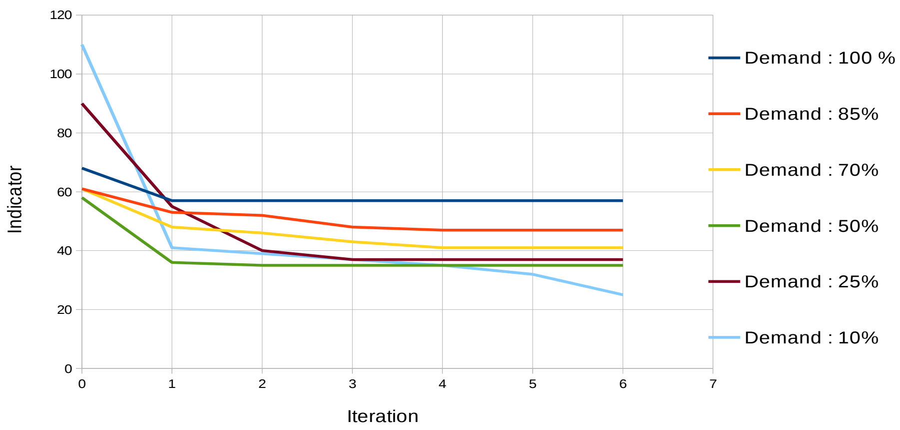


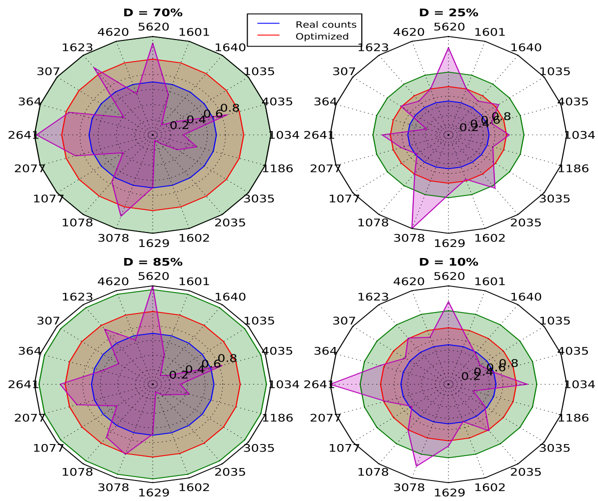
Figure 3.11: GSA: Different initialization strategies

Evolution of the indicator, function of the iteration, for different initialization strategies. The indicator is the mean over the streets of the ***relative*** *error in the traffic counts, function of the iteration, for different initial ODMs.*

In the following experiments, the proposed heuristic has been used to initialize the algorithm. The figure 3.12 presents the results of the GSM on the case study, for different demand levels. We observed that the GSM reached the intra-day spread for small or medium demands only. The best result achieved with this technique with the real demand is shown in figure 3.13



(a) *GSA: Different demand levels*

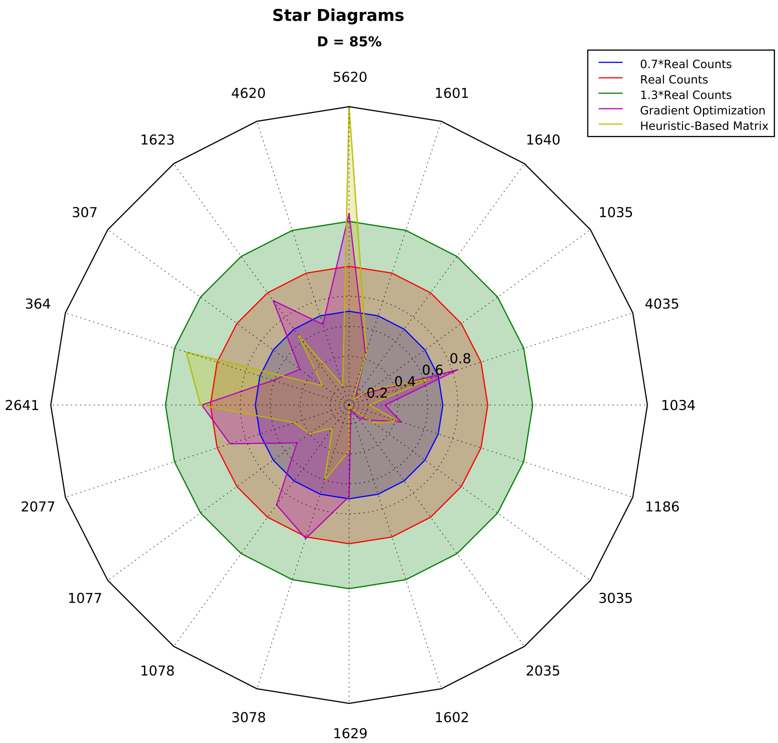


(b) *GSA: Star diagram representation*

Figure 3.12: GSA: Different demand levels.

a) Evolution of the indicator as a function of the iteration for different demand levels. Those demand levels have been obtained by taking a proportion of the actual demand observed for the congested case. Thus, the initial value of the indicator is not relevant. Divided the demand by two might not divide the observed counts by two, especially for the congested case. The indicator is the mean over the streets of the ***relative*** *error in the traffic counts.*

b) Star diagrams representing the ***relative*** *error in the traffic counts, for each monitored street of the network, after the GSA, for different demand levels. The search is inoculated with the heuristic-based ODM.*



(a) Best solution for the Gradient Search Method

Figure 3.13: Star diagram representing the relative error in the traffic counts, for each monitored street of the network, after the GSM optimization inoculated with the proposed ODM. The red curve represents the real traffic counts, the green and blue curves represent the range of values within the intra-day spread and the purple curve represents the simulated traffic counts.

The results for this part might be summarized as:

- The GSM benefits from the inoculation with the proposed heuristic-based initial point.
- The GSM fails to reach the desired performances on the congested network but succeeds for small or medium demands (non-congested case)
- the GSM achieves important improvements within a small number of call to the evaluation function.

3.4 Optimization: Stochastic Search

3.4.1 Presentation

The GSM is a deterministic method which achieved good performances for non-congested networks. One of the drawbacks of this technique is it's likelihood to get stuck in local minimum. Hence, we used a SSA to solve the ODM estimation problem, namely a GA. This approach is a contribution of this thesis.

The SSA proposed in the paper might be seen as a $(1+\mu)$ GA. Given an ODM, small random mutations are applied to obtain an offspring population, this population is evaluated using the traffic simulator and a selection operator is applied to keep the best individual.

The evolutionary algorithm is stochastic, the individuals produced during different runs will be different. To address this variability each 'experiment' discussed below represent several independent 'runs' of the optimization process led with different 'random seeds'.

The flowchart of the SSA is presented below:

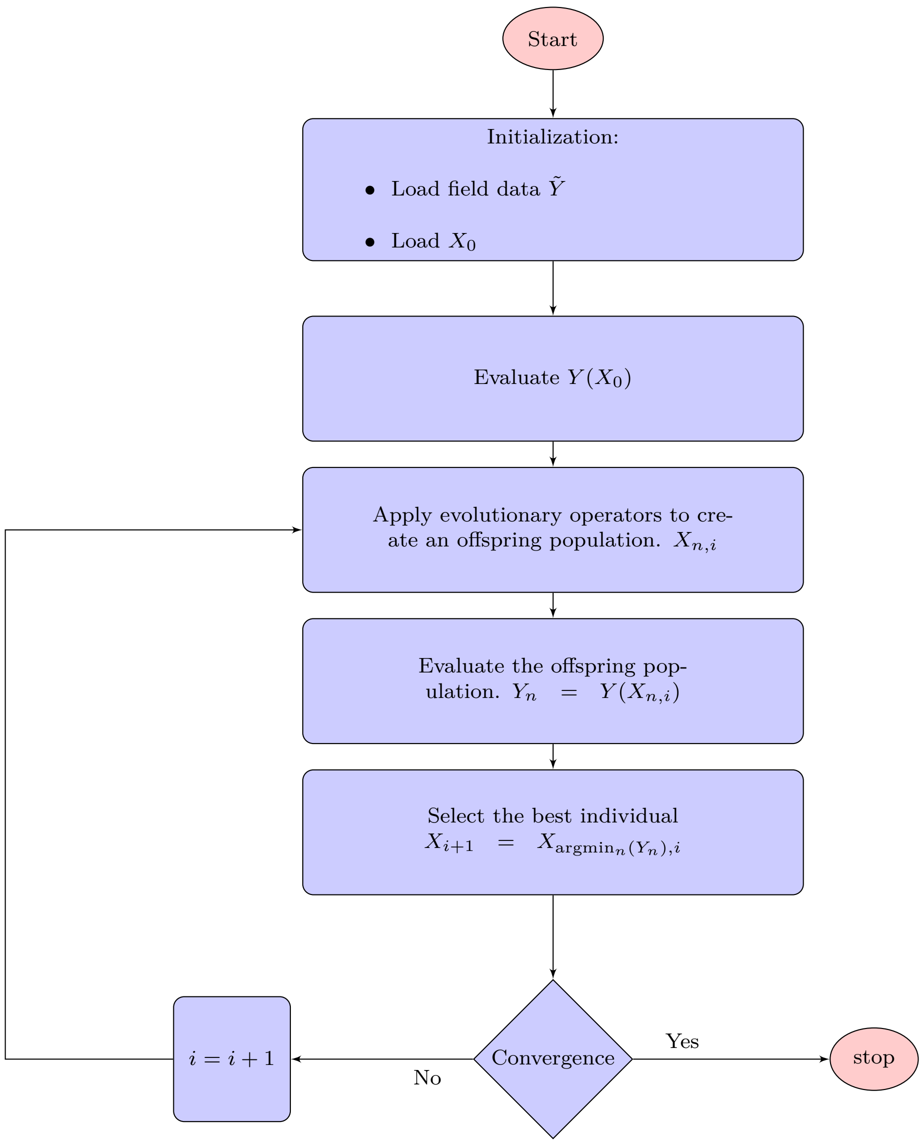


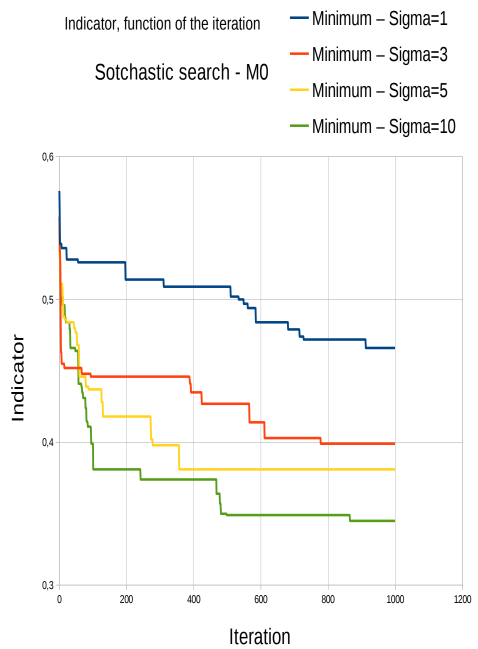
Figure 3.14: Flowchart of the SSA

Comparatively with the GSM, the SSA does not try to explicitly find the terms in the ODM responsible for the error in the traffic counts. There is no direct 'feedback'. the running time might thus be important and the convergence slow. One of the main drawbacks of this technique is the cost of the evaluation function. The action of turning an ODM into traffic counts grows exponentially with the number of vehicle and the simulation time.

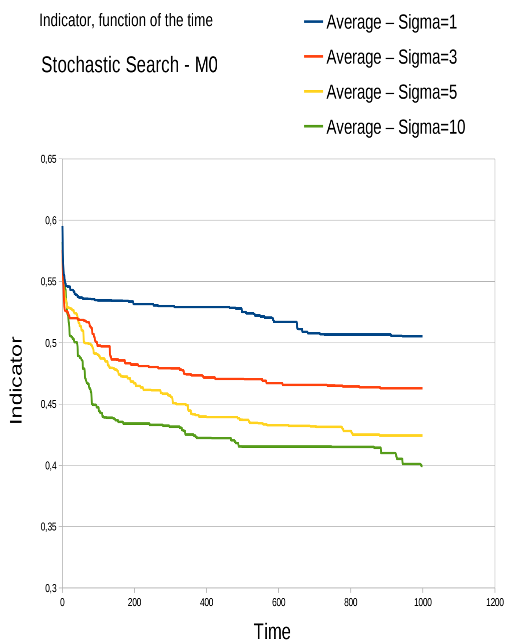
The result of the SSA is highly dependent on both the mutation strategy and the initial point. Several experiments have been conducted to study the impact of those parameters.

3.4.2 Results

The mutation strategy has been tested for the SSA in figure 3.15. We observed that a large mutation parameter achieved better performances. (the amplitude of an atomic mutation should thus be important).



(a) Minimum over the distribution

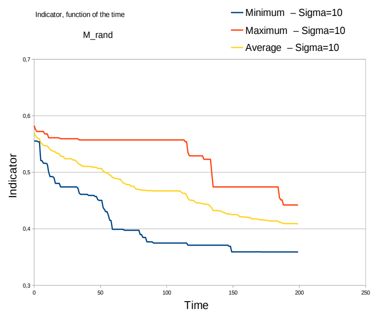


(b) Average over the distribution

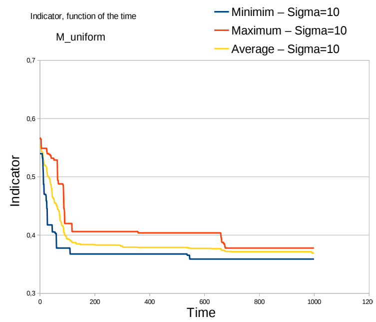
Figure 3.15: SSA: Different mutation strategies

Evolution of the indicator as a function of the time for different mutation strategies for the SSA. The indicator is computed as the mean over the streets of the ***relative*** *error in the traffic counts, function of the iteration. The traffic counts are the observed ones. This experiment is the result of several runs led with different random seed. a) Shows the minimum value observed over the different runs, b) Show the average over the different runs.*

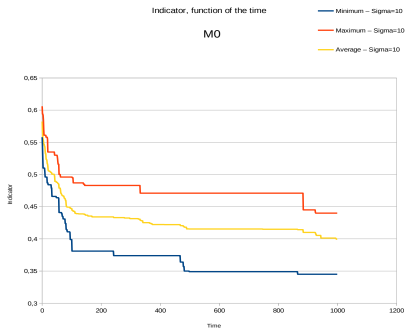
With this setting, an experimental campaign has been led to study the impact of the initialization. Figures in 3.16 present the results of those experiments.



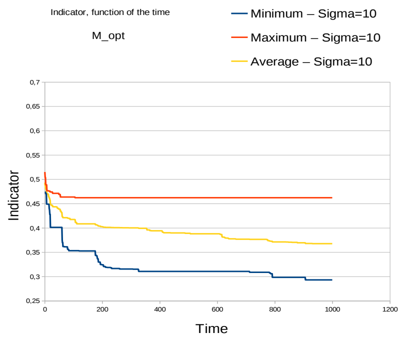
(a) Initial ODM: Random ODM



(b) Initial ODM: Uniform ODM



(c) Initial ODM: Heuristic-based ODM



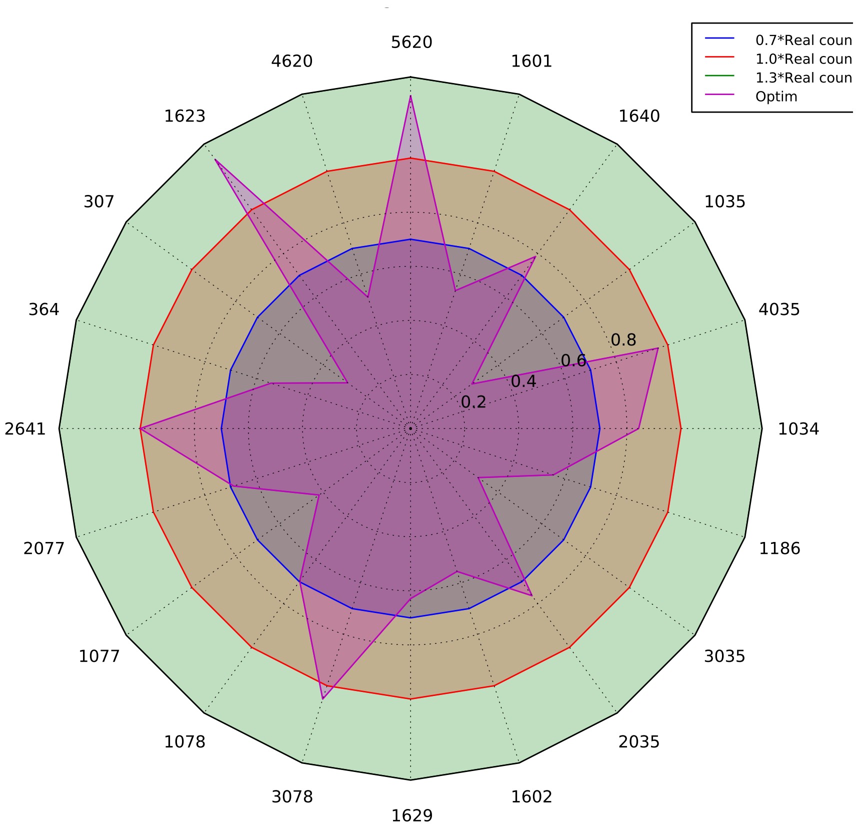
(d) Initial ODM: Optimized ODM

Figure 3.16: SSA: Initialization with different initial points.

Evolution of the indicator as a function of the generation, for different initialization strategies. The indicator is the mean over the streets of the ***relative*** *error in the traffic counts.*

We observed that neither the random, nor the uniform, nor the heuristic-based initial point achieved the desired performance. The results obtained with the SSA were always better than those of the GSM. Yet, the SSA was much slower. One iteration of the GSM and one generation of the SSA require about the same number of evaluation functions.

Figure 3.17 presents the best results obtained with this technique.



(a) Best solution for Stochastic Search Algorithm

Figure 3.17: Star diagram representing the relative error in the traffic counts, for each monitored street of the network, after the SSA. The red curve represents the real traffic counts, the green and blue curves represent the range of values within the intra-day spread and the purple curve represents the simulated traffic counts.

The results for this part are that the SSA has proved to always outperform the GSM in terms of final error in the traffic counts. Yet the time required to reach this result might be prohibitory.

- Large mutation parameters achieved better for the SSA
- The SSA achieved better than the GSM in every cases. Yet it required more time
- Whichever the initial point among 'random ODM','uniform ODM',or 'Heuristic-based ODM', the SSA failed to reach the desired performance

3.5 Optimization: Memetic Search

3.5.1 Presentation

The results collected so far suggest that the GSM improved an initial point in few iterations but failed to reach the desired performances. This might be due to a number of modeling errors but also on the dimensionality of the problem. Also, the SSA improved significantly the initial point but the running time might be considered too important.

Inoculation

In order to take profit of the advantages of both of the algorithms, the SSA has been inoculated using the result of the GSM. The term 'Inoculation' usually refers to the injection of a substance (usually a vaccine) for a patient. In order to keep the genetic analogy, 'Inoculation' will refer to the injection of a carefully-chosen initial point within the initial population of an evolutionary algorithm (Such as the SSA).

Inspired by both Darwinian principles of natural evolution and Dawkins' notion of a meme, the term 'Memetic Algorithm' (MA) was introduced by Moscato (1989) [54] in his technical report. He viewed MA as being close to a form of population-based hybrid GA coupled with an individual learning procedure capable of performing local refinements.

Simple Memetic Search Algorithm

While the GSM performed a fast local search using all the available knowledge on the considered problem, the SSA blindly tries to improve the solution. In the following we consider a MSA that tried to take profit of both techniques without being stuck in poor local minimum within a reasonable time. See figure 3.18. This approach is a contribution of this thesis.

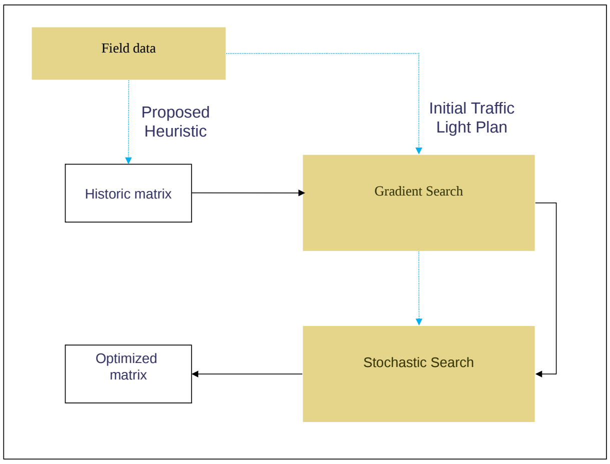
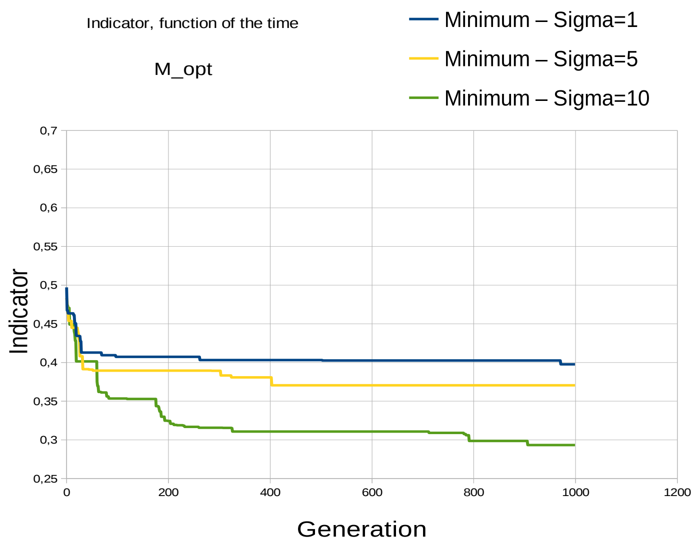


Figure 3.18: Memetic Search Algorithm

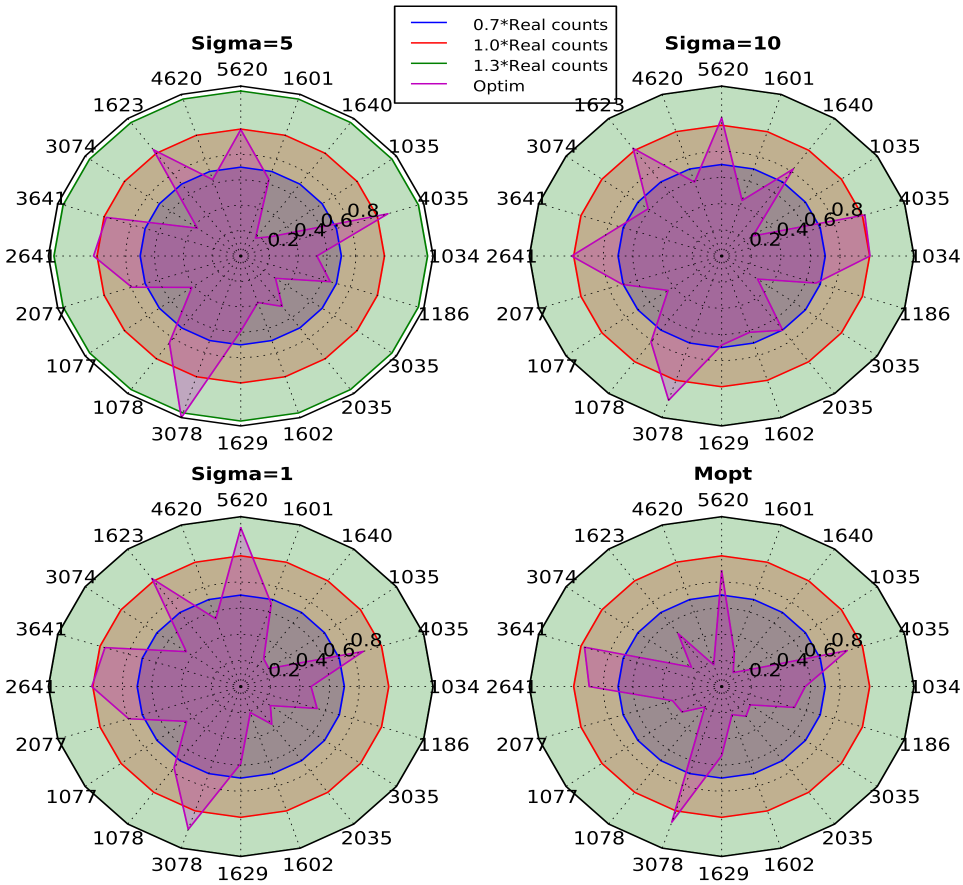
Also, memetic algorithm are usually designed so that the local refinements come after the SSA. This paper proceeded in a reverse fashion. See figure 3.18. The local refinement step (performed with the GSM) took place before the SSA. This strategy was adopted to position the initial solution of the SSA in a relevant area of the search space in few iterations before spending an important time to optimize it.

3.5.2 Results

The SSA process inoculated with the result of the GSM achieved the desired performances. Figures 3.19 and 3.20 show the results obtained with this technique.



(a) Different inoculation strategies

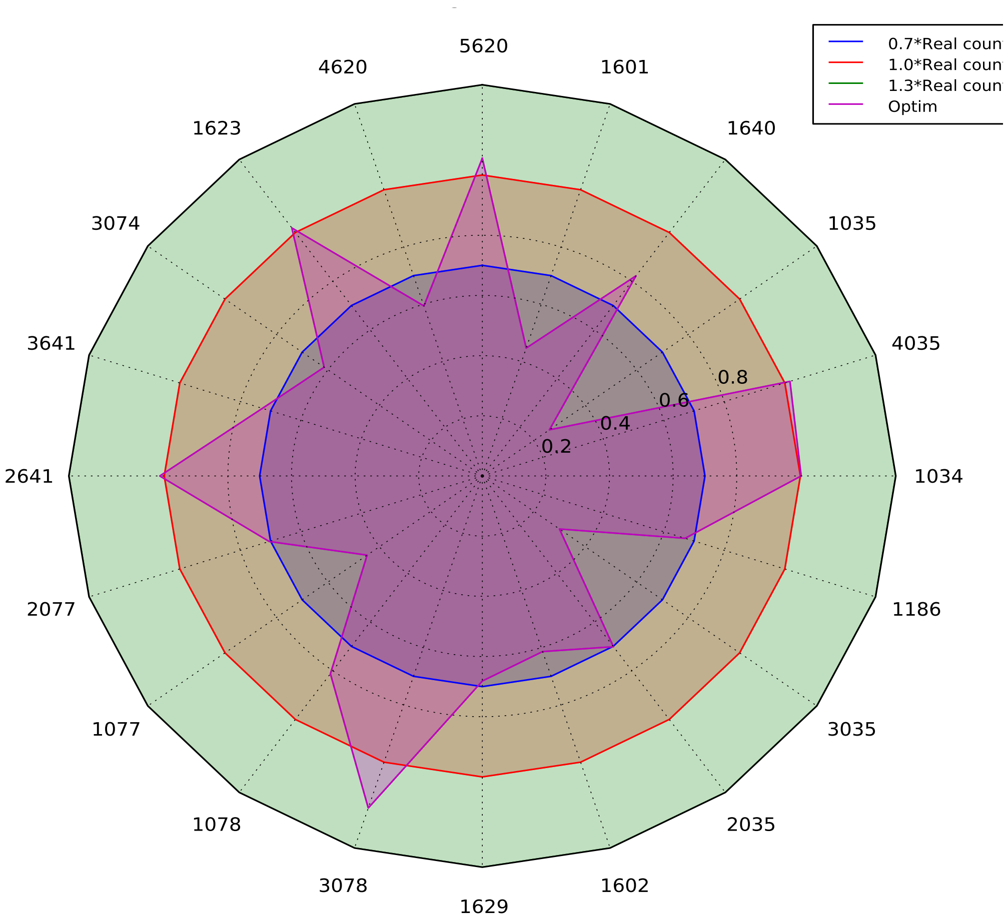


(b) Star diagrams

Figure 3.19: Memetic Search

a) Evolution of the values of the indicator(avg) during the optimization for different mutation strategies.

b) Star diagrams representing the relative error in the traffic counts, for each monitored street of the network, after the SSA, for different mutation strategies. The search was initialized with the results of the GSM optimization.



(a) Best solution for the Memetic Search Algorithm

Figure 3.20: Star diagram representing the relative error in the traffic counts, for each monitored street of the network, after the MSA. This diagram represents the counts obtained with the best ODM obtained with the MSA. The red curve represents the real traffic counts, the green and blue curves represent the range of values within the intra-day spread and the purple curve represents the simulated traffic counts.

The result for the MSA might be stated a follow:

- The MSA has been successfully applied for the considered network, during the peak-hour of the morning.
- The MSA benefits from the heuristic-based ODM initialization.

3.6 Conclusion

As a conclusion, both the GSM and the SSA failed to reach a satisfactory behavior for the simulator within a reasonable time for a medium sized network with the considered success criteria. Yet, a hybrid algorithm (the MSA) succeeded in reaching this behavior.

This result has yet to be challenged for other benchmarks. Indeed, this conclusion has been based on the study of a single benchmark and additional studies should be conducted.

In addition, the heuristic-based initialization techniques has proved to be useful for the GSM and thus also for the MSA.

3.7 Discussion

In order to further increase the reliability of the simulator, several directions might be considered.

Modeling errors have been observed in the network especially at the entrance of a complex roundabout. Indeed, most of the streets with poor performances regarding the reliability indicator were located around this roundabout. Even though this roundabout might considered absurdly complex, Figure 3.21 a deeper look in the simulator behavior might be required to address this question.

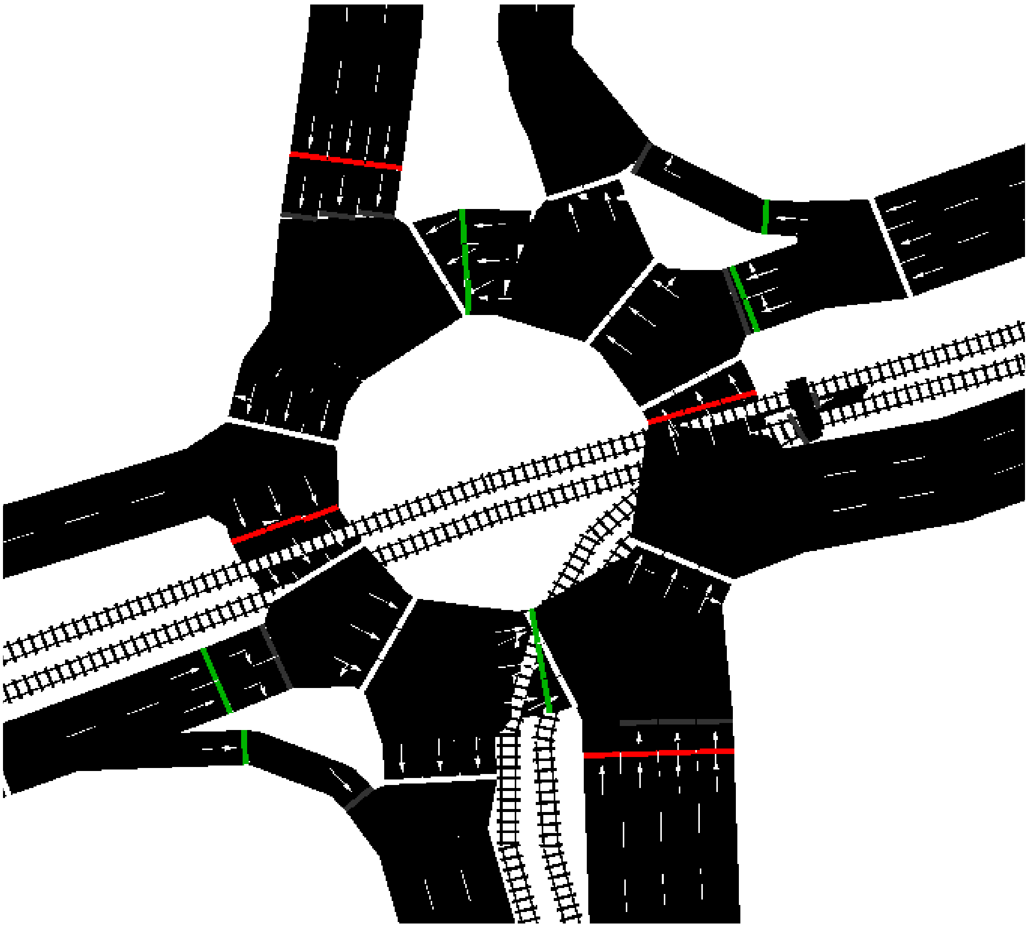


Figure 3.21: View of one of the roundabout in the considered network

The drivers-related parameters might be tuned to decrease the modeling errors such as by Ronaldo (2012) [52] where the reaction time and the drivers imperfection have been studied.

Other simulator-related topic might be considered:

- The route choice model.
- The distribution law used by the simulator to create the demand from the ODM.
- The description of the vehicles might be improved to meet a realistic situation. (10% of buses, 5% of trucks etc..)

Work can also be done to reduce the modeling errors induced by the Search Algorithm itself. For instance the GSM implicitly makes some assumptions regarding the cars injection. e.g: The GSM assumes that the simulator actually worked with the desired ODM whereas the simulator tries to best fit this ODM.

Other Search Algorithm might also be considered. For instance a full MSA might be considered. See figure 3.22

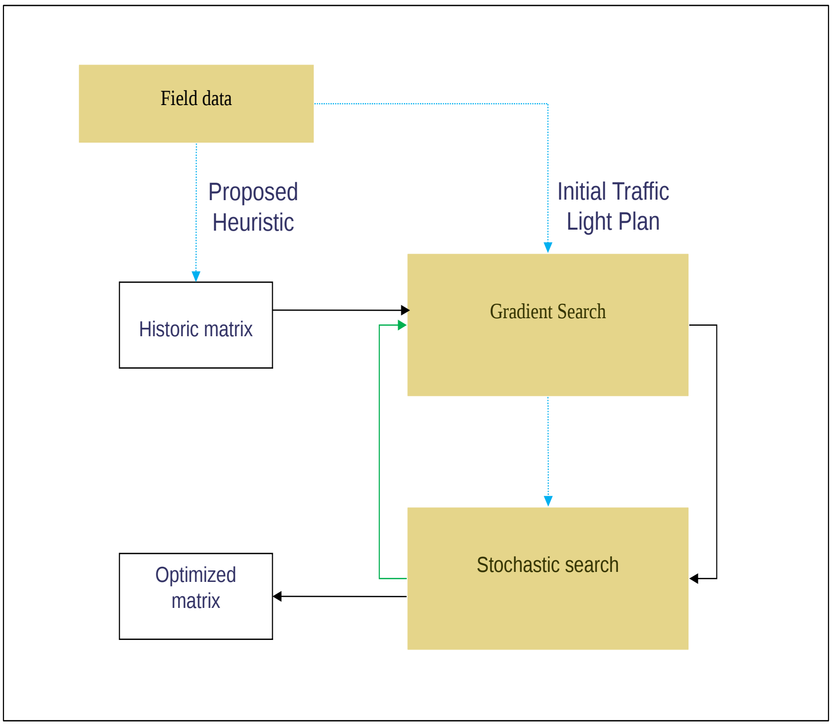


Figure 3.22: Full Memetic Search Algorithm

Chapter 4

Conclusions

This study developed a framework to optimize the duration of the green light phases for a medium sized network. This framework integrates the simulator SUMO and a NSGA-II based algorithm. The aim was to minimize several objectives such as the total waiting time and the total pollutant emissions for several pollutants such as: CO, CO_2, HC, NOx, PMx, Fuel in a reasonable amount of time. Moreover it has been demonstrated on a benchmark of 11 intersections with 168 traffic lights and 40 possible turning movements.

The framework has been shown to successfully find several tradeoffs that outperforms the initial traffic light plan used on the real network for the different objectives within a reasonable amount of time.

This has been achieved by adding several features to the algorithm. Those features are:

- Choosing the objectives that are to be optimized.
- Finding the optimal set of parameters for the GA.
- Dynamically adapt the parameters during the optimization process
- Seeding the algorithm with previously computed solutions.

The second part of this study successfully calibrated the demand related model of SUMO in order to reach a satisfactory behavior within a reasonable time for a medium-sized network using. The dimensionality of the problem made it too difficult to solve using only the GSM. Thus, a MSA algorithm resulting of the combination of a GSA and a SSA has been implemented and allowed us to reach a satisfactory behavior for the simulator within a reasonable time for a medium sized network. In addition, the heuristic-based initialization techniques has proved to be useful for the GSM and also for the MSA.

Those results have yet to be challenged for other benchmarks. Indeed, those conclusions have been based on the study of a single benchmark and additional studies should be conducted.

Bibliography

[1] C.S-W. The cost of traffic jams. *The Economist*, Nov 2014.

[2] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *Trans. Evol. Comp*, 1(1):67–82, April 1997.

[3] Matti Pursula. Simulation of traffic systems-an overview. *Journal of Geographic Information and Decision Analysis*, 3(1):1–8, 1999.

[4] D.I. Robertson. *TRANSYT: A Traffic Network Study Tool.* RRL report. Road Research Laboratory, 1969.

[5] trafficware. *Synchro studio user guide.* U, U.

[6] Sharon Adams Boxill and Lei Yu. *An Evaluation of Traffic Simulation Models for Supporting ITS.* Houston, 2000.

[7] Alexander Skabardonis, M Cassidy, Adolf D May, and S Cohen. *Application of simulation to evaluate the operation of major freeway weaving sections.* Transportation Research Board, 1989.

[8] M McDonald, M Brackstone, and D Jeffery. Data requirements and sources for the calibration of microscopic motorway simulation models. In *Vehicle Navigation and Information Systems Conference, 1994. Proceedings., 1994*, pages 233–238. IEEE, 1993.

[9] Aleksandar Stevanovic, Jelka Stevanovic, and Cameron Kergaye. Optimizing signal timings to improve safety of signalized arterials. *Third international Conference on Road Safety and Simulation*, 2011.

[10] Darrell Whitley. A genetic algorithm tutorial. *Statistics and computing*, 4(2):65–85, 1994.

[11] Leena Singh, Sudhanshu Tripathi, and Himakshi Arora. Time optimization for traffic signal control using genetic algorithm. *International Journal of Recent Trends in Engineering*, 2(2):4–6, 2009.

[12] Sahar Rahmani, Seyed Mahmoud Mousavi, and MJ Kamali. Modeling of road-traffic noise with the use of genetic algorithm. *Applied Soft Computing*, 11(1):1008–1013, 2011.

[13] Rao Qian, Zhang Lun, Yang Wenchen, and Zhang Meng. A traffic emission-saving signal timing model for urban isolated intersections. *Procedia - Social and Behavioral Sciences*, 96(0):2404 – 2413, 2013.

[14] R.K. Abushehab, B.K. Abdalhaq, and B. Sartawi. Genetic vs. particle swarm optimization techniques for traffic light signals timing. In *Computer Science and Information Technology (CSIT), 2014 6th International Conference on*, pages 27–35, March 2014.

[15] Javier J. Sánchez Medina, Manuel J. Galán, and Enrique Rubio. Applying a traffic lights evolutionary optimization technique to a real case: Las Ramblas area in santa cruz de tenerife. *IEEE Trans. Evolutionary Computation*, 12(1):25–40, 2008.

[16] G Iscaro and G Nakamiti. A supervisor agent for urban traffic monitoring. *Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2013 IEEE International Multi-Disciplinary Conference*, 2013.

[17] Carlos M. Fonseca and Peter J. Fleming. Genetic algorithms for multiobjective optimization: Formulationdiscussion and generalization. In *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 416–423, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

[18] Eckart Zitzler. *Evolutionary algorithms for multiobjective optimization: Methods and applications*, volume 63. Citeseer, 1999.

[19] Eckart Zitzler, Marco Laumanns, Lothar Thiele, Eckart Zitzler, Eckart Zitzler, Lothar Thiele, and Lothar Thiele. Spea2: Improving the strength pareto evolutionary algorithm, 2001.

[20] N. Srinivas and Kalyanmoy Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2:221–248, 1994.

[21] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and Tanaka Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. *Lecture notes in computer science*, 1917:849–858, 2000.

[22] Jürgen Branke, Peter Goldate, and Holger Prothmann. Actuated traffic signal optimization using evolutionary algorithms, 2007.

[23] Danny Robles. Optimal signal control with multiple objectives in traffic mobility and environmental impacts. Master's thesis, KTH, Traffic and Logistics, 2012.

[24] Yan Li, Lijie Yu, Siran Tao, and Kuanmin Chen. Multi-objective optimization of traffic signal timing for oversaturated intersection. *Mathematical Problems in Engineering*, 2013, 2013.

[25] MSc. A.E. Ohazulike and ir. T. Brands. Multi-objective optimization of traffic externalities using tolls: A comparison of genetic algorithm with game theoretical approach. In IEEE, editor, *Proceedings of IEEE Conference on Evolutionary Computation, Cancún, Mexico, June 20-23 2013.*, pages 2465 – 2472. IEEE, 2013.

[26] Frank DiCesare, PaulT. Kulp, Michael Gile, and George List. The application of petri nets to the modeling, analysis and control of intelligent urban traffic networks. In Robert Valette, editor, *Application and Theory of Petri Nets 1994*, volume 815 of *Lecture Notes in Computer Science*, pages 2–15. Springer Berlin Heidelberg, 1994.

[27] Michel dos Santos Soares and Jos Vrancken. A modular petri net to modeling and scenario analysis of a network of road traffic signals. *Control Engineering Practice*, 20(11):1183 – 1194, 2012. Special Section: Wiener-Hammerstein System Identification Benchmark.

[28] Thomas L. Thorpe and Charles W. Anderson. Traffic light control using sarsa with three state representations. Technical report, IBM Corporation, 1996.

[29] Tim Brys, Tong T. Pham, and Matthew E. Taylor. Distributed learning and multi-objectivity in traffic light control. *Connection Science*, 26(1):65–83, 2014.

[30] Ana L.C. Bazzan, Denise de Oliveira, and Bruno C. da Silva. Learning in groups of traffic signals. *Engineering Applications of Artificial Intelligence*, 23(4):560 – 568, 2010.

[31] Mohamed A. Khamis and Walid Gomaa. Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework. *Eng. Appl. Artif. Intell.*, 29:134–151, March 2014.

[32] Ivana Dusparic and Vinny Cahill. Research issues in multiple policy optimization using collaborative reinforcement learning. In *Unknown*, Minneapolis, USA., May 2007.

[33] Ella Bingham. Reinforcement learning in neurofuzzy traffic signal control. *European Journal of Operational Research*, 131(2):232–241, 2001.

[34] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. Recent development and applications of SUMO - Simulation of Urban Mobility. *International Journal On Advances in Systems and Measurements*, 5(3-4):128–138, December 2012.

[35] S.A. Boxill and L. Yu. *A evaluation of traffic simulation models for supporting ITS development*. Center for Transportation Training and Researcj, Texas Southern University, 2000.

[36] E.J. Hughes. Evolutionary many-objective optimisation: many once or one many? In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 1, pages 222–227 Vol.1, Sept 2005.

[37] Robin C Purshouse and Peter J Fleming. Evolutionary many-objective optimisation: An exploratory analysis. In *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, volume 3, pages 2066–2073. IEEE, 2003.

[38] Vineet Khare, Xin Yao, and Kalyanmoy Deb. Performance scaling of multi-objective evolutionary algorithms. In *Evolutionary Multi-Criterion Optimization*, pages 376–390. Springer, 2003.

[39] Thomas Bäck and Martin Schütz. Intelligent mutation rate control in canonical genetic algorithms. In Zbigniew W. Ras and Maciej Michalewicz, editors, *ISMIS*, volume 1079 of *Lecture Notes in Computer Science*, pages 158–167. Springer, 1996.

[40] Schoenauer. M Semet. Y. On the benefits of inoculation, an example in train scheduling. *Genetic and Evolutionary Computation Conference*, 2006.

[41] Thomas Back. Self-adaptation in genetic algorithms. In *Proceedings of the First European Conference on Artificial Life*, pages 263–271. The MIT Press, Cambridge, MA, 1992.

[42] A. D May. *Traffic flow fundamentals*. Prentice Hall, 1990.

[43] Ennio Cascetta and Sang Nguyen. A unified framework for estimating or updating origin/destination matrices from traffic counts. *Transportation Research Part B: Methodological*, 22(6):437–455, 1988.

[44] Hai Yang, Tsuna Sasaki, Yasunori Iida, and Yasuo Asakura. Estimation of origin-destination matrices from link traffic counts on congested networks. *Transportation Research Part B: Methodological*, 26(6):417–434, 1992.

[45] Ennio Cascetta, Domenico Inaudi, and Gerald Marquis. Dynamic estimators of origin-destination matrices using traffic counts. *Transportation science*, 27(4):363–373, 1993.

[46] Ennio Cascetta and Maria Nadia Postorino. Fixed point approaches to the estimation of o/d matrices using traffic counts on congested networks. *Transportation Science*, 35(2):134–147, 2001.

[47] Christian Gawron. An iterative algorithm to determine the dynamic user equilibrium in a traffic simulation model. *International Journal of Modern Physics C*, 9(03):393–407, 1998.

[48] Michael Behrisch, Daniel Krajzewicz, and Yun-Pang Wang. Comparing performance and quality of traffic assignment techniques for microscopic road traffic simulations. In *DTA2008 International Symposium on Dynamic Traffic Assignment*, 2008.

[49] C.S. Fisk. On combining maximum entropy trip matrix estimation with user optimal assignment. *Transportation Research Part B: Methodological*, 22(1):69 – 73, 1988.

[50] Hai Yang, Tsuna Sasaki, Yasunori Iida, and Yasuo Asakura. Estimation of origin-destination matrices from link traffic counts on congested networks. *Transportation Research Part B: Methodological*, 26(6):417 – 434, 1992.

[51] T. Kolechkina and T. Toledo. *Off-line Estimation of Dynamic Origin-destination Trip Matrices*. Technion-I.I.T., Transportation Research Institute, 2010.

[52] Aji Ronaldo and M taufiq Ismail. Comparison of the two micro-simulation software aimsun & sumo for highway traffic modelling. Master's thesis, Linköping UniversityLinköping University, Communications and Transport Systems, The Institute of Technology, 2012.

[53] H Spiess. A descent based approach for the od matrix adjustment problem. *Special Publication*, 693, 1990.

[54] Pablo Moscato et al. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech concurrent computation program, C3P Report*, 826:1989, 1989.

[55] Sharminda Bera and K. V. Krishna Rao. Estimation of origin-destination matrix from traffic counts: the state of the art. *European Transport Trasporti Europei*, pages 2–23, 2011.

[56] Ennio Cascetta. Estimation of trip matrices from traffic counts and survey data: A generalized least squares estimator. *Transportation Research Part B: Methodological*, 18(4–5):289 – 299, 1984.

[57] Carlos M. Fonseca and Peter J. Fleming. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3:1–16, 1995.

[58] Rao Qian, Zhang Lun, Yang Wenchen, and Zhang Meng. A traffic emission-saving signal timing model for urban isolated intersections. *Procedia - Social and Behavioral Sciences*, 96(0):2404 – 2413, 2013.

[59] Mariano Gallo and Luca D'Acierno. Comparing algorithms for solving the local optimisation of the signal settings (loss) problem under different supply and demand configurations. *Procedia - Social and Behavioral Sciences*, 87(0):147 – 162, 2013. SIDT Scientific Seminar 2012.

[60] Htin Lin, Khin Muyar Aye, Hla Myo Tun, Theingi, and Zaw Min Naing. Design and construction of intelligent traffic light control system using fuzzy logic. *AIP Conference Proceedings*, 1052(1):237–239, 2008.

[61] Yu-Chiun Chiou and Yen-Fei Huang. Genetic fuzzy logic traffic signal control with cell transmission modeling. *Journal of the Chinese Institute of Engineers*, 37(4):446–460, 2014.

[62] Giovanni Felici, Giovanni Rinaldi, Antonio Sforza, and Klaus Truemper. A logic programming based approach for on-line traffic control. *Transportation Research Part C: Emerging Technologies*, 14(3):175 – 189, 2006.

[63] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto evolutionary algorithm. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.

[64] Daniel Krajzewicz, Elmar Brockfeld, and Jürgen Mikat. Simulation of modern traffic lights control systems using the open source traffic simulation sumo. *Proceedings of the 3rd Industrial Simulation Conference 2005*, June 2009.

[65] Eusebio Angulo, Francisco P. Romero, Ricardo García, Jesús Serrano-Guerrero, and José A. Olivas. An adaptive approach to enhanced traffic signal optimization by using soft-computing techniques. *Expert Systems with Applications*, 38(3):2235 – 2247, 2011.

[66] Mifa Kim, Tomoyuki Hiroyasu, Mitsunori Miki, and Shinya Watanabe. Spea2+: Improving the performance of the strength pareto evolutionary algorithm 2. In *Parallel Problem Solving from Nature - PPSN VIII*, volume 3242 of *Lecture Notes in Computer Science*, pages 742–751. Springer Berlin Heidelberg, 2004.

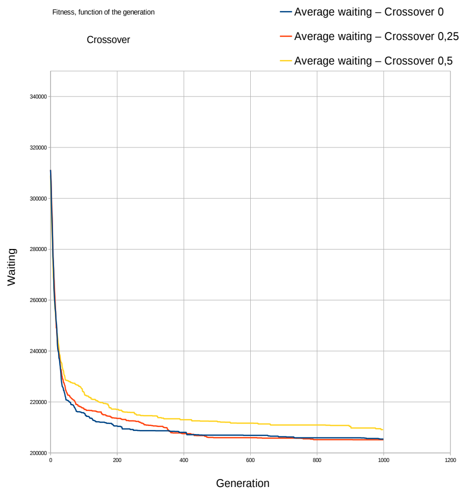
[67] Kesur and Khewal Bhupendra. Improving precision and reducing runtime of microscopic traffic simulators through stratified sampling. *Journal of Engineering*, 2013.

[68] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau, and Christian Gagné. DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13:2171–2175, jul 2012.

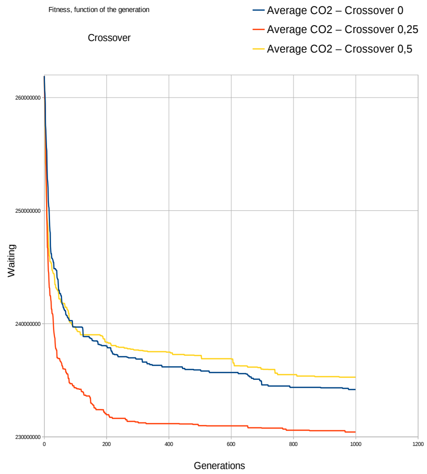
[69] Kaveh Amouzgar. Multi-objective optimization using genetic algorithms. Master's thesis, Jönköping University, School of Engineering, 2012.

Appendix A

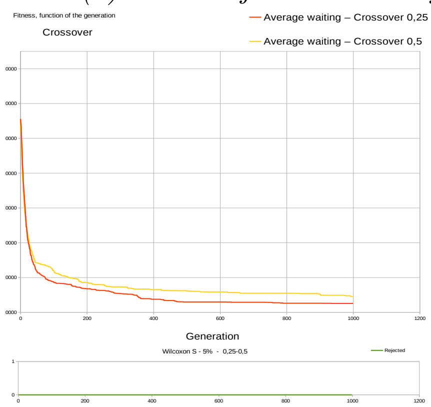
Traffic Optimization - Parameter tuning



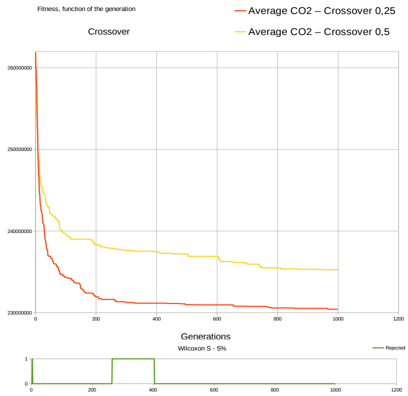
(a) Mono-objective: Waiting



(b) Mono-objective: CO2



(c) Mono-objective: Waiting

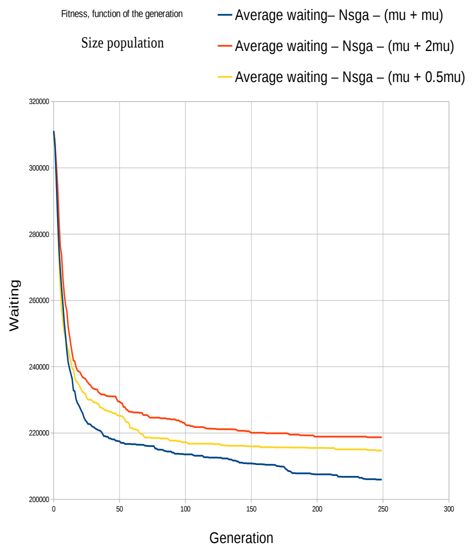


(d) Mono-objective: CO_2

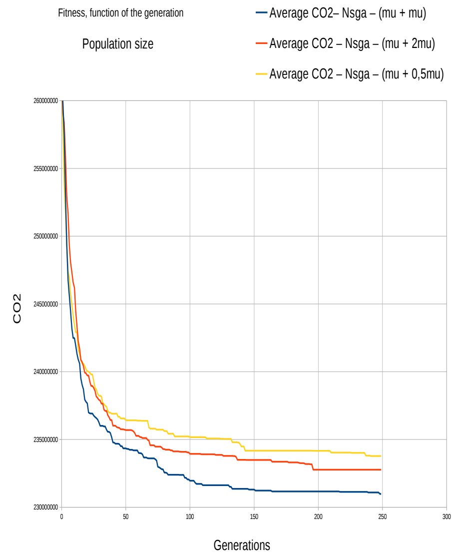
Figure A.1: Mono-objective: Different crossover strategies.

Comparison of different mutation strategies for the mono-objective evolutionary process for two objectives (separately).

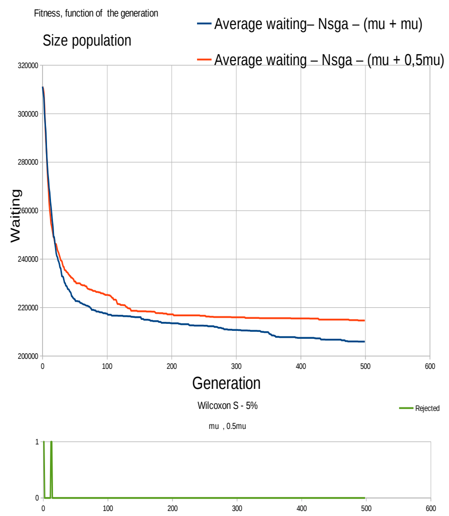
The first graph represent the evolution of the indicator as a function of the generation. The second graph represents a statistical test on the different runs for each experiment. A value of 1 indicates a statistically significant difference between the two distributions. The test used in this part is the test of Wilcoxon S at 5



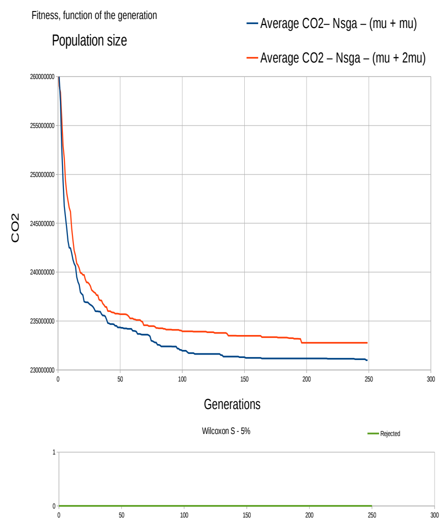
(a) Mono-objective: Waiting



(b) Mono-objective: CO_2



(c) Comparison: Waiting

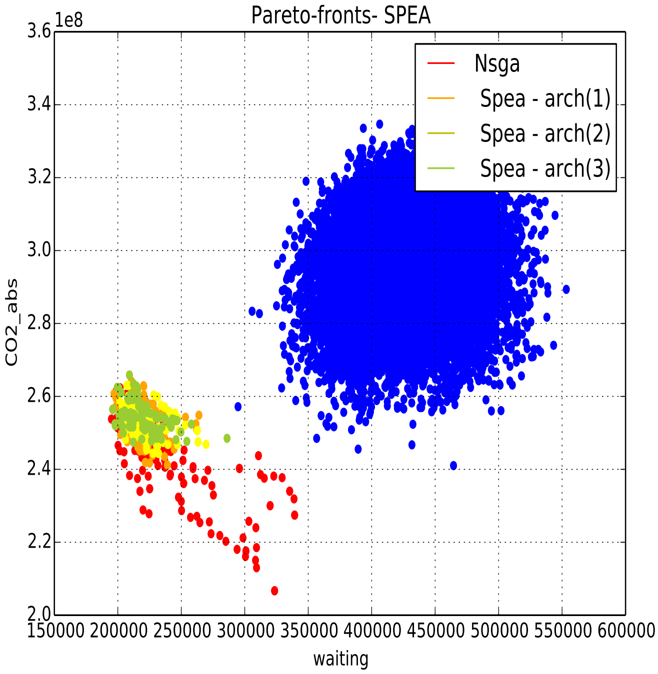


(d) Comparison: CO_2

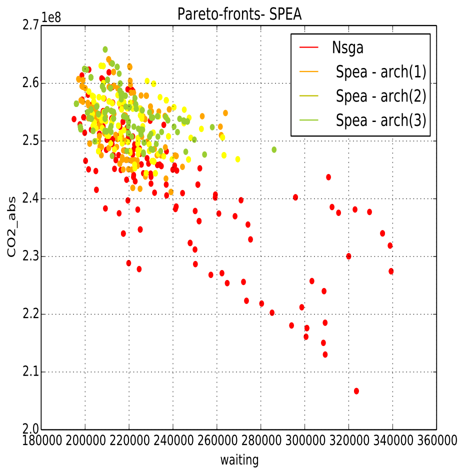
Figure A.2: Mono-objective: Population size parameter.

Comparison of different population sizes for the mono-objective evolutionary process for two objectives (separatly).

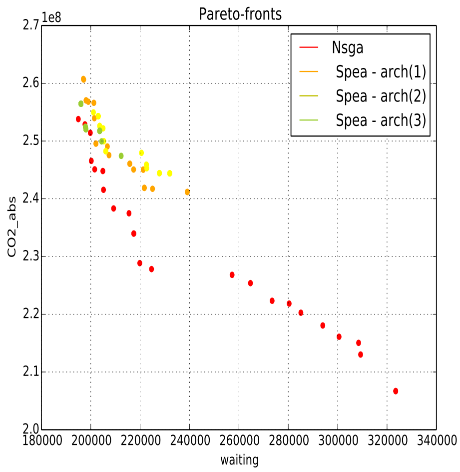
The first graph represent the evolution of the indicator as a function of the generation. The second graph represents a statistical test on the different runs for each experiment. A value of 1 indicates a statistically significant difference between the two distributions. The test used in this part is the test of Wilcoxon S at 5%.



(a) Multi-Objective optimization: Random-clouds and Pareto-fronts



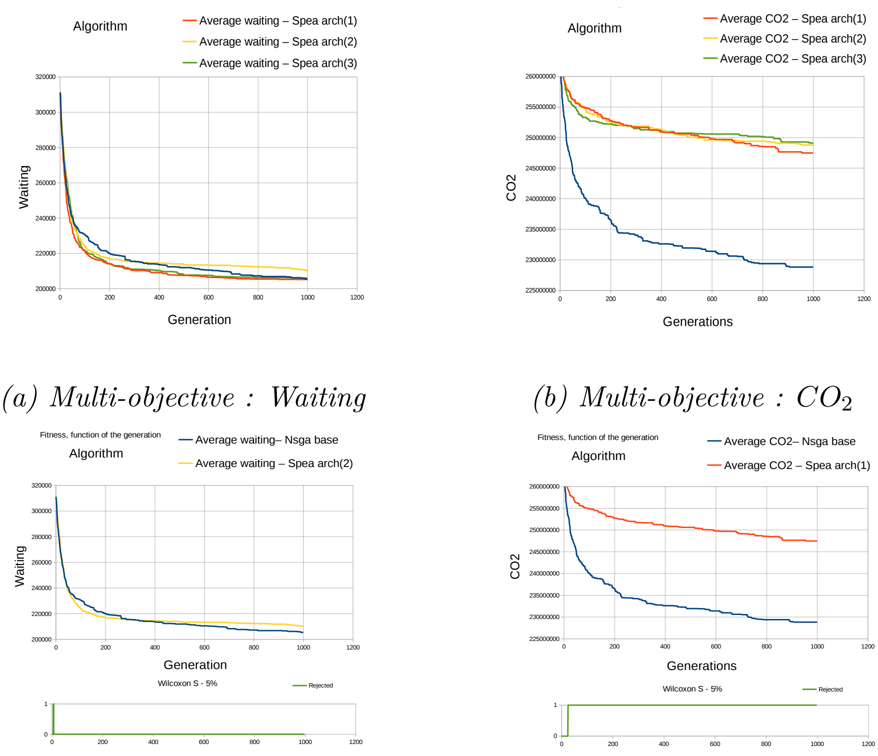
(b) Pareto-fronts for each run.



(c) Non-dominated points.

Figure A.3: Multi-objectives: Comparison of several algorithms.

Pareto-fronts of two versions of the algorithm used in the multi-objective evolutionary process for two objectives. In red Nsga-II, in orange, green and yellow different version of Spea-II with different size for the archive



(a) Multi-objective : Waiting

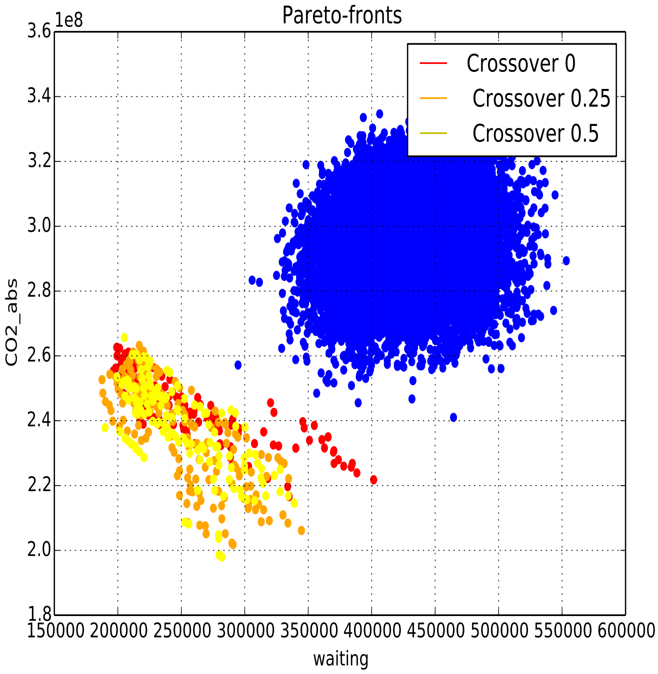
(b) Multi-objective : CO_2

(c) Comparison : Waiting

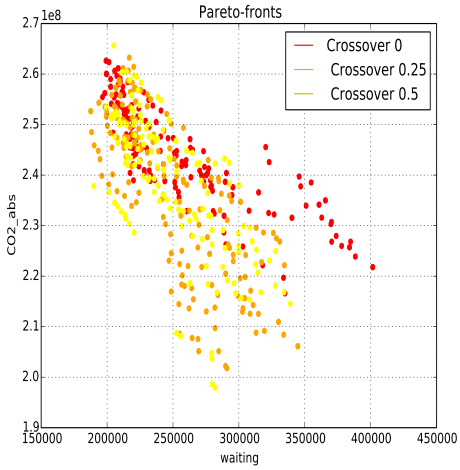
(d) Comparison : CO_2

Figure A.4: Multi-objectives: Comparison of several algorithms.

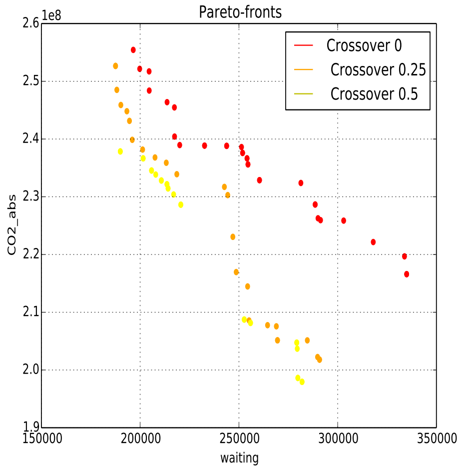
The first graph represents the evolution of the indicators as a function of the generations. The second graph represents a statistical test on the different runs for each experiment. A value of 1 indicates a statistically significant difference between the two distributions. The test used in this part is the test of Wilcoxon S at 5%. Those graph represent the results for different algorithms. In red Nsga-II, in orange, green and yellow different version of Spea-II with different size for the archive



(a) Multi-Objective optimization: Random-clouds and Pareto-fronts



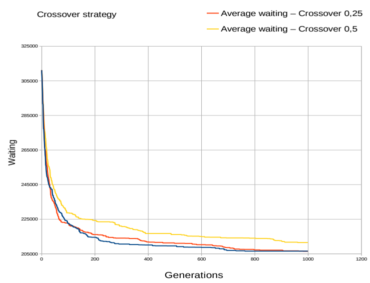
(b) Pareto-fronts for each run.



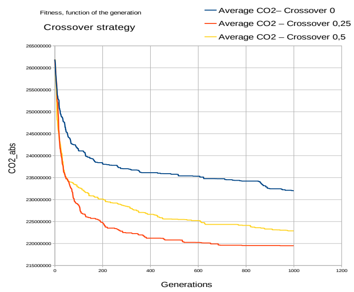
(c) Non-dominated points.

Figure A.5: Multi-objectives: Comparison of several crossover strategies.

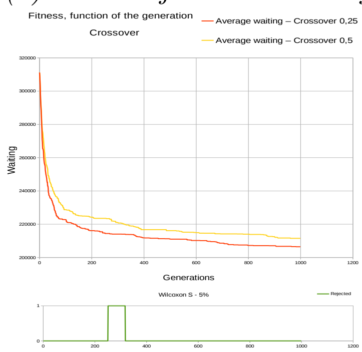
Pareto-fronts of two versions of the algorithm used in the multi-objective evolutionary process for two objectives. In red a crossover parameter of $p_m = 0$ has been used, in yellow $p_m = 0.25$ and in orange $p_m = 0.5$



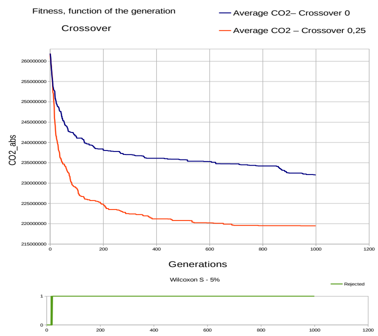
(a) *Multi-objective : Waiting*



(b) *Multi-objective : CO_2*



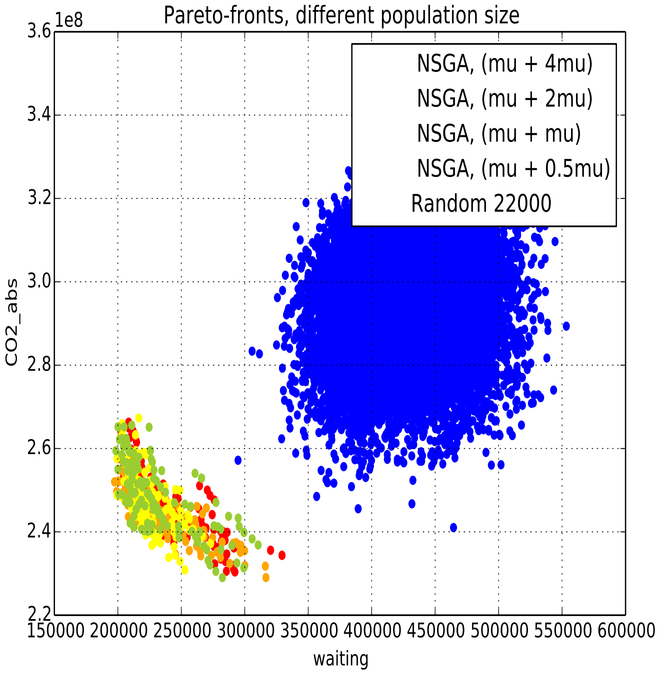
(c) *Comparison : Waiting*



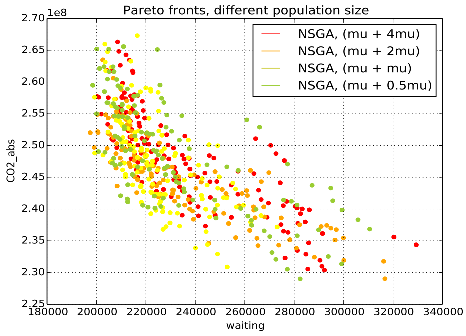
(d) *Comparison : CO_2*

Figure A.6: Multi-objectives: Comparison of several crossover strategies.

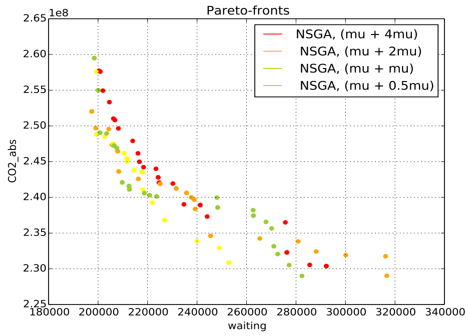
The first graph represents the evolution of the indicators as a function of the generations. The second graph represents a statistical test on the different runs for each experiment. A value of 1 indicates a statistically significant difference between the two distributions. The test used in this part is the test of Wilcoxon S at 5%. Those graph represent the results for different crossover parameters.



(a) *Multi-Objective optimization: Random-clouds and Pareto-fronts*



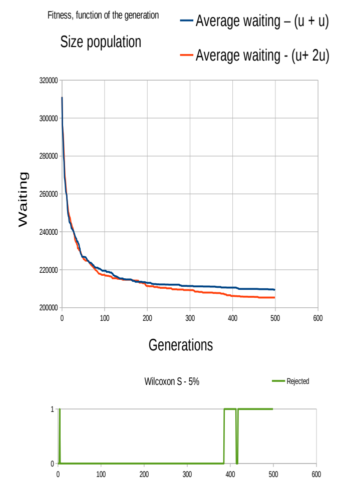
(b) *Pareto-fronts for each run.*



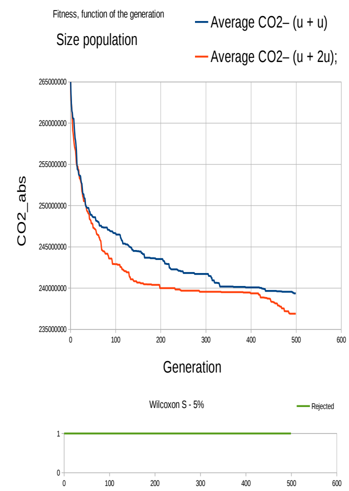
(c) *Non-dominated points*

Figure A.7: Multi-objectives: Comparison for different population sizes (λ)

Pareto-fronts of different versions of the algorithm used in the multi-objective evolutionary process for different size for the offspring population λ.



(a) Multi-objective : Waiting



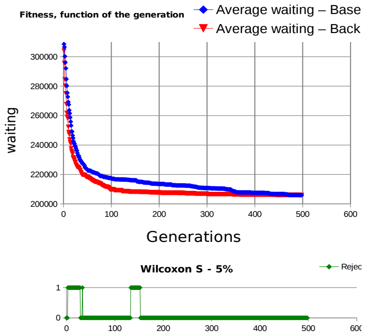
(b) Multi-objective : CO2

Figure A.8: Multi-objectives: Comparison for different population sizes (λ)

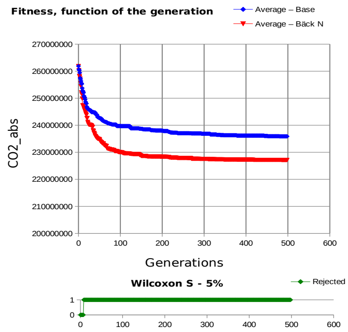
The first graph represents the evolution of the indicators as a function of the generations. The second graph represents a statistical test on the different runs for each experiment. A value of 1 *indicates a statistically significant difference between the two distributions. The test used in this part is the test of Wilcoxon S at 5%. Those graph represent the results for different crossover parameters.*

Appendix B

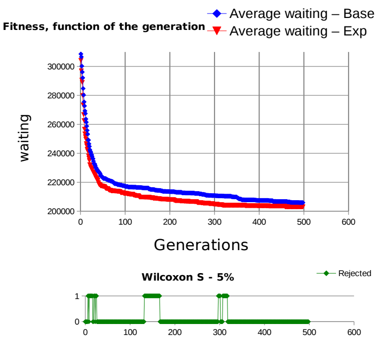
Traffic Optimization - Control strategies



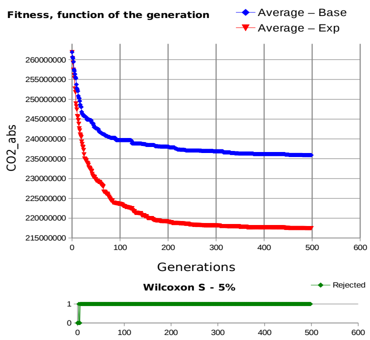
(a) *Mono-objective: Waiting. Comparison: Base VS Back*



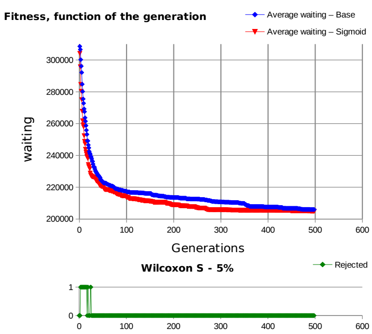
(b) *Mono-objective:* CO_2*. Comparison: Base VS Back*



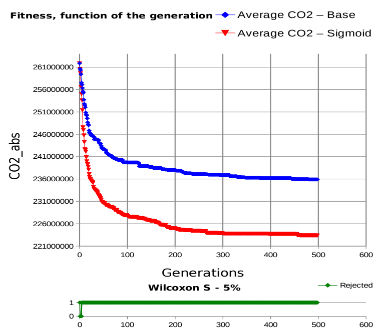
(c) *Mono-objective: Waiting. Comparison: Base VS Exponential*



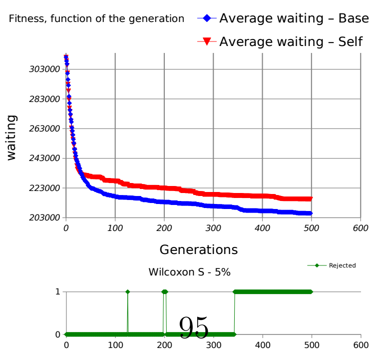
(d) *Mono-objective:* CO_2*. Comparison: Base VS Exponential*



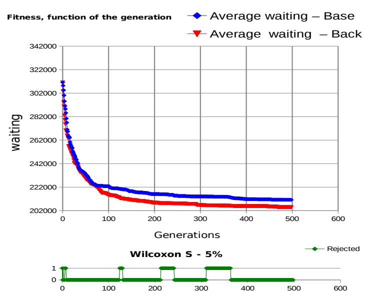
(e) *Mono-objective: Waiting. Comparison: Base VS Sigmoid*

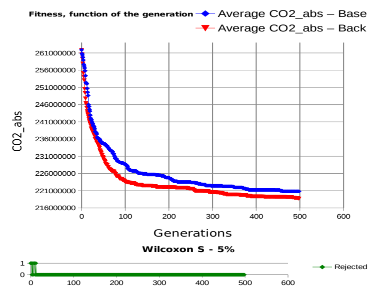


(f) *BMono-objective:* CO_2*. Comparison: Base VS Sigmoid*



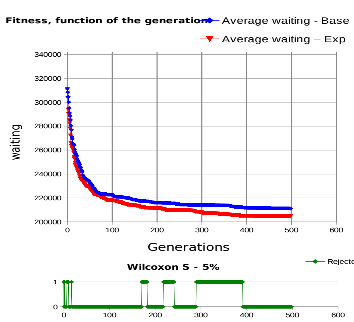
(g) *Mono-objective: Waiting. Comparison: Base VS Self-adaptive*

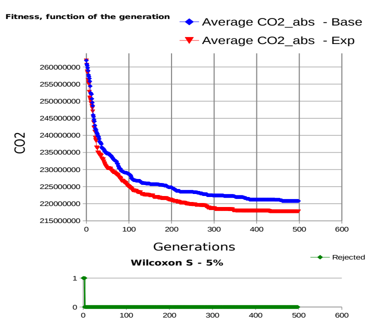




(a) Multi-objective: Waiting. Comparison: Base VS Back

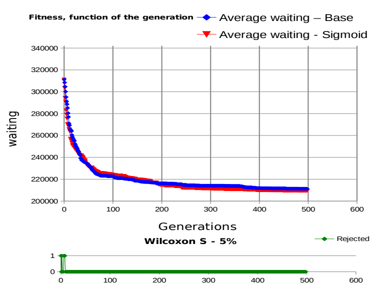
(b) Multi-objective: CO_2. Comparison: Base VS Back

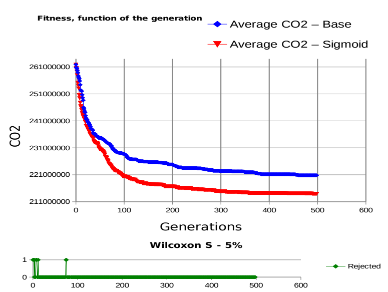




(c) Multi-objective: Waiting. Comparison: Base VS Exponential

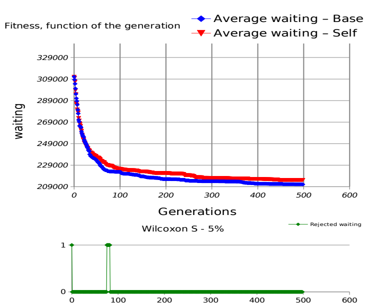
(d) Multi-objective: CO_2. Comparison: Base VS Exponential

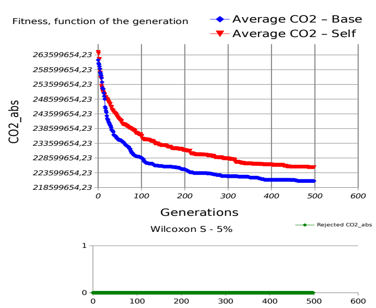




(e) Mono-objective: Waiting. Comparison: Base VS Sigmoid

(f) Multi-objective: CO_2. Comparison: Base VS Sigmoid





(g) Multi-objective: Waiting. Comparison: Base VS Self-adaptive

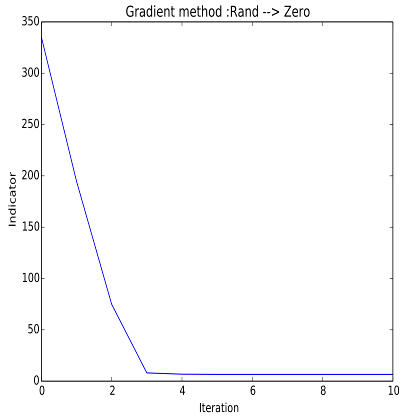
(h) Multi-objective: CO_2. Comparison: Base VS Self-adaptive

Figure B.2: Multi-objectives - Control Strategies.

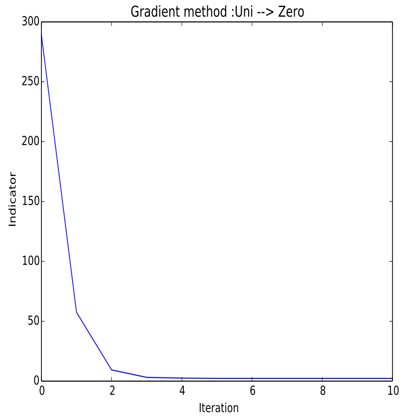
The first graph represents the evolution of the indicators as a function of the generations. The second graph represents a statistical test on the different runs for each experiment. A value of 1 indicates a statistically significant difference between the two distributions. The test used in this part is the test of Wilcoxon S at 5%. Those graph

Appendix C

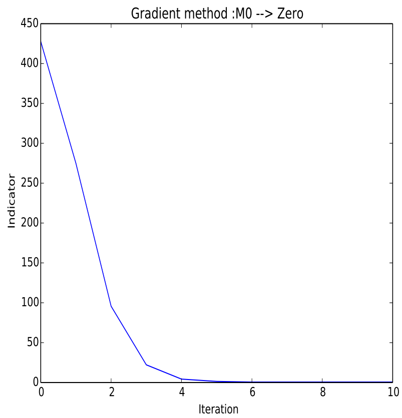
Traffic Modeling - GSA



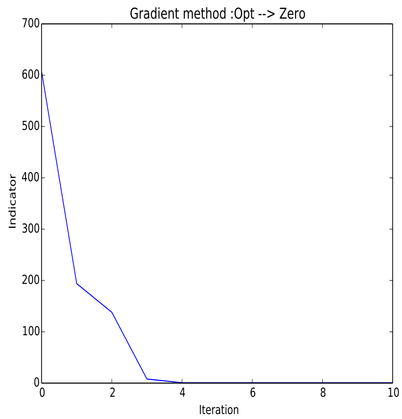
(a) *Initial ODM: Random ODM*



(b) *Initial ODM: Uniform ODM*



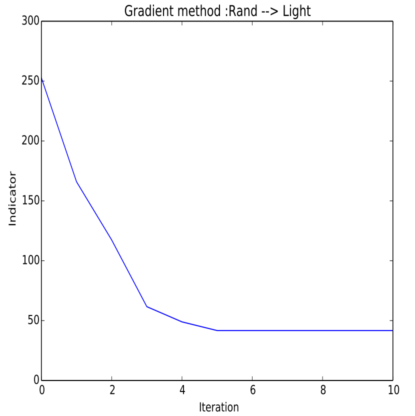
(c) *Initial ODM: Heuristic-based matrix*



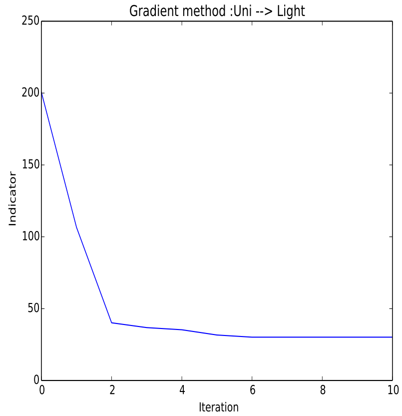
(d) *Initial ODM: optimized ODM*

Figure C.1: GSA - Simple case - No demand.

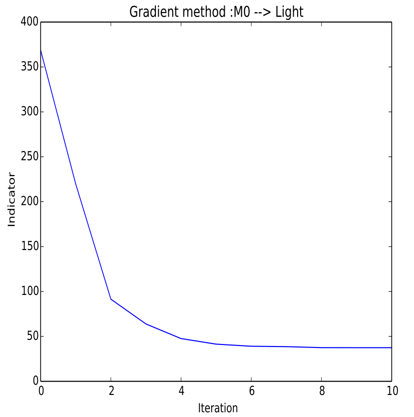
Evolution of the indicator as a function of the iteration of the GSA for a scenario with a low demand for different initial ODM. The indicator is the mean over the streets of the ***absolute*** *error in the traffic counts. In this scenario the demand is set to 0.*



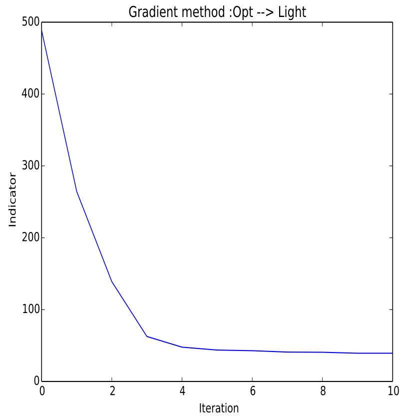
(a) Initial ODM: Random ODM



(b) Initial ODM: Uniform ODM



(c) Initial ODM: Heuristic-based ODM



(d) Initial ODM: Optimized ODM

Figure C.2: GSA - Simple case - three flows.

Evolution of the indicator as a function of the iteration of the GSA for a scenario with a low demand for different initial ODM. The indicator is the mean over the streets of the ***absolute*** *error in the traffic counts. In this scenario the demand is set to* 0 *except for three (randomly chosen) path.*