<http://www.diva-portal.org>

Postprint

This is the accepted version of a paper published in *Advances in Engineering Software*. This paper has been peer-reviewed but does not include the final publisher proof-corrections or journal pagination.

Citation for the original published paper (version of record):

Jin, J., Ma, X., Konsonen, I. [Year unknown!]

A stochastic optimization framework for road traffic controls based on evolutionary algorithms and traffic simulation.

Advances in Engineering Software

Access to the published version may require subscription.

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-210160>

A stochastic optimization framework for road traffic controls based on evolutionary algorithms and traffic simulation

Junchen Jin^a, Xiaoliang Ma^{a,*}, Iisakki Kosonen^b

^a*System Simulation & Control (S2CLab), Department of Transport Science,
KTH Royal Institute of Technology, Teknikringen 10, Stockholm 10044, Sweden.*

^b*Department of Civil and Environmental Engineering,
Aalto University, P.O. Box 12100, FI-00076, Finland.*

Abstract

Traffic flow is considered as a stochastic process in road traffic modeling. Computer simulation is a widely used tool to represent traffic system in engineering applications. The increased traffic congestion in urban areas and their impacts require more efficient controls and management. While the effectiveness of control schemes highly depends on accurate traffic model and appropriate control settings, optimization techniques play a central role for determining the control parameters in traffic planning and management applications. However, there is still a lack of research effort on the scientific computing framework for optimizing traffic control and operations and facilitating real planning and management applications. To this end, the present study proposes a model-based optimization framework to integrate essential components for solving road traffic control problems in general. In particular, the framework is based on traffic simulation models, while the solution needs extensive computation during the engineering optimization process. In this study, an advanced genetic algorithm, extended by an external archive for storing globally elite genes, governs the computing framework, and it shows superior performance than the ordinary genetic algorithm because of the reduced number of fitness function evaluations in engineering applications. To evaluate the optimization algorithm and validate the whole software framework, this paper illustrates a detailed application for optimization of traffic light controls. The study optimizes a simple road network of two intersections in Stockholm to demonstrate the model-based optimization processes as well as to evaluate the presented algorithm and software performance.

Keywords: simulation-based optimization, archived genetic algorithm, road traffic controls, traffic light control, object-oriented software framework.

1. Introduction

Nowadays, traffic congestion causes important problems in urban areas including the undermined mobility efficiency and increased fuel consumption and air pollution because of the increasing traffic demand. Road traffic controls are important for mitigating congestion and negative environmental impacts. According to [Papageorgiou et al. \(2003\)](#), different traffic control strategies are mainly provided for three areas: urban road networks (e.g. intersection and network traffic light controls), freeway networks (e.g. ramp metering, lane control, and variable speed limit) and route guidance. The effectiveness of traffic control strategies greatly depends on their abilities to react to the live traffic conditions. Therefore, advanced traffic modeling becomes an essential part of the latest traffic control strategies.

While emerging technologies in intelligent transport systems (ITS) have the potential to change the existing infrastructure, there is still a rather long way ahead before they overwhelm the existing ones and become economically more feasible. For road traffic control systems, municipality administrations around the world often apply optimization methods to determine the settings of traffic control measures. In the engineering practice, traffic control parameters are usually pre-tuned according to historical traffic demands. Meanwhile, current policy often requires

*Corresponding author. Tel: +46 87908426; Email: liang@kth.se.

multiple planning objectives not only for better mobility efficiency, but also for considering more efficient energy usage and environmental sustainability.

As the processes in road transport system are stochastic and nonlinear with many parameters, nature-inspired optimization approaches have been commonly used to solve the optimization problems for traffic control systems. For instance, traffic light control is one of the most widely used traffic control instruments. Moreover, genetic algorithms (GAs), one of the nature-inspired optimization techniques, probably are the most frequently used approach for the applications of traffic light optimization (e.g. [Teklu et al., 2007](#); [Park and Lee, 2009](#); [Sánchez-Medina et al., 2010](#); [Ma et al., 2014b](#); [Stevanovic et al., 2015](#)). Some studies, including the one by [Stevanovic et al. \(2015\)](#), have brought utilities for the practice in traffic engineering concerning different aspects including mobility, safety, and environmental impacts. However, the proposed approaches are limited to the dedicated traffic models, and few of them introduced the detailed implementations, especially concerning engineering software development and more efficient optimization algorithms.

This study intends to propose a general optimization framework for road traffic controls. The implementation of such a computing framework is due to the engineering requirements in several R&D projects for optimizing traffic lights, lane settings and other measures in collaboration with the Swedish national and municipality transport administrations. A literature review from the aspect of simulation-based optimization for traffic controls is presented in the next section. Whereas the computing framework is introduced in the [section 3](#), this paper presents more implementation details for a traffic light control application in a later section. A case study is then performed to evaluate the optimization algorithm and demonstrate the capability of the proposed software. Finally, the paper is finalized by summary and perspectives on future research.

2. Literature review

In general, [Bierlaire \(2015\)](#) presented the role of simulation in traffic and transportation research, with a focus on simulation-based optimization. When it comes to traffic flows, they can be represented by macroscopic models, or by mesoscopic models, or by microscopic models ([Barceló et al., 2010](#)). Whereas macroscopic simulation model represents traffic flows from an aggregated point of view based on a hydrodynamic analogy, traffic flow is described from the dynamics of the individual particles (the vehicles) in microscopic simulation models. Mesoscopic models, based on a simplification of vehicular dynamics, represent an intermediate alternative for modeling traffic flows.

A large amount of research effort has been put into optimizing traffic controls using aggregated simulation models. For example, [Ukkusuri et al. \(2010\)](#) applied a macroscopic simulation model, cell transmission model (CTM), to represent traffic dynamics in a traffic light optimization problem. [Liu et al. \(2015\)](#) investigated an optimal variable speed limit (VSL) system capable of optimizing the system designs when the variable message signs (VMSs) are movable using a GA approach. [Dell’Orco et al. \(2014\)](#) proposed an artificial bee colony algorithm to find the optimal settings of a traffic light controller in a simulation environment enabled by TRANSYT-7F which integrates a macroscopic simulation model.

Mesoscopic models have been increasingly developed, such as [Burghout \(2004\)](#) and [Celikoglu and Dell’Orco \(2007\)](#), due to the significant decreases of the efforts involved in traffic modeling compared to microscopic models, and the higher fidelity of traffic system representation than macroscopic models. Nevertheless, previous studies seldom investigated the effects of traffic control systems when traffic flows are mesoscopically modeled. [Di Gangi et al. \(2016\)](#) has pioneered a network optimization strategy for traffic light control system based on a mesoscopic traffic flow model. The employed mesoscopic model allows simulating queue and spillback phenomena.

Macroscopic and mesoscopic traffic models usually do not provide sufficient details to characterize the operations of advanced traffic controls or the emerging data where vehicle-level information is required to explicitly model the operations. Over the past decade, different microscopic traffic tools (e.g. VISSIM ([Fellendorf and Vortisch, 2010](#)), AIMSUN ([Barceló and Casas, 2005](#)), and SUMO ([Krajzewicz et al., 2012](#))) have been applied for optimization of the traffic control systems when detailed vehicle information is required. For instance, [Khondaker and Kattan \(2015\)](#) presented a variable speed limit (VSL) control system in a connected vehicle environment, which focused on individual driver’s behavior. A multi-objective optimization function with respect to the control parameters was formulated to find a balanced trade-off among the mobility, safety and environmental benefits. [Osorio and Bierlaire \(2013\)](#) proposed a simulation-based optimization framework for solving the traffic light control problems in large-scale urban areas.

Their optimization algorithm is based on an analytical approximation of the objective function combining the information from AIMSUN. However, there is a lack of research effort on proposing a general model-based optimization framework that considers the varieties of traffic simulation models and is applicable to the optimization of different traffic control instruments.

The most challenging problem of microscopic simulation-based optimization is the heavy computational burdens since nature-inspired optimization algorithms inherently impose iterative simulations. In each simulation run, detailed states (including position, speed and acceleration) of each vehicle are computed every simulation step. The total computational time can be reduced if traffic simulations are executed in parallel. One example study was conducted by [Sánchez-Medina et al. \(2010\)](#) who implemented an optimization framework in the context of using microscopic simulations based on cluster computing. Few studies, however, have focused on the enhancement of the optimization algorithm to reduce the computational load when microscopic traffic simulation models are incorporated.

3. Model-based optimization framework

This section introduces a general computing framework for optimizing traffic control measures. It has three essential elements: the optimization engine, the models of traffic system (including traffic simulator and control measures) and the system performance estimator. We begin with the introduction of the optimization engine.

3.1. GA-based optimization algorithms

While the optimization technique may vary in the proposed framework, the scope of this study only considers GA algorithm, an evolutionary algorithm that has been frequently used in the practice of traffic controls. GA starts with a randomly generated population composed of feasible solutions that evolves through its generations towards the global solution. To create a new generation, GA usually performs *selection*, *crossover*, and *mutation* operators. Here, parents are described by the strings of bits.

3.1.1. Basic operators

Before the operations, an encoding scheme, often binary encoding, transforms those traffic parameters (often integers) to a string of 0 or 1 before the population is further manipulated. The size of the string is predefined. The lower bound value in decimal corresponds to all zero digits in a bit string, whereas the upper bound value in decimal is represented by all one digits in a bit string. The values between the lower bound and the upper bound are linearly scaled and associated to the corresponding binary strings. Such an encoding of a control parameter ϕ could be analytically represented by

$$\tilde{\phi} = (b_l \dots b_i \dots b_1)_2 = fl\left(\frac{\phi - \phi^{min}}{\phi^{max} - \phi^{min}} \times (2^l - 1)\right) \quad (1)$$

where $\tilde{\phi}$ denotes the signal parameter that is encoded in a bit string, and b_i represents the i^{th} bit in the chromosome; l represents the size of the bit string; $fl(\cdot)$ is the floor function mapping a real number to the largest previous following integer; ϕ^{min} and ϕ^{max} are the minimum value and the maximum value of ϕ , respectively.

[Fig. 1](#) illustrates the process of GA operators for each generation. [Fig. 1\(a\)](#) represents the selection procedure for parents (P_A and P_B). The selection procedure may differ due to real applications, such as random selection, tournament selection and truncation selection ([Thierens and Goldberg, 1994](#)). The applied selection procedure is described in the next subsection in detail. The process of generating offspring population is driven by the operators of uniform crossover and bit-flip mutation. Each part of the father's bit string has an opportunity to exchange with the counterpart of the mother's bit string in the uniform crossover procedure. For example, the bits in the gray squares in [Fig. 1\(c\)](#) carry out the crossover operation. [Fig. 1\(d\)](#) shows the bit-flip mutation procedure for the chromosome $C_{A'}$. The mutation process produces random changes in various bit strings spontaneously, and such changes are represented by inverting the bit (0 changes to 1 and vice versa).

After reproducing the new population, chromosomes are inversely decoded from bit string to integer and is interpreted as control parameters. For instance, $C_{A'}$ changes to the signal parameter g (shown in [Fig. 1\(e\)](#)). The decoding of a chromosome $(b'_l \dots b'_i \dots b'_1)_2$ can be presented by

$$\phi' = fl\left(\frac{\sum_{i=1}^l b'_i 2^{i-1}}{2^l - 1} \times (\phi^{max} - \phi^{min}) + \phi^{min}\right) \quad (2)$$

Fig. 1. Demonstration of the GA operators applied in the computational framework.

where ϕ' is the decoded signal parameter.

In order to obtain the fitness values for a newly created population, evaluations must be carried out. In traffic control applications, traffic models are applied to estimate the system performance measures. This study focuses on stochastic models of traffic system, usually requiring multiple simulation runs during the evaluation process. An inventory database has been created to store evaluation results for a new set of control parameters (see subsection 3.2).

3.1.2. Archived GA algorithm

In stochastic traffic optimization, the computational time spent for performing the GA operators is often negligible compared to the time for performance estimation using simulation models. Consequently, it is of high importance to enhance the ordinary GA algorithm by reducing the total computational time for evaluating fitness functions. The latest research in computational intelligence proposed an archive-based genetic algorithm (AGA) that employs a selection procedure which uses a very small population size together with a large-size external archive (Tiwari et al., 2011). The external archive stores the global elites ever found to accurately approximate the best solution in applications. The selection procedure benefits from the search history (i.e., the external archive) of the algorithm and attempts to minimize the number of function evaluations required to achieve the desired convergence. The previous numerical study of the AGA approach shows that it attains the desired convergence faster than ordinary GA for benchmark optimization problems. The detailed description of AGA can be found in Tiwari et al. (2011).

Because of the advantages of AGA in computational efficiency, the algorithm is implemented for our application and it plays the central role in finding optimal solution. In addition, considerations on the quality of search space have been made in this study. Firstly, randomly distributed initial population, covering global search space, is created by applying the Latin Hypercube (LH) sampling algorithm (Loh et al., 1996). For each variable, evenly-sized segments are created. A real number bounded by the respective segment is generated according to a uniform distribution. The number within one segment is randomly assigned to an individual in the initial population as one of the variables. This process is proven to assure that the resulted population, spanning the entire exploration space, is sufficiently random while being free from any bias.

Fig. 2 summarizes the conceptual steps of archived GA-based optimization to facilitate control parameters by the pseudo-code. At the beginning, an inventory database D is created for storing the evaluated results so that repeated evaluation of similar traffic control parameters by simulation can be avoided (line 2 in the pseudo-code). The size of parents n and the size of archive m are then defined (lines 3-4 in the pseudo-code). After that, the algorithm handles the initial population (lines 6-9 in the pseudo-code). First, the initial population is selected randomly by the LH sampling algorithm from the parameter space. Fitness values of the initial population are estimated by running traffic simulation evaluations. In addition, solutions in the initial population are copied to the archive and also saved into the inventory database. At the beginning of each generation, a handful of well-performed solutions, namely micro-elitism solutions, are extracted from the archive and used as a part of the parent population.

After such an initialization process, the archived GA runs for pre-defined iterations for evolution. In each gen-

```

/* Initialization: */
1  $D$ : database storing evaluation results
2  $n$ : size of parents for each generation
3  $m$ : size of archive

/* Handle initial population: */
4  $P_0 \leftarrow \text{get\_initial\_population}()$ 
5  $F_0 \leftarrow \text{get\_evaluation\_from\_parallel\_simulation}(P_0)$ 
6  $D \leftarrow \text{update\_database}(P_0, F_0)$ 
7  $E \leftarrow \text{update\_archive}(P_0, F_0)$ 

/* Evolution: */
8 while termination criteria are not satisfied do
    /* Create parent population from the archive: */
    9  $P_{micro} \leftarrow \text{extract\_micro\_elitism\_solutions}(E, n)$ 
    /* Manipulate population to produce offsprings: */
    10 for  $i = 1$  to  $n$  do
        11  $Q \leftarrow \text{tournament\_selection}(E)$ 
        12  $\text{binary\_encoding}(Q, P_{micro}[i])$ 
        13  $P_{imp}[i] \leftarrow \text{uniform\_crossover}(Q, P_{micro}[i])$ 
        14  $P_{new}[i] \leftarrow \text{bit\_flip\_mutation}(P_{imp}[i])$ 
        15  $\text{decoding}(P_{new}[i])$ 
    16 end
    /* Get new fitness values: */
    17 for  $i = j$  to  $n$  do
        18  $\text{data\_flag} \leftarrow \text{check\_in\_database}(P_{new}[j], D)$ 
        19 if  $\text{data\_flag} == \text{True}$  then
            20  $F_{new}[j] \leftarrow \text{get\_evaluation\_from\_database}(P_{new}[j], D)$ 
        21 else
            22  $F_{new}[j] \leftarrow \text{get\_evaluation\_from\_parallel\_simulation}(P_{new}[j])$ 
        23 end
    24 end
    /* Update archive using the new parents: */
    25  $D \leftarrow \text{update\_database}(P_{new}, F_{new})$ 
    26  $E \leftarrow \text{update\_archive}(P_{new}, F_{new})$ 
27 end

```

Fig. 2. Pseudo-code of Archived GA-based Optimization.

eration, n individuals (P_{micro}) are selected from the micro elitism solutions which have the best fitness values in the archive (E). Every individual in the P_{micro} is in turn used as one of the parents to reproduce children with the other parent (Q) using a tournament selection (lines 15-21 in the pseudo-code). In the tournament selection, a certain number of individuals are randomly chosen. The selected individuals participate into a tournament, and the individual with best fitness value becomes the winner. This process is repeatedly until a pre-defined number of winners are collected. To be regarded as the parents in the following generation, the winners are required to be mutually different from each other.

Then, the introduced binary encoding, uniform crossover, bit flip mutation, and decoding operators are employed to obtain the offspring (P_{new}). In the algorithm, adaptive crossover and mutation probabilities are used to adjust the balance between the capacity to explore new distant regions and the ability of exploiting the regions within the neighborhood around the previously visited points. Thus, the premature convergence (too much exploitation) and blind random search (too much exploration) phenomena can be avoided. The applied methods for adapting the crossover

Fig. 3. The generalized optimization framework for traffic control problems.

and mutation probabilities are presented in McGinley et al. (2011). The two probabilities vary in pre-defined ranges based on the standard population diversity (SPD) describing a population's solution space diversity. In particular, the parent population is divided into an exploration section and an exploitation section according to the SPD. The mutation operator is employed with a high probability in the exploration section to explore potentially unvisited areas while is employed with a low probability in the exploitation section for local-search mechanism.

The fitness values F_{new} of the newly obtained offspring are obtained one by one (lines 23-33 in the pseudo-code). Each individual in the offspring is checked whether the same set of control parameters has been stored in the database D . If the individual is ever stored in the database, the stored fitness value is assigned to its fitness value. Otherwise, the fitness values are extracted from traffic simulations. At the end of each generation, the database is updated using the newly created population. The archive is updated by the parents with relatively better fitness values.

3.2. Generalized computing framework

After presenting the GA-based optimization engine and enhancement, we can take a close look of the general computing framework. Fig. 3 shows the component diagram with relevant data flow. The whole framework consists of three essential components (blocks), *Software-in-the-loop Simulation*, *Evolutionary Optimizer* and *Performance Estimator*. The optimization process starts with the *Evolutionary Optimizer* module, sending initial population of control parameters to the *Software-in-the-loop Simulation* (SILS) module. *Traffic Controller* and *Traffic Simulator* are included into the SILS module.

The SILS module has three major components: *Traffic Simulator*, *Traffic Controller* and *Simulator-Controller Program*. *Simulator-Controller Program* is responsible for the communication between *Traffic Controller* and *Traffic Simulator*. Control indications from *Traffic Controller* are imported into the traffic model in *Traffic Simulator* through the interface program whereas the traffic states are provided to the controller for making control decisions.

Different traffic controls can be implemented in the *Traffic Controller* module. For example, if the *Traffic Controller* module represents a fixed-time traffic light control, the control indications can refer to green durations for traffic lights. Loop detectors are usually deployed at signalized intersections which send the detected traffic states to traffic light controller. In addition, if a speed limit control is the employed instrument, control indications can be the value of limited speed on each traffic sign. The detected traffic states may include the information of traffic flow, density, and mean speed on the detection spot.

Traffic Simulator is a module that describes the dynamics of traffic system. In real applications, different models can be applied. Traffic models can be classified according to their fidelity when representing real-world traffic events. The models with the lowest fidelity are called macroscopic models which describe traffic flow analogously to liquids or gases in motion. Such models are aggregated and normally governed by analytical equations. On the other hand, microscopic models describe individual driver behavior, such as acceleration, braking and lane-changing actions, as well as detailed vehicle movements. Besides, there are also intermediate approaches. For example, researchers have combined partial attributes of microscopic and macroscopic models to so called mesoscopic approaches. For traffic control, a convenient approach is to apply high-fidelity models since vehicle trajectory information is often required for the evaluation of traffic measures. Besides, the microscopic approach becomes popular due to the adoption of different simulation software by traffic planners, industrial consultants and researchers in traffic control projects.

Dynamic traffic states are generated by traffic model and registered in computer memory. When the execution of traffic simulation is completed, simulated traffic states are subsequently treated as the inputs to *Performance Estimator*. Based on the pre-defined optimization objectives, performance measures in mobility efficiency, traffic safety, pollution and energy efficiency are then calculated using such characteristics data. Then, the estimated performance measures are provided to the optimizer so as to generate new control parameters for further optimization. The whole process is repeated until the termination criteria are finally met.

In an optimization process, multiple simulation runs must be performed for each set of traffic parameters in order to obtain statistical significant evaluation results. Therefore, the requirement spends most of the available computing resources. However, the proposed software framework is equipped with the capacity to distribute traffic simulation runs to a different process in a system with symmetric multiprocessing (SMP). This functionality has proved to be effective for solving large-scale optimization problems in real application.

4. Software and application in traffic light control

While the objective of this paper is to propose a generalized computing framework that is not confined to a particular traffic control application, this section presents an application of the proposed model-based optimization framework for traffic light control. Signal control is one of the most used road facilities to manage traffic movements in an efficient mode. Therefore, the computing framework is extended by incorporating traffic light control schemes and interfacing between traffic light controller and traffic simulator.

Fig. 4 illustrates a Unified Modeling Language (UML) class diagram for the detailed software design pattern and implementation of traffic light control in the proposed framework. *SimulationBasedOptimizationProgram* represents a singleton class that plays an essential role of coordinating the three components of the computing framework that are implemented as *SimulatorControllerProgram* class, *PerformanceEstimator* class and *EvolutionaryOptimizer* class. Moreover, the *SimulatorControllerProgram* class is a composite of the *SignalController* class and *TrafficSimulator* class. The rest of this section illustrates the detailed models being incorporated in those component classes for traffic light controller, traffic simulator and performance estimator.

4.1. Traffic light control schemes

Strategies of traffic light control can be defined from three perspectives: signal timing, signal phasing, and signal coordination. Signal timing refers to the method how the durations of traffic light indications are allocated. In a real signal operation, compatible traffic movements are possible to be given the same traffic light indications all the time. Signal phasing determines how compatible traffic movements are formed in an operation. Additionally, closely spaced intersections are usually coordinated to generate "green wave" on prioritized road or direction. In a "green wave" scenario, vehicles may progress along prioritized road through several signalized intersections without any stops. When the length of a signal duration is adjustable in such a coordinated system, some specifications on signal controllers are also required to be certain that the progression feature is not disrupted. In Fig. 4, signal timing, phasing and coordination strategies have been implemented using the object-oriented methodology.

Fixed-time (FT) and vehicle-actuated (VA) schemes are two commonly applied strategies for signal timing. Compared to FT timing, VA timing is more flexibility on allocating signal durations (Day et al., 2008). FT timing assigns pre-defined durations for traffic movements whereas the durations determined by VA timing depend on the real-time vehicle presences with the aid of on-street detectors. On the other hand, signal phasing can be classified into stage-based phasing (SP) and group-based phasing (GP) approaches. The GP-based phasing approach can dynamically

Fig. 4. UML class diagram for the traffic light control application.

integrate compatible traffic movements during the signal operation whereas the SP-based phasing approach assigns the right-of-way to pre-defined combinations of compatible movements. Nowadays, a growing number of GP-based controllers have been deployed due to its flexibility in phase generations. GP-based signal control has the potential to outperform the counterpart in the SP-based mode in many aspects, such as reductions in travel delays and fuel consumptions (Jin and Ma, 2014b).

4.2. Traffic simulator

Traffic simulator is used to describe the stochastic patterns of traffic system. Due to the different signal control schemes, microscopic traffic models provide a direct approach of generating vehicle movement patterns based on detailed driver behavior modeling. In transport engineering applications, several commercial or open-source microscopic traffic simulation tools have been popular for planners, researchers and other professionals, including VISSIM, AIMSUN, TransModeler, SUMO and so on. While all these simulators can be integrated in the optimization framework, SUMO, an open-source simulation software, has been the main tool for our study. SUMO provides a convenient interface, called TraCI, to communicate and obtain real-time vehicle states using socket connection. Traffic light control in SUMO adopts, by default, the fixed-time stage-based (FTSB) scheme.

Car following model is the essential component, defining the speed of the following vehicle related to the vehicle ahead, in microscopic traffic simulation software. SUMO applies Krauss's car following model (Krauss et al., 1997) generating vehicle acceleration based on safe speed profiles. The instantaneous vehicle speed for individual vehicle

at each time step Δt is determined by

$$v(t + \Delta t) = \max(0, v^{des}(t) - a(t) \cdot \Delta t) \quad (3)$$

where v^{des} denotes the desired speed at each time step t ; $a(t)$ refers to the acceleration caused by driver imperfection. v^{des} is taken as the smallest value among the maximum speed, the speed using the vehicle's maximum acceleration ability, and the safe speed, i.e.,

$$v^{des}(t) = \min(v^{max}, v(t) + a^{max} \cdot \Delta t, v^{safe}(t)) \quad (4)$$

where v^{max} and a^{max} refer to the maximum speed and acceleration ability. $v^{safe}(t)$ is the safe speed, defined as

$$v^{safe}(t) = v^l(t) + \frac{g(t) - v^l(t)\zeta}{(v(t) + v^l(t))/2a^{min} + \zeta} \quad (5)$$

where $v^{safe}(t)$ is the safe speed at time t ; $v^l(t)$ is the speed of the preceding vehicle; g is the gap between vehicle and its preceding vehicle; ζ is the reaction time of the driver in vehicle, and a^{min} is the maximum deceleration. The driver imperfection is modeled as a stochastic deceleration, i.e.,

$$a(t) = r \cdot a^{max} \cdot \epsilon \quad (6)$$

where r is a random number generated from a uniform distribution between 0 and 1 and $\epsilon \in [0, 1]$ is an input parameter depending on the degree of imperfection.

4.3. Performance estimator

Performance estimator provides a functional class in Fig. 4 to evaluate the system performance under traffic control measures. Travel delay and fuel consumption are considered as the main indicators in our analysis. Travel delay is explicitly computed using instantaneous vehicle states obtained from microscopic traffic simulation. Average travel delay (seconds/vehicle) can be derived as follows:

$$\bar{d} = \frac{\sum_{j=1}^N d_j}{N} \quad (7)$$

where d_j denotes the travel delay for vehicle j and N is the number of vehicles. The travel delay is defined as the difference between the actual travel time and the ideal travel time to complete the journey. The ideal travel time is estimated as the time of finishing the journey with desired speed. This is derived as follows:

$$d_j = \Delta t_j - \frac{td_j}{v_j^{des}}, \quad (8)$$

where Δt_j denotes the time interval between two actions, td_j is the traveled distance within the time interval Δt_j . v_j^{des} refers to the desired speed of vehicle j .

Fuel consumption is calculated in this study by a micro-scale emission model, Comprehensive Modal Emission Model (CMEM), using vehicle trajectories. Accordingly, CMEM considers the modal operation of a vehicle and predicts instantaneous emissions of vehicle as a function of driving modes. The *PerformanceEstimator* class calls CMEM using a pre-calculated look-up table to retrieve the fuel consumption result. In real application, CMEM computations also can be distributed to other machines to save computing resource (Ma et al., 2014a).

4.4. Optimization process

Fig. 5 shows a sequence diagram representing the whole optimization process in a use case. The relations of the functions involved in the sequence diagram are defined in the class diagram in Fig. 4. The optimization process starts by instantiating a *SimulationBasedOptimizationProgram* object as well as calling the function *run_opt*. The initialization of the *sim_opt* object triggers the constructor of a *SimulatorControllerProgram* object and sets signal parameters in *SignalController*. Then, a traffic simulation run is started. During the simulation, traffic light indications

Fig. 5. Demonstration of the optimization process for traffic light control application using a sequence diagram.

are obtained at each time step from the signal controller by calling *run_one_step_con*. Traffic data are saved every simulation step. After a simulation run is finished, *sim_opt* invokes *sim_opt* again to get all vehicle states. The fitness function for GA optimization is computed by a *PerformanceEstimator* object. The *EvolutionaryOptimizer* class is instantiated and called to run the optimization algorithm based on the signal control parameters and the fitness function evaluation. Then, the new control parameters are sent back to the *sim_opt* object and replace the old parameters. The program iteratively updates a population of control parameters until the termination criteria are finally met. When the whole evolution process is terminated, optimal solution indicated by the fitness value is eventually obtained.

5. Case study

5.1. Problem formulation

To demonstrate the proposed optimization framework, computational experiments are set up for traffic light control at a small road network in Stockholm, Sweden. A group-based vehicle actuated control system, called LHOVRA, has been widely deployed for isolated intersections in Sweden and many other European countries (Kronborg and

Fig. 6. Illustration of vehicle-actuated timing scheme.

Fig. 7. An example of group-based phasing technique.

Davidsson, 1993). Fig. 6 shows a typical configuration of detectors in the LHOVRA system. That is, one short detector and one long detector are designated by their distances from the stop line. Specifically, $D80$ represents that the detector is 80 meters far away from the stop line. The extension signal will be authorized if the time gap between the passing vehicles is smaller than the defined gap threshold. For example, the green time is extended from point A to point B in Fig. 6. Usually, the vehicle is able to reach $D10$ within the extension period. Then, the traffic light keeps green if the long detector, $D10$, is occupied.

5.1.1. Signal group

The basic unit of LHOVRA control is a signal group which is defined as a collection of traffic movements with a same inbound direction. The traffic lights associated with a signal group always show the same indications. A phase is a set of signal groups that can be operated concurrently without conflicts among major movements. In practice, some traffic movements are, however, allowed to proceed during a phase even though they cause conflicts. For example, pedestrians are commonly allowed to proceed across intersections even though right-turn movements are occurring.

Fig. 7 gives an example on how group-based control may provide a more flexible phase structure. All compatible signal groups are possible to form a phase. Conflict matrix (on the right side of the figure) is used to represent the conflicts between signal groups. Value 0 represents that signal groups can be served simultaneously. Inter-green times between the signal groups are assigned into gray squares. Also, the left diagram in Fig. 7 gives an example of the operation for group-based phasing. Assume signal group $SG1$ is activated. Signal group $SG1$ is able to combine with

$SG2$, $SG3$ as well as $SG4$. Thus, three possible combinations are represented by phase $PH1$, phase $PH2$ and phase $PH3$. In practice, the decision of a signal group to be combined with depends on the detection information. If phase $PH1$ is determined, the following phase can be either phase $PH3$ or phase $PH2$ or phase $PH4$ depending on which one of the signal groups in phase $PH1$ is terminated. In principle, in the case of multiple conflicting requests, an ordered sequence of signal groups is used by taking account into the detection information. Consequently, the phase sequence is capable of being flexibly generated in real-world traffic signal operations.

5.1.2. LHOVRA timing control

Usually, a signal group is defined to be terminated after executing green time, yellow time and all-red time. As LHOVRA applies group-based vehicle-actuated timing technique, the length of green time depends on the pattern of vehicle actuations during the cycle. A minimum green time, $g_{i,u}^{min}$, is required to assign to the signal group u at intersection i . $g_{i,j,u}^{ext}$ denotes the green extension time for the signal group u at j th cycle. The duration of green time is computed by the summation of minimum green time and the total green extension time, i.e.,

$$g_{i,j,u} = g_{i,u}^{min} + g_{i,j,u}^{ext}, \quad g_{i,u}^{min}, g_{i,j,u}^{ext} \in \mathbb{N}. \quad (9)$$

The minimum green time is often pre-defined due to safety considerations. The green extension time is determined by the real-time detector information and control logic. In LHOVRA, the green extension time will be continuously offered if the time headway between two vehicle detections is sufficiently small and meanwhile the maximum green time for the signal group has not been reached. This can be analytically expressed by

$$g_{i,j,u}^{ext} = F(q_{i,j,u}) \quad (10)$$

where $q_{i,j,u}$ denotes the detection information obtained during the j th cycle for signal group u . $F(\cdot)$ is the logic function that relates independent detection information to decision on green extension time.

Additionally, the green duration in LHOVRA is bounded by the parameter of maximum green time i.e.

$$g_{i,j,u} \leq g_{i,u}^{max} \quad (11)$$

where $g_{i,u}^{max}$ refers to the maximum green time for signal group u at intersection i . The maximum green time for each signal group is a main decision variable in vehicle-actuated timing technique that highly influences on the efficiency of signal operations. In fact, maximum green time for a signal group is in practice set between the lower bound $g_{i,u,-}^{max}$ and the upper bound $g_{i,u,+}^{max}$. Such a constraint on the maximum green time can be analytically represented by:

$$g_{i,u,-}^{max} \leq g_{i,u}^{max} \leq g_{i,u,+}^{max}, \quad g_{i,u}^{max} \in \mathbb{N}. \quad (12)$$

In Jin and Ma (2014a), we proposed a coordination approach for LHOVRA and other group-based VA controls. Normally, a master intersection and the coordination direction should be specified in advance. When traffic lights run in the coordination mode, the beginning time of the first phase at one intersection should be properly set beforehand such that vehicles can pass through several intersections without stopping. This scenario is so-called "green wave". Offset value is used to represent the beginning of signal in relative to the master intersection. The constraint on offset δ_i is hence defined by

$$\delta_{i,-} \leq \delta_i \leq \delta_{i,+}, \quad \delta_i \in \mathbb{N}. \quad (13)$$

where $\delta_{i,+}$ and $\delta_{i,-}$ are the upper bound and lower bound of the offset value associated with intersection i respectively.

In signal coordination, a common cycle length is required to be defined in a coordination system such that "green wave" is able to be achieved over cycles. Let C , C_- and C_+ be a common cycle length, the lower bound of the common cycle length and the upper bound of the common cycle length, respectively. The constraints of a common cycle length are represented as follows

$$C_- \leq C \leq C_+, \quad C \in \mathbb{N}. \quad (14)$$

Since LHOVRA is vehicle actuated, individual cycle length at each intersection may have a small difference from the predefined cycle length C . When individual cycle length is smaller than C , it is called "early-return". When individual cycle length is larger, the last phase of the cycle is forced to be off at the time when the common cycle length is achieved. Such a scenarios is called "force-off".

Fig. 8. Layout and signal groups of the study traffic network.

In summary, as the LHOVRA control scheme has been implemented in the proposed software, the simulation-based stochastic optimization problem can be generally formulated by a multi-objective function

$$\min_{\theta} \mathbb{E}_{\mathbf{r}}[\mathbf{P}(\theta, \mathbf{x}, \mathbf{r})] \quad (15)$$

where $\mathbf{P}$ is a vector of performance indexes that depend on θ , $\mathbf{x}$ and $\mathbf{r}$; θ is a vector of parameters being optimized; $\mathbf{x}$ is a vector of input variables to the traffic model other than the control parameters; $\mathbf{r}$ is a set of random seeds for traffic simulator. Equation 9 to Equation 14 are considered as constraints for the optimization problem.

5.2. Experiment setup

The numerical study focus on a road network displayed in Fig. 8. It consists two neighboring intersections, Hornsgatan-Ringvägen and Hornsgatan-Rosenlundsgatan. In reality, traffic lights located at the two intersections are either operated by isolated group-based vehicle-actuated (GBVA) control or coordinated group-based fixed-time (GBFT) control. The first one is a vehicle-actuated control without coordination whereas the latter one is a simple fixed-time control with coordination.

Traffic simulation model is built in accordance with a blueprint of real signal plans provided by Stockholm Traffic Administration. In our study, all traffic simulation runs are performed for a 60-minute period, excluding a warm-up time of 15 minutes to avoid the unstable loading effects in the beginning. Parallel traffic simulation runs are implemented with the shared memory architecture on a computing node in a high-performance computing (HPC) environment. The computing power is defined by four hexa-core AMD Opteron 8425HE CPUs and 32GB RAM. Once the process of simulation-based optimization is completed, the best signal plan is further evaluated through 30 simulation runs carried out using different random seeds.

The first experiment is to examine the potential benefits brought by using the AGA algorithm. Later, two control approaches (the isolated GBVA and coordinated GBFT) after optimization are compared. Finally, the coordinated GBVA control is compared with the isolated GBVA and coordinated GBFT controls. Identical traffic demand pattern has been applied in the computational experiments. For the coordination cases, Hornsgatan-Ringvägen intersection (shown on the left) is set as the major intersection with priority in coordination. Table 1 presents the resulted optimal parameters for the tested three controllers when travel delay and fuel consumption are set as the objective function, respectively.

5.3. Performance of AGA

The AGA algorithm is compared with an ordinary GA (OGA) optimizer, whose operators have been carefully selected. Since the settings of control parameters for GA are problem-dependent, different sets of control parameters

Table 1

The optimal control parameters for the isolated GBVA, coordinated GBFT and coordinated GBVA controllers when travel delay and fuel consumption are the objective function, respectively.

Signal parameters	Isolated GBVA		Coordinated GBFT		Coordinated GBVA	
	Delay	Fuel	Delay	Fuel	Delay	Fuel
$G1$	16	18	16	18	18	17
$G2$	16	18	16	18	18	17
$G3$	16	18	16	18	18	17
$G4$	29	48	26	49	30	51
$G5$	29	48	26	49	30	51
$G6$	29	48	26	49	30	51
$G7$	29	48	26	49	30	51
$G8$	10	12	12	12	11	13
$G9$	10	12	12	12	11	13
$G10$	40	63	26	49	30	51
$G11$	40	63	26	49	30	51
$G12$	40	63	26	49	30	51
$G13$	40	63	26	49	30	51
$G14$	11	14	28	30	29	30
Offsets	–	–	17	18	16	16

Note: In the first column, $G1$ - $G14$ represent the (maximum) green times for the associated signal group in Fig. 8.

Fig. 9. Performance comparison between OGA and AGA for coordinated GBFT control across generations.

were examined and the well-established ones for the two optimizers are listed in Table 2. Specifically, the size of bit string depends on the the bounds of parameters according to Equation 1. Since the differences between upper bound and lower bound for control parameters are all smaller than 128, a value of 7 is chosen as the size of bit string for both optimizers. As introduced in subsubsection 3.1.2, both crossover and mutation probabilities are adaptive. The same ranges of the probabilities, suggested by McGinley et al. (2011), are set for the two optimizers for comparison purpose. The size of population in each generation is set to 30 for OGA, while a relatively small size of population (i.e., 6) is set for AGA. Moreover, the total number of simulation evaluations is identically set for both algorithms. The number of simulation evaluations is equal to the size of the population in each generation multiplied by the number of generation.

To provide insights into the performance of the algorithms, the profiles of best fitness values for each generation

Table 2

Summary of control parameters of GAs.

Control Parameters	Optimizers	
	Ordinary genetic algorithm (OGA)	Archived genetic algorithm (AGA)
Elite size	2	–
Archive size	–	100
Bit string size	7	7
Tournament pool size	4	4
Crossover probability	0.4-0.8	0.4-0.8
Mutation probability	0.0-0.5	0.0-0.5
Initial population size	30	100
Population size in each generation	30	6
Number of generation	200	1,000

Table 3

Evaluation results of optimal signal plans for coordinated GBFT control.

Objectives	Performance measures			
	Avg. Delay (sec/vehicle)	Avg. Fuel (g/km)	Avg. Total Emissions (g/km)	Avg. No. of Vehicles Caught in Dilemma Zone
Avg. Delay	25.09	87.94	4.59	158.60
Avg. Fuel	27.74	85.99	4.36	82.75

are presented for the case that coordinated GBFT control is applied in Fig. 9. Both travel delay and fuel consumption are, respectively, set as the optimization objective. The figure reveals that both optimization algorithms eventually succeed in finding the solution, assumed to be the near global optimum. Two typical stages can be observed for the search process using genetic algorithms. One stage is where the population is relatively further away from the final solution, and the other stage is where the population is close to the final solution.

In the figure, the resulted best performance measure by the AGA algorithm constantly reduces at the first stage. The example in Fig. 9 shows that AGA keeps reducing the best performance measures until approximately 750 evaluation functions when the average delay is the optimization objective. Whereas, the reduction speed of the best performance measures in each generation for OGA becomes much lower after 300 evaluation functions. The plot also reveals a similar finding if the average fuel consumption is the optimization objective. The above phenomenon is caused by the mechanism of the AGA algorithm. Since the ever found good solutions across the generations (the archive) are utilized for generating new population in the AGA algorithm, the movements towards better solutions are accomplished using potentially fewer search paths compared to OGA.

When the optimization process enters the second stage, most individuals in the population are close to the near global optimum. The population with good individuals resembles a good approximation to the final solution. The number of function evaluations, therefore simulation runs, can be reduced by focusing only on those points that are relatively close to the near global optimum and by including only them in the parent population. Such a strategy tends to fill the gaps to the near global optimum in new population by generating solutions around the near global optimum in the final stage. It is desirable to include only a handful of solutions rather than the entire population as the parent. This equips AGA with the ability to speed up the convergence rate with the aid of large-size archive and manipulation of micro-size elitism solutions. Fig. 9 shows the comparison result when the average delay is used as objective for coordinated GBFT control system. It implies that 1,050 evaluation functions are needed to locate the near global optimum for AGA, whereas OGA needs 1,800 evaluations to reach the solution.

Table 4
Evaluation results of optimal signal plans for isolated GBVA control.

Objectives	Performance measures			
	Avg. Delay (sec/vehicle)	Avg. Fuel (g/km)	Avg. Total Emissions (g/km)	Avg. No. of Vehicles Caught in Dilemma Zone
Avg. Delay	24.92	84.65	4.29	101.04
Avg. Fuel	26.01	81.23	3.90	64.22

Fig. 10. Comparison of the optimal results among different signal controllers (Objective functions are set to average travel delay on the left bar and fuel economy on the right bar, respectively).

5.4. Evaluation results for different control strategies

Table 3 summarizes performance measures for optimal signal plans when coordinated GBFT control is in operation. Likewise, Table 4 presents the corresponding performance evaluation of isolated GBVA control strategy. In these two tables, apparent trade-off effect can be observed between fuel consumption and average travel delay, and between emissions and travel delay. For example, a gain of 2.2% reduction in fuel economy and 5.0% decrease in total emission rates can be obtained by setting average fuel consumption as the objective compared to the case that average travel delay is minimized when coordinated GBFT control is deployed.

The difference on average delay between coordinated GBFT and isolated GBVA in terms of traffic mobility is not as significant as expected, probably due to the relatively simple and fixed demand pattern in the network. Nevertheless, a significant improvement on sustainability can be achieved when isolated GBVA is deployed. Specifically, it suggests an average more than 5% reduction in fuel economy, and total emissions when fuel consumption is regarded as the optimization objective. This result also indicates that such coordination strategy between the two intersection brings benefits on average travel delay, but sacrificing fuel and emission outputs.

Vehicle-actuated timing scheme contributes significantly to the findings above. If a driver decides to proceed when he/she is in dilemma zone (20 - 80 meters far from the stop line) and subsequently finds that he or she cannot pass through the intersection before traffic light indication changes to red. He or she must apply hard braking to stop the vehicle. Wu et al. (2010) revealed that such unnecessary hard braking is a rapid release of fuel consumption. Therefore, it is reasonable to understand that the more vehicles are caught in dilemma zone, the higher probability that more fuels are consumed. When vehicle-actuated timing is operated, the number of vehicles in dilemma zone is dramatically reduced (see Table 3 and Table 4) so that fuel consumption is reduced accordingly.

Another hypothesis in our study is that the coordinated group-based control has potential to bring benefits to traffic system. Hence, the coordinated GBVA is compared to real-world operations, isolated GBVA and coordinated GBFT. The optimization results shown in Fig. 10 are in line with our expectation. Independent of the selected optimization objective, coordinated GBVA control shows significantly better performance over all the other control strategies in

the computational experiments. For example, about 3.0% reduction of travel time can be achieved by coordination for GBVA controls. In addition, the optimal signal plan with coordination leads to around 100 gram fewer fuel consumptions per 100 kilometers per vehicle for GBVA when fuel economy is considered as the policy goal. Therefore, the coordinated GBVA is suggested as the best performed control strategy in the case study.

6. Conclusions

Road congestion is a worldwide problem causing not only travel delays but also reduced energy efficiency and additional pollutants. There are strong requirements in traffic engineering applications to develop optimization tools, facilitating traffic management decisions. In this regards, the paper outlines a general model-based optimization framework for traffic control problems. The computing framework is governed by an evolutionary optimizer, in which an archived GA algorithm is implemented and plays a central role for the optimization process. Because traffic system is complex and stochastic, simulation model is required in the framework to describe random traffic patterns, e.g., in terms of detailed vehicle trajectories. In addition, the software architecture provides flexibility to implement independent control schemes and define their communication with traffic simulator.

As repeated simulation runs are necessary, the computational process is expensive in nature. However, the archived GA provides merits concerning two aspects. One is the capacity to achieve faster convergence by using an external archive and micro elitism population manipulation. The other is the capability of exploring global search space. Furthermore, this study presents an implementation of the proposed framework in the application of traffic light control. The application integrates an open-source microscopic traffic simulator, SUMO, and a self-developed traffic light controller module.

A case study of optimal traffic signal control has been carried out to validate the algorithm performance, demonstrate the proposed optimization framework, and document the comparison results for different control strategies. The performance of the archived GA-based optimization method is investigated through experiments. While the previous study evaluated the approach using standard testing problems in computational intelligence, this study serves as a more concrete engineering application of the AGA algorithm. It highlights that the archive-based population maintenance accelerates the convergence towards the near global optimum and the computational efficiency is significantly enhanced when comparing to the conventional ordinary GA method.

Besides, the performance measures of different control strategies in mobility and sustainability are evaluated with the aid of the proposed stochastic optimization framework. Different control approaches are implemented in the software framework and compared in the case study. One main finding is that the coordinated GBVA control strategy shows better performance in the case study concerning not only average travel delay but also fuel efficiency and emissions.

Although an innovative GA approach has been applied in the optimization, the application still requires thousands of simulation runs. The parallelism of the simulation runs during the evolutionary optimization plays an important role to increase the computational efficiency. Further research has looked into the model-based evolutionary algorithms, which have potential to reduce the computational time in application. Meanwhile, it is planned to incorporate other nature-inspired optimization approaches in the current optimization software so that the tool can be used for a wider spectrum of traffic optimization problems. Finally, our latest development starts integrating low-fidelity traffic simulation models and dynamic assignment models in the software framework so that it can optimize traffic control measures for large-scale networks.

Acknowledgment

The implementation of the presented computing framework was partly supported by J. Gust. Richert Foundation (PIAH/12:24) and the Swedish Transport Administration (Dnr TRV 2013/16116). The corresponding author would like to acknowledge the TRENoP funding for supporting his international research visits during 2012 and 2014. The authors would like to thank three anonymous reviewers for their constructive comments from different aspects that help us improve the quality of the paper significantly.

References

- Barceló, J., Casas, J., 2005. Dynamic network simulation with aimsun. In: *Simulation approaches in transportation analysis*. Springer, pp. 57–98.
- Barceló, J., et al., 2010. *Fundamentals of traffic simulation*. Vol. 145. Springer.
- Bierlaire, M., 2015. Simulation and optimization: A short review. *Transportation Research Part C: Emerging Technologies* 55, 4–13.
- Burghout, W., 2004. Hybrid microscopic-mesoscopic traffic simulation. Ph.D. thesis, KTH.
- Celikoglu, H. B., Dell’Orco, M., 2007. Mesoscopic simulation of a dynamic link loading process. *Transportation Research Part C: Emerging Technologies* 15 (5), 329–344.
- Day, C. M., Smaglik, E. J., Bullock, D. M., Sturdevant, J. R., 2008. Quantitative evaluation of fully actuated versus nonactuated coordinated phases. *Transportation Research Record: Journal of the Transportation Research Board* 2080 (1), 8–21.
- Dell’Orco, M., Başkan, Ö., Marinelli, M., 2014. Artificial bee colony-based algorithm for optimising traffic signal timings. In: *Soft Computing in Industrial Applications*. Springer, pp. 327–337.
- Di Gangi, M., Cantarella, G. E., Di Pace, R., Memoli, S., 2016. Network traffic control based on a mesoscopic dynamic flow model. *Transportation Research Part C: Emerging Technologies* 66, 3–26.
- Fellendorf, M., Vortisch, P., 2010. Microscopic traffic flow simulator vissim. In: *Fundamentals of traffic simulation*. Springer, pp. 63–93.
- Jin, J., Ma, X., 2014a. Implementation and optimization of group-based signal control in traffic simulation. In: *Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 2517–2522.
- Jin, J., Ma, X., 2014b. Stochastic optimization of signal plan and coordination using parallelized traffic simulation. In: *Proceedings of the Second SUMO Conference: Modeling Mobility with Open Data*. DLR German Aerospace Centre, Berlin-Adlershof, Germany, pp. 115–126.
- Khondaker, B., Kattan, L., 2015. Variable speed limit: A microscopic analysis in a connected vehicle environment. *Transportation Research Part C: Emerging Technologies* 58, 146–159.
- Krajzewicz, D., Erdmann, J., Behrisch, M., Bieker, L., 2012. Recent development and applications of SUMO—simulation of urban mobility. *International Journal On Advances in Systems and Measurements* 5 (3 and 4), 128–138.
- Krauss, S., Wagner, P., Gawron, C., 1997. Metastable states in a microscopic model of traffic flow. *Physical Review E* 55 (5), 5597.
- Kronborg, P., Davidsson, F., 1993. MOVA and LHOVRA: traffic signal control for isolated intersections. *Traffic Engineering and Control* 34 (4), 195–200.
- Liu, H., Zhang, L., Sun, D., Wang, D., 2015. Optimize the settings of variable speed limit system to improve the performance of freeway traffic. *IEEE Transactions on Intelligent Transportation Systems* 16 (6), 3249–3257.
- Loh, W.-L., et al., 1996. On Latin hypercube sampling. *The annals of statistics* 24 (5), 2058–2080.
- Ma, X., Huang, Z., Koutsopoulos, H., 2014a. Integrated traffic and emission simulation: a model calibration approach using aggregate information. *Environmental Modeling & Assessment* 19, 1–12.
- Ma, X., Jin, J., Lei, W., 2014b. Multi-criteria analysis of optimal signal plans using microscopic traffic models. *Transportation Research Part D: Transport and Environment* 32, 1–14.
- McGinley, B., Maher, J., O’Riordan, C., Morgan, F., 2011. Maintaining healthy population diversity using adaptive crossover, mutation, and selection. *IEEE Transactions on Evolutionary Computation* 15 (5), 692–714.
- Osorio, C., Bierlaire, M., 2013. A simulation-based optimization framework for urban transportation problems. *Operations Research* 61 (6), 1333–1345.
- Papageorgiou, M., Diakaki, C., Dinopoulou, V., Kotsialos, A., Wang, Y., 2003. Review of road traffic control strategies. *Proceedings of the IEEE* 91 (12), 2043–2067.
- Park, B. B., Lee, J., 2009. Optimization of coordinated-actuated traffic signal system. *Transportation Research Record: Journal of the Transportation Research Board* 2128 (1), 76–85.
- Sánchez-Medina, J. J., Galán-Moreno, M. J., Rubio-Royo, E., 2010. Traffic signal optimization in "La Almozara" district in saragossa under congestion conditions, using genetic algorithms, traffic microsimulation, and cluster computing. *IEEE Transactions on Intelligent Transportation Systems* 11 (1), 132–141.
- Stevanovic, A., Stevanovic, J., So, J., Ostojic, M., 2015. Multi-criteria optimization of traffic signals: Mobility, safety, and environment. *Transportation Research Part C: Emerging Technologies* 55, 46–68.
- Teklu, F., Sumalee, A., Watling, D., 2007. A genetic algorithm approach for optimizing traffic control signals considering routing. *Computer-Aided Civil and Infrastructure Engineering* 22 (1), 31–43.
- Thierens, D., Goldberg, D., 1994. Convergence models of genetic algorithm selection schemes. In: Davidor, Y., Schwefel, H., Männer, R. (Eds.), *Parallel Problem Solving from Nature - PPSN III, Lecture Notes in Computer Science*. Vol. 866. Springer.
- Tiwari, S., Fadel, G., Deb, K., 2011. AMGA2: improving the performance of the archive-based micro-genetic algorithm for multi-objective optimization. *Engineering Optimization* 43 (4), 377–401.
- Ukkusuri, S. V., Ramadurai, G., Patil, G., 2010. A robust transportation signal control problem accounting for traffic dynamics. *Computers & Operations Research* 37 (5), 869–879.
- Wu, G., Boriboonsomsin, K., Zhang, W.-B., Li, M., Barth, M., 2010. Energy and emission benefit comparison of stationary and in-vehicle advanced driving alert systems. *Transportation Research Record: Journal of the Transportation Research Board* 2189 (1), 98–106.