This is the published version of a paper published in .

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

Optimizing local and global objectives for sustainable mobility in urban areas
*Journal of Urban Mobility*, 2: 100012
https://doi.org/10.1016/j.urbmob.2021.100012

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-307177
Optimizing local and global objectives for sustainable mobility in urban areas

Amita Singh a, *, Jannicke Baalsrud Hauge a, Magnus Wiktorsson a, Utkarsh Upadhyay b

a KTH Royal Institute of Technology, Kvarnb ergagatan 12, S ödert lje 151 36, Sweden
b Max Planck Institute for Software Systems, Paul Ehrlich Str. 26, 67663 - Kaiserslautern, Germany.

A R T I C L E   I N F O

Keywords:
Stakeholder participation
Sustainable urban mobility
Multi-objective optimization
Simulation-based optimization

A B S T R A C T

Cities are growing and sustainable urban mobility planning (SUMP) is gaining in importance with it. The problems in the domain often involve multiple stakeholders with conflicting or competing objectives. The stakeholders and objectives can be local to certain neighborhoods or apply to the global city-wide scale. We present a methodology to address such problems with the help of modern simulators and multi-objective evolutionary algorithms. The methodology brings all stakeholders to the table and presents to them a near optimal set of alternatives to choose from. As an example, we consider the problem of minimizing vehicular noise in a particular neighborhood while also minimizing city-wide emission for heavy vehicles. We describe the requirements and capabilities of the simulator and the optimization algorithm in detail and present a methodology to model both local (noise reduction) and global (emissions) objectives simultaneously. We apply our methodology on two large city scale case studies and present our findings.

1. Introduction

Growing size of cities and increasing population mobility has led to a surge in the number of vehicles on roads (Berry, 2008; Eurostat, 2019; Lerner, van Franã et al., 2012). The growing motorisation has led to increase in traffic congestion, noise, carbon emissions and concerns of road safety resulting in social, environmental and economic costs (Okraszewska et al., 2018; van Wee & Ettema, 2016). These problems are broadly classified under the umbrella term sustainable urban mobility planning (SUMP).

A key complexity in these problems is number of stakeholders involved. The stakeholders often have different point of views of the problem. For example, the local/community representatives may be biased towards improving their neighborhoods (Liu, Liao, & Mei, 2018; Xu & Lin, 2020) while the policy makers usually have a mandate coming from the state or national level. The business representatives may be focused primarily on the efficiency of the logistics while environmental activists may give more weight to the sustainable solutions. A typical example of such competing and conflicting objectives is the desire to reduce the noise levels in a particular neighborhood while also minimizing the city-wide emissions.

In literature, participatory modelling has majorly been used as reflexive, descriptive or normative study (Andersson, Olsson, Arheimer, & Jonsson, 2008; Hare, Letcher, & Jakeman, 2003; Jones et al., 2009; Malekpour, de Haan, & Brown, 2013). However, with the increase in computational power available today, analytical studies to support the decision-making have come to the forefront of participatory modelling (Grogan, 2021; Middya, Roy, Dutta, & Das, 2020). For complete literature in the field of simulation-based participatory modelling, the reader is referred to Singh, Baalsrud Hauge, & Wiktorsson (2021).

In this paper, we present an approach combining (i) simulation and (ii) Multi-Objective Evolutionary Algorithms (MOEA). The objective of the combined approach is to support the stakeholders develop a common understand and agree upon a compromise. We believe that this methodology will lend itself readily to solving other multi-stakeholder problems in the domain of urban mobility.

A primary component of the solution we propose is a simulator. Changing transport and mobility behaviour in the real world not only involves significant monetary costs and is time-consuming, but usually also has negative consequences for citizens when building or testing the new scenarios. For example, the new scenarios created might not be the most cost effective route, in terms of time and money, for citizens living in the area as it disrupts the regular optimized traffic flow (Okraszewska et al., 2018; Péres, Ruiz, Nesmachnow, & Olivera, 2018). This is where the ability to simulate traffic on a large scale with the help of traffic simulators (Behrisch, Bieker, Erdmann, & Krajzewicz, 2011; Cameron & Duncan, 1996; Holm, Tomich, Sloboden, Lowrance et al., 2007; Milo- jicic, 2018; Santana, Lago, Kon, & Milojevic, 2017; Simulator, 2005) can aid the process. Microscopic (Li, Yu, Tao, & Chen, 2013; Sánchez, Galán, & Rubio, 2008; Zhou & Cai, 2014) and macroscopic (McCreas &

* Corresponding author.
E-mail address: amitasa@kth.se (A. Singh).

https://doi.org/10.1016/j.urbmob.2021.100012
Received 5 March 2021; Received in revised form 8 December 2021; Accepted 14 December 2021
2667-0892/© 2021 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)
Moutari, 2010) representation of traffic views have been done in many simulation studies. It has become increasingly clear that microscopic simulators offer superior fidelity than macroscopic simulators, esp. with a large number of intersections and vehicles and when the effects of having multiple vehicles is either super-linear (e.g., increase in waiting time due to congestion) or sub-linear (e.g., amount of noise produced by multiple vehicles crossing a road simultaneously) (Ratrout & Rahman, 2009). The microscopic simulators can model a wider variety of scenarios and emergent phenomenon than a macroscopic one and, hence, we present our methodology for such a microscopic simulator.

Simulation is chosen over analytical solutions because even a simple act like blocking of roads can have second order effects which can make analytical solutions inaccurate by violating the assumptions they make. For example, while an analytical solution may assume that the vehicles will choose the shortest path for each vehicle, with the help of a simulator, it is possible to capture the evolving nature of vehicles re-routing as the traffic on the roads dynamically changes.

The other component of our solution is Multi-Objective Evolutionary Algorithm for optimization and for finding a landscape of solutions. Introduction of multiple objectives from different stakeholders means that there is likely no single solution that is optimal for everyone. Hence, the optimization algorithm has to find a set of solutions (Abraham & Jain, 2005). This set (called the Pareto optimal front) will allow the stakeholders to come together, see the problem from the point of view of others, and help them agree on a compromise which serves all of them well. Over the past few years, efforts have been made in combining microscopic traffic simulation with evolutionary algorithms to optimize for vehicular emissions together with traditional costs metrics (De Coensel, Can, Degraeuwe, De Vlieger, & Botteldooren, 2012; García-Nieto, Olivera, & Alba, 2013; Olivera, García-Nieto, & Alba, 2015; Pères et al., 2018; Wu, Deng, Du, & Ma, 2014). However, the use of evolutionary algorithms in large and heterogeneous case studies is still an open issue (Kutz, 2004). In this work, we show that with judicious use of parallelization, we can perform this optimization at a city scale.

In a nutshell, the primary contributions of our work are:

• Presenting a methodology for solving complex problems in the emergent domain of sustainable urban mobility that go beyond optimising for the traditional cost metrics.

• We show how multiple stakeholders can be brought to the table and be presented with several Pareto optimal solutions to choose between. Importantly, the stakeholders can be global or local but their constraints and desires can be adequately represented in a uniform manner.

• Lastly, we present two case studies applying our methodology to solving a multi-stakeholder problem, viz. reducing noise in certain neighborhoods while also minimizing the emission on the scale of an entire city.

The article is organized as follows: Section 2 presents the system definitions and mathematical formulation of the problem. Section 3 has an overview of the research methodology followed in this work. Later in Section 4, case studies are described followed by the discussion of results. Section 5 concludes the article by presenting the possible future works while briefly touching on the limitations of the proposed approach.

2. System description

2.1. Definitions

An urban traffic network is composed of intersections and roads which connect them. These roads have multiple physical (e.g., number of lanes, paved/concrete, etc.) and logical attributes (e.g., one-way traffic, heavy vehicles allowed, speed limits, etc.) associated with them. A network is represented as a directed graph \( G = (\mathcal{J}, \mathcal{E}, A, \mathcal{A}, o) \), where the set \( \mathcal{J} \) contains the junctions, the set \( \mathcal{E} \) contains the roads and \( \mathcal{A} \) is the set of all attributes that the edges can have and \( \mathcal{A} \) is the set of values that the attributes can take. In addition, there is a mapping \( g : \mathcal{E} \times A \rightarrow \mathcal{A} \), which lists the attribute values for all edges. These logical attributes of the roads are under the control of the city and roads are often reclassified temporarily, e.g., disallowing heavy vehicles to allow for constructions or blocking complete traffic for public manifestations. Such a reclassification of a subset \( \mathcal{S} \subseteq \mathcal{E} \) of edges is represented as using a partial map \( x : \mathcal{C} \times A \rightarrow \mathcal{A} \). The resulting reclassified network is represented using the notation \( \mathcal{G}(x) \), where the roads in \( \mathcal{C} \) have some of their attributes changed, but other roads retain their original attributes.

Similar to roads, the junctions \( J \) also have attributes (as shown in Fig. 1). One of the most familiar to the readers may be Traffic Light Control phase vectors. While they can easily be considered as an extension of this work, it is omitted to keep our exposition simple.

The traffic on the roads is composed of individual vehicles moving along the network. These vehicles also have physical (e.g., weight, engine type, position, etc.) and logical (e.g., commercial vehicle classification, starting time, etc.) attributes associated with them. The set of vehicles in the network is represented as \( \mathcal{V} = (V, B, \mathcal{B}, b) \) where \( \mathcal{V} \) is the set of available vehicles, \( B \) is the set of attributes of the vehicles, and \( \mathcal{B} \) is the set of values that the attributes can take. Additionally, \( b : \mathcal{V} \times B \rightarrow \mathcal{B} \) is the mapping between vehicles and their attributes. Some attributes of the vehicles (e.g., departure time) are under the control of the individual inhabitants of the city or the commercial transport operators in the city.

The route a vehicle takes through the network is dictated by a complex interaction between objectives of each vehicle and constraints imposed by the city on the roads. We make an assumption that the objective of all vehicles is to minimise the time it takes to travel from the source to the destination and it may re-route if the traffic conditions on their path deteriorate due to congestion. The constraints are often imposed through the logical attributes set on the roads, disallowing flow of traffic in one direction, or limiting speed on others, etc.

The traffic as it drives through the city, produces emissions and noise, and suffers delays due to congestion in the city (or due to certain routes being unavailable due to restrictions imposed by the city on the roads). The emission cost of the traffic is defined as \( f : \mathcal{G} \rightarrow \mathcal{R} \). An example of such a cost is the amount of emissions \( \text{NO}_x \) produced by the traffic on the network \( \mathcal{G} \). The local cost of the traffic observed at an edge \( e \in \mathcal{E} \) in the city is defined with \( h : \mathcal{G} \times \mathcal{E} \rightarrow \mathcal{R} \), for example, the amount of noise produced on a particular road. Finally, using \( g : \mathcal{G} \rightarrow \mathcal{R} \), we denoted the cost of the attributes of the roads in the network. For example, the cost of the attributes of the roads in the network may be the number of vehicles which are able to reach their destination during the course of the simulation. This simple cost is a good proxy of the amount of congestion on the roads of the city. Other measures which capture different aspects of the same metric may be the total waiting time experienced by the cars on the road or the overall fuel consumed. For our example problem, we focus only on the binary property of roads, which is whether trucks are allowed or not allowed on a particular road. We do not consider other properties of the network or roads, for example, taking into consideration the maximum speed allowed on a particular road, etc.

In this section, we have presented a general methodology which can both (i) capture a wide variety of problems in the SUMP domain, and (ii) be easily translated to a program which can run on a simulator. We will describe the details of the simulation tool in later sections. The model can capture the concerns of local as well as global stakeholders, can model hard constraints that the stakeholders impose, and allow the stakeholders to manipulate the logical as well as physical attributes of the network to run what-if scenarios.

2.2. Mathematical formulation

The task of determining the attributes of roads in \( \mathcal{G} \) now can be framed as a multi-objective optimisation problem as:

\[
\min \mathcal{f}(x) = \{ f(\mathcal{G}(x)), g(\mathcal{G}(x)) \} \cup \{ h(\mathcal{G}(x), e) \} e \in \mathcal{C} \subseteq \mathcal{E}
\]
where:

- \( x \) is the state of the traffic network which we want to evaluate. It encodes, for example, which roads allow trucks and which do not.
- \( G(x) \) is the traffic network which respects the state \( x \).
- \( f(G(x)) \) is the emission cost of one day of traffic in network \( G(x) \). It is clearly a function of the state of the network, e.g., disallowing trucks on certain paths may increase the emission by making the trucks take a longer path.
- \( g(G(x)) \) is the global cost of one day of traffic in the network \( G(x) \). This maps to the traditional costs which are often the objective of dynamic vehicle routing problems (VRP). An example is the total number of vehicles reaching their destination (as in this article) or total waiting time: Changing which paths disallow trucks results in different congestion patterns which may limit the number of vehicles reaching their destination.
- \( C' \) is the subset of roads that the stakeholders are particularly interested in. For example, a hospital or a school may be interested in the noise being produced on the road(s) on which it is situated.
- \( h(G(x)) \) is the local cost of one day of traffic in the network \( G(x) \) affecting the stakeholders. In this article, we have considered the total noise produced on a particular road, i.e., \( \int_0^1 \) harmonoise \( dt \) where harmonoise is defined in Harmonoise (SUMO documentation) (2021).

At this point, we can describe the objective functions and the setup of a typical SUMP problem, which includes the description of the constraints. In the later sections, we will describe how we can let an optimization algorithm modulate the setup and pass input to a simulator and how we can process the output of the simulator to calculate our objective functions. However, we would like to stress at this point that the simulator would make some additional assumptions about the operation of the vehicles on the network, i.e., which route they take after starting. There are certain rules that the simulator imposes on the traffic by the virtue of how the simulator is built. This includes the intrinsic properties of the simulator itself, for example, the model of how vehicles flow in the simulation. In particular, we enforce the following assumptions on the simulator:

- Vehicles always obey traffic rules such as speed limits and avoid collisions.
- Vehicles take the path that takes them the shortest amount of time to reach their destination.
- All vehicles recalculate their chosen route after a period of \( \pi \) seconds.

The last assumption approximates the realities of the present world where periodically the information on navigation apps updates to reflect the true congestion on the route. Incidentally, this is what also allows our vehicles to re-route through the network in case their initially desired path is no longer available, e.g., for a truck when a road is re-classified to disallow heavy vehicles.

With these constraints on the routing of the vehicles, a fully specified multi-objective optimization problem is obtained. However, since the optimization problem involves multiple objectives which may be mutually incompatible, it is unlikely that there will be a single point which will minimize all costs simultaneously.

### 2.3. Pareto optimal front

Before we can determine what we seek as a result of the optimization procedure, we need a few formal definitions.

**Definition 1 (Dominance operator \( \prec \))** Define a point \( x \in \mathbb{R}^n \) as being dominated by point \( y \in \mathbb{R}^n \) if

1. \( \forall i \in \{1, \ldots, n\}, y_i \leq x_i \) and,
2. \( \exists i \in \{1, \ldots, n\}, y_i < x_i \) i.e.,

\( x \) is element-wise less than or equal to \( y \), but at least at one element, strict inequality holds. If \( x \) is dominated by \( y \), we refer to it as \( x \prec y \).

With this definition of dominance, points are defined in the control space of an optimization problem which are dominant in terms of their fitness, or costs.

**Definition 2 (Dominant point \( \prec y \))** Given a vector function \( \tilde{f}(x) : D \to \mathbb{R}^n \), and two points \( x \in D \) and \( y \in D \), we say \( x \prec y \) if and only if \( \tilde{f}(x) < \tilde{f}(y) \).

**Definition 3 (Pareto optimal point).** For a given vector valued function \( f : D \to \mathbb{R}^n \), a point \( x \in D \) is called in its domain Pareto optimal if and only if \( \forall y \in D. \prec(y \prec f(x)) \).

---

**Fig. 1.** A sample of data used and its corresponding structure.
A set of finite members is referred to in the domain $D$ as a population $\mathcal{X} = \{x_i | x_i \in D\}$. With Definition 3, the output of our optimization process as a set of points $\mathcal{X}' \subset D$ which are Pareto-optimal according to the relationship $\preceq$ is defined where

$$\bar{f}(x) = \{f(G(x)), g(G(x))\} \oplus \{h(G(x), c) | c \in C\} \tag{2}$$

is a vectorized version of all our objective functions from Eq. (1).

The set of points obtained as a result of the optimization is called the Pareto optimal front. These points represent the characteristics of different network configurations. The Pareto optimal points give the stakeholders the possibility to understand what each network configuration i.e. blocking and unblocking of roads would mean in terms of costs, for example, waiting times and fuel consumption together with CO₂, NOₓ, PM and noise emissions. For additional reading on multi-objective optimization and the Pareto optimal front, the reader is referred to Abraham & Jain (2005).

We highlight here that such a Pareto front is very useful for discussions among stakeholders. The city administration and its policy makers may look for environmentally best solutions, operations and logistics management from companies require most cost efficient solutions, and local stakeholders may want the solution which is best for their particular neighbourhoods. However, their best possible compromise will necessarily be a solution on the Pareto optimal front. These mutually competing demands require different stakeholders to come to a discussion and decide the best plausible solution(s). We will describe its use with concrete results from our experiments in Section 4.4.

In the ensuing Sections, we describe how to calculate the function $\bar{f}(x)$, i.e., $f(G(x)), g(G(x))$, and $\{h(G(x), c) | c \in C' \subset E\}$, using a simulator (SUMO) for a given choice of $x$, and how we can then iteratively uncover the Pareto optimal front $\mathcal{X}'$.

3. Research methodology

In the previous section, we described how to formulate a general SUMP problem, along with a particular problem of noise minimization by allowing/disallowing trucks on roads. In this section, we describe how we represent the control variables, evaluate the objective functions, and perform the optimization.

3.1. Control variables

Control variable are the partial reclassification of a (fixed) subset of roads $C \subseteq E$ on the network, denoted by $x : C \times A \rightarrow \alpha$. Changing $x$ allows us to simulate various what-if scenarios: in one mapping, we may disallow trucks on certain subset of roads (as done in this article), or increase the speed limit on certain roads, while in another, we may allow the trucks, or decrease the speed limits. However, representing $x$ as a function, while flexible, does not allow us to directly use optimization methods to uncover it. In the most general setting, the mapping can be embedded as a vector with binary, categorical, ordinal and floating point numbers. The embedding scheme will differ from problem to problem, and will play a crucial role in the interpaly between the optimization algorithm and the simulator. The control variables in the example problem we have chosen are intentionally limited to allowing or disallowing trucks on roads. Hence, the mapping $x$ is represented as a binary vector, where each bit corresponds to an edge in the subset $C$. Trucks are allowed to travel on the edge in the simulation only if the bit corresponding to it is 0. The task of the optimizer is to uncover various values of $x$ which result in solutions which lie on the Pareto optimal front. In the next section, we describe how $x$ is conveyed to the simulator and is optimized using evolutionary algorithms.

3.2. Function simulator: SUMO

Our SUMP problem is to minimize pollution emissions along with noise in a neighbourhood via multi-objective optimization. We have seen in earlier sections how to formulate this problem mathematically. In this section, we will describe the simulator that we used, its structure, capabilities, and limitations.

The simulator should have the capability to model both traditional costs metrics and environmental sustainability aspects on a city-wide scale. Keeping these criteria in mind, SUMO emerged as a good fit for the choice of simulator to model common SUMP problems. In particular, it can completely model the example problem at hand.

Simulation of Urban Mobility (SUMO) (Behrisch et al., 2011) is an open source and highly portable software which allows different modes of traffic simulation including pedestrians and a large set of tools for creating scenarios. It integrates well with other software and also has a programmable API (Krajzewicz et al., 2005). SUMO accepts as input the parameters to run a simulation in form of configuration files, runs the simulation as per the specification, and then writes the requested output files back to disk, as shown in Fig. 2. Helpfully, SUMO supports describing the road network $\mathcal{G}$ (including full attribute mapping $a : \mathcal{E} \times \Delta \rightarrow \alpha$) in the OpenStreetMap (Haklay & Weber, 2008) (OSM) format.

In simulation each vehicle $v \in W$ moves individually through the network. The characteristics $b(v) \in \mathcal{B}$ of each vehicle $v$ can be customized individually as well. To simplify this process, each vehicle can be represented with the help of existing vehicle types (e.g., car, truck, etc.) and further can be customized with respect to vehicle configuration, e.g., vehicle speed, acceleration, deceleration, etc. The core logical attributes which determine the path of a vehicle $v$ are referred to as a trip which consists of a start edge, an end edge and a starting time. We will describe how we arrive at these logical attributes for the vehicles while describing our exact experimental setup below. Such a trip can be transformed into a route, which is a start edge, end edge and a starting time together with all the edges $e \in E$ that a vehicle traverse from start- ing point to end point. This conversion can be done statically once at the beginning of the simulation (using tools shipped with SUMO, e.g., duarouter), or dynamically while SUMO is running by periodically allowing vehicles to modify their routes given the existing congestion on the network. For a comprehensive review of the routine algorithms, the reader is referred to Golden, Raghavan, & Wasil (2008). As mentioned before in Section 2.2, vehicles in the simulation follow dynamic routing and choose the route which minimizes the time to destination from the current location of the vehicle and the vehicles recalculate the routes with a period of $x$ seconds to account for changing congestion conditions.

An additional file (.additional.xml) can be included in the configuration to obtain better visualizations, to provide additional vehicle attributes and for SUMO to output additional data for processing. SUMO can produce a variety of output from the simulation it runs. This work concentrates on the following outputs:

1. summaryData output to calculate the overall emission cost $f(G(x))$
2. tripsData to determine how far each vehicle $v \in W$ progressed toward its respective destination, and, hence, calculate $g(G(x))$
3. edgeData output for all edges $e \in C'$ to be able to calculate the edge specific costs $h(G(x), c)$

It is worth mentioning that SUMO can produce much more detailed output for any simulation in form of full emissions output and full output, from which these objectives could be extracted. For even small simulation runs, the size of the full emissions output can get prohibitively large (e.g., 2GB for a 1 day of simulated time) such that more CPU compute is spent processing the file instead of actually running the simulation. Hence, we opted for the alternate summary output

---

1 $\oplus$ acts as the concatenation operator.
2 See Pères et al. (2018) for examples of how real values can be embedded, e.g., for controlling traffic light phases and offsets.
3 [https://sumo.dlr.de/docs/Simulation/Output.html](https://sumo.dlr.de/docs/Simulation/Output.html)
formats for our experiments. However, there are methods available e.g., see TraCI (Krajzewicz, Erdmann, Behrisch, & Bieker, 2012) which can be used to extract data as the simulation is running via sockets instead of reading it via disk. Hence, SUMO can act as an effective evaluator for almost any arbitrary objective function. Investigating which class of objective functions SUMO cannot be used to evaluate is an interesting area of future work which could be pursued to improve SUMO itself. An example of such an objective is light pollution.

Following the structure of network adopted by SUMO, the blocking and non-blocking of roads can easily be encoded by means of a vector of binary values where each binary value represent the state of a particular edge from the provided list of edges where “0” means trucks are allowed and “1” means trucks are not allowed on a particular edge as shown in Fig. 2.

3.3. Evolutionary algorithms

Evolutionary algorithms (EAs) (Abraham & Jain, 2005) start with an initial population \( X_p = \{ x_i : x_i \in D \} \) of feasible assignments to the control variables. This initial population could be randomly generated or hand-crafted, or a mix thereof. For the problem at hand, for example, each member of the populate \( x \) is encoded as a bit-string with each position \( x[i] \) encoding the blocked or unblocked nature of a particular lane/edge \( e_i \in C \) (the mapping of the index \( i \) to the edge \( e_i \) can be arbitrary). Then the domain of the control variables is \( D = \{0, 1\}^n \) where \( n = |C| \) is the number of bits in each \( x \).

Starting with this initial assignment \( X_p \), the algorithms proceed in an iterative manner with each step producing a set of points called a generation. Given a population \( X_\iota \) at generation \( \iota \) of assignments of control variables, first the fitness of each member \( x \) in \( X_\iota \) is \( f(x) \) is calculated. For the example problem we have chosen, the fitness depends on the emissions and noise levels on certain roads along with how diverse the solutions are in the solution space. This fitness influences the likelihood of that member being selected for producing an offspring, or being included as-is in the next generation. The exact mechanics of the mutation and cross-over operations depend on the flavor of EA being used. Then the algorithm produces the next generation of assignments \( X_{\iota+1} \). Additional considerations are usually taken into account to ensure a level of diversity among the members of the population to both explore the control variable space \( D \) better (e.g., random mutations), as well as to provide as precise description of the Pareto optimal front \( X^* \) as possible (e.g., crowding distance (Deb, Pratap, Agarwal, & Meyarivan, 2002)). After a fixed number of generations \( G \), a set of the fittest members over the entire history are produced as the output. They represent an approximation to the Pareto optimal front of solutions for our stakeholders.

The simulator SUMO is treated as a black box in this setting: the EA is completely unaware of how a control variable \( x \) is mapped to its fitness \( f(x) \). This makes the genetic algorithms well suited for our problem setting: the EA is oblivious to the choice of the evaluator and simulators allows us to capture the nuances of the changes with more granularity than an analytical model while still being computationally tractable as compared to deploying the solution in the real world. The interplay between the optimization algorithm and the simulator, hence, is mediated by the embedding of the control variables \( x \). The optimization algorithm uses the runs so far to determine which \( x \) to explore next and the simulator returns the value of the objective functions at those \( x \). In doing so, the onus lies on the optimization algorithm to ensure that (i) the points yield diverse values for the objective functions, and, (ii) improve the objective functions (iteratively).

Finally, we choose NSGA-II (Deb et al., 2002) as it is a general purpose optimization algorithm as it produces high quality Pareto fronts for a variety of difficult optimization problem. It remains an interesting area for future work to evaluate how the choice of the optimization algorithm will affect the quality and tractability of the problem.

4. Case studies

To experimentally evaluate our method, we run large scale realistic experiments for two cities A and B. This section will describe the experimental setup, followed by details on the two case studies performed.

4.1. Experimental setup

We first fix the maximum number of vehicles running in the cities by using daily averages number of vehicles (Eurostat, 2019; Turku Statistical Data, 2019). We configure each simulation to run for 86,400 s (1 day). For each vehicle \( v \in W \), the following attributes \( h : W \times B \rightarrow \mathbb{R} \) are assigned (all dependent on a random seed):

1. starting position as an edge chosen at random from \( E \)
2. ending position as another edge chosen at random from \( E \)
3. starting time as a random second, i.e., \( t_{start} \in [0, 86,400] \)
4. vehicle type \((vtype)\) as one of \{car, truck\} (equally probable)

A script called randomTrips.py which comes bundled with SUMO is used to generate these easily. Three different set of these attributes \( b_1, b_2, b_3 \) by using three different seeds as input to the script are generated. The \( f(x) \) value calculated for each \( x \) are the average of the values generated for \( b_1, b_2, b_3 \). This prevents the optimization algorithm from over-fitting for one particular distribution of starting and ending points. In each city, we select 21 salient roads for \( C \) which saw significant traffic as the edges which we will reclassify to allow or disallow vehicles of type truck. We also selected one central road \( C' = \{c\} \) for each case study for minimizing the total noise. To measure the aggregate noise level on the street, we calculate the noise level at each second interval on the road in dB (note that noise does not scale linearly with the number of vehicles driving on the road, but SUMO can make a good approximation using internal algorithms Krajzewicz, Erdmann, Behrisch, & Bieker, 2012), and aggregate it to come up with an average measure \( h(x, c) = \text{AvgNoise}(G(x), c) \). The other metrics we optimized for were \( g(x) = \text{NumVehicles}(G(x)) \) and \( f(x) = \text{CO2Emitted}(G(x)) \), resulting in the final objective function:

\[
\begin{align*}
\mathbf{f}(x) &= [\text{NumVehicles}(G(x)), \text{CO2Emitted}(G(x)), \text{AvgNoise}(G(x), c)]
\end{align*}
\]
so that maximizing \( f(x) \) maximizes the number of vehicles reaching destination (i.e., reduces congestion), minimizes \( CO_2 \), and noise produced on the edge \( e \).

We evaluated a generation \( X_g \) of assignments in parallel by using a Grid-cluster and a dedicated machine for pre and post-processing data. A single process spawns \(|X_g| \) processes, and sends each process one member \( x_i \in X_g \) to evaluate the fitness of, i.e., \( f(x) \). Each process starts three instances of the simulator SUMO on a Grid cluster, and gives it as input files generated for the different seeds, i.e., for \( x_i \) and \( b_1 \), \( b_2 \), \( b_3 \). After the SUMO simulator completes, the same process collect the output written to disk, process it, and return \( f(x_i) \) to the main process. A brief illustration of the setup is shown in Fig. 3.

The implementation of the method is done in Python using the DEAP library (De Rainville, Fortin, Gardner, Parizeau, & Gagné, 2012) and the variant of the Evolutionary Algorithm we employed was Non-Sorting Genetic Algorithm (NSGA-II) (Deb & Jain, 2013). We fix the population size to be 40 and we run the experiment for a total of 40 generations, i.e., \( \forall g \in [0, \ldots, 40] \). \(|X_g| = 40\) with crossover probability \( p_c = 0.9 \) and mutation probability of \( p_m = 0.1 \). We experimented with different values of population and probabilities of crossover and mutation and it was found that with the above mentioned values the optimization was the best trade-off in terms of time and computational power. Note that this requires at most \( 40 \times 40 = 1600 \) runs of the simulator as opposed to \( 2^{21} \approx 2 \times 10^6 \) executions which would be needed for a brute force search of the domain of the control variables.

The experiments were run on two cities with different parameters in terms of their size and characteristics. Details of these two cities are given as City A and City B are given below.

4.2. City A case study

City A is a medium-sized town in northern Europe with a population of approximately 70,000 inhabitants. The city is inherently an industrialized area with major industries from different sectors. See Fig. 4 to see which roads in \( C \), and \( C' \).

The results of the optimization are shown in Figs. 5 and 6. The evaluation of one generation for City A took \(~60\) min. Fig. 5 shows that as the number of vehicles reaching the destination increases from 1760 to 2900, which indicates that the congestion on the roads overall falls, the noise level on the considered road also increases, albeit non-linearly, from 20.1 dB to 25.3 dB. Fig. 6 shows that while the noise falls from 25.3 dB to 20.1 dB, the total \( CO_2 \) emitted increases from \( 0.91 \times 10^6 \) kg to \( 1.16 \times 10^6 \) kg.

4.3. City B case study

In comparison City B is a large city in northern Europe with a population of 200,000 inhabitants and an area of 250 sq. km. The city is characterized by a major harbour and serves as input to influx of both import/export and tourists from neighbouring countries. The city is currently undergoing a renovation plan and there are new routes being planned for the city. The optimization model proposed in this work, thus, helps in understanding the different scenarios for to-be built roads and provides stakeholders with a tool to discuss the trade-offs associated with them (Fig. 7).

The results of the optimization are shown in Figs. 8 and 9. The eval-

Algorithm 1 Finds non-dominated points \( X^* \) for a given multi-objective fitness function \( f(x) \).

1: Input: Population size \( P \), No. of Generations \( G \), Prob. of mutation \( p_m \), Prob. of crossover \( p_c \)
2: Requirement: \( P \) is divisible by 4
3: Output: Non-dominated Points
4: \( X_0 \leftarrow \text{RANDOMPOPULATION}(P) \) \( \triangleright \) Initialize using random binary vectors
5: \( F_0 \leftarrow \text{EVALUATE}(X_0) \)
6: for \( g \in [0, \ldots, G - 1] \) do
7: \( U \leftarrow \text{SELTOURNAMENT}(X_g, F_g) \) \( \triangleright \) Off-spring Tournament selection\(\sim(\text{Deb et al., 2002})\)
8: \( X_{g+1} \leftarrow \emptyset \)
9: \( Z_1, Z_2 \leftarrow \text{EXTRACT}(U, P/2), U \) \( \triangleright \) Partition \( U \) into 2 equal sets
10: for \( i \in [1, \ldots, P/2] \) do
11: \( o_1, o_2 \leftarrow \text{EXTRACT}(Z_1, 1), \text{EXTRACT}(Z_2, 1) \)
12: \( X_{g+1} \leftarrow X_{g+1} \cup \{\text{MATE}(o_1, o_2)\} \) if RANDOM() < \( p_c \) else \{\( o_1, o_2 \)\}
13: end for
14: \( X_{g+1} \leftarrow \{\text{MUTATE}(o) \text{ if RANDOM() < } p_m \text{ else } o \text{ given } o \in X_{g+1}\} \) \( \triangleright \) Random binary noise
15: \( F_{g+1} \leftarrow \text{EVALUATE}(X_{g+1}) \) \( \triangleright \) Get the fitness of the generation
16: end for
17: return \( \text{PARETO}\left(\bigcup_{g=0}^{G} \{X_g\}\right) \)
noise falls from 26.5 dBA to 20.94 dBA, the total CO₂ emitted increases from $1.201 \times 10^6$ kg to $2.25 \times 10^6$ kg. The status-quo, which allows trucks on the all roads can also be seen in the graphs to not be the optimal solution in any of the metrics. The results of the simulations have currently been presented twice to expert groups. The first presentation was carried out as a part of a meeting with the project board on September 9, 2020. Here a first draft of the model, the possible options and how it could be used was briefly outlined. The second meeting was organised as a workshop with the problem owner on September 17, 2020. Two experts participated here, and presented additional information related to routes and vehicle restrictions. Furthermore, they confirmed that the overall approach was reasonable and provided valuable insights.

In Figs. 5 and 8, it can be seen that paradoxically blocking some roads often leads to a better solution on all fronts than not blocking any roads. It can be understood as a variant of the famous Braess’s Paradox (Braess, 1968; Braess, Nagurney, & Wakolbinger, 2005), where adding an additional road (i.e., unblocking a road in our case) to a network may reduce overall traffic flow.

Another curious thing to note is that there is a much stronger inflection point in case of City B seen in Figs. 8 and 9 than for the corresponding Pareto optimal front for City A. This can be attributed to larger effect blocking of seminal roads has in the city. This inflection point is common for both the Noise / vehicles Pareto optimal front, as well as the Noise / CO₂ front in Figs. 8 and 9.
4.4. Results and discussion

The experiments performed in this work simultaneously optimize the local and global objectives, thus enabling bringing together the stakeholders from the two levels. The results from the optimization are presented in the form of approximate Pareto optimal curves which serve as the basis for discussion towards blocking/non-blocking of roads in the cities. On the one hand, stakeholders like residents, schools, hospitals can come together and discuss the local objectives in the form of noise pollution. On the other hand, the global objectives of minimizing the costs together with minimizing the carbon emissions serve as the decision-making criteria for stakeholders like personnel responsible for logistics from companies, policy makers for the city, traffic management system owners, etc. We indeed were able to find multiple solutions on the front with different compromises. The simulations ran in reasonable amount of time for both cities and, for City B, the results were validated by talking to experts.

Now we discuss some shortcomings of the approach for our method for solving this particular concrete problem, especially those which may generalize to other applications of our approach. It is of utmost importance that the roads identified for the encoding of blocking and unblocking are seminal roads for the city and it should be discussed among all stakeholders which roads are significant in this regard. These roads have to be decided before designing the experiments and thus requires preparatory work. The same will be true for other problems as well: deciding the control variables, and the constraints upfront with the stakeholders will be of utmost importance. The algorithm parameters have to be chosen such that the solution is a diverse set of points in the solu-
A. Singh, J. Baalsrud Hauge, M. Wiktorsson et al.  
Journal of Urban Mobility 2 (2022) 100012

Fig. 8. Graph of total number of vehicles reaching destination (before the simulation ends) and noise produced at the chosen edge.

Fig. 9. Graph of total CO₂ produced in the complete simulation and noise produced at the chosen edge.

tion space, i.e., the diversity of solutions does not disappear after some generations. This usually requires some hand-tuning though a judicious choice of the algorithm which automatically does this (e.g., NSGA-II) can address this problem. Another weakness of the method is shared with all other simulation based approaches: that of having access to high quality input data. The experiments for our example problem were performed for the daily average traffic numbers to simulate a typical day, but we spread the vehicles uniformly at random throughout the day and gave them random start and end points in the city. Real traffic does not behave so: there are clearly defined peak hours, and the vehicles follow a set route in the morning and a different one in the evening. These can be addressed by having access to improved data about the traffic. Another limitation which springs from the nature of evolutionary algorithms is that the Pareto optimal front found iteratively cannot be guaranteed to contain the best solutions always. We can mitigate this by running the optimization for more generations; there is a trade-off between the computational power required by the optimization and its quality. Lastly, we have made an assumption about the routing policies followed by the vehicles and have exclusively looked at regulations for heavy vehicles. For other problems, we will have to rigorously test the assumptions of the simulator similarly.

As the next steps, adding the type of engines and vehicles to the control variables can be done to permit a particular types of engine, e.g., Euro 5 or Euro 6 engines (Keller, Hausberger, Matzer, Wüthrich, & Notter, 2010), on a road. In the proposed methodology, network or road attributes play a crucial role. However, we have limited our work to changing the binary properties of roads i.e., whether heavy vehicles are allowed or a particular road or not. In further research, changing other
road attributes (e.g., speed limits) for roads would also be an interesting area.

5. Conclusions

In this work, we have presented a methodology for formulating complex multi-stakeholder SUMP problems and shown how they can be solved using simulators and Multi-Objective Evolutionary Algorithms. In particular, we presented an evolutionary algorithm based multi-objective optimization of the competing objectives of minimizing vehicular noise in a particular neighborhood together with minimizing city wide emissions. The research methodology was developed for the simulation-based optimization approach and was further demonstrated with the help of two case studies. Results from the experiments were discussed with the help of the non-dominated solution set on the approximate Pareto optimal front. This set of near-optimal solutions allows different stakeholders including the policy makers, traffic management and city administration to discuss the different options from the solution set and quantify the noise and carbon emission levels trade-offs. 

While we believe that our methodology, as well as tool/algorithm selection, lends itself well to other urban mobility problems, there may be better simulators or algorithms which could be utilized for particular problems. Therefore, it would be of interest to look at the performance and tractability of different evolutionary algorithms. Similarly, other simulators may be faster or better at simulating certain objective functions or constraints. We believe that applying supervised learning, reinforcement learning, Bayesian optimization and particle swarm optimization on the problem may also result in interesting insights. Another important direction of research could be to try to reduce the search space by including more domain specific knowledge into the model.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The research is funded by Interreg Baltic Sea Region (SS010EU) R105 HUPMOBILE project

References


