Towards a Realistic Optimization of Urban Traffic Flows

F. Angius\textsuperscript{1}, M. Reineri\textsuperscript{2}, C.-F. Chiasserini\textsuperscript{2}, M. Gerla\textsuperscript{1}, G. Pau\textsuperscript{1}

Abstract—In spite of recent advances in Intelligent Transport, vehicular traffic dynamics are still hard to represent and analyze. Most of the previous work on traffic regards highways or single lanes where vehicles interact in one dimension. Models for multi-dimensional vehicle-to-vehicle interactions and models for urban intersections are quite complicated and hardly applicable on a large scale. Nonetheless, urban traffic jams are an actual problem that requires a solution. This paper proposes a method to optimize urban traffic layout using basic heuristics and computationally efficient simulations. Instead of modeling an entire urban map with hundreds of intersections, each typology of intersection is simulated in order to understand how it responds to different traffic patterns and intensities. This knowledge is leveraged to allow the computation of minimal delay route on the complete road map. In order to validate our model, we use the solution obtained with our heuristic to derive the average travel delay through simulation on realistic Manhattan topologies with different intersection types.

I. INTRODUCTION

Traffic congestion is one of the major problems in urban areas. It impacts directly people’s everyday lives and causes several billion dollars lost revenues each year [1]. Traffic congestion can at times be tackled by resizing roads and junctions so that they can serve more vehicles. The roadway expansion problem reduces to classical bottleneck analysis which are widely exploited in computer systems and network design. Roadwork can then be planned empirically or as an optimization problem with respect to costs, time, space and user requirement [2]. Unfortunately, even when feasible, the increase of road capacity only mitigates the problem until the next inevitable demand growth. An alternative solution consists in efficient traffic control and management. Specifically, traffic control deals with tuning traffic lights, green waves and access ramp rates, so that the traffic density is sustainable for the infrastructure [3]. Traffic management proactively directs vehicles towards alternative routes in order to avoid jams [4]. Both approaches are driven by the goal of minimizing the delay perceived by the drivers.

Our work is inspired by computer network research that already dealt with congestion and optimization back in the days of the early Internet [5]. However, there is no satisfactory model for traffic flow on roadways as there is for computer network traffic. The complexity stems from the fact that vehicles do not operate independently but they constantly influence each other. Moreover, the problem is complicated by the unpredictability of the human behavior. The challenge addressed in this paper is to understand the behavior of vehicular flows across urban intersections. In particular, the paper clarifies how our urban traffic optimization has to deal with the following problems:

- The use of mathematical frameworks, such as linear or convex optimization, is not trivial because: (1) there is not a closed form solution to calculate the average delay of a vehicle on a route; and (2) the dimension of the optimization problem grows exponentially with the number of lanes and intersection that the system takes into consideration;
- It is computational expensive to reproduce traffic on a city-wide scale thus an optimal organization of traffic flows can hardly be found with an exhaustive search.

The paper is organized as follows. Sections II and III describe the idea and its application. Section IV explains the difficulty of representing urban traffic as a flow. Section V gives the details of our framework. Section VI contains the experimental results, while section VII discusses the previous works which are most related to our study. Finally, section VIII contains conclusion and future work.

II. CONTRIBUTION

The goal of this paper is to find the optimal routing (i.e., minimum overall delay) in a urban grid that includes intersections. Tackling the problem of vehicular flow optimization as a whole is not feasible due to many variables. In the one hand, previous works show that the methods that account for intersection details do not provide a closed form solution required for efficient multi-commodity flow optimization. On the other, existing flow optimization algorithms are suitable only for low-density highway traffic, indeed without intersections. Henceforth this work formalizes how to break the problem into smaller tasks that can be completed using simple tools and basic algorithms.

The underline idea consists in categorizing road segments so that they can be simulated and modeled individually. First, the traffic behavior at different types of intersection under various conditions is derived, then this information is exploited to compute optimal routing scheme for a given traffic demand. Based on this knowledge, the model of the system has been designed and used to minimize the average travel delay of the drivers.

This approach is promising for the following reasons:

- the approach does not depend on a specific simulator or traffic model. The choice of the simulator is intention-
ally left to the user as there are many simulators that can be used, but none that can successfully represents every scenario.

- In selecting the optimization algorithm it is possible to trade computational power for precision. Nevertheless, the results in this paper show how a simple greedy algorithm can optimize traffic flows on a realistic urban scenario.
- The knowledge base is created just once and it is updated only if some intersections change. Note that in this case the new model of intersection can be simulated individually and the knowledge base can be updated incrementally. This is an important aspect since it might take days to simulate small urban area while it takes less than an hour to simulate a single intersection.

### III. Example Scenario

The aim of this work is to accommodate the new generation of car navigators, in particular that ones that can receive remote instructions to use low congestion detours if the primary route is heavily congested. Today’s GPS navigators take in consideration only distance and traffic updates. Unfortunately, this method does not work during rush hours because all vehicles on the same congested segment take the same detour causing route flapping [6]. The next generation navigator will assign drivers to different detours, reducing the probability of congestion. For example consider a rush hour situation when the same people must drive on the same routes from home to work. Via a smart navigator, before leaving his garage, the driver is asked to choose the neighborhood closest to his destination. With that information, the Traffic Authority randomly assigns the vehicle to one of the possible routes that have been precomputed for the daily traffic demand. Note that this route might be longer than the straight path solution. However it has the advantage of being (1) decongested and (2) faster than traveling on the shortest path in case of traffic jam.

#### A. Feasibility of the implementation

The whole idea of letting a computer (e.g., the navigator) deciding the route to the destination might sound excessive and even scary for the driver. However, the impact will be beneficial to society, considered that the probability that an individual is consistently routed toward longer routes is negligible. On average the miles driven will be about the same than before but the total amount of time spent in the car will be much less. The main problem is the degree of penetration: this technology needs to be used by most of the drivers to be cost effective. To this aim it is possible to offer incentives to people who are willing to collaborate, for instance routes that are not suggested by the navigator for a particular driver are considered toll roads. Otherwise, people can be threatened with a fine if they do not respect their commitment. Note that it is not hard to verify if a vehicle is on its route, for instance it is possible to install at the main intersections RFID readers so that the presence of the vehicle can be recorded once it passes by. Viceversa, vehicles can send a beacon to base stations along the path to notify their presence.

### IV. Vehicles behavior in a urban scenario

The goal of this paper is to find the optimal routing (i.e. min overall delay) in an urban grid that includes intersections. For the sake of simplicity, only scenarios at full penetration rate with no selfish driver are considered in this work. A future extension will study the sensitivity of the approach to the different penetration rates. The previous works (see Section VII) indicates that the method that account for intersection details (e.g., cellular automata schemes) do not yield the closed form solution required for efficient multi-commodity flow optimization. On the other hand, existing flow optimization algorithms apply only to low-density highway traffic (no intersections). They cannot directly handle surface roads and residential scenarios. More specifically, conventional flow optimization methods cannot adequately approximate the impact of density impacts on: (1) the traveling time on each lane and (2) the waiting time at each intersection. Therefore the result is an oversimplified model that leads to misleading results.

In the following, a brief description of the vehicle interactions on the lanes and at the intersection is provided based on our analysis.

#### A. Vehicles on the lanes

![Fig. 1.](image)

The road segment $d_i$ is the space where normally vehicles are traveling whereas $\sigma$ is the space where they start slowing down to approach the intersection. Accordingly to the majority of traffic models, vehicle interfere with each other by stretching the inter-vehicle distance $s$ depending on the driver’s aggressiveness. An aggressive driver tends to stay closer to the vehicle ahead of him whereas a safe driver slows down to keep more space in between. By slowing down he might force another vehicle behind him to slow down causing a chain reaction that propagates backward on the lane.

Vehicles on the same lane influence each other by accelerating and decelerating as dictated by drivers habits. It is known, for example, that more aggressive drivers tend to maintain a smaller inter vehicle distance whereas safe drivers tend to slow down when approaching another car. Luckily, microscopic models such as Newell’s car-following [7] successfully capture all these aspects allowing to reproduce traffic dynamics such as jams and shockwaves. These phenomena cannot be noticed when considering vehicles as a flow. However, by taking into account their impact, on the flow model average delay, it is possible to anticipate them and slow down the traffic before severe congestion set in. Note that it is not feasible to redirect congested traffic in terms of flows. In case of congestion, vehicles must be rerouted individually and not as a flow. How vehicles interact on a single lane is shown in Fig. 1.


B. Vehicles in the intersection

Modeling intersections is even more complicated than modeling lanes, since vehicles interact in more than one dimension. Besides depending on drivers actions, the performance of an intersection is strongly influenced by:

* Capacity. Intersections have an implicit capacity bounded by the average time needed by a vehicle to move from one street to another. This value can be found with on-site measurements or can be computed from the average speed at the intersection. Note that, even knowing the capacity of an intersection, queue analysis still remains hardly applicable since the capacity is not uniformly distributed to each incoming lane. Instead, it varies with respect to factors such as geometry, policies, traffic density etc. Fig. 2 gives a graphical insight to these assumptions. Fig. 5 shows some results about how intersection policies affect the capacity of each street lane. In presence of a traffic light, the service time of the intersection is time shared among all the lanes in such a way that these almost never interfere with each other. In this particular case convex optimization can be used to optimize the traffic demand.

* Environment. The environment influences the way drivers approach the intersection and consequently its overall performance. This aspect can be better observed with real measurements rather than simulation although factors such as narrow roads, blind spots, bicycle lanes or pedestrian crossing impact sensibly the performance of the intersection.

* Policies. The delay experienced by vehicles at an intersection strongly depends on the policy of the intersection, e.g., right-before left, no right turn on red, traffic lights etc., and the amount of vehicles traveling on each incoming lane. When streets with priority are subject to high traffic volumes all the other incident streets become congested because the high priority streets are capturing the entire capacity of the intersection.

V. Model

This model has been developed by assuming that waiting time at an intersection is much smaller than the traveling time on the lanes between two intersections. However, if one or more intersections are jammed, the intersection becomes the bottleneck and the time spent on the lanes is of secondary importance. With this key assumption the following model has been formulated. A urban map is considered as $M = (I, L)$, where $I$ is the set of the intersections and $L$ is the set of lanes on the map. Each lane $l_i \in L$ is assumed to have (1) known length $d_i$ and (2) a known vehicle average speed $v_i$. Traffic demand is expressed as:

1. A set of users $U = \{u_k : u_k \in I \times I\}$. Because flows are considered from intersection to intersection.

2. A function $\Phi(x) : U \mapsto \mathbb{R}$ that maps users to their vehicles/hour demand.

Each user can be served by a number of routes from a set $R = \{r_i : r_i \subset L\}$. From now on, $r_i \subset u_k$ signifies that route $r_i$ serves user $u_k$. Analogously $l_i \subset r_j$ means that route $r_j$ contains the lane $l_i$. Two vectors $f = \{f_1, f_2, ...f_{|L|}\}$ and $y = \{y_1, y_2, ..., y_{|L|}\}$ represent the current state of the system, $f_i$ contains the amount of flow on lane $l_i$ whereas $y_j$ contains the amount of flow on the route $r_j$.

By design, this model does not handle heavy, congested traffic as the latter cannot be properly represented as a continuous steady state flow. However, if applied incrementally, this approach delays the congestion build-up until the capacity of the road system is reached.

Formally, it works under the assumption that, given a lane $l_i$, vehicles never stop before having driven a distance $d_i - \sigma$ on it, where $\sigma$ is the space of the lane used by vehicles to decelerate and approach the intersection. The reason is that given a lane $l_i$, the time spent on it, i.e., $T_i(f)$, must be split into two components: the amount spent at the intersection $w_i$ and the time spent moving on the street segment $l_i$. This leads to the expression:

$$T_i(f) = \frac{d_i - \sigma}{v_i} + w_i(f)$$

Thus, the total time spent on a route $r_j$ is equal to:

$$T(r_j, f) = \sum_{l_i \subset r_j} T_i(f)$$

Thus, the minimum average delay for the drivers can be written as:

$$\min_f \left( \frac{1}{\gamma} \sum_{l_i \in L} T_i(f) \cdot f_i \right)$$

where $\gamma$ is equal to $\sum_{r_j \in R} y_j$. Note that, if this function were convex the problem could be solved with the flow deviation method or similar frameworks. Unfortunately, as the results in Fig. 6 show, the function does not remain convex due to $w_i(f)$. The time spent at an intersection depends on the load of each incident lane and, to the best of our knowledge, beside [8] there is no previous work for a synthetic representation of the different types of urban intersections. In order
to complete the task, our model considers the time spent at an intersection with \( n \) incident lanes as a generic function \( \mathbb{R}^n \to \mathbb{R}^n \) where the domain is the amount of flows on the incident lanes and the codomain is the delay on each lane. An estimate can be obtained by simulation.

![Image](96x536 to 257x663)

Fig. 3. Example of recognizable urban patterns in Santa Monica (CA). Note the symmetry of the areas in red, yellow, purple and blue. By zooming in, intersections can be recognized to have the same geometry. Differently, in the area colored in green each intersection is different from the others and clearly need to be inspected further.

A. Simulation

Before running the simulations, intersections must be classified with respect to their geometry. This task can be easily done via software even though some urban areas can simply be visually inspected. For example, in Fig. 3, it is clear which intersections have the same geometry and which do not. The process of deriving a delay function can be summarized as follows:

1) The process starts with an initial minimum flow on each lane. Then, at each iteration one of the flow is incremented of a given step until all the lanes are saturated. The smaller is the step, the more precise is the resulting function.

2) Simulation run so that both the time a vehicle arrives at a distance \( \sigma \) from the intersection and the time the vehicle enters into another lane are recorded. The difference between the two is then averaged with basic statistic.

It is important to verify that vehicles depart at random instants of time before approaching the intersection. The reader is owed of an explanation of why simulations are ran only on intersections and not on the entire lanes. In particular:

- The performance of streets can be usually approximated dividing their length by the average speed. Intersections, instead, follow complex dynamics that do not have a closed form solution. Although, on the other hand, intersections with similar geometry, and same strategy, show similar performance, as in [9]. This paper aims at minimizing time and costs of traffic management by reducing the number of simulations needed (simulation requires expensive hardware for computation).

- If there is no traffic jam, the time that a car travels before approaching the intersection is quite accurate.

- It is relatively easy to categorize intersections on a map by inspecting their geometry.

B. Algorithm

At last an heuristic must be chosen to compute optimal routes. As a matter of fact, there are many algorithms that can be used, each one with a different tradeoff between performance and precision. Here we present a very intuitive greedy algorithm that helped us to prove the validity of our assumptions and gave good results in practice, as shown in Fig. 4. The algorithm simply loops over each user \( u_k \) and distributes a fraction \( \delta \) of its traffic demand \( \Phi(u_k) \) on the route that would increase the total average delay the least. This can be formalized as:

\[
\text{opt}(u_k, \mathbf{f}) = r_i \quad \text{s.t.} \quad \frac{\partial T(\mathbf{f})}{\partial x_i} = \min_{j \in u_k} \left( \frac{\partial T(\mathbf{f})}{\partial x_j} \right)
\]

The pseudo-code is the following:

**Algorithm 1 Greedy Algorithm**

1) while \( \text{tot} < \gamma \)
2) \( \mathbf{y} \leftarrow [0, 0, \ldots, 0] \)
3) \( \mathbf{p} \leftarrow [0, 0, \ldots, 0] \)
4) for each \( u_k \in U \)
5) \( \text{min} = \text{opt}(u_k, \mathbf{f}) \)
6) if \( \Phi(u_k) < \Phi(u_k) \)
7) \( \gamma[\text{min}] = \gamma[\text{min}] + \delta \)
8) \( p[k] = p[k] + \delta \)
9) \( \text{tot} = \text{tot} + \delta \)

The time complexity is \( O(U) \); however, as most of the greedy algorithm, this is an approximated algorithm which can deviate from the optimal solution. A theoretical bound to the quality of the approximation is part of the future work, while the experimental results can be found in the next section. In any case, the approach described throughout the paper can be used with any other heuristic that can optimize the objective function (3).

VI. Evaluation

For the experiments we used the open source simulator SUMO (Simulation of Urban MOBility) [10]. Unfortunately, due to the complexity of parsing SUMO map files, experiments could be run only over Manhattan grids with \( 11 \times 11 \), \( 21 \times 21 \) and \( 31 \times 31 \) intersections of the same type. Future work will include real urban maps as the one in Fig. 3. Each road segment on the grid is 400 m long and vehicles travel on it with an average speed of 60 Kmh/h. Traffic demand consists of two orthogonal flows moving respectively from East to West and from North to South.

The experiments used two different kinds of intersections:

- **Priority**: two parallel lanes out of four have absolute priority on the others; this implies that parallel lanes do not interfere with each other. Also, when the priority
Vehicles moved on the map using the car-following model as described in [11].

A. Comparison

The proposed greedy scheme has been compared against three different routing policies:

- **Shortest Path (SP):** traffic demand is directed toward its shortest route. Obviously, this policy initially leads to the minimum average delay, but it causes congestion earlier. Hence, it has been used as a lower bound.
- **Load Balancing (LB):** traffic demand is equally split on its possible routes. Note that, on a Manhattan grid, load balancing would be the routing policy that delays congestion the most if traffic could move as a simple fluid.
- **Mixed Strategy (SP+LB):** this policy combines the two previous approaches. The flow moving from North to South is directed on the shortest path, whereas the other is equally divided on all the available routes.

B. Results

The graphs in Fig. 4 show that even by taking into account only the delay introduced by the intersections the results produced by our “greedy” optimization far outperform the typical load balance and shortest path strategies. This is due, in great part, to the fact that our model, albeit approximate, still reproduces the critical vehicle-to-vehicle interactions in a realistic way. Eventually, also the greedy approach will lead to congestion for large enough offered loads. One may argue that better heuristic solutions could be obtained by recalculating the entire traffic layout after each increment in demand. Obviously, the time complexity of the latter method would be much higher. A possible compromise is to adopt a randomized approach to search for local minima, an option that we are currently investigating.

VII. RELATED WORK

Here, we discuss previous work dealing with traffic models and traffic management which are mostly related to our study.

A. Traffic Models

Mobility models are classified under three categories, as by Fiore et al in [12], that offer different tradeoffs between performance and complexity: Macroscopic, Mesoscopic and Microscopic.

**Macroscopic models** look at traffic as a continuous flow of vehicles. This is the highest level of abstraction and it is hardly applicable to an urban scenario that has intersections and cross traffic. The aim of this paper is to use this level of abstraction to optimize delays in an urban scenario without losing precision. Noticeable contributions are:

- From Prigogine et al. [13]. This model investigates the interactions of two traffic flows in a urban scenario. It is a valuable contribution although it is quite far from modeling realistic intersections.
- Fluid Traffic Model (FTM) [14]. This model adapts the speed of vehicles according to traffic density. It considers a single lane without accounting for multi-flow interactions.
- Car following models. In this kind of models vehicle motion is described by Ordinary Differential Equations (ODE) or a Algebraic Differential Equations (ADE). Popular examples are the Intelligent Driver Model (IDM), the GHR Model [15] and the Krauss Model [11].
- Nagel and Schreckenberg [16] is a stochastic discrete automaton model to simulate freeway traffic. This work consists of a discrete model that allows to represents important phenomena such as (1) traffic jam and (2) the transition from laminar to start-stop-traffic.
- Biham et al. in [17] describes how traffic flows interact on a two dimensional space. In spite of the simplicity of the model, the results show a sharp transition that separates a low-density dynamical phase in which all cars move at a maximum speed and a high-density static phase in which they get stuck in a global traffic jam.

**Mesoscopic models** compromise between the simplicity of a macroscopic approach and the precision of microscopics. Vehicles usually move in groups or clusters (e.g., platoons) so that the probability of a specific vehicle being in a certain place at a specific time instant can be bounded. This solution offers a tradeoff between car level precision and the complexity of microscopic models, although it can deviate significantly from the real scenario.

B. Traffic Management

**Reactive schemes.** They are usually studied from the control theory point of view: traffic is seen as a dynamic system. A feedback mechanism triggers a controller to tune the traffic regulators (e.g., traffic lights, access ramp controls, etc). In [18], two schemes are used: a local balancing scheme and a global scheme. The global scheme assumes global traffic knowledge at each node, which is unlikely to happen in a real environment. Moreover, in the latter case the routing is done by each vehicle independently and thus it is subject to route flapping. The local scheme, on the other hand, assumes that the On Board Navigator has heard from its peers in the adjacent road segments and uses that information to perform local route balancing. Results in [18] state that Global Optimization is not any better than
local load balancing. However, these results must be taken with a grain of salt since they assume that the traffic is randomly distributed. This implies that there are no major hot spots, such as a congested freeway access ramp, which is unlikely. In [19] Mohandas et al. propose to use the Adaptive Proportional Integral rate controller as it has been done for the Internet to deal with links congestion. The work showed the applicability of the method although there has been no experimentation in the urban scenario. Likewise, in [20] there is an extensive theory for vehicular traffic control without
Flow 2 is the one with priority, while in (b) flow 1 has the right of way against flow 2. Note how, even if in both the configurations there is only one flow with priority, the performances are not the same (in the first performances drop at 800 vehicles/h, whereas in the other at 700). This is an example of how small factors can impact the performance of urban intersections.

**Proactive schemes.** They consist in preemptively computing a best solution for traffic using a knowledge base created with real measurements or via modeling and simulation. For example, Liu et al. in [21] use historical records to calculate the optimal routes for the drivers. Other proposals, such as [22], suggests the use of bayesian networks to infer traffic dynamics.

**Flow-based schemes.** Many proposals are inspired by previous work on computer networks optimization, such as [4] [23] [24]. For instance, Kim et al. in [4] use the Flow Deviation algorithm for load balancing traffic demand over alternative routes. This model has appeal because it leads to a closed form expression of delay as a function of road segment parameters, and can be used to obtain optimal routes using convex optimization. Lanes act as the links of a network and intersections act as the routers. Each lane $l_i$ has a maximum capacity $C_i$, measured in vehicles/hour, and a traveling time $T_p$ (analogous to the propagation time of networks links) measured in seconds; intuitively vehicles move from link to link like packets do in a computer network. From [5], the average delay of a vehicle is equal to the time spent in a M/M/1 queue with service rate $\mu_i = C_i$ and arrival rate $\lambda_i$ equal to the intensity of incoming traffic on the lane. The total delay on the lane has two components: $T_d = T_p + T_q$ where $T_p = \frac{1}{C_i - \lambda_i}$ is the average queuing time spent at the ending intersection of lane $l_i$ and $T_p = \frac{\text{Lane Length}}{\text{Average Speed}}$ is the average traveling time on the lane. The average delay perceived by vehicles can then be rewritten as:

$$T = \sum_{l_i \in L} \frac{\lambda_i}{\gamma} T_d(i)$$

where $n$ is the number of lanes in the map, $\gamma$ is the total vehicle arrival rate to the system and $T_d(i)$ is the average delay of lane $l_i$. Unfortunately, vehicles interact in a different way than packets thus the result that can be obtained with this model are quite far from what can be observed in reality. For example, packets in a router can be switched from input to output port with minimal overhead, while vehicles suffer the most severe delays at intersections. These models are based on the unrealistic assumption that a road lane can be approximated with basic queues, such as an M/M/1. Other works, such as [25], dug more into advanced queueing theory for a finer representation of the problem. Unfortunately, while simple queue models fail in describing the real dynamics of vehicles, the advanced queue models are too complicated to be applied to a real world scenario. Instead, our paper proposes an hybrid approach that leads to realistic results using relatively simple algorithms and inexpensive simulators.

**VIII. Conclusions and future work**

In this paper we described an efficient yet simple way of optimizing traffic flows on a realistic urban scenario. The proposed solution is computationally efficient, accurate and
inexpensive to be deployed. As future work, we would like to optimize traffic on a real map and to study the algorithm sensitivity for different penetration rates.

IX. ACKNOWLEDGMENTS

The authors gratefully acknowledge NSF Grant CNS-1111971 and the Fulbright Program that supported Massimo Reineri with a Fulbright Scholarship during his stay at UCLA.

REFERENCES