



# Intersections management for autonomous vehicles: a heuristic approach

Victor Silva<sup>2</sup> · Clairton Siebra<sup>1</sup>  · Anand Subramanian<sup>2</sup>

Received: 18 June 2020 / Revised: 21 June 2021 / Accepted: 29 November 2021 /  
Published online: 9 January 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

## Abstract

Roads intersections are one of the main causes of traffic jams since vehicles need to stop and wait for their time to go. Scenarios that only consider autonomous vehicles can minimize this problem using intelligent systems that manage the time when each vehicle will pass across the intersection. This paper proposes a mathematical model and a heuristic that optimize this management. The efficiency of this approach is demonstrated using traffic simulations, with scenarios of different complexities, and metrics representing the arrival time, CO<sub>2</sub> emission and fuel consumption. The results show that the present approach is scalable, maintaining its performance even in complex real scenarios. Moreover, its execution time is maintained in milliseconds, what suggests this approach as a candidate for dealing with real-time and dynamic scenarios.

**Keywords** Autonomous vehicles · Optimization · Heuristic

## 1 Introduction

Nowadays, traffic lights and roundabouts are the main solutions to assist drivers in road intersections. However, these methods require that drivers stop their vehicles, waiting for their time to continue their journeys. This behavior usually causes a delay in the expected traffic flow, increasing fuel consumption and air pollution (Medina et al. 2015). Moreover, the slow flow and the frequent stop-and-go traffic behavior increase the stress of drivers, creating a prone situation for accidents (Lamouik et al.

---

✉ Clairton Siebra  
clairton@ci.ufpb.br

<sup>1</sup> LIAA, Universidade Federal da Paraíba, Rua dos Escoteiros s/n, Mangabeira, João Pessoa, PB 58058-600, Brazil

<sup>2</sup> Centro de Informática, Universidade Federal da Paraíba, Rua dos Escoteiros s/n, Mangabeira, João Pessoa, PB 58058-600, Brazil

2017; Wu et al. 2013). The study in Kamal et al. (2014) shows other disadvantages in terms of time wasted, fuel consumption, and total cost caused by traffic jams.

The future scenario, where the majority of the vehicles tend to have autonomous resources (Zhu and Ukkusuri 2015), brings opportunities to the development of intelligent strategies for the management of intersections. The literature already discusses technologies related to connected vehicles such as *Vehicle-to-Vehicle* (V2V), *Vehicle-to-Infrastructure* (V2I), and vehicle to other digital components (V2X) (Gong et al. 2016). These technologies allow individual vehicles to access online information about the traffic, acquire the current situation of the roads, and interact with other vehicles to obtain collaborative decisions.

This paper proposes a V2I strategy to manage traffic intersections as a solution to mitigate the problem of traffic jams and, consequently, its related issues (e.g. higher CO<sub>2</sub> emission and fuel consumption). The main idea is to generate a joint global plan that indicates the speed that each vehicle must maintain in different parts of their routes, avoiding in such way conflicts regarding the use of intersections. The mathematical model used to create this global plan was initially specified using Mixed Integer Linear Programming (MILP) and the sum of arrival times as the main attribute to be optimized, considering all the vehicles that are currently active in the scenario. We also propose a heuristic algorithm as a strategy to support scalability, therefore maintaining the executing time in levels that allow the future use of this model in dynamic environments.

We conducted a set of experiments using the SUMO simulator (Lopez et al. 2018) to evaluate this approach and show its advantages over the traditional use of traffic lights for intersections control. Therefore, the experiments used the same scenarios with two different configurations. The first configuration contained traffic lights to control the flow of vehicles. These vehicles use the maximum speed limit allowed for each lane, and only the traffic lights modify this situation. Thus, the sum of arrival times is always as low as possible for such a configuration. The second configuration removed all the traffic lights from the scenarios, and a global plan was used to control the vehicles' behavior. The results of both configurations were given in the form of (1) sum of the arrival times of all vehicles, (2) total of CO<sub>2</sub> emission over the simulation, and (3) total of fuel consumed over the simulation.

We employed synthetic and real scenarios with crescent levels of complexity (number of vehicles and intersections) to show the scalability and efficiency of this approach. Real scenarios represent regions generated using *OpenStreetMaps*, whose format is compatible with SUMO. While the evaluation of previous approaches is limited to simple domains (from 1 to 12 intersections and a few vehicles), as far we know, this is the first approach that considers complex simulations that involve, for example, more than one thousand intersections.

The remainder of the paper is organized as follows. Section 2 presents the related work. Section 3 describes the mathematical formulation and the heuristic approach. Section 4 contains the results of the computational experiments. Finally, Sect. 5 concludes this study with its main remarks, limitations and research directions.

## 2 Related work

The approaches for intersection management are divided into centralized and distributed strategies. The centralized strategy uses a central controller to coordinate the traffic of vehicles, while the distributed strategy is based on the information sharing among vehicles. The main examples of these two approaches are discussed considering five criteria, if they are available: (i) complexity of the map, (ii) technique applied, (iii) limitations, (iv) simulator, and (v) use of measures.

### 2.1 Centralized approaches

Hausknecht and colleagues (Hausknecht et al. 2011) extended the initial work of Dresner and Stone (Dresner and Stone 2014) beyond the case of an individual intersection. Their experiments employed 4 intersections, 12 roads with 2 lanes in each direction; and the A\* algorithm was used to dynamically alter initial planned paths. This technique caused some delays that negatively affected the traffic and the vehicles could only change their decisions outside of the limit of 300 m to the intersection. An evaluation environment was developed together with the project and questions related to the reduction of fuel consumption or CO<sub>2</sub> emission were not considered.

Zhu and Ukkusuri (2015) developed a linear programming formulation for autonomous intersection control. Their evaluations employed a matrix  $4 \times 3$  with 12 intersections. This approach was not accident free so they could observe some collisions. Questions related to the reduction of fuel consumption or CO<sub>2</sub> were also not considered.

Sharon and Stone (2017) proposed a new protocol, called *Hybrid Autonomous Intersection Management*, which is able to sense approaching vehicles and use this information to improve the process of path reservation within the intersection. Their evaluations employed a region with 1 intersection and 4 roads, each of them with 3 lanes in both directions. They developed their own simulating environment for autonomous and non-autonomous vehicles. The use of non-autonomous vehicles could cause undesired behaviors and the controller should be able to deal with these situations. Questions related to the reduction of fuel consumption or CO<sub>2</sub> were also not considered.

Budan et al. (2018) used the concepts of vehicle-to-infrastructure (V2I) communications and multi-agent systems as central to achieving a robust and reliable traffic-light-free intersection control. Their evaluations employed a matrix with 1 intersection, 4 roads with 2 lanes each in both directions. A total of 4 conventional vehicles, with distinct behaviors, were used to generate diverse situations in the intersection. The *PTV Vissim* simulator (PTV Group 2013) was used and the results show a reduction of 42% regarding the consumption of fuel.

Zheng et al. (2017) presented a cooperative vehicle intersection control scheme, which was based on the *Predictive Model Control Theory*. Their simulations employed 1 intersection, and four roads with 1 lane in each direction. The experiments in the *PTV Vissim* simulator showed that their model was not able to ensure the viability of the optimal solution. In the best scenario, the evaluations returned a reduction of 58.6% of CO<sub>2</sub> emission and 52.4% of fuel consumption.

Ashtiani et al. (2018) formalized the intersection management as a problem of optimization using MILP. After receiving subscription requests and the status of approaching vehicles, the intersection control calculates an arrival schedule that ensures safety while reduces the number of stops and intersection delays. Their evaluations employed a 3x3 matrix, where each intersection had four roads, each of them with 1 lane in both directions. The MILP was used considering the exchange of information between the neighborhood intersections. The group developed their own simulator in Java and further measurements were not considered.

## 2.2 Distributed approaches

Belkhouche (2017) presented a collaborative method for collision avoidance that expresses the collision conditions in terms of the vehicles' speed ratios. This scenario is then formulated and solved as an optimization problem with the safety constraints represented by the speed ratios, and the cost function is defined in terms of the deviation from the current speeds. The evaluation employed 1 intersection and 4 roads, each of them with a lane in both directions. The author discusses the possible scalability of the approach, but only 3 vehicles were used during the experiments. Further metrics such as consumption and level of CO<sub>2</sub> emission were not considered.

Campos et al. (2014) proposed a solution that is based on a receding horizon formulation with a pre-defined decision order. In this approach, local problems are formulated for each vehicle and divided into a finite-time optimal control problem, where collision avoidance is enforced as terminal constraints, and an infinite horizon control problem that can be solved offline. Their evaluation employed 1 intersection and 4 roads, each of them with 1 lane in both directions. A total of 3 vehicles were used but the paper does not give further details about the simulator used and other metrics.

Makarem and Gillet (2011) considered the decentralized coordination of point-mass autonomous vehicles at intersections using navigation functions. Their main contribution is to consider the inertia of the vehicles to enable onboard energy optimization for crossing. This approach defines a limited field of vision where information exchange and vehicle detection can be conducted. However, the authors did not indicate the decisions that are taken in case of failures in this communication. Their evaluation employed 1 intersection and 4 roads, but the number of lanes was not specified. The paper does not give further details about the simulator and other metrics.

Lim et al. (2018) proposed a hierarchical trajectory planning based on the integration of a sampling and an optimization method for urban autonomous driving. These algorithms were evaluated through experiments in various scenarios of an urban area involving the presence of intersections. Their evaluation employed 3 scenarios with a variation of a few roads with curves and 3 lanes. A real autonomous vehicle (A1 - University of Hanyang) was used in such an evaluation and metrics of consumption and emission were not considered.

Isele et al. (2017) explored the Deep Reinforcement Learning technique to handle intersection problems. The idea was to learn policies that surpass the performance of a commonly-used heuristic approach in several metrics including task completion time and goal success rate and have limited ability to generalize. Their evaluation employed 1 intersection with 6 lanes. The training stage of this learning approach was conducted using data from 10 thousand executions for each scenario. The simulator used was SUMO, but its metrics of consumption and emission were not considered.

de Campos et al. (2017) presented a decentralized coordination approach that combines optimal control with model-based heuristics. Their evaluation employed 1 intersection with 4 roads, with 1 lane each. Other details such as time to generate the solutions, the simulator used, and further metrics are not given.

Fayazi et al. (2017) proposed an optimal scheduling of autonomous vehicle arrivals at intersections, which is formulated as a MILP instance. Their evaluation employed 1 intersection with 4 roads, each of them with 1 lane in each direction. Other details such as time to generate the solutions, the simulator used, and further metrics are not given.

### 2.3 Discussion

The analysis of the centralized and distributed approaches show they are only focused on very limited scenarios. In order, the majority of the centralized works considered 1 intersection with 4 roads, while the number of cars used during the simulations is not well-detailed. Issues that could appear due to the use of only one central controller (e.g. bottleneck) are mentioned, but their analysis is considered out of the scope. The main metric is associated with the average time to conclude the routes and all approaches show that such time is reduced when compared with the traditional approaches (use of traffic lights and roundabouts). However, important sustainability metrics such as fuel consumption and CO<sub>2</sub> emission are not usually considered. The distributed approaches also present these similar features. Moreover, their implementation seems more complex if we consider their transition to the real world. For example, the reliability and failure tolerance regarding the communication must be higher than the required in centralized approaches.

In contrast to these studies, the present centralized approach considers evaluations in complex scenarios, using a MILP approach with safety constraints that ensure the nonexistence of collisions while the mathematical model tries to reduce the arrival time of the vehicles. To ensure the scalability of this approach, we also proposed a heuristic to better deal with the conflicts and allow the evaluation of this approach in scenarios with several intersections and vehicles. While this strategy does not ensure the optimal solution, it is conflict-free and its resolution time was maintained in milliseconds, as better detailed in the results section. As it is not ensured that the decrease of the arrival time reduces the fuel consumption and CO<sub>2</sub> emission, we also considered such metrics during our evaluations.

### 3 Methods and tools

#### 3.1 Premises

This study considers the next set of premises: (i) there is a central component that receives all the requests for journeys from the vehicles, executes the schedule, and sends the speed instructions back to the vehicles; (ii) each request is a 2-tuple  $\langle \text{origin}, \text{destination} \rangle$ ; (iii) each instruction is a sequence of speed values for each lane that composes the route of a vehicle; (iv) all vehicles of the simulation are autonomous and strictly follow the indications of speed for each road; (v) the simulation does not consider inertial questions so only similar vehicles are allowed rather than motorcycles or trucks; (vi) the scenarios are static so other vehicles do not join after the beginning of the simulation; (vii) transit lights and roundabout were not used to control the intersections flow; (viii) each road or lane has a maximum value for speed and the management system knows such information; and (ix) the vehicle only stops after reaching the last intersection of its corresponding route.

Based on this information and considering that each vehicle will receive a fix speed value for each road, the management system is able to define the time interval that each vehicle will be in an intersection. The system also uses the *First Come First Serve* approach to include the requests into the agenda and, after this definition, the time of each journey is returned as the result of an integer linear programming formulation.

#### 3.2 Optimization problem and mathematical formulation

The mathematical formulation that describes the optimization problem considered in this work has the following definitions:

- $V$ —set of nodes (intersections);
- $A$ —set of arcs (transitions from a node to another);
- $K$ —set of vehicles;
- $K_j$ —set of vehicles that pass across  $j \in V$ ;
- $K_{ij}$ —set of vehicles that pass along  $(i, j) \in A$ .

The input data is represented as:

- $I$ —Minimum interval between two vehicles in an arc;
- $T_{ij}^{min}$ —Minimum time of  $(i, j) \in A$ ;
- $T_{ij}^{max}$ —Maximum time of  $(i, j) \in A$ ;
- $\pi^k$ —Sequence of arcs that compose the route of a vehicle  $k \in K$ .

And the next following decision variables were used:

- $x_{ij}^k$ —Time that a vehicle  $k \in K$  spends to pass along  $(i, j) \in A$ ;
- $y_j^k$ —Arrival time of the vehicle  $k \in K$  and  $j \in V$ .

For example,  $\pi^k = (1, 2, 7, 8, 9, 10)$  is the sequence of nodes that represent the route illustrated in Fig. 1. Then,  $\pi^k(2) = 2$  is the second node that will be crossed by the vehicle. Considering that  $|\pi^k|$  is the cardinality of the nodes sequence, then  $|\pi^k| = 6$ ,

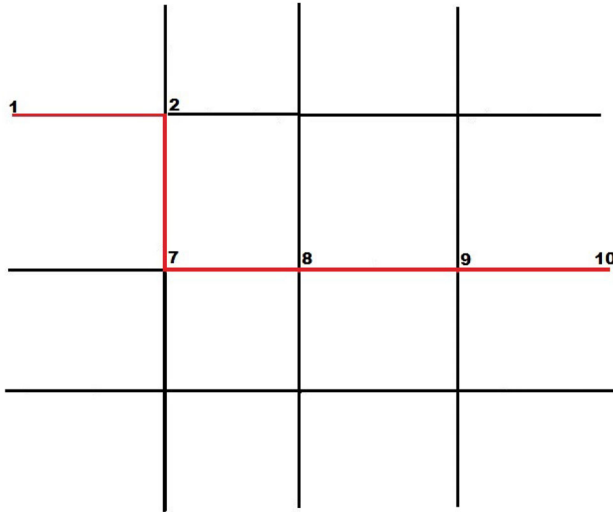


Fig. 1 Example of a route

while  $\pi^k(|\pi^k|) = 10$  represents the last node (destination) that will be reached by the vehicle in this example.

The proposed mathematical formulation is described as follows.

$$\min \sum_{k \in K} y_{\pi^k(|\pi^k|)}^k \tag{1}$$

Subject to:

$$y_{\pi^k(i)}^k = y_{\pi^k(i-1)}^k + x_{\pi^k(i-1)\pi^k(i)}^k, \quad \forall k \in K, i = 2, \dots, |\pi^k| \tag{2}$$

$$|y_j^k - y_j^l| \geq I, \forall j \in V, \forall k \in K_j, \quad \forall l \in K_j, k \neq l \tag{3}$$

$$T_{ij}^{min} \leq x_{ij}^k \leq T_{ij}^{max}, \forall k \in K_{ij}, \quad (i, j) \in A \tag{4}$$

$$y_j^k \geq 0, \quad \forall j \in V, \forall k \in K_j \tag{5}$$

The objective function (1) aims to reduce the sum of the arrival times at the last intersection of each route. Constraints (2) ensure that the time of each intersection, which composes a route, is scheduled considering the speed of the vehicle. To define a safety margin while a vehicle is crossing the intersection, constraints (3) guarantee that the time difference between two vehicles that will cross the intersection is higher than a predefined interval. This means that, while some vehicle is crossing the intersection, another vehicle will only be allowed to cross this same intersection after a predefined time.

To maintain the model as linear, constraints (3) were replaced by (6) and (7), which use a binary variable  $\underline{b}$  and a parameter  $M$  sufficiently large.

$$y_j^k - y_j^l + M\underline{b} \geq I, \quad \forall i \in V, \forall k \in K_j, \forall l \in K_j, k \neq l \quad (6)$$

$$y_j^k - y_j^l + M\underline{b} \leq -I + M, \quad \forall i \in V, \forall k \in K_j, \forall l \in K_j, k \neq l. \quad (7)$$

After defining and associating the speed of a vehicle to a road, such a vehicle must maintain this speed while it is on this road. Thus, a vehicle can present different speeds but only on different roads as stated by constraints (4) were imposed. Finally, constraints (5) ensure the non-existence of negative times in the model.

### 3.3 Heuristic

The proposed heuristic allows for solving the problem in areas with a high amount of intersections and vehicles, as the scenarios described in the next section. Its idea is to schedule the early arrival time possible for each intersection that composes the route of a vehicle, as described in Algorithm 1. The procedure receives as input the list of *routes* in a predefined order and the safety time interval (*sTime*) (line 1). The structure *departureTime* stores the departure time of vehicle (route)  $i$  at intersection  $j$  (line 2). For each route, the heuristic tries to schedule the earliest possible departure time at each intersection, using the procedure described in Algorithm 2, in a sequential order (lines 3–45). The departure time of the first intersection is determined in lines 7–19. Variables *tFirst*, *tAux*, and *tAcc* store the departure time of the first intersection, the departure time of the current intersection other than the first, and the accumulated time, respectively. Once the departure time of the first intersection is determined, then those of the remaining intersections are computed in lines 20–40. After all intersections of a given route have been completely scheduled, then the corresponding departure times are added to the list of times (*tScheduled*) associated with each intersection (lines 41–44). Finally, the solution (i.e., the departure time associated with every intersection of each route) is returned.

In our solution approach, Algorithm 1 is actually called two times. In the first, we consider the list of routes in their original form, whereas in the second we consider the list of routes in the reverse order, as shown in the Main Algorithm 3. The objective function of both solutions are computed using Eq. (1), and the solution that yields the smaller value is selected. Note that because of premise (ix) described in Sect. 3.1, that is, the vehicle only stops after reaching the last intersection, it can be assumed that the departure and arrival times of a vehicle at a given intersection are the same, thus allowing one to use Eq. (1) in this case.

Consider the illustration depicted in Fig. 2 as the map for an example that contains 4 vehicles with different routes. The first route is composed of the intersections  $\pi^1 = (4,5,6,7,8)$ , the second route is  $\pi^2 = (1,5,10,15,19)$ , the third is  $\pi^3 = (3,7,12,17,21)$ , and the fourth is  $\pi^4 = (14,15,16,17,18)$ . As a form to simplify this example, also consider that all roads have the same length (150 m), the minimum time to go along the road is 7 s, while the maximum time is 9 s. This means that the vehicles will have a speed around 80 km/h and 60 km/h, respectively. The safety interval is 5 s.



**Algorithm 1:** Heuristic for conflict resolution

---

```

1 ConflictResolution(routes[], sTime)
2 departureTime[][]  $\leftarrow$  0
3 for k  $\leftarrow$  1 to routes.size() do
4   i  $\leftarrow$  routes[k]
5   tAcc  $\leftarrow$  tFirst  $\leftarrow$  0
6   for j  $\leftarrow$  1 to i.size() do
7     if j = 1 then
8       d  $\leftarrow$  intersection(j)
9       scheduled  $\leftarrow$  tryToSchedule(tFirst, d.tScheduled[], sTime)
10      if scheduled then
11        tAcc  $\leftarrow$  tAcc + tFirst
12        departureTime[i][j]  $\leftarrow$  tAcc
13        j  $\leftarrow$  j + 1
14      end
15      else
16        j  $\leftarrow$  1
17        tFirst  $\leftarrow$  tFirst + 1
18      end
19    end
20    else
21      o  $\leftarrow$  intersection(j - 1)
22      d  $\leftarrow$  intersection(j)
23      for time  $\leftarrow$  tMin[o][d] to tMax[o][d] do
24        tAux  $\leftarrow$  tAcc + time
25        scheduled  $\leftarrow$  tryToSchedule(tAux, d.tScheduled[], sTime)
26        if scheduled then
27          tAcc  $\leftarrow$  tAcc + tAux
28          departureTime[i][j]  $\leftarrow$  tAcc
29          scheduled  $\leftarrow$  true
30          j  $\leftarrow$  j + 1
31          break
32        end
33      end
34      if scheduled = false then
35        j  $\leftarrow$  1
36        tAcc  $\leftarrow$  tAux  $\leftarrow$  0
37        tFirst  $\leftarrow$  tFirst + 1
38      end
39    end
40  end
41  for j  $\leftarrow$  1 to Size(route) do
42    d  $\leftarrow$  intersection(j)
43    add departureTime[i][j] to d.tScheduled[]
44  end
45 end
46 return departureTime

```

---

---

**Algorithm 2:** Try to schedule intersection
 

---

```

1 tryToSchedule(timeToSchedule, tScheduled[], sTime)
2 if tScheduled.size() = 0 then
3   | return true
4 end
5 foreach time ∈ tScheduled[] do
6   | if |time - timeToSchedule| < sTime then
7     | | return false
8   | end
9 end
10 return true

```

---



---

**Algorithm 3:** Main
 

---

```

1 Main(routes, sTime)
2 departureTime1 ← ConflictResolution(routes[], sTime)
3 routes[] ← list of routes[] in reversed order
4 departureTime2 ← ConflictResolution(routes[], sTime)
5 if departureTime1.objectiveValue() ≤ departureTime2.objectiveValue() then
6   | return departureTime
7 end
8 else
9   | return departureTime2
10 end

```

---

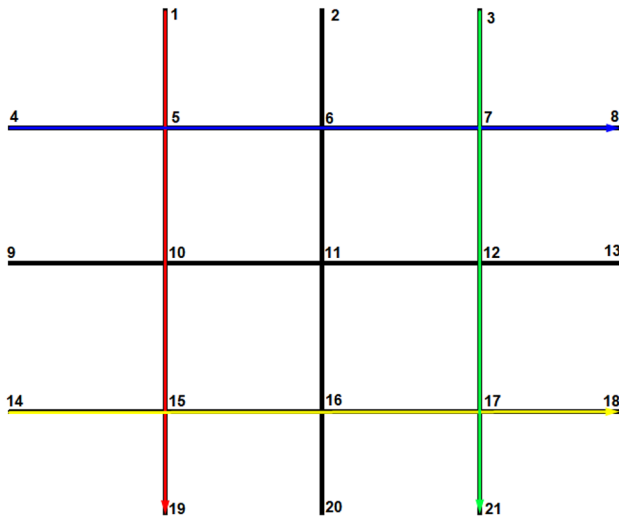


Fig. 2 Example of routes with conflicting intersections

In this scenario, the heuristic tries to schedule the routes from the first to the fourth vehicle, in this sequence. At the initial moment, all intersections are free so the times in  $\pi^1$  are allocated considering the maximum speed, which means 7 s to go along each road of its route. Hence, the time sequence for  $\pi^1$  is (0,7,14,21,28). While the first intersection of  $\pi^2$  is normally scheduled, a conflict appears in  $\pi^1(2) = \pi^2(2) = 5$  when the heuristic tries to schedule the second intersection because the safety constraint is not satisfied (line 26, Algorithm 1) since there is a vehicle there at the instant 7. Thus, the heuristic tries to schedule the next times until the maximum value allowed is achieved (line 23, Algorithm 1). However, the conflict persists in the instants 8 and 9.

As the heuristic was not able to schedule this intersection, all the previous times are removed and 1 time unit (1 s) is added to the initial time (lines 34–37, Algorithm 1). This process continues until the route is completed. In this case, the heuristic can only complete the route considering 5 as initial time and the final sequence of times for  $\pi^2$  is (5,12,19,26,33). If the same process is conducted with the other routes, several other conflicts will be identified. At the final, the sequence of times for  $\pi^3$  is (0,7,14,21,28), while for  $\pi^4$  is (5,12,19,26,33).

Therefore, the value of objective function will be the sum of the arrival times at the last intersection of each of the four routes, that is,  $28 + 33 + 28 + 33 = 122$  s. In this case, due to the symmetry of the map in this particular example, the values of the objective function are the same when considering the list of routes in its original and reverse orders, respectively.

## 4 Results

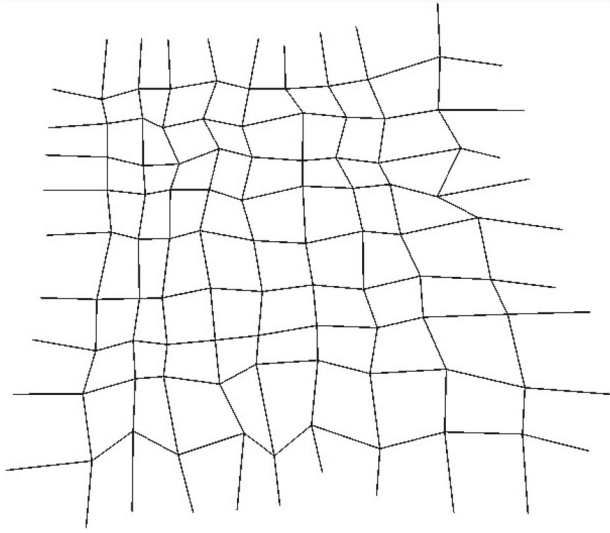
The simulations were conducted in a notebook Intel i3 with 4 Gb of RAM, and CPLEX was used to solve the model. This restricted configuration was important to show that the approach does not need special hardware to properly execute. The comparisons were conducted using the SUMO resources (traffic light control). The configuration files of these simulations are public and available<sup>1</sup> for replications and improvements of this approach.

### 4.1 SUMO setup

SUMO receives two files as input: the scenario configuration and the traffic demand. The scenario configuration files contain the representation of every road (arc) as a collection of lanes, including their positions, shapes, and speed limits; the connections between lanes at junctions (nodes); and the traffic lights referenced by junctions (optional feature). Our experiments used the same five scenarios in two different situations (with traffic lights and without traffic lights).

The traffic demand files contain the identification of each vehicle, together with its departure point, time of departure, route until the arrival point, and speed in each lane

<sup>1</sup> <http://ci.ufpb.br/docentes/claurton/datasetav/ds.html>.



**Fig. 3** Map as a grid  $9 \times 9$

that composes the route. According to the situation (with/without traffic lights), the files are specified as follows:

- For experiments with scenarios containing traffic lights, the speed of each vehicle always receives the speed limit of the lane where it is. Thus, we ensure that the arrival time is always as early as possible, considering the compulsory stops (traffic lights) along the routes;
- For experiments using the same scenarios but without traffic lights, our approach sets the departure times and speeds along the route for each vehicle so they avoid conflicts in the use of intersections.

For this second situation, the departure time can only be changed to a value later than the original time. Moreover, the routes of all vehicles in both situations, for the same scenario, are always the same. These restrictions ensure a fair analysis between experiments that compare these two situations using the same scenario.

## 4.2 Mathematical model

### 4.2.1 Case Study 1

The first case study considered a small artificial map that is, however, bigger than the maps used in other studies discussed in Sect. 2. To create several intersections, this map was configured as a  $9 \times 9$  grid (Fig. 3), which was deformed to create a heterogeneity among the roads and their lengths. The roads were also configured to have two lanes in opposite directions.

This map employed 36 vehicles whose routes start in an extreme node of the network and finish in the opposite extreme node. The simulation showed a reduction of 853 s,

**Table 1** Results of the simulation in the  $9 \times 9$  grid

Control	Sum of arrival times (s)	Total of CO <sub>2</sub> emission (mg/s)	Total of fuel consumption (ml/s)
Traffic lights	7786.00	29,268,179.50	12,568.23
Model	6933.00	2,198,4066.87	9433.00

**Fig. 4** Real map representing a district of a large city**Table 2** Results of the simulation in the district map

Control	Sum of arrival times (s)	Total of CO <sub>2</sub> emission (mg/s)	Total of fuel consumption (ml/s)
Traffic lights	3584.00	10,909,621.55	4680.50
Model	3091.00	11,074,781.90	4760.73

in the sum of the vehicles' arrival times, when our model, rather than the traffic lights strategy, controls the vehicles' behavior. Moreover, our model also reduced the fuel consumption and CO<sub>2</sub> emission in about 25% (Table 1).

#### 4.2.2 Case Study 2

The second case study used a real scenario that represents a district of a large city. This simulation had 1167 roads and 463 intersections, as illustrated in Fig. 4, and the execution time was 20 min. The simulation used 73 vehicles (routes) and, using this configuration, the model could almost return the optimal solution (maximum speed to all vehicles). This means, the model returned the sum of all arrival times just as 1% higher than its optimal value.

Table 2 presents the results for this simulation, where we can observe a reduction of 493 s in the sum of the arrival times when the model is employed. This represents an improvement of 14% in the average time that each vehicle spends in their journeys. However, the CO<sub>2</sub> and fuel consumption had a slight increase of about 1%.

## 4.3 Mathematical model versus heuristic

### 4.3.1 Prologue

These two previous case studies have used scenarios with less than 100 routes. As we could expect, the execution time for this problem increases according to the number of intersections and vehicles used in the scenarios. Thus, the use of a heuristic for more complex scenarios was necessary because the mathematical model was spending a long time to return a solution and, after a certain point, no solution was returned (last line in Table 3).

As a form to compare the two versions, i.e., mathematical model and heuristic, and understand their results regarding the use of different scenarios, we designed several maps to generate a high number of possible conflicts. The complexity of these scenarios was gradually increased (Table 3) so we could identify possible thresholds concerning the resolution of the model.

Table 3 shows (1) the number of intersections and routes (Inters/Routes) of each scenario, (2) the resultant sum of arrival times when only the model is used, (3) the time required to generate the plan using only the model, (4) the resultant sum of arrival times when the model is used together with the heuristic, (5) the time required to generate the plan when the model is used together with the heuristic, and (6) the percentage increase in the sum of arrival times when the heuristic is used.

These results show that both the model heuristic basically return the same results regarding the sum of arrival times of the vehicles. However, the important benefit of this heuristic is to significantly reduce the total execution time. The longest time was just 234 ms when the map presented 169 intersections and 104 routes (vehicles). The limit of the model was the scenario with 441 intersections and 168 routes, when the model was not able to return a solution. In this case, the heuristic could still return a solution in 228 ms of execution.

### 4.3.2 Case Study 3

As demonstrated in Table 3, the heuristic can quickly produce feasible solutions even in scenarios with more than 100 routes. Therefore, this case study used a real map (Fig. 5) that represents a second district of a large city and employs 125 routes to evaluate the heuristic.

As we can observe in Table 4, the heuristic obtained a reduction of about 7% in the sum of the arrival times, decreasing such time in 256 s. The conflicts were successfully managed and any collision was identified during the simulation. Moreover, the reduction of the required stops in intersections ensured improvements of about 20% in both fuel consumption and CO<sub>2</sub> emission.

### 4.3.3 Case Study 4

The next scenario (Fig. 6) creates a more complex situation to evaluate the scalability of the heuristic, using a real map of a third district with 200 routes. Using this higher number of routes, it was possible to generate more conflicts and stress the resolution

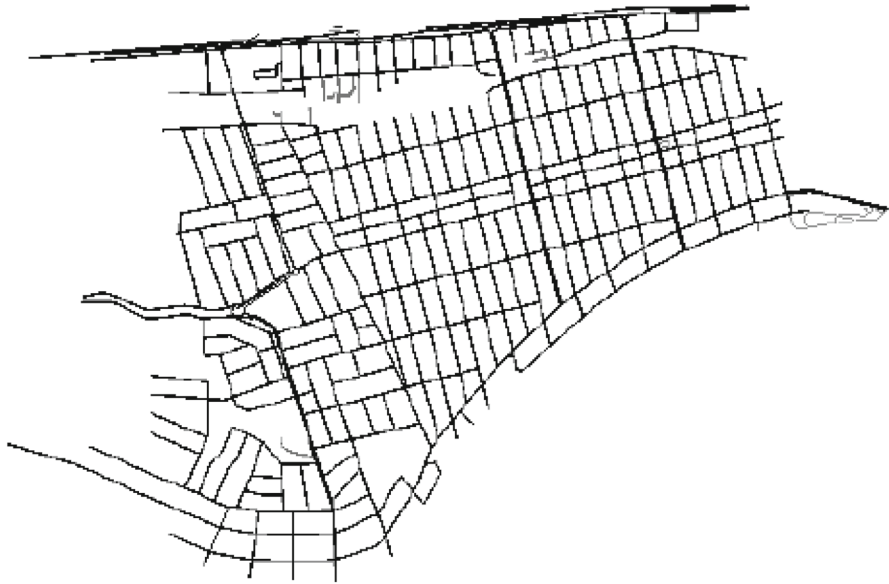
**Table 3** Comparison between the mathematical model and the heuristic algorithm

Inters/routes	Arrival times sum (model)	Execution time (model)	Arrival times sum (heuristic)	Execution time (heuristic) (ms)	Increase in arrival time sum (%)
25/20	1260	1.19 s	1260	34	0
25/40	2713	10 min	2812	63	1.52
36/24	1832	5 min	1856	48	1.31
36/48	3840	10 min s	4040	69	5.21
49/28	2596	0.7 s	2596	54	0
49/56	5461	10 min	5586	37	2.29
81/36	4412	4.42 s	4412	61	0
81/72	9174	10 min	9328	84	1.68
100/40	5448	0.13 s	5488	119	0.73
100/80	11,168	10 min	11,460	120	2.61
121/44	6708	6.7 s	6708	148	0
121/88	13,853	10 min	14,036	134	1.32
169/52	9484	10.88 s	9484	86	0
169/104	19,536	10 min	19,631	234	0.49
196/112	10,984	0.41 s	11,040	182	0.51
196/112	22,422	10 min s	11,040	182	0.51
225/60	12,740	11.19 s	12,740	152	0
225/120	26,133	10 min	26,237	133	0.40
256/64	14,472	0.47 s	14,536	86	0.44
256/168	29,476	10 min	29,757	154	0.95
289/68	16,476	16.07 s	16,476	124	0
289/136	33,712	10 min	33,807	224	0.28

Table 3 continued

Inters/routes	Arrival times sum (model)	Execution time (model)	Arrival times sum (heuristic)	Execution time (heuristic) (ms)	Increase in arrival time sum (%)
324/72	18,440	0.77 s	18,512	76	0.39
324/144	37,479	10 min	37,930	123	1.20
361/76	20,692	23.88 s	20,692	104	0
361/114	31,388	10 min	31,469	158	0.26
400/80	22,888	1.03 s	22,968	146	0.35
400/160	46,559	30 min	46,858	196	0.64
441/84	25,228	30.10 s	25,388	206	0.20
441/168	–	–	51,883	228	–





**Fig. 5** Real map representing a second district of a large city

**Table 4** Results of the simulation using a second district map

Control	Sum of arrival times (s)	Total of CO <sub>2</sub> emission (mg/s)	Total of fuel consumption (ml/s)
Traffic lights	3709.00	13,268,861.76	5709.65
Heuristic	3453.00	10,563,535.57	4541.80

of the model. For example, the heuristic had to apply several increases at the initial time of the routes to satisfy the constraints of the problem.

However, even considering such increases, the final results (Table 5) showed the advantage of the heuristic procedure, which provided a reduction of 7% regarding the sum of the arrival times when compared with the SUMO traffic lights approach. Moreover, the simulation showed a reduction of about 10% in the CO<sub>2</sub> emission and 11% in the fuel consumption.

#### 4.3.4 Case Study 5

The final evaluation of the heuristic was conducted in a fictitious map with 1024 intersections and about 4000 roads, which were deformed to ensure a higher level of aleatory regarding their lengths (Fig. 7).

This simulation had 256 routes and each route was composed of more than 60 intersections to be managed. Despite this significant number of intersections and routes, the execution time was about 300 ms. Thus, the heuristic algorithm could solve small and big problems, always presenting improvements regarding the traditional intersec-



**Fig. 6** Real map representing a third district of a large city

**Table 5** Results of the simulation using a third district map

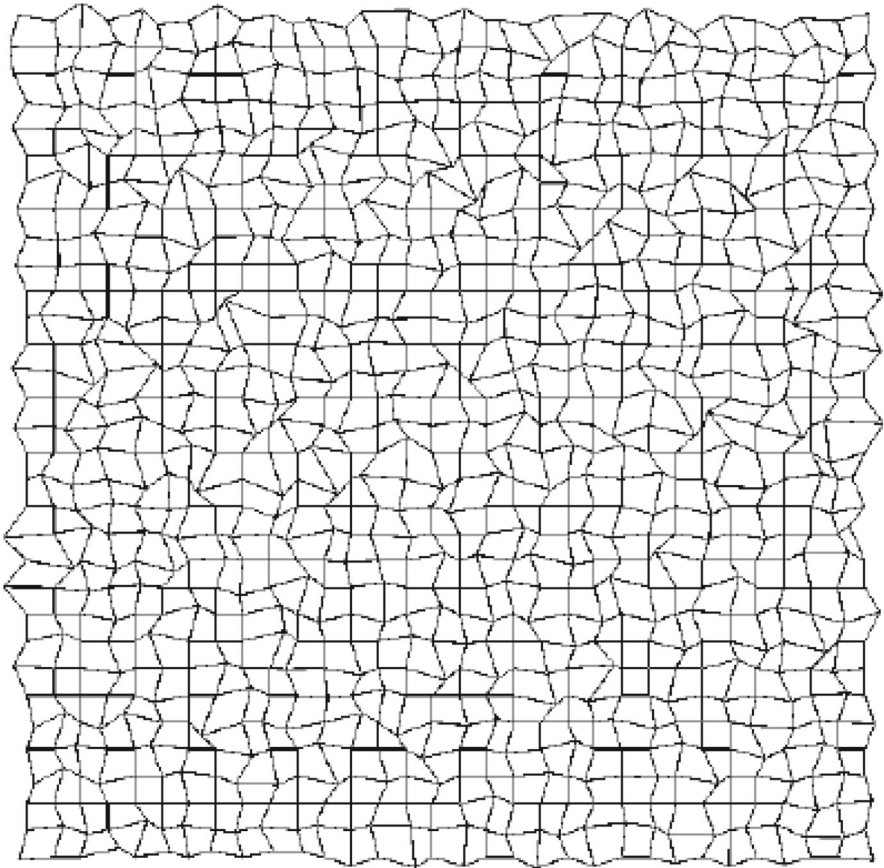
Control	Sum of arrival times (s)	Total of CO <sub>2</sub> emission (mg/s)	Total of fuel consumption (ml/s)
Traffic lights	9616.00	31,671,179.73	13,611.98
Heuristic	8937.00	25,217,755.41	10,835.47

tion management system. Table 6 shows that our approach still presents advantages regarding the sum of arrival times, fuel consumption, and CO<sub>2</sub> emission.

## 5 Conclusion

### 5.1 Main contributions

This paper presented a proposal for the optimization of intersections management in scenarios with autonomous vehicles. Its main contribution is the definition of a mathematical model and a heuristic algorithm, where the later is capable of dealing with complex scenarios, which are not discussed in the literature. While the studies in this area are focused only on one or a few intersections, we in turn have evaluated our approach from small to complex (1024 intersections) configurations. Even in such



**Fig. 7** Scenario with 1024 intersections and 4000 roads

**Table 6** Results of the simulation using a scenario with 1024 intersections

Control	Sum of arrival times (s)	Total of CO <sub>2</sub> emission (mg/s)	Total of fuel consumption (ml/s)
Traffic lights	140,163.00	450,411,233.93	194,003.43
Heuristic	130,491.00	332,133,716.66	142,772.07

more complex scenarios, the heuristic algorithm could find a solution in milliseconds, also considering specific constraints (safety interval) to ensure the safety of the system against collisions. Constraints were also used to limit the maximum speed of vehicles on each of the roads.

The simulations demonstrated that the heuristic can improve the traditional approaches for intersection management regardless of the scenarios complexity. However, the gain in simple scenarios is lower because the traffic lights approach always ensures the maximum speed of the vehicles until they are a few meters of the intersections. Differently, our model defines a constant speed for each lane that is part of a

route. This speed is always lower or equal than the lane speed limit, and it ensures that the vehicle does not need to stop in the intersections. Thus, the gain of the heuristic will be proportional to the number of intersections.

Apart from such a contribution, this study also considered two other metrics (fuel consumption and CO<sub>2</sub> emission) that are not usually discussed in other studies, but are important in the sustainability context. Finally, all the configurations used in our experiments are available and can be used as a benchmark to support future research replications and extensions.

## 5.2 Limitations

This proposal only considers static rather than real-time scenarios. This means, after the plan creation, other vehicles cannot join the simulation. The simulations also showed that the heuristic has low gains when the scenario presents few routes. Moreover, while the heuristic could return a solution for all scenarios, it cannot ensure that such a solution is optimal.

As far as we know, this is the first study that proposes a mathematical model and heuristic to deal with scenarios that involve hundreds of intersections. Thus, we only compared this approach with the implementation of the SUMO simulator, which uses the traffic lights to manage the intersections.

## 5.3 Research directions

The main research direction of this study is to extend its approach so it can be used in real-time scenarios. Currently, this approach can generate a solution in milliseconds for complex scenarios, using a basic computational platform. If a new vehicle asks to join the scenario, then the heuristic should run again to create a new scheduling that considers this vehicle. For example, the central controller knows exactly the current positions of all vehicles and could use these positions as the initial nodes of each vehicle. In this case, it just needs to run again the heuristic algorithm. However, this situation is not so easy and there are several issues involved in this approach. Their formalization is part of our future studies.

**Acknowledgements** Funding was provided by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Grant No. 001).

## References

- Medina, A., Van de Wouw, N., Nijmeijer, H.: Automation of a T-intersection using virtual platoons of cooperative autonomous vehicles. In: *IEEE 18th International Conference on Intelligent Transportation Systems*, pp. 1696–1701 (2015)
- Lamouik, I., Yahyaouy, A., Sabri, M.: Smart multi-agent traffic coordinator for autonomous vehicles at intersections. In: *International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, pp. 1–6 (2017)
- Wu, J., Perronnet, F., Abbas-Turki, A.: Cooperative vehicle-actuator system: a sequence-based framework of cooperative intersections management. *IET Intell. Transp. Syst.* **8**(4), 352–360 (2013)

- Kamal, M., Imura, J., Hayakawa, T., Ohata, A., Aihara, K.: A vehicle-intersection coordination scheme for smooth flows of traffic without using traffic lights. *IEEE Trans. Intell. Transp. Syst.* **16**(3), 1136–1147 (2014)
- Zhu, F., Ukusuri, S.: A linear programming formulation for autonomous intersection control within a dynamic traffic assignment and connected vehicle environment. *Transp. Res. Part C: Emerg. Technol.* **55**, 363–378 (2015)
- Gong, S., Shen, J., Du, L.: Constrained optimization and distributed computation based car following control of a connected and autonomous vehicle platoon. *Transp. Res. Part B: Methodol.* **94**, 314–334 (2016)
- Lopez, P.A., et al.: Microscopic traffic simulation using sumo. In: 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE (2018)
- Hausknecht, M., Au, T., Stone, P.: Autonomous intersection management: multi-intersection optimization. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4581–4586 (2011)
- Dresner, K., Stone, P.: Multiagent traffic management: a reservation-based intersection control mechanism. In: *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, vol. 2, pp. 530–537 (2014)
- Sharon, G., Stone, P.: A protocol for mixed autonomous and human-operated vehicles at intersections. In: *International Conference on Autonomous Agents and Multiagent Systems*, pp. 151–167 (2017)
- Budan, G., Hayatleh, K., Morrey, D., Ball, P., Shadbolt, P.: An analysis of vehicle-to-infrastructure communications for non-signalised intersection control under mixed driving behavior. *Analog Integr. Circuits Signal Process.* **95**(3), 415–422 (2018)
- PTV Group, Ptv vissim 7 user manual. PTV AG, 240 (2013)
- Zheng, Y., Jin, L., Jiang, Y., Wang, F., Guan, X., Ji, S., Xu, J.: Research on cooperative vehicle intersection control scheme without using traffic lights under the connected vehicles environment. *Adv. Mech. Eng.* **9**(8), 1–13 (2017)
- Ashtiani, F., Fayazi, S., Vahidi, A.: Multi-intersection traffic management for autonomous vehicles via distributed mixed integer linear programming. In: *Annual American Control Conference (ACC)*, pp. 6341–6346 (2018)
- Belkhouche, F.: Control of autonomous vehicles at an unsignalized intersection. *Am. Control Conf.* **2017**, 1340–1345 (2017)
- Campos, G., Falcone, P., Wymeersch, H., Hult, R., Sjöberg, J.: Cooperative receding horizon conflict resolution at traffic intersections. In: *53rd IEEE Conference on Decision and Control*, pp. 2932–2937 (2014)
- Makarem, L., Gillet, D.: Decentralized coordination of autonomous vehicles at intersections. In: *emphIFAC Proceedings Vol. 44, no. 1*, pp. 13046–13051 (2011)
- Lim, W., Lee, S., Sunwoo, M., Jo, K.: Hierarchical trajectory planning of an autonomous car based on the integration of a sampling and an optimization method. *IEEE Trans. Intell. Transp. Syst.* **19**(2), 613–626 (2018)
- Isele, D., Cosgun, A., Subramanian, K., Fujimura, K.: Navigating intersections with autonomous vehicles using deep reinforcement learning. *arXiv preprint arXiv:1705.01196* (2017)
- de Campos, G., Falcone, P., Hult, R., Wymeersch, H., Sjöberg, J.: Traffic coordination at road intersections: autonomous decision-making algorithms using model-based heuristics. *IEEE Intell. Transp. Syst. Mag.* **9**(1), 8–21 (2017)
- Fayazi, S., Vahidi, A., Luckow, A.: Optimal scheduling of autonomous vehicle arrivals at intelligent intersections via MILP. *Am Control Conf* **2017**, 4920–4925 (2017)