

Universidade Federal da Paraíba
Centro de Informática
Programa de Pós-Graduação em Informática

Gerenciamento de Interseções para Otimização do Tráfego de
Carros Autônomos

Victor Hugo

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Informática da Universidade Federal da Paraíba como parte dos requisitos necessários para obtenção do grau de Mestre em Informática.

Área de Concentração: Ciência da Computação
Linha de Pesquisa: Sistemas de Computação | Sinais, Sistemas Digitais
e Gráficos

Claurton Siebra (Orientador)
Anand Subramanian (Coorientador)

João Pessoa, Paraíba, Brasil
© Victor Hugo, 19 de Julho de 2019

Resumo

Interseções tornaram-se, devido ao rápido crescimento da população das cidades, um local propenso para acidentes automobilísticos, sendo também a principal causa de congestionamento do tráfego. Este trabalho apresenta um sistema virtual capaz de gerenciar e otimizar o tráfego de uma região em um ambiente de carros autônomos. Esta abordagem introduz restrições que garantem a segurança dos veículos autônomos, de modo que eles possam usufruir das vantagens oferecidas por essa nova tecnologia, sem perda do nível de confiança. Foi proposta a retirada de todos os semáforos e placas de sinalizações presentes nas interseções, atribuindo ao sistema a responsabilidade de gerenciar o tráfego. Os resultados utilizando o simulador SUMO demonstram que o sistema foi capaz de otimizar o tráfego dos veículos nas diversas regiões utilizadas como estudo de casos.

Palavras-Chave: Otimização, Carros Autônomos, Simulação Computacional.

Abstract

Intersections have become, due to the rapid growth of the population of the cities, a prone place for auto accidents and also the main cause of traffic congestion. This work presents a virtual system capable of managing and optimizing the traffic of a region in an environment of autonomous cars. This approach introduces constraints that ensure the safety of stand-alone vehicles, so that they can take advantage of the benefits offered by this new technology without losing confidence. We proposed to remove all traffic lights and signboards present at the intersections, giving the system the responsibility to manage traffic. The results using the SUMO simulator demonstrate that the system was able to optimize vehicle traffic in the various regions used as case studies.

Keywords: Optimization, Autonomous Cars, Computational Simulation.

Conteúdo

1	Introdução	1
1.1	Motivação	1
1.2	Problema da Pesquisa	3
1.3	Objetivos	5
1.3.1	Objetivo Geral	5
1.3.2	Objetivos Específicos	5
1.4	Estrutura da Dissertação	6
2	Fundamentação Teórica	7
2.1	Simulation of Urban Mobility (SUMO)	7
2.1.1	Características do Movimento	8
2.1.2	Relatórios	8
2.1.3	Emissões	9
2.2	Open Street Map (OSM)	9
2.3	Programação Linear Inteira	11
2.4	Heurística	12
3	Trabalhos Relacionados	13
3.1	Sistemas Centralizados	14
3.2	Sistemas Descentralizados	18
3.3	Considerações	21
3.3.1	Sistemas Centralizados	22
3.3.2	Sistemas Descentralizados	23
3.3.3	Análise	25

4	Metodologia	27
4.1	Sistema Virtual	27
4.1.1	Modelagem Matemática	28
4.2	Heurística	30
5	Resultados	36
5.1	Modelo Matemático	36
5.1.1	Estudo de Caso 1	36
5.1.2	Estudo de Caso 2	38
5.1.3	Discussão	39
5.2	Heurística	39
5.2.1	Comparação Modelo Matemático x Heurística	39
5.2.2	Estudo de Caso 3	41
5.2.3	Estudo de Caso 4	42
5.2.4	Estudo de caso 5	43
6	Conclusão	44
6.1	Contribuições	44
6.2	Limitações	45
6.3	Trabalhos Futuros	46
	Referências Bibliográficas	53
A	Passo a passo da resolução da Heurística	54

Lista de Símbolos

SUMO : *Simulation of Urban Mobility*

OSM : *Open Street Map*

PLI : *Programação Linear Inteira*

HVs : *Human Operated Vehicles*

CAVs : *Connected and Autonomous Vehicles*

HV : *Human Vehicle*

FIFO : *First In, First Out*

AIM : *Autonomous Intersection Management*

V2I : *Vehicle-to-Infrastructure*

V2V : *Vehicle-to-Vehicle*

Lista de Figuras

2.1	Página inicial do OpenStreetMap	10
2.2	Mapa gerado pelo SUMO	11
4.1	Exemplo de rota utilizada	29
4.2	Exemplo de rotas conflitantes	32
5.1	Região 9x9 em forma de grade	37
5.2	Região dos Bancários utilizada na simulação	38
5.3	Região do bairro Bessa	41
5.4	Região do conjunto dos bairros	42
5.5	Super Região: 1024 cruzamentos	43

Lista de Tabelas

4.1	Percentagem dos valores conseguidos com a heurística em relação aos resultados obtidos com o SUMO (1)	34
4.2	Percentagem dos valores conseguidos com a heurística em relação aos resultados obtidos com o SUMO (2)	34
4.3	Percentagem dos valores conseguidos com a heurística em relação aos resultados obtidos com o SUMO (3)	34
5.1	Descrição dos valores da simulação do mapa 9x9	37
5.2	Descrição dos valores da simulação do mapa Bancários	38
5.3	Descrição dos valores das comparações entre o Modelo e a Heurística	40
5.4	Descrição dos valores da simulação do mapa Bessa	41
5.5	Descrição dos valores da simulação da região dos bairros	42
5.6	Descrição dos valores da região com 1024 cruzamentos	43

Capítulo 1

Introdução

Este capítulo é destinado a descrever os principais aspectos que motivaram a realização deste trabalho (Seção 1.1), bem como o problema de pesquisa tratado nesta dissertação (Seção 1.2), os objetivos do trabalho em questão (Seção 1.3) e, por fim, a organização dos demais capítulos deste documento (Seção 1.4).

1.1 Motivação

Um carro autônomo é um veículo que tem a capacidade de dirigir sozinho sem intervenção humana. Existem cinco funções básicas que impulsionam o carro autônomo: percepção, localização, planejamento, controle e gerenciamento do sistema [Jo et al. 2014]. Em outras palavras, é um carro que navega da posição atual (início), para a posição de destino com base no entendimento do ambiente e na direção do veículo [Lim et al. 2018]. Enquanto os veículos autônomos hoje são meros demonstradores das capacidades tecnológicas e realizações dos fabricantes de automóveis e universidades, eles provavelmente penetrarão no mercado em larga escala no futuro [Hult et al. 2015]. É importante notar que esse conceito não está em um futuro muito distante de ser aplicado no mundo real. Conforme previsto em algumas publicações, os sinais de trânsito serão eliminados e 75% dos veículos serão autônomos até o ano de 2040 [Zhu e Ukkusuri 2015].

Um subconjunto das pesquisas realizadas com carros autônomos está direcionado a sistemas para coordenar o tráfego em interseções. Atualmente, semáforos e sinais de parada ou sinais de prioridades auxiliam os motoristas humanos a atravessar interseções com segu-

rança [Makarem e Gillet 2012]. Esses métodos exigem que os veículos fiquem parados e esperem pela sua vez de cruzar a interseção ou entrar na rotatória, obstruindo o fluxo de veículos, que por sua vez limita o rendimento da interseção [Medina, Wouw e Nijmeijer 2015], tornando-se assim também a principal causa do congestionamento do tráfego, fazendo com que os motoristas se sintam estressados e frustrados enquanto aguardam à espera de sinais vermelhos ou sinais de parada [Lamouik, Yahyaouy e Sabri 2017], [Wu, Guizani e Hsieh 2016]. Os congestionamentos também levam ao desperdício de uma enorme quantidade de tempo, combustível e dinheiro, e o custo total devido ao congestionamento do tráfego está aumentando a cada ano [Kamal, Imura e Ohata 2013].

Embora diferentes fatores contribuam para colisões de veículos, como problemas mecânicos e mau tempo, o comportamento do motorista é considerado a principal causa de mais de 90% de todos os acidentes devido à distração humana e/ou erro de julgamento [Zohdy e Rakha 2012]. Interseções tornaram-se, devido ao rápido crescimento da população das cidades, um local propenso para acidentes automobilísticos, representando um perigo potencial para os usuários da via, uma vez que são os locais em que as trajetórias de veículos se cruzam [Medina, Wouw e Nijmeijer 2015]. Existem dois principais problemas que afetam a segurança do tráfego em cruzamentos. Um deles é a abordagem de visualizar apenas a interseção como um ponto único, sem pensar na região como um todo, de modo que os motoristas não possam avançar quando deveriam por conta da solução utilizada. O outro é a sinalização de trânsito inadequada em áreas com uma série contínua de cruzamentos [Milanés et al. 2012].

Cenários em que todos os veículos são autônomos e comunicantes oferecem a possibilidade da remoção dos atuais mecanismos de coordenação de trânsito, prevendo um aumento no rendimento nas interseções, minimizando o tempo que os veículos ficam parados. Em particular, as tecnologias de veículos conectados, incluindo veículo-para-veículo (V2V), veículo-para-infra-estrutura (V2I) e veículo para outras instalações digitais (V2X), permitem que veículos individuais acessem informações de tráfego on-line, visualize com precisão as condições de trânsito nas proximidades e comuniquem-se com outros veículos para estabelecer uma condução coordenada ou cooperativa [Gong, Shen e Du 2016]. Com a utilização de cenários compostos apenas por carros autônomos, é previsto uma melhora no comportamento comparado com cenários que possuem apenas condutores humanos, permitindo assim lidar com mais tráfego, menores atrasos, menos poluição e, certamente, melhor conforto para

o condutor e para os passageiros. No entanto, a complexidade do controle distribuído de centenas de milhares de carros não pode ser feita de modo simples.

Diante do que foi contextualizado, o objetivo da presente dissertação de mestrado é propor um sistema virtual capaz de otimizar o tráfego dos veículos autônomos de uma região, utilizando Programação Linear Inteira, levando em consideração o problema do gerenciamento em interseções, de modo a minimizar o tempo final de chegada de todos os carros. Foi proposto a retirada de cada semáforo ou placas de trânsito nas interseções, tornando responsabilidade do sistema fazer o gerenciamento das rotas dos veículos. Antes de começar sua viagem, cada veículo comunica sua rota ao sistema que agenda os tempos dos veículos cadastrados. Cada veículo recebe como resposta os tempos de chegada previstos para atravessar cada interseção presente na rota. Simulações utilizando o SUMO (<http://sumo.dlr.de/>) demonstraram a redução dos tempos de chegada em relação aos sistemas tradicionais de controle do tráfego de veículos, além da redução do consumo de combustível e emissão de CO₂.

1.2 Problema da Pesquisa

O controle de interseção tem um papel importante na gestão do tráfego urbano para garantir a segurança, um fluxo elevado, e para evitar o congestionamento, podendo ser visto como um recurso compartilhado onde o acesso exclusivo é dado por um período de tempo limitado. O controle fornece informações uniformes a todos os usuários presentes nas interseções, de forma a ser uma das principais razões para o sucesso no controle do tráfego. Porém, devido à importância crescente das preocupações ambientais e dos limitados recursos econômicos e físicos, os sistemas tradicionais de controle de tráfego não podem continuar a ser resposta às crescentes necessidades de transporte e mobilidade das sociedades modernas [Wu, Perronet e Abbas-Turki 2014]. Nota-se que o controle *stop-and-go* em uma interseção tem sido um paradigma dominante. Nas interseções controladas por semáforos, os veículos em cada aproximação que recebem sinais verdes são assegurados pela passagem segura da interseção, mas com os inconvenientes adicionais de paradas frequentes e em marcha lenta até que o direito de ir seja obtido [Lee e Park 2012]. Nesse sentido, em um cenário onde apenas carros autônomos são considerados, propomos um sistema virtual capaz de otimizar o tráfego de

uma região utilizando os benefícios presentes nos carros autônomos, apresentando oportunidades para o desenvolvimento de redes rodoviárias mais seguras, limpas e eficientes [Sharon e Stone 2017].

Nos cenários utilizados, todos os semáforos foram retirados, assim como qualquer forma de sinalização presentes em cruzamentos, tornando-se em um dos problemas mais desafiadores para veículos autônomos, lidar com interseções não sinalizadas em ambientes urbanos [Isele et al. 2017]. Um dos desafios nessa área de pesquisa é encontrar métodos de coordenação que melhorem o desempenho dos veículos nas interseções. Geralmente, existem duas abordagens diferentes para resolver esse problema. Uma abordagem é projetar um controlador centralizado para todo o sistema de veículos autônomos, ou seja, sistemas onde existe um controlador único que gerencia todas as tomadas de decisões. E o descentralizado, no qual os próprios carros fazem o gerenciamento e a troca de informações utilizando técnicas de coordenação entre si [Makarem e Gillet 2012]. Se as interseções forem controladas por um desses algoritmos cooperativos autônomos que possam otimizar a sequência de cruzamento para todos os veículos próximos, a interseção deve ser mais segura, pois nenhum fator humano causaria acidentes. A interseção também deve ser mais eficiente em termos de tempo e consumo de energia, já que o algoritmo decidirá a ordem de cruzamento e a velocidade dos veículos para obter a máxima eficiência. Aspectos como minimizar a desaceleração, respectivamente a aceleração, bem como reduzir o tempo total para o cruzamento de interseção, poderiam ser levados em consideração para o cálculo da sequência ótima de cruzamento [Riegger et al. 2016].

Para demonstrar a eficiência dos carros autônomos com o sistema proposto, foram utilizados mapas reais dos bairros de João Pessoa obtidos do *OpenStreetMap* (OSM). Nestes mapas foi observado que o desempenho dos carros foi melhorado, acarretando uma redução no tempo do percurso percorrido. Consequentemente houve um ganho no tempo que cada carro levou para finalizar sua rota, como também, uma diminuição no tempo total da viagem de todos os carros. Nesse contexto, o sistema demonstrou ser superior às abordagens atuais de controle de tráfego, contribuindo para o avanço do desenvolvimento de um trânsito mais eficiente.

Apesar do sistema ser proposto para fazer o gerenciamento de veículos autônomos em uma determinada região, o sistema pode ser adaptado para realizar outros gerenciamentos,

como por exemplo, o sistema de transporte subterrâneo de alta performance proposto por Elon Musk [The Guardian] e [The Washington Post] . O sistema tem como proposta a entrada de um carro autônomo em um ponto da cidade no qual daria acesso a rede subterrânea, dessa maneira seria possível trafegar pela rede até o ponto desejado. O sistema proposto terá várias redes que se conectam, que caso não haja o devido gerenciamento, vários conflitos poderão acontecer. Se for levado em consideração a entrada do sistema como um carro autônomo sinalizando o tempo de saída, e indicando um ponto de chegada, seria possível descobrir os pontos pelos quais o veículo passará, fazer o devido gerenciamento, e solucionar os conflitos que possam existir com a devida modificação do sistema.

1.3 Objetivos

1.3.1 Objetivo Geral

O objetivo do presente trabalho é apresentar um sistema virtual capaz de gerenciar o tráfego de uma região utilizando otimização e gerenciamento dos veículos em interseções. Para isso, apenas carros autônomos são levados em consideração.

1.3.2 Objetivos Específicos

Os objetivos específicos deste trabalho são:

- Desenvolver um modelo matemático para resolução do problema dos conflitos e agendamentos dos veículos autônomos em cruzamentos.
- Validar o modelo desenvolvido utilizando Programação Linear Inteira;
- Validar os resultados obtidos;
- Desenvolver e avaliar uma heurística que permita a escalabilidade do sistema em termos do número de rotas e cruzamentos.
- Validar a heurística utilizando mapas reais.

1.4 Estrutura da Dissertação

O restante desta dissertação está estruturada da seguinte forma. O Capítulo 2 apresenta a fundamentação teórica necessária para a compreensão do problema abordado neste trabalho de mestrado. O Capítulo 3 apresenta algumas das propostas dos trabalhos relacionados presentes na literatura. No Capítulo 4 é apresentada a proposta do sistema desenvolvido, assim como o método utilizado para validar o trabalho. O Capítulo 5 discute os resultados obtidos nos experimentos utilizando o simulador. O Capítulo 6 aborda a conclusão do presente trabalho.

Capítulo 2

Fundamentação Teórica

Este capítulo apresenta uma breve fundamentação dos assuntos necessários para a compreensão do problema abordado nesta dissertação. A Seção 2.1 apresenta uma visão geral sobre o simulador utilizado para obter os resultados. A Seção 2.2 aborda como foram obtidos os mapas utilizados. A Seção 2.3 apresenta como foi a otimização utilizada para gerenciar os tempos de todas as rotas. A Seção 2.4 apresenta a solução utilizada para resolução dos conflitos e agendamentos em áreas com grandes volumes de rotas e cruzamentos.

2.1 Simulation of Urban Mobility (SUMO)

Simulações de tráfego facilitam a avaliação de mudanças na infraestrutura, bem como mudanças políticas antes de implementá-las na estrada. Por exemplo, a eficácia de zonas ambientais ou algoritmos de controle para semáforos podem ser testados e otimizados em uma simulação antes de serem implantados no mundo real.

O SUMO é uma suíte de simulação de tráfego livre e aberta que está disponível desde 2001. O SUMO permite a modelagem de sistemas de tráfego intermodal, incluindo veículos rodoviários, transporte público e pedestres. Incluído no SUMO está uma riqueza de ferramentas de suporte que lidam com tarefas como localização de rotas, visualização, importação de rede e cálculo de emissões. O SUMO pode ser aprimorado com modelos personalizados, fornecendo também o uso de várias APIs para controlar remotamente a simulação (<http://sumo.dlr.de/>).

A utilização do SUMO se dá através de linhas de comando. Instalar o programa, simular

um veículo para percorrer uma via, configurar uma malha viária, entre outros, dependem destas linhas de comando. As entradas e saídas do programa são realizadas diretamente em arquivos XML. O SUMO é implementado em C++ e usa apenas bibliotecas portáteis.

2.1.1 Características do Movimento

Apesar do simulador utilizar valores para a aceleração do veículo durante a rota desejada, nas simulações feitas foram utilizados valores constantes de velocidade em cada rota para atender as especificações do modelo desenvolvido (Subseção 4.1.1). Em outras palavras, em cada rua a velocidade não é modificada e segue exatamente o que foi especificado pelo modelo. Dessa forma, é possível agendar todos os tempos de acordo com a resolução das rotas e velocidades desejadas. Para que a simulação do SUMO não ficasse em desvantagem perante o modelo, foi modificado a velocidade máxima das ruas com o valor máximo que o modelo poderia agendar há um determinado carro, dessa forma, os carros controlados pelo SUMO poderia atingir a velocidade máxima da via, assim como o modelo é capaz. Como a velocidade no modelo precisa ser constante, nas simulações do SUMO foi especificado aos veículos a saída do ponto inicial como a velocidade máxima permitida pela rua. Em outras palavras, o carro não sai do instante inicial com a velocidade zero, mas sim com a especificada pelo modelo para a rua.

Para controlar as informações de velocidade no SUMO foi utilizado a opção do *variableSpeedSign* através de um arquivo de definições que é utilizado por simulação, possibilitando a especificação que no momento x, a velocidade na rua y será z.

2.1.2 Relatórios

O SUMO permite gerar uma quantidade considerável de resultados. Estes valores são escritos em arquivos texto ou através de *sockets* para os usuários. Os relatórios mais comuns, como as posições dos veículos, informações de viagens, rotas de veículos e estatísticas, estão disponíveis através de recursos na linha de comando. Os demais relatórios devem ser definidos através de arquivos adicionais, tais como, informações baseadas nos veículos, detectores simulados, valores para arestas ou faixas, estatísticas sobre o estado corrente da simulação e relatórios de semáforos [Hallmann 2011].

Nas simulações foram utilizadas as opções *emission-output* para gerar as informações referentes às emissões dos gases poluentes emitidos, e a opção *tripinfo-output* para obter os valores dos tempos levados por cada veículo para completar a rota previamente especificada, assim como o consumo de combustível durante esse tempo.

2.1.3 Emissões

O projeto COLOMBO desenvolveu um conjunto de modernas aplicações cooperativas de vigilância e controle de tráfego que visam diferentes objetivos relacionados a transporte, como o aumento da mobilidade, eficiência de recursos e respeito ao meio ambiente. O projeto é baseado em modelos de simulações que permitem comparar o desempenho das aplicações. Para obter medições de emissão precisas a partir das simulações, o projeto COLOMBO acoplou ao simulador SUMO o modelo microscópico de emissão de veículos PHEMlight [Blokpoel et al. 2015].

Para obter as informações através da opção *emission-output* é necessário especificar o tipo de projeto que será utilizado para calcular as emissões. No presente trabalho, foi utilizado do PHEMLight o modelo PKW_D_EU4, o qual permitiu o cálculo das emissões das frotas dos veículos com base nas trajetórias e velocidades simuladas por modelos de tráfego integrados ao simulador.

2.2 Open Street Map (OSM)

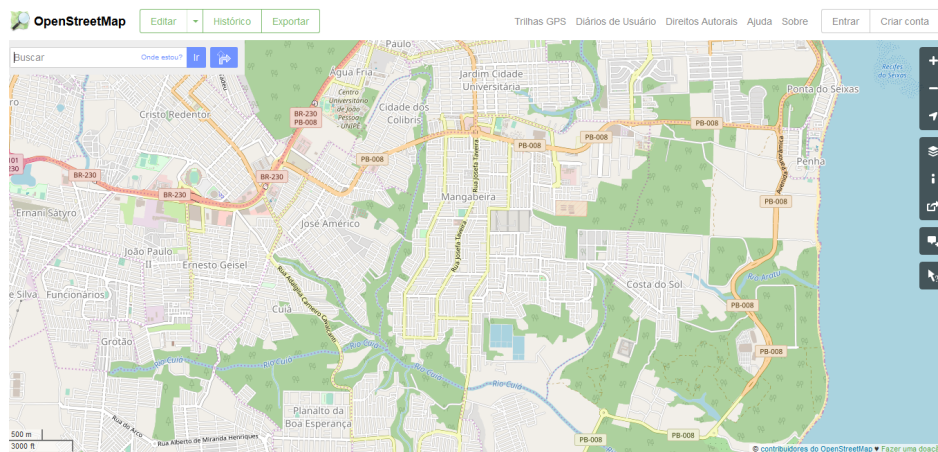
O projeto OSM, fundado em 2004 na *University College London*, tem como objetivo criar um banco de dados gratuito com informações geográficas de todo o mundo. Uma infinidade de dados espaciais, como estradas, prédios, áreas de uso da terra ou pontos de interesse, é inserida no banco de dados do projeto. Semelhante a outros projetos baseados na comunidade na Internet, qualquer usuário pode começar a contribuir para o projeto e editar os dados após um breve registro on-line [Neis e Zipf 2012].

As informações geográficas no banco de dados do OSM, como estradas, informações sobre o uso da terra ou edifícios, são armazenadas usando três tipos de objeto: Nós, Caminhos e Relações. Um “Nó” no banco de dados contém informações de localização de um ponto na forma de coordenadas de latitude e longitude. Linhas como estradas e polígonos são ar-

mazenadas como “Relações” e definem conexões lógicas ou geográficas entre os objetos. Cada objeto contém informações adicionais, como um número de versão, um ID, o nome do editor, a data em que foi criado ou modificado pela última vez e, é claro, outros atributos, os chamados pares Tag / Valor.

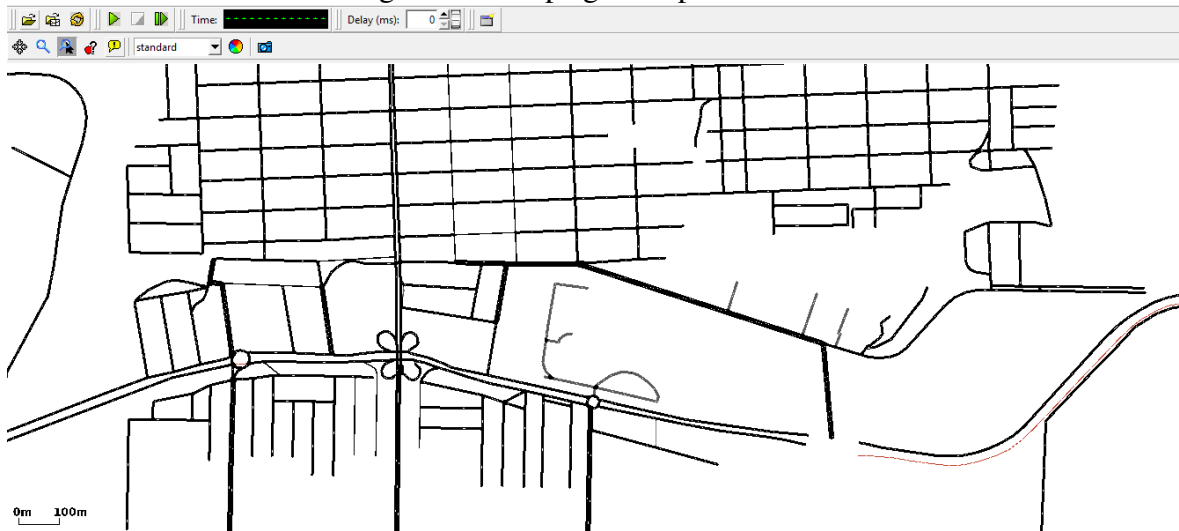
À medida que os usuários arrastam o mapa, a extensão visível é atualizada e novos blocos de mapa são solicitados em segundo plano sem recarregar toda a página HTML. Uma função de pesquisa (implementada como um serviço da Web externo) permite que os usuários encontrem rapidamente cidades, aldeias ou outros pontos de interesse no banco de dados [Haklay e Weber 2008]. A Figura 2.1 exemplifica a página inicial do OpenStreetMap.

Figura 2.1: Página inicial do OpenStreetMap



Uma vez definida a região no qual se irá trabalhar, é necessário apenas clicar em exportar e o OSM irá gerar um arquivo *.osm* em formato xml com as informações sobre o local. Com o arquivo é possível importar e utilizar no SUMO como pode ser visto na Figura 2.2.

Figura 2.2: Mapa gerado pelo SUMO



2.3 Programação Linear Inteira

Problemas de otimização discreta aparecem em profusão na vida real e aplicações ocorrem nas mais diversas áreas, tais como energia, transportes, telecomunicações, circuitos eletrônicos, medicina e saúde, criptografia, entre outros. Variáveis discretas importantes nessas aplicações envolvem, por exemplo, decidir se um produto é fabricado ou não em um período, escolher a melhor sequência de itens a serem processados em uma máquina, escolher o melhor local a instalar uma nova fábrica, determinar as melhores rotas de distribuição de produtos, assim como as melhores rotas a serem seguidas no trânsito [Arenales et al. 2007].

Um problema de Programação Linear Inteira é um problema de Programação Linear em que todas ou alguma(s) das suas variáveis são discretas (têm de assumir valores inteiros). Quando todas as variáveis estão sujeitas à condição de integralidade estamos perante um problema de Programação Linear Inteira Pura; e se apenas algumas o estão, trata-se de um problema de Programação Linear Inteira Mista [Alves e Delgado 1997]. Os modelos de PLI estão sujeitos a restrições adicionais indicando que algumas ou todas as variáveis são discretas, conforme se pode ver no exemplo seguinte:

$$z = \max \mathbf{c}\mathbf{x}$$

$$\mathbf{A}\mathbf{x} \leq \mathbf{b}$$

$$\mathbf{x} \in \mathbb{Z}_+^n$$

em que $\mathbf{A}$ representa uma matriz $(m \times n)$, $\mathbf{c}$, um vetor $(1 \times n)$, o vetor de variáveis é representado por $\mathbf{x}$, e $\mathbf{b}$ um vetor $(m \times 1)$ e Z_+^n representa o espaço dos vetores com n componentes inteiras não-negativas.

2.4 Heurística

Métodos heurísticos são algoritmos exploratórios que buscam resolver problemas. Alguns dos métodos envolvem a implementação de um algoritmo computacional sobre um conhecimento especializado de forma a resolver um problema pontual para o qual foi desenvolvido. Nos métodos empregados, a solução ótima do problema nem sempre é o alvo das heurísticas, tendo como ponto de partida uma solução viável, baseando-se em aproximações direcionadas a um ponto lógico. Em outras palavras, os métodos heurísticos são uma busca contínua e empírica, com possíveis várias ótimas resposta para o problema, cujo resultado é o melhor que se pode encontrar sob determinadas condições.

A heurística utilizada nesse trabalho é encontrar uma solução (não necessariamente a ótima) para a resolução dos conflitos de áreas com um grande volume de rotas e cruzamentos. Para resolver os conflitos, a heurística segue as condições do modelo matemático utilizado (Subseção 4.1.1) garantindo que não haja conflitos nas rotas utilizadas, assim como nenhum excesso na velocidade limite imposta por cada rua.

Capítulo 3

Trabalhos Relacionados

Um dos desafios presente nessa área de pesquisa é encontrar métodos de coordenação que melhorem o desempenho dos veículos nas interseções. Geralmente, existem duas abordagens diferentes para se resolver o problema. Uma abordagem é representada pelos sistemas centralizados, os quais utilizam um controlador central para coordenar todo o tráfego de veículos (Seção 3.1), semelhante por exemplo ao semáforo existente atualmente. Dessa forma, prevemos que os veículos e um controlador de interseção possam cooperativamente trabalhar juntos para melhorar as operações de tráfego em uma interseção quando todos os veículos estiverem totalmente automatizados [Lee e Park 2012]. Além disso, alguns trabalhos conseguem otimizar o movimento de maneira segura e eficiente, sem comprometer a qualidade da experiência da perspectiva dos passageiros [Dai et al. 2016].

A outra abordagem utilizada são os sistemas descentralizados, os quais coordenam seus veículos utilizando outras técnicas (Seção 3.2). O objetivo da tecnologia é a realização de um carro autônomo que possa dirigir sem colisões, percebendo o ambiente ao redor, com uma navegação segura, planejando a trajetória ideal para evitar os obstáculos dentro dos limites físicos do veículo [Jo et al. 2015]. O potencial total dos veículos autônomos pode ser, no entanto, explorado se todos os veículos rodoviários forem autônomos, suas trajetórias forem otimizadas usando critérios adequados e (a fim de alcançar a otimização) eles compartilhem algumas informações [Charalampidis e Gillet 2014].

3.1 Sistemas Centralizados

As pesquisas em [Dresner e Stone 2008] propuseram um novo protocolo de controle para interseções denominado Gerenciamento de Interseção Autônomo, do inglês, *Autonomous Intersection Management* (AIM). O AIM obteve reduções significativas no atraso do tráfego quando comparado aos sinais de trânsito tradicionais. O protocolo AIM define dois tipos de agentes autônomos: gerenciadores de interseção (um por interseção) e agentes controladores (um por veículo). Os gerenciadores de interseção são responsáveis por dirigir os veículos pelas interseções, enquanto os agentes controladores são responsáveis por controlar o CAV (*Connected and Autonomous Vehicles*) ao qual estão atribuídos. Esse controle é baseado no mecanismo de reserva, atribuindo uma reserva de espaço e tempo a cada veículo que é controlado por um agente controlador. A política de controle na interseção é baseada em FIFO (*First In, First Out*). Agentes dos veículos autônomos informam ao gerenciador da interseção sua ida através de uma mensagem de *REQUEST*, a qual inclui a direção pretendida do agente do veículo autônomo e estimativas de seu tempo e velocidade de chegada. O gerente de interseção usa essas informações, juntamente com uma política de controle para decidir se deve conceder a reserva. Para conceder, ele responde com uma mensagem de *CONFIRM* contendo restrições que o veículo deve obedecer. O gerente de interseção também pode usar essas restrições para fazer uma nova oferta caso o pedido não possa ser obedecido. Para rejeitar a solicitação, o gerente de interseção responde com uma mensagem *REJECT*. Nenhum veículo pode entrar na interseção sob nenhuma circunstância sem uma reserva. Um veículo com reserva é garantida a passagem segura, desde que atravesse de acordo com os requisitos informados. Se o agente motorista não puder cumprir a reserva, ele enviará uma mensagem *CANCEL*.

Mais do que apenas aplicar atribuição de tráfego dinâmico a interseções baseadas em AIM, [Hausknecht, Au e Stone 2011] consideraram as propriedades exclusivas do AIM que permitem o controle dinâmico, de baixa granularidade, de veículos autônomos por meio de uma grade de interseções. Especificamente, foi demonstrado como os veículos individuais podem conseguir as informações já disponíveis através do protocolo AIM para prever os tempos de viagem ao longo de várias rotas e alterar dinamicamente seus caminhos planejados. Os autores assumiram que cada carro é um agente autônomo interessado em minimizar

seu próprio tempo de viagem individual, e modificaram o protocolo AIM básico, exigindo que os veículos enviem uma mensagem de notificação antes de solicitar uma reserva.

No trabalho de [Zhu e Ukkusuri 2015] foi proposto um novo modelo de fluxo de tráfego baseado em faixa. Para explicar os movimentos de veículos livres de conflitos em interseções autônomas, os autores introduziram a restrição de complementaridade ao modelo, resultando em uma formulação não-linear e em dois níveis. Em seguida, a formulação de otimização de dois níveis é transformada na versão de programação linear (LPAIC), relaxando as restrições não-lineares com um conjunto de equações lineares. Em cada interseção será instalada um controlador distribuído que é capaz de acessar as informações protegidas por privacidade (por exemplo, velocidade, posição, destino) da aproximação dos veículos conectados através da comunicação V2I. O controlador então aplica a formulação LPAIC para obter soluções de fluxo de tráfego. Em seguida, o controlador comunica os tempos programados de volta aos veículos conectados próximos, que ajustarão a velocidade e obedecerão ao cronograma especificado. Finalmente, os veículos conectados passam pela interseção com segurança e eficiência, sem a necessidade de semáforos.

Os autores em [Sharon e Stone 2017] apresentam um gerenciador autônomo de interseção híbrido, do inglês, *Hybrid Autonomous Intersection Management* (H-AIM). O H-AIM é capaz de detectar veículos que se aproximam, tornando necessário apenas comunicação V2I. Se nenhum HV for observado em uma dada faixa, presume-se que as trajetórias originadas daquela faixa não serão ocupadas pelas HVs, permitindo maior flexibilidade dos CAVs na obtenção de reservas. Foi utilizado uma interseção com 4 ruas, onde cada rua possui 3 faixas, permitindo 3 possibilidades de manobras aos veículos (seguir reto, dobrar a direita ou a esquerda). Supõe-se que o tempo do sinal seja conhecido, de modo que o protocolo seja capaz de prever antecipadamente as trajetórias verdes. Quando isso acontece, é introduzido o termo trajetórias verdes ativas do inglês, *active green trajectories*, indicando por qual faixa o carro irá prosseguir, possibilitando a passagem dos CAVs em sinal vermelho quando não é detectada nenhuma colisão, o que é permitido fazer em alguns países, como o Estados Unidos da América por exemplo.

As pesquisas em [Milanés et al. 2012] apresentam um controlador fuzzy automático para gerenciar os veículos que atravessam a interseção, permitindo que um veículo ceda a outro que tenha o direito de passagem ou cruze se a velocidade do veículo que chega

permitir. Uma rede *wireless* é utilizada para comunicar as informações necessárias para avaliar as decisões *fuzzy*. O controlador desenvolvido será responsável pelo gerenciamento dos pedais do acelerador e do freio (ou seja, o controle longitudinal) na tomada de decisões sobre redução, manutenção ou aumento da velocidade do veículo autônomo quando os dois carros estiverem se aproximando do cruzamento.

A abordagem proposta em [Kamal, Imura e Ohata 2013] e [Kamal et al. 2015] é a coordenação global, considerando os estados de todos os veículos juntos, com base na prevenção de seus riscos de colisões cruzadas projetados em torno da interseção. Para simplificar, foi assumido que todos os veículos têm forma idêntica e cruzam a intersecção na velocidade especificada em condições muito ideais. Uma unidade de coordenação de interseção, do inglês, *Intersection Coordination Unit* (ICU), que supostamente é instalada na interseção, usa comunicação bidirecional para receber informações básicas de direção dos veículos próximos, enviando assim orientações para eles depois de calcular suas entradas de controle. Neste cálculo, considerando os estados de todos os veículos, um problema de otimização não-linear restrito que inclui uma função de risco é resolvido em um modelo de estrutura de controle preditivo para permitir que eles cruzem a interseção com segurança sem muita redução em suas velocidades.

Os autores em [Fayazi, Vahidi e Luckow 2017] formalizaram o gerenciamento de interseção como um problema de otimização, no qual o principal objetivo é encontrar a sequência ótima e o tempo de chegada (*taccess*) para cada veículo. Para tal finalidade, foi utilizada Programação Linear Inteira Mista, do inglês, *Mixed Integer Linear Program* (MILP). Os autores consideraram a interação entre veículos em microsimulações, garantindo que possam atender o tempo de chegada programado por um algoritmo de rastreamento robusto. O gerenciador de interseção reside em um servidor computacional e recebe informações de todos os veículos que se aproximam e, em seguida, otimiza e programa regularmente o tempo de acesso da interseção para cada veículo. Os tempos de acesso agendados (*arrival-times*) são enviados para todos os veículos. Dessa forma, eles podem ajustar suas velocidades de acordo com o especificado. Para realizar a otimização, os autores levaram em considerações três tempos: tempo em que a frente do veículo entra na área de interseção; tempo em que a traseira do veículo deixa a área de interseção e o tempo em que a frente do veículo atinge uma distância de acesso até a chegada na interseção. Em um trabalho posterior [Ashtiani, Fayazi

e Vahidi 2018] também foi utilizada a mesma abordagem, com o diferencial de que os gerenciadores de interseção processam informações dos veículos vizinhos otimizando o acesso dos mesmos através da utilização de MILP, dessa forma, comunicando a decisão tomada aos veículos.

Os autores em [Pourmehrab et al. 2017] desenvolveram um algoritmo de otimização baseado em trajetória aplicável a uma interseção isolada em tráfego misto. O algoritmo otimiza o controle do sinal e as trajetórias dos AVs (*Automated Vehicles*), assumindo que não há mudança de faixa além do ponto de comunicação inicial do veículo e do controlador de interseção. Este estudo considerou um ambiente de veículos mistos, visando melhorar o controle de sinal adaptativo e aproveitando a presença de AVs para ajustar suas trajetórias e minimizar ainda mais o tempo total de viagem, evitando paradas desnecessárias dentro do alcance de comunicação. O modelo de trajetória proposto minimiza o tempo de deslocamento dos veículos automatizado e estima o movimento dos veículos convencionais. O algoritmo implementa uma lógica de controle adaptativa modificada baseada em trajetórias computadas, fornecendo dois conjuntos de saídas: o padrão de sinalização, e a trajetória de cada AV para otimizar o tempo de viagem do local de detecção até a interseção.

O trabalho em [Budan et al. 2018] desenvolveu um controle de interseção centralizado cooperativo para realizar a troca de informações, ou seja, posições, dinâmica, e outros atributos, entre o Agente de Controle de Interseção, do inglês, *Intersection Control Agent* (ICA), e os Agentes de Veículos, do inglês, *Vehicle Agents* (VA). Primeiro, os VAs planejam sua trajetória de cruzamento na interseção, incluindo as faixas de chegada e saída, o tempo de chegada à área crítica e o perfil de velocidade do cruzamento quando estão dentro da faixa de controle da interseção. Em segundo lugar, os VAs transmitem o conjunto de dados para o ICA, a fim de reservar uma janela de tempo de cruzamento. Para isso foram implementados quatro diferentes comportamentos de condução dos VAs, modificando a dinâmica do veículo, tais como perfis de velocidade, posicionamento de faixa, parâmetros de aceitação no *gap* entre os veículos para analisar a capacidade da estrada, segurança, emissões de carbono e atrasos de veículos sob controle de interseção não sinalizada.

No trabalho de [Zheng et al. 2017] é apresentado um esquema de controle de interseção para veículos cooperativos sem usar nenhum semáforo, baseado no modelo da teoria do controle preditivo. Os pressupostos são que todos os veículos são totalmente automatizados

e que uma unidade de controle de interseção, do inglês *Intersection Control Unit*, seja instalada na interseção para coordenar os movimentos dos veículos. Cada veículo transmite suas informações de direção, incluindo posição precisa, velocidade instantânea, e a taxa de aceleração para as infraestruturas que contenham o controlador central. Se os conflitos forem esperados, a ICU divide todos os veículos em uma série de pares conflitantes com base nos estados dos veículos. No entanto, as técnicas de otimização nem sempre podem garantir a viabilidade de soluções ótimas. Caso não seja obtida a solução ótima ou o sistema falhe, a ICU entrará no modo de emergência, disseminando o comando lento para guiar os veículos a desacelerar.

Os autores em [Ahn, Colombo e Vecchio 2014] abordam dois principais problemas: verificação da segurança de todos os veículos nos cruzamentos, e o problema de verificação e gestão dos insumos de veículos controlados (Problema de Supervisão). Para resolver os problemas foi utilizado um controlador (supervisor) que gerencia os veículos controlados para evitar colisões de interseção na presença de veículos não controlados. Os autores trabalham com a ideia de agendamento, com um vetor de horários de início para todos os veículos onde todas as restrições em um problema são atendidas. Os autores focaram nos agendamentos de tempo ocioso inserido, do inglês *inserted idle-time* (IIT), no qual é considerado como um intervalo de tempo em aberto quando uma máquina é deliberadamente mantida ociosa enquanto pelo menos uma tarefa está aguardando para ser processada, dessa forma determinando se as entradas atuais levam a uma colisão de interseção. Na pesquisa foram considerados n veículos movendo-se ao longo de caminhos unidirecionais que se cruzam em um ponto. Foi assumido que um subconjunto desses veículos pode ser controlado enquanto o restante não pode.

3.2 Sistemas Descentralizados

O objetivo de [Belkhouche 2017] é otimizar o fluxo de tráfego em uma interseção não sinalizada, garantindo segurança, focando na otimalidade das soluções e no comportamento colaborativo para evitar a colisão entre veículos autônomos. Presumiram que os veículos podem se comunicar e compartilhar sua posição, velocidade e tamanho em tempo real. O processo de comunicação é iniciado quando os veículos estão a uma distância de 50 metros

da interseção. Foram utilizadas as restrições e a função de custo introduzidas pelo autor no trabalho, dessa forma, resolvendo conflitos, otimizando o fluxo de tráfego, e o esforço de controle.

No trabalho de [Campos et al. 2014], foram abstraídos os problemas de implementação focando apenas nos aspectos fundamentais nas tomadas de decisões subjacentes aos problemas. A solução depende de uma cooperação com ordem de decisão pré-determinada (permitindo uma ordem sequencial) combinada com uma implementação de horizonte calculando ótimas trajetórias sem colisões. Para cada veículo, os autores assumiram que um caminho é dado e conhecido; o caminho designado é perfeitamente seguido; a aceleração ao longo do caminho pode ser variada; todos os veículos têm relógios sincronizados e estão localizados antes da interseção no instante inicial. No contexto, uma ordem de decisão definirá a sequência na qual os diferentes agentes resolverão seus problemas de otimização local.

Os autores em [Makarem e Gillet 2011] desenvolveram um algoritmo flexível e descentralizado computacionalmente eficiente para veículos em interseções. Foram consideradas as diferentes inércias dos veículos e sua prioridade de passagem, bem como sua capacidade de frear e acelerar nas interseções. A função de navegação proposta é baseada na distância de cada veículo com seu destino (que pode ser um ponto de movimento) e com outros veículos e/ou obstáculos. Uma zona de visibilidade é definida em torno de cada veículo para simular uma faixa real de detecção e comunicação. No problema de navegação, o objetivo principal é modificar a função de navegação para levar em conta as características dinâmicas dos veículos. Desta forma, espera-se obter comportamentos autônomos que possam imitar a reação típica dos motoristas humanos na interseção.

No trabalho de [Makarem e Gillet 2012] foi considerado um sistema composto por N veículos autônomos. O objetivo de cada veículo é cruzar a interseção com um desvio de velocidade nominal mínimo e sem ter nenhuma colisão com outros veículos. O caminho do veículo é predefinido, e para coordenar os veículos, foi introduzida uma função de navegação para cada veículo, habilitando assim o controle descentralizado em um controlador que garante a segurança do veículo e alta capacidade na interseção sob as restrições mencionadas do mundo real relacionadas à aceleração, velocidade e frenagem. A função de navegação proposta evita a desaceleração simultânea de todos os veículos na interseção. Além disso, o uso dessa função de navegação descentralizada melhora significativamente a aplicabilidade

e a escalabilidade do tráfego.

O trabalho de [Lim et al. 2018] propõe um planejamento de trajetória hierárquica baseado na integração de uma técnica de amostragem e otimização numérica para a condução autônoma urbana. O planejador de trajetória consiste em dois módulos: um de planejamento da trajetória comportamental, e um módulo de planejamento de trajetória de movimento. O primeiro módulo gera uma trajetória macroscópica (ou uma trajetória de manobra) usando o método baseado em amostragem. Com base na trajetória comportamental, o planejamento do movimento busca uma trajetória microscópica para controlar o veículo. Este percurso considera principalmente a dinâmica do veículo e a prevenção de colisão com objetos adjacentes através da otimização numérica. O planejamento da trajetória hierárquica proposto foi implementado em um veículo de teste e o desempenho foi avaliado em vários cenários de direção.

No trabalho de [Isele et al. 2017] foi proposto um método de aprendizado *offline* baseado em *Deep Reinforcement Learning*. O problema de coordenação dos veículos autônomos em interseções foi visto como um problema de aprendizado, onde foi utilizado *Deep Q Network* (DQN) para aprender o valor de estado da função Q. O veículo autônomo começa a partir de uma posição inicialmente parada, o caminho é assumido como sendo dado por um processo de nível mais alto, e a abordagem proposta é encarregada de determinar o perfil de aceleração ao longo do caminho. Foram combinadas várias técnicas recentes de *deep learning*, a fim de aumentar a velocidade de aprendizagem e melhorar o desempenho. A tarefa de aprendizado por reforço foi formulada como um Processo de Decisão de Markov. A utilização de DQN mostrou-se superior sobre algumas técnicas utilizadas (como por exemplo TTC - *Time-to-Collision*), considerando métricas de segurança, como taxa de colisão e métricas de eficiência, taxa de sucesso, e tempo total para concluir a interseção e interrupção do tráfego. Os autores também observaram que a utilização de DQN treinadas podem ser utilizadas em outros cenários.

No trabalho de [Sadigh et al. 2016] foi modelada uma interação entre um carro autônomo e um motorista humano como um sistema dinâmico, no qual as ações tomadas têm consequências imediatas no estado do carro, mas também nas ações humanas. Os autores modelaram essas consequências aproximando o humano como um planejador ideal, com uma função de recompensa adquirida através da Aprendizagem por Reforço Inverso, do in-

glês, *Inverse Reinforcement Learning*. Foi desenvolvido um método baseado em otimização para planejar o comportamento do veículo autônomo de maneira que esteja ciente dos efeitos que ele terá sobre as ações do motorista humano. O controlador do carro a cada iteração computa a sequência de ações para maximizar sua recompensa, em seguida executará a melhor e replanejará as ações.

No trabalho de [Campos et al. 2017] foi apresentada uma abordagem descentralizada para o problema de coordenação do tráfego nas interseções, combinando controle ótimo com heurísticas baseadas em modelos. Foi considerado um cenário onde múltiplos veículos precisam se coordenar de forma autônoma para garantir o acesso a interseção. Os autores projetaram um algoritmo de resolução de conflitos para evitar colisões laterais, capturando assim os principais desafios e aspectos de segurança do problema de coordenação nas interseções. Além disso, assumiram que a condição inicial do sistema de múltiplos veículos é tal que existe no tempo zero uma entrada de controle viável para resolver o problema de coordenação, formulando-o dentro de uma estrutura de controle ótimo restrita, permitindo acomodar convenientemente argumentos de desempenho e segurança.

Em [Campos, Falcone e Sjöberg 2015] foi considerada uma solução totalmente descentralizada adequada para veículos completamente autônomos. A solução adotada baseia-se na ordem de decisão pré-determinada cooperativamente (permitindo a tomada de decisões sequenciais) para o cálculo de trajetórias ideais livres de colisões, que se concentram em sequências de cruzamento viáveis e não de otimização. Em particular, os autores exploraram uma abordagem cooperativa e sequencial de resolução dos conflitos com base em uma ordem de decisão predefinida. Os autores utilizaram uma formulação de controle ótima, demonstrando como a coordenação pode ser assegurada resolvendo-se dois problemas locais, onde a prevenção de colisão é aplicada como restrição de estado dependentes do tempo. Os autores abstraíram as questões de implementação e se concentraram nos aspectos fundamentais dos problemas da tomada de decisão subjacentes.

3.3 Considerações

A fim de demonstrar a relevância do presente estudo na resolução dos agendamentos e conflitos nos cenários de carros autônomos em relação à literatura abordada, foram elencados 5

critérios de comparação: A complexidade do mapa utilizado no estudo, a técnica utilizada, limitações, o simulador e redução no consumo de combustível e/ou emissão de gases.

3.3.1 Sistemas Centralizados

Os autores em [Hausknecht, Au e Stone 2011] utilizaram a malha com 4 cruzamento, 12 ruas, e cada rua com 2 faixas em ambos os sentidos. Foi utilizado atribuição dinâmica do tráfego baseado em A*. A técnica empregada pode ocasionar delay no tráfego dos veículos, e tem o limite máximo de 300 metros para troca de informações. Utilizou o AIM GUI para as simulações. Não foi especificado se houve redução de combustível e/ou CO₂.

Em [Zhu e Ukkusuri 2015] foi utilizado uma malha com 12 nós, ou seja 4x3 ruas. Foi empregado formulação de programação linear para o controle das interseções. Apresentou acidentes nas simulações. Não foi especificado o simulador utilizado. Não foi especificado se houve redução de combustível e/ou CO₂.

Em [Sharon e Stone 2017] foi utilizado 1 cruzamento com 4 ruas, cada rua com 3 faixas em ambos os sentidos. Foi utilizado AIM Híbrido para carros autônomos e não autônomos. A utilização dos veículos não autônomos podia apresentar comportamentos indesejados do esperado pelo controlador. As simulações ocorreram no AIM *Simulator*. Não foi especificado se houve redução de combustível e/ou CO₂.

No trabalho de [Budán et al. 2018] foi utilizado 1 cruzamento com 4 ruas, cada rua com 2 faixas em ambos os sentidos. Foram utilizados 4 CAV com comportamentos distintos para analisar as diferentes características dos veículos em interseções não sinalizadas. Por apresentar veículos não autônomos, o controlador fica a mercê do bom comportamento desses veículos. Utilizou o PTV Vissim como simulador. Apresentou uma redução de 42% de combustível nas simulações.

Em [Zheng et al. 2017] foi utilizado 1 cruzamento com 4 ruas, e cada rua com 1 faixa em ambos os sentidos. Teoria do modelo de controle preditivo foi empregado para resolução dos conflitos. O modelo nem sempre garante a viabilidade da solução ótima. Vissim foi utilizado como simulador. No melhor cenário apresentou redução de 58.6% na emissão de CO₂ e 52.4% no consumo de combustível.

No trabalho de [Ahn, Colombo e Vecchio 2014] não foi especificado uma interseção ou uma grade, mas que n veículos poderiam se aproximar da interseção, para isso um controla-

dor supervisorio foi utilizado. O caso de veículos não autônomos é considerado, tornando o controlador sujeito ao bom comportamento do motorista. Não foi especificado o simulador utilizado. Não foi especificado se houve redução de combustível e/ou CO₂.

Em [Ashtiani, Fayazi e Vahidi 2018] foi utilizado uma grade de 3x3, no qual cada cruzamento tinham 4 ruas, e cada rua uma faixa em ambas as direções. Foi utilizado programação linear inteira mista com troca de informações entre os cruzamentos vizinhos. Essa abordagem está propensa a possíveis percas de informações nas trocas de mensagens com os nós vizinho. Foi utilizado um simulador próprio desenvolvido em Java. Não foi especificado se houve redução de combustível e/ou CO₂.

3.3.2 Sistemas Descentralizados

No trabalho de [Belkhouche 2017] foi utilizado o cenário de 1 cruzamento com 4 ruas, cada rua com 1 faixa em ambos os lados. Através de modelagem matemática e troca de informações entre os veículos foi possível prevenir colisões. Apesar da possibilidade de escalabilidade, foi utilizado o cenário com apenas 3 carros em rotas de conflitos para estudo de caso. Não foi especificado onde a simulação foi feita, assim como se houve redução no consumo de combustível e/ou emissão de gases.

Em [Campos et al. 2014] foi utilizado o cenário com apenas 1 cruzamento de 4 ruas, cada rua com 1 faixa em ambas as direções. A resolução dos conflitos foi feita no cálculo baseado na ordem de decisão cooperativamente predeterminada. Foi utilizado o cenário com apenas 3 carros tentando atravessar a interseção sem ocasionar conflitos, apesar de ser dito que a utilização não perde a generalidade. Assim como o simulador, os gastos de combustível e/ou as emissões não foram especificados.

Foi utilizado o cenário de 1 cruzamento em [Makarem e Gillet 2011], assim como os demais, o cruzamento possui 4 ruas, porém nesse trabalho não foi especificada a quantidade de faixas, sendo provavelmente 1 em ambas as direções. A resolução de conflitos foi feita através de um algoritmo que considera a inércia dos veículos assim como a prioridade de passagem, e a capacidade de frear/acelerar na interseção. O algoritmo necessita de um campo de visão para trocar informações e detecção de veículos, porém não foi especificado as decisões a serem tomadas em caso de falhas com o campo de visão. Não foi especificado o simulador, assim como se houver redução de combustível, ou emissão de gases.

Em [Lim et al. 2018] foi utilizado como cenário a variação de ruas com várias curvas e 3 faixas, dessa forma podendo validar a solução dos conflitos de carros autônomos. Planejamento de trajetória hierárquica baseado na integração de amostragem em conjunto com método de otimização para a condução autônoma urbana foi utilizado como técnica da resolução de conflitos. Para a validação da pesquisa, o carro autônomo A1 da Universidade de Hanyang foi utilizado, equipado com sensores e lidars. O estudo foi realizado em 3 cenários, mudança de faixa, desvio de direção com obstáculos, e cruzamento de interseção. Foi abstraído se houve redução de combustível, assim como a emissão de gases.

No trabalho de [Isele et al. 2017] foi utilizado o cenário de 1 cruzamento no qual tinha 6 faixas a serem atravessadas, o carro saía de um ponto inicial parado e tinha que chegar ao destino. Foi utilizado um método de Deep Reinforcement Learning, e para se tornar efetivo os dados, foi necessário rodar mais de 10 mil vezes cada cenário. Foi utilizado o SUMO como simulador para obter os resultados, porém não foi especificado se houve redução no consumo de combustível e/ou emissão de CO₂.

Em [Campos et al. 2017] foi utilizado o cenário de 1 cruzamento com 4 ruas, e 1 faixa em ambas as direções. A técnica de resolução dos conflitos empregada foi a combinação de controle ótimo com modelo baseado em heurística. Não foi especificado o tempo de solução do modelo para resolução dos conflitos nos casos de testes. Não foi especificado o simulador, e se houve redução no consumo de combustível ou emissão de gases.

Foi utilizado em [Campos, Falcone e Sjöberg 2015] o cenário de 1 cruzamento com 4 ruas, e 1 faixa em ambas as direções. Foi utilizada a função de controle ótimo para pre definir a ordem de decisão afim de solucionar os conflitos que possam vir a existir. Não foi especificado o tempo de solução dos casos estudados, assim como o simulador utilizado e se houve redução no consumo de combustível, e emissão de gases poluentes.

Dentre todos os trabalhos citados, os autores em [Fayazi, Vahidi e Luckow 2017] apresentaram uma abordagem semelhante a nossa, porém, foi especificado a solução para apenas uma interseção. Em um trabalho futuro baseado na utilização de MILP, os autores em [Ash-tiani, Fayazi e Vahidi 2018] apresentaram uma solução para um cenário de malha com 3x3 ruas para demonstrar que é possível solucionar problemas em rotas com mais de 1 cruzamento. Em ambos os casos, foi utilizado um simulador próprio desenvolvido em Java baseado na ideia de *Software-in-loop*. Não foi especificado se houve reduções no consumo de

combustível e/ou emissões.

3.3.3 Análise

Depois de abordar a literatura, foi possível identificar que tanto entre os trabalhos centralizados e descentralizados, o tema de gerenciar e otimizar uma grande região com cenários apenas de carros autônomos não foi abordado da maneira devida, apenas sendo explicitado a possibilidade da adaptação do estudo de caso para maiores áreas ou cruzamentos. Identificado isso, o presente trabalho tem o intuito de propor uma solução para esses casos que possam vir a acontecer. Foi utilizado Programação Linear Inteira com algumas restrições de segurança para garantir a inexistência de colisões, além da otimização dos tempos nas rotas utilizadas. Para cenários com muitos cruzamentos assim como uma grande quantidade de rotas, é apresentada uma heurística para resolver os conflitos presentes. Apesar de não garantir a solução ótima, a heurística provou ser uma opção de gerenciamento devido à solução dos conflitos independente do cenário abordado.

Com a utilização da heurística foi demonstrado que é possível resolver os casos que possuem grandes áreas assim como uma grande quantidade de cruzamentos. O tempo de resolução dos problemas apresentados foram de alguns milissegundos, e a solução encontrada é livre de conflitos, o que pode ser uma opção para trabalhos futuros nos quais o fator tempo real é explorado. nesse trabalho esses casos não foram abordados.

Nos trabalhos centralizados houve um senso comum na utilização apenas de 1 cruzamento com 4 ruas, e cada rua geralmente possuindo 2 faixas. O tráfego utilizado varia conforme os trabalhos. A presença apenas de um controlador central também é abordado nos casos estudados, porém não foi especificado os casos onde acontecem erros. A garantia de diminuição nos tempos do tráfego utilizado é demonstrada, porém, não foi abordado de maneira devida se houve ou não uma redução no consumo do combustível, assim como a emissão de CO₂.

Nos trabalhos descentralizados, a utilização de várias técnicas foram abordadas, porém, há a necessidade de padronização dos carros com vários equipamentos que podem vir a inviabilizar a solução proposta para os dias atuais, e em alguns casos, não são totalmente tolerantes a erros e falhas, o que deixa em aberto os casos em que conflitos possam acontecer nos casos apresentados.

Com a otimização dos tempos, além de diminuir a chegada no destino desejado, o presente trabalho demonstra que foi possível diminuir o consumo de combustível, assim como a emissão de gases poluentes, reduzindo o impacto ao meio ambiente onde a solução é aplicada.

Capítulo 4

Metodologia

4.1 Sistema Virtual

Para demonstrar a capacidade do trabalho com a utilização do sistema em conjunto com os carros autônomos, foi assumido que nos cenários utilizados não haveria pedestres e nenhum outro obstáculo presente nas simulações. Além disso, apenas carros autônomos são permitidos, logo, caminhões, bicicletas e motocicletas não foram considerados nesse estudo. Os mapas utilizados são estáticos, o que significa dizer que nenhum outro carro aparecerá em algum tempo aleatório na simulação além dos que foram programados. Os mapas utilizados foram extraídos do *OpenStreetMap* (<https://www.openstreetmap.org/>).

Como foi dito anteriormente, o presente trabalho propõe retirar todos os sinais de trânsito e todas as placas de sinalizações que são utilizadas em interseções, substituindo-as por um sistema virtual capaz de fazer o gerenciamento dos carros autônomos. O sistema será capaz de conhecer as características da região a ser otimizada, por exemplo, o tamanho das ruas, o limite de velocidade, e os cruzamentos presentes. Dessa forma, será possível localizar os pontos de interseção onde possam ocorrer conflitos entre as rotas dos carros autônomos.

Cada carro cadastra sua rota (para o sistema, uma rota é descrita como uma sequência de ruas) no sistema incluindo o tempo inicial da viagem. Para cada carro que tem sua rota cadastrada, o sistema agenda o tempo em todas as interseções presentes por onde o carro há de passar, resolvendo assim os conflitos que possam existir. O sistema é capaz de identificar os tempos máximos e mínimos utilizados para fazer o agendamento baseado nos limites impostos em cada rua utilizando velocidade média. Foi utilizada a ideia de *First Come First*

Serve, significando dizer que a ordem do agendamento dos tempos no sistema foi a ordem em que as rotas foram cadastradas pelos veículos.

O sistema retorna os tempos das rotas como resposta utilizando a formulação de Programação Linear Inteira. Através da modelagem matemática, o sistema virtual foi capaz de identificar um possível limite mínimo para os tempos (*lower bound*). Quando o modelo não conseguia identificar os tempos que satisfizessem as restrições impostas, eram utilizados tempos nos quais a solução do somatório de chegada de todos os veículos ficassem próximo ao limite inferior previamente estabelecido, garantindo assim uma resposta satisfatória às condições dos tempos impostos previamente pelos carros.

4.1.1 Modelagem Matemática

Definições

Neste trabalho, a seguinte representação foi utilizada:

- V – Conjunto de vértices (interseções);
- A – Conjunto de arcos (transição de um vértice a outro vértice);
- K – Conjunto de veículos;
- K_j – Conjunto de veículos que passam por $j \in V$;
- K_{ij} – Conjunto de veículos que passam por $(i, j) \in A$.

Dados de entrada utilizados:

- I – Intervalo de segurança mínimo entre dois veículos em um vértice;
- T_{ij}^{min} – Tempo mínimo para atravessar o arco $(i, j) \in A$;
- T_{ij}^{max} – Tempo máximo para atravessar o arco $(i, j) \in A$.

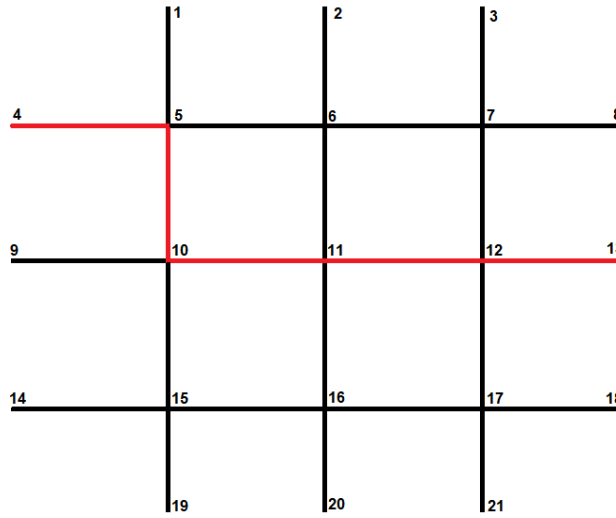
Foram utilizadas as seguintes variáveis:

- x_{ij}^k – Tempo que o veículo $k \in K$ leva para passar por $(i, j) \in A$;
- y_j^k – Tempo de chegada do veículo $k \in K$ em $j \in V$;

- π^k – Sequência de vértices presentes na rota do carro $k \in K$.

Por exemplo, $\pi^k = (4, 5, 10, 11, 12, 13)$, como sendo a sequência de vértices presentes na rota demonstrada na Figura 4.1, ou seja, $\pi^k(2) = 5$, o segundo vértice pelo qual o carro irá passar. Assumimos também que $|\pi^k|$ é a cardinalidade da sequência dos vértices, no exemplo acima $|\pi^k| = 6$. Então $\pi^k(|\pi^k|) = 13$, o último vértice pelo qual o carro irá passar.

Figura 4.1: Exemplo de rota utilizada



$$\min \sum_{k \in K} y_{\pi^k(|\pi^k|)}^k \quad (4.1)$$

Sujeito a:

$$y_{\pi^k(i)}^k = y_{\pi^k(i-1)}^k + x_{\pi^k(i-1)\pi^k(i)}^k, \quad \forall k \in K, i = 2, \dots, |\pi^k| \quad (4.2)$$

$$|y_j^k - y_j^l| \geq I, \quad \forall j \in V, \forall k \in K_j, \forall l \in K_j, k \neq l \quad (4.3)$$

$$T_{ij}^{min} \leq x_{ij}^k \leq T_{ij}^{max}, \quad \forall k \in K_{ij}, (i, j) \in A \quad (4.4)$$

$$y_j^k \geq 0, \quad \forall j \in V, \forall k \in K_j \quad (4.5)$$

Na função objetivo (4.1) é proposta a minimização dos tempos finais de todas as rotas, dessa forma, diminuindo o somatório dos tempos de chegada para cada rota previamente cadastrada. As restrições (4.2) garantem que os tempos de cada interseção presentes na rota sejam agendados de forma que os tempos designados sejam satisfatórios às veloci-

des previamente explicitadas pelo veículo. Para que houvesse uma margem de segurança ao atravessar a interseção, a restrição (4.3) garante que a diferença dos tempos de dois carros ao atravessarem uma interseção conflitante seja maior que o intervalo de segurança, ou seja, enquanto algum veículo estiver cruzando uma interseção conflitante, o outro veículo só poderá entrar na interseção depois que o tempo de segurança for obedecido. Para que o modelo permanecesse linear, a restrição (4.3) foi substituída por:

$$y_j^k - y_j^l + M\underline{b} \geq I, \quad \forall i \in V, \forall k \in K_j, \forall l \in K_j, k \neq l, \quad (4.6)$$

$$y_j^k - y_j^l + M\underline{b} \leq -I + M, \quad \forall i \in V, \forall k \in K_j, \forall l \in K_j, k \neq l \quad (4.7)$$

foi utilizado uma variável binária $\underline{b}$ para que fosse possível a resolução dos conflitos utilizando as novas restrições, e um parâmetro M suficientemente grande.

Para que se assemelhasse ao cenários do mundo real, nos mapas estudados foi utilizado o limite de velocidade imposto por rua, o que significa dizer que o veículo poderá ter velocidades limites diferentes de acordo onde estiver na rota designada ao sistema. Para isso, a restrição (4.4) é imposta. A restrição (4.5) garante que não haja tempos negativos presentes no modelo.

4.2 Heurística

A fim de se ter uma perspectiva melhor em relação a áreas com grandes quantidades de rotas, foi desenvolvida uma heurística baseada nas regras do modelo apresentadas anteriormente. A heurística tem como objetivo agendar o menor tempo para cada cruzamento das devidas rotas, conseqüentemente, garantindo a velocidade mais rápida para o veículo.

Para exemplificar o funcionamento da heurística, vamos utilizar o mesmo mapa apresentado anteriormente, porém com novas rotas que apresentam conflitos, conforme pode ser visto na figura Figura 4.2. Foram simulados 4 carros com rotas distintas, a rota do primeiro carro é formada pelos seguintes cruzamentos $\pi^1 = (4, 5, 6, 7, 8)$ representada pela seta azul, a rota do segundo carro pelos cruzamentos $\pi^2 = (1, 5, 10, 15, 19)$ representada pela seta vermelha, a rota do terceiro carro é formada pelos cruzamentos $\pi^3 = (3, 7, 12, 17, 21)$ representada pela seta verde, e a rota do quarto carro é formada pelos seguintes cruzamento $\pi^4 =$

Algoritmo 1: Heurística Utilizada

```

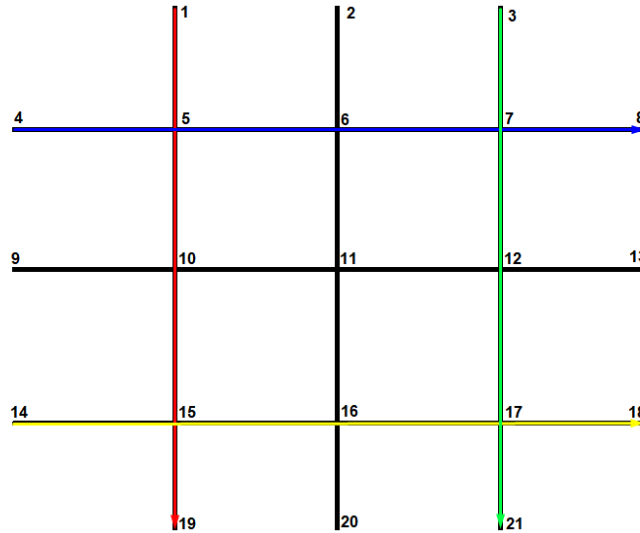
1  Resolução de Conflitos(configFile)
2  qtdeRotas  $\leftarrow$  configFile.qtdeRotas
3  qtdeCruzamentos  $\leftarrow$  configFile.qtdeCruzamentos
4  para  $i \leftarrow 0$  to qtdeRotas faça
5      rotaAtual  $\leftarrow$  configFile.rotas(i)
6      tempoAcumulado  $\leftarrow$  0
7      tempoInicial  $\leftarrow$  0
8      para  $j \leftarrow 0$  to Tamanho(rotaAtual) faça
9          se  $j = 0$  então
10             agendado  $\leftarrow$  agendaTempo(tempoInicial, cruzameto(j) )
11             se agendado então
12                 atualizaTempoAcumulado()
13                 atualizaTempo(cruzameto(j), tempoInicial)
14                  $j++$  ; // Agendar o próximo cruzamento
15             senão
16                  $j = 0$  ; // Começar novamente
17                 tempoInicial++ ; // Dando shift no tempo inicial
18             senão
19                 tempoMinimo  $\leftarrow$  lêTempoMinimo()
20                 tempoMaximo  $\leftarrow$  lêTempoMáximo()
21                 para tempo  $\leftarrow$  tempoMinimo to tempo  $\leq$  tempoMaximo faça
22                     tempoAuxiliar  $\leftarrow$  tempoAcumulado + tempo
23                     se agendaTempo(tempoAuxiliar, cruzameto(j) ) então
24                         atualizaTempoAcumulado(tempoAuxiliar)
25                         atualizaTempo(cruzamento(j), tempoAcumulado)
26                         agendado  $\leftarrow$  true
27                          $j++$ 
28                         break
29                 se agendado = false então
30                      $j \leftarrow 0$  ; // começar novamente
31                     tempoAcumulado  $\leftarrow$  0
32                     tempoAuxiliar  $\leftarrow$  0
33                     tempoInicial++
34 tempo  $\leftarrow$  somatorioTemposRotas()
35 temposRotas  $\leftarrow$  tempoRotasAtualizados
36 retorna temposAtualizados

```

(14, 15, 16, 17, 18) representada pela seta amarela.

Para simplificar o entendimento, vamos supor que todas as ruas tem o mesmo tamanho de 150 metros, que o tempo mínimo para percorrer a rua é de 7 segundos, e o máximo de 9 segundos, equivalente mais ou menos a 80 Km/h e 60 Km/h respectivamente. O intervalo de segurança utilizado foi de 5 segundos. Para a simulação, cada carro não teve restrição sobre o tempo inicial de saída.

Figura 4.2: Exemplo de rotas conflitantes



Dadas as seguintes informações, a heurística tentará agendar as rotas conforme especificadas, do primeiro até o quarto carro. No momento inicial, todos os cruzamentos estão livres de agendamentos. Dessa maneira, os tempos em π^1 são alocados utilizando a velocidade máxima permitida, ou seja, 7 segundos. Logo os tempos na sequência π^1 são (0, 7, 14, 21, 28).

Notemos que na segunda rota há um cruzamento conflitante com a primeira $\pi^2(2) = 5$. A heurística consegue agendar o primeiro cruzamento da segunda rota com o tempo inicial 0, visto que não há conflitos nesse cruzamento para o tempo desejado (linha 10). Uma vez bem sucedido o agendamento, a heurística agenda ao cruzamento especificado o tempo inicial não conflitante (linha 13), e logo em seguida tentará agendar o próximo (linha 14).

Ao tentar agendar o segundo cruzamento com a velocidade máxima, a restrição de segurança não é satisfeita (4.3) (linha 23). Conforme agendamento prévio, já há um veículo passando pelo cruzamento 5 no instante 7. Nesse caso, a heurística tentará agendar os próximos tempos até que chegue no máximo permitido para a rua (linha 21), porém, o conflito ainda persiste com os tempos de 8 e 9 segundos. Como não foi possível o agendamento desse cruzamento na rota especificada, a heurística dispensa os tempos já agendados para essa rota, e acrescenta 1 segundo ao tempo inicial (respectivamente da linha 29 até linha 33). Nesse momento, a heurística começa a rota com o tempo inicial 1, e ao tentar agendar os tempos no segundo cruzamento, a restrição de segurança ainda não é satisfeita, e o mesmo processo anterior é repetido.

Para a resolução dessa rota, a heurística só consegue resolver o conflito utilizando o tempo inicial 5. Dessa maneira, a diferença entre os tempos agendados no cruzamento conflitante satisfazem a condição de segurança imposta. Nos outros cruzamentos não há conflitos, e os tempos de π^2 ficam (5, 12, 19, 26, 33), garantindo a velocidade máxima na rota, porém começando no instante 5.

No rota 3 há um cruzamento também presente na rota 1 $\pi^3(2) = 7$, porém, ao tentar agendar os tempos começando com o instante inicial 0, a restrição de segurança imposta é satisfeita, e não há restrições com os agendamentos da rota. Dessa maneira, os tempos da rota ficam (0, 7, 14, 21, 28). Na rota 4, há um conflito com a rota 3 $\pi^4(4) = 17$. A heurística passará pelo mesmo problema antes abordado ao tentar agendar os tempos começando por 0, e os tempos em π^4 ficam (5, 12, 19, 26, 33). Todos os passos descritos anteriormente podem ser melhor observados visualizando a sequência de execução presente no Apêndice A.

Apesar de não estar explícito na heurística, o mesmo processo de resolução dos conflitos é feito novamente, só que ao invés de percorrer as rotas da primeira até a última (linha 4), é percorrido da última até a primeira (*for* $i \leftarrow \text{tamanho}(\text{qtdeRotas})$ *to* $i \geq 0$). Uma vez conhecido o menor valor do somatório dos tempos das rotas, é retornado o melhor tempo no qual todos os cruzamento foram agendados, obedecendo todas as restrições, evitando assim possíveis colisões nas rotas utilizadas.

Para identificar os cenários nos quais a heurística seria melhor utilizada, foram feitos vários testes comparando com os resultados obtidos no SUMO. Foram feitas comparações do tempo total de chegada dos veículos, o consumo de combustível, e a emissão de gases CO₂. A simulação foi aumentando de forma gradativa na quantidade de rotas, assim como na de ruas como demonstrado na coluna Mapa. Os valores das colunas Tempo, CO₂ e Combustível representam o ganho obtido em cima dos valores da simulação, como pode ser observado na Tabela 4.1.

Foram testados alguns casos nos quais não foram utilizadas todas as ruas presentes no mapa, assim como nem todos os cruzamentos estavam presentes nas rotas utilizadas, investigando assim o impacto no gerenciamento dos casos nos quais há rotas de tamanhos variados. Os valores obtidos podem ser observados na Tabela 4.2.

Foram feitos alguns testes utilizando mapas que continham um grande número de ruas, assim como o de rotas, investigando assim a possibilidade do impacto gerado com o ge-

Tabela 4.1: Percentagem dos valores conseguidos com a heurística em relação aos resultados obtidos com o SUMO (1)

Mapa	Tempo (%)	CO ₂ (%)	Combustível (%)
25x20	1,58	0,29	0,10
36x24	0,83	2,85	3,11
49x28	1,26	5,96	6,22
64x32	1,97	5,39	5,66
81x36	0,57	6,23	6,49
100x40	2,70	6,71	6,96
121x44	1,99	7,74	7,99
144x48	3,53	8,01	8,26
169x52	2,39	8,29	8,54

Tabela 4.2: Percentagem dos valores conseguidos com a heurística em relação aos resultados obtidos com o SUMO (2)

Mapa	Tempo (%)	CO ₂ (%)	Combustível (%)
25x20	1,76	2,67	0,36
36x21	0,26	1,82	2,11
49x24	1,16	5,82	6,09
64x26	3,20	8,77	9,02
81x30	3,07	8,38	8,63
100x30	2,52	7,35	7,61
625x85	9,21	19,17	19,19

renciamento em regiões que podem apresentar muitos conflitos. Os resultados podem ser observados na Tabela 4.3.

Tabela 4.3: Percentagem dos valores conseguidos com a heurística em relação aos resultados obtidos com o SUMO (3)

Mapa	Tempo (%)	CO ₂ (%)	Combustível (%)
625x100	6,58	20,52	20,63
625x150	6,64	21,31	21,40
625x200	6,95	17,86	17,87
900x120	9,91	20,24	20,26
900x240	9,18	19,10	19,11

Como pode ser observado nas tabelas anteriores, apesar de fazer o gerenciamento e obter ganhos nos resultados em relação ao SUMO em todos os cenários, nos que apresentaram grandes quantidades de ruas e cruzamentos foram os que a heurística se saiu melhor. Nesses casos, a redução no tempo ocioso das paradas desnecessárias nos cruzamentos foi evitado. Consequentemente, houve uma diminuição no tempo de chegada dos veículos no destino

desejado, e redução no consumo do combustível e emissão de gases.

Capítulo 5

Resultados

5.1 Modelo Matemático

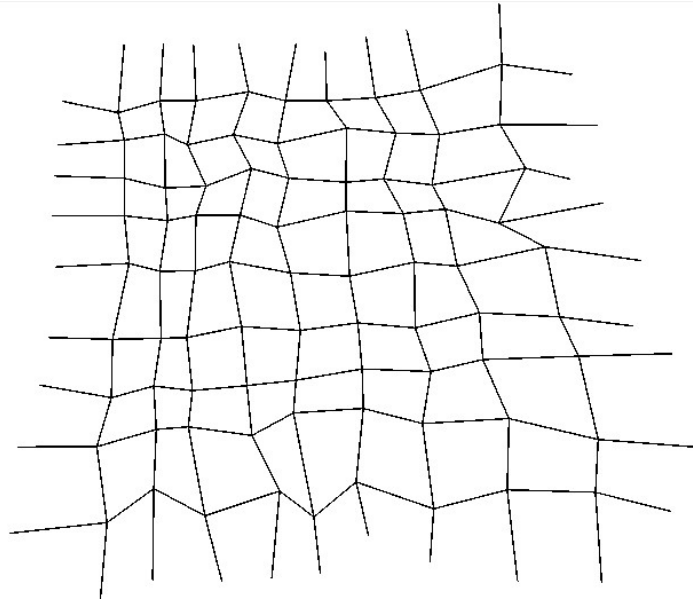
5.1.1 Estudo de Caso 1

Para validar o modelo, pequenas simulações foram criadas utilizando o SUMO. Para que houvessem o máximo de cruzamentos possíveis, os mapas tinham o mesmo número de ruas na horizontal e vertical, ou seja, uma região em forma de grade, como demonstrado na Figura 5.1. Nas simulações foram retirados todos os semáforos ou placas de trânsito que identificavam alguma preferência (por exemplo, uma placa de pare antes de um cruzamento), deixando apenas o controle dos cruzamentos sendo realizado pelas informações que o sistema informou previamente aos carros autônomos. Para a realização dos testes, foi utilizado um notebook com 4Gb de memória RAM, processador intel core i3, o CPLEX para resolver o modelo, com a utilização de um código suporte desenvolvido em C++.

É importante notar que os cenários aqui utilizados apesar de serem limitados a um veículo por pista, não excluem a possibilidade da existência de dois veículos cruzando a mesma interseção ao mesmo tempo, contanto que ambos estejam viajando em direções opostas, o que permitiria a travessia sem problemas. No entanto, esse caso não foi considerado, porém, é possível projetar a entrada para a resolução dos conflitos permitindo que isso se torne possível.

Para cada rua, colocamos um carro fazendo a rota de um ponto inicial da malha até um final. Deformamos a malha para que houvesse a heterogeneidade encontrada nas regiões

Figura 5.1: Região 9x9 em forma de grade



atuais. Ou seja, ruas com tamanhos e sentidos diferentes, tornando assim a simulação mais próxima da realidade. Nessa simulação foram utilizadas 36 rotas diferentes com veículos iniciando suas rotas em todas as direções possíveis da região acima. As ruas da simulação tinham as direções de ida e volta permitidas (ou mão e contramão), possibilitando ao carro fazer a manobra de curva para algum dos lados, ou seguindo em frente na rota desejada.

Ao fazer o gerenciamento dos tempos das rotas dos veículos autônomos através do sistema virtual, houve uma redução no somatório dos tempos de chegada de todos os veículos em 853 segundos conforme Tabela 5.1. Também houve uma redução em cerca de 25% tanto no consumo de combustível quanto na emissão do CO₂.

Tabela 5.1: Descrição dos valores da simulação do mapa 9x9

	Duração (segundos)	CO ₂ (mg/s)	Combustível (ml/s)
SUMO	7786.00	29268179.50	12568.23
Sistema Virtual	6933.00	21984066.87	9433.00

Os valores acima foram obtidos da simulação utilizando as opções *-emission-output* para o consumo de combustível e emissão de CO₂, e *-tripinfo-output* para a duração dos tempos em que os carros finalizaram suas rotas. Em relação ao tempo de simulação, foi utilizada a opção padrão de 1 segundo como parâmetro em *-step-length*. Como a simulação do SUMO é estática, não foi necessário executar mais de uma vez a simulação do mapa utilizado para

obter os resultados. Isso seria necessário apenas se houvesse alguma mudança, tal como o acréscimo de algum veículo, ou mudança na rota de algum dos veículos.

5.1.2 Estudo de Caso 2

Para obtermos uma visão mais realística do problema enfrentado, decidimos fazer rotas em um mapa de uma região verídica. Para isso, utilizamos uma parte do bairro dos Bancários localizado na capital da Paraíba, João Pessoa. A simulação possui 1167 ruas e 463 cruzamentos. Figura 5.2 demonstra a região utilizada.

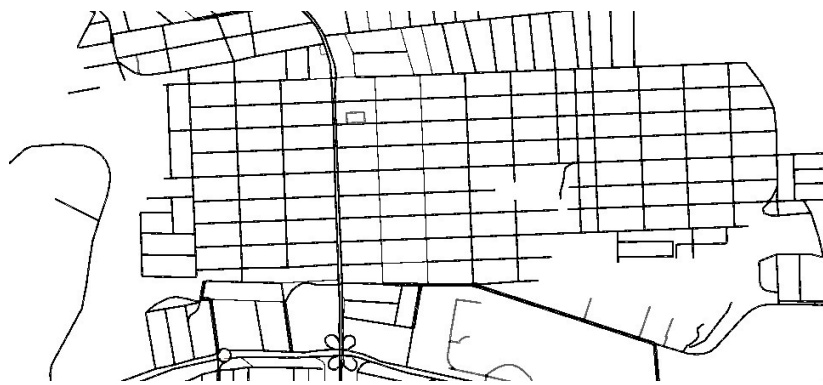


Figura 5.2: Região dos Bancários utilizada na simulação

Nessa simulação foram utilizadas 73 rotas diferentes de forma que fossem obtidos diversos conflitos nos cruzamentos, tornando impossível o tráfego dos veículos de forma segura sem o devido gerenciamento. O modelo executou por 20 minutos (tempo limite especificado pelo autor). O valor da solução retornado com os tempos totais de viagens ficou cerca de 1% da solução ótima. Conforme a Tabela 5.2, podem ser observados os valores obtidos utilizando a mesma configuração dos parâmetros da simulação anterior.

Tabela 5.2: Descrição dos valores da simulação do mapa Bancários

	Duração (segundos)	CO ₂ (mg/s)	Combustível (ml/s)
SUMO	3584.00	10909621.55	4680.50
Sistema Virtual	3091.00	11074781.90	4760.73

Através do gerenciamento feito pelo sistema virtual, foi possível reduzir o somatório dos tempos de chegada dos veículos em 493 segundos, apesar da solução não ser o ótimo para a instância utilizada, o sistema virtual obteve uma melhora em cerca de 14% do tempo gasto diariamente no trânsito. Porém, o CO₂ e o combustível consumidos tiveram um aumento em

cerca de um pouco mais de 1%. Conforme Subseção 2.1.3, o modelo utilizado calcula as emissões utilizando a região desejada. Nessa simulação, nem todas as ruas foram utilizadas como rotas diferentemente da anterior, ficando apenas a região mais central com cruzamentos por onde as rotas passam. A simulação também contou com rotas nas quais a quantidade de cruzamentos não foi elevado (e.g. 4), o que impactou no cálculo da simulação.

5.1.3 Discussão

Todos os testes realizados até o momento utilizaram instâncias pequenas (menos de 100 rotas). Para utilizar instâncias maiores, foi necessário desenvolver uma heurística para otimizar e gerenciar os tempos das rotas presentes no sistema, pois o modelo matemático não foi capaz de retornar em tempo hábil (segundos) a solução que satisfizesse o agendamento, algumas vezes demorando minutos tentando otimizar a solução atrás do ótimo do problema. Em outras palavras, o tempo de solução do problema cresce conforme a quantidade de cruzamentos aumentam, também assim como a de veículos presentes na instância.

5.2 Heurística

5.2.1 Comparação Modelo Matemático x Heurística

Para comparar os tempos retornados da heurística com o modelo, foram feitos mapas que pudessem gerar a maior quantidade de conflitos possíveis, com rotas vindas de todos os lugares, gerando assim um cenário caótico sem o devido gerenciamento, fazendo-se necessária a resolução do mesmo para que não fosse ocasionado nenhum acidente. O tamanho dos mapas foi crescendo gradativamente para tentar identificar um possível ponto limite para a resolução do modelo. Assim como a quantidade de ruas, a quantidade de rotas também foi crescendo a fim de identificar a influência nos resultados obtidos. Pode ser visto na Tabela 5.3 os resultados da resolução dos arquivos de configuração passados para o modelo, tanto quanto para a heurística.

Na coluna Mapa é descrita a quantidade de cruzamentos x rotas, Modelo (Total) e Heurística (Total) é descrito o somatório dos tempos retornados pela resolução das rotas passadas respectivamente, Tempo descreve os valores máximos de cada simulação, e Comparação

significa a porcentagem da diferença obtida entre os tempos.

Tabela 5.3: Descrição dos valores das comparações entre o Modelo e a Heurística

Mapa	Modelo (Total)	Tempo	Heurística (Total)	Tempo (ms)	Comparação (%)
25x20	1260	1.19 s	1260	34	0
25x40	2713	10 min	2812	63	1.52
36x24	1832	5 ms	1856	48	1.31
36x48	3840	10 min	4040	69	5.21
49x28	2596	0.7 s	2596	54	0
49x56	5461	10 min	5586	37	2.29
81x36	4412	4.42 s	4412	61	0
81x72	9174	10 min	9328	84	1.68
100x40	5448	0.13 s	5488	119	0.73
100x80	11168	10 min	11460	120	2.61
121x44	6708	6.7 s	6708	148	0
121x88	13853	10 min	14036	134	1.32
169x52	9484	10.88 s	9484	86	0
169x104	19536	10 min	19631	234	0.49
196x56	10984	0.41 s	11040	182	0.51
196x112	22422	10 min	11040	182	0.51
225x60	12740	11.11 s	12740	152	0
225x120	26133	10 min	26237	113	0.40
256x64	14472	0.47 s	14536	86	0.44
256x168	29476	10 min	29757	154	0.95
289x68	16476	16.07 s	16476	124	0
289x136	33712	10 min	33807	224	0.28
324x72	18440	0.77 s	18512	76	0.39
324x144	37479	10 min	37930	123	1.20
361x76	20692	23.88 s	20692	104	0
361x114	31388	10 min	31469	158	0.26
400x80	22888	1.03 s	22968	146	0.35
400x160	46559	30 min	46858	196	0.64
441x84	25338	30.10 s	25388	206	0.20
441x168	-	-	51883	228	-

Conforme pode ser observado nos resultados, a heurística demonstrou ser melhor em termos de resolver os conflitos e retornar os tempos das soluções. A partir do mapa 441x168 o modelo começou a não responder mais aos comandos da máquina utilizada, tornando difícil as comparações entre o modelo e a heurística.

5.2.2 Estudo de Caso 3

Para validar a resolução dos conflitos utilizando a heurística em uma área com grandes quantidades de rotas, bairros de João Pessoa foram utilizados nas simulações. Na Figura 5.3 foi feito uma simulação com um total de 125 rotas.

Figura 5.3: Região do bairro Bessa

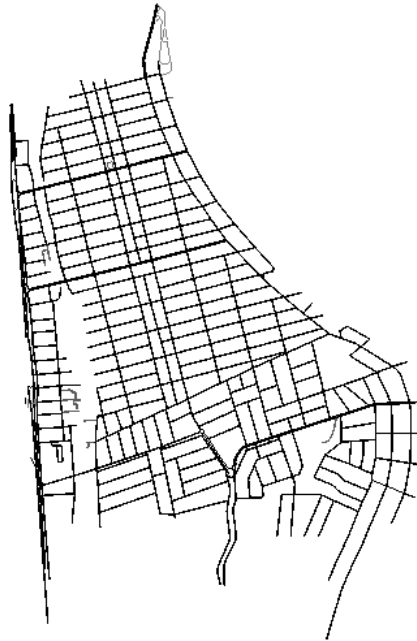


Tabela 5.4: Descrição dos valores da simulação do mapa Bessa

	Duração (segundos)	CO ₂ (mg/s)	Combustível (ml/s)
SUMO	3709.00	13268861.76	5709.65
Sistema Virtual	3453.00	10563535.57	4541.80

Como pode ser observado na Tabela 5.4, a heurística obteve uma redução em cerca de 7% no somatório dos tempos, totalizando uma redução de 256 segundos no tempo final das rotas. Os conflitos foram gerenciados e nenhuma colisão foi apresentada perante o simulador. Em algumas rotas foi necessário acrescentar algum tempo no cruzamento inicial para que todas as restrições fossem obedecidas, dessa forma, nem todas as rotas saíram do instante inicial 0. Com o gerenciamento das rotas, a diminuição das paradas desnecessárias nos cruzamentos foi alcançada, garantindo assim uma melhora em quase 20% no consumo do combustível, assim como a emissão de gases CO₂.

5.2.3 Estudo de Caso 4

Para gerar mais conflitos, uma grande área foi utilizada permitindo gerar 200 rotas, como pode ser visto na Figura 5.4. As rotas foram obtidas dos bairros ao redor da Eptácio Pessoa, parte de Tambaú e Torre.

Figura 5.4: Região do conjunto dos bairros



Tabela 5.5: Descrição dos valores da simulação da região dos bairros

	Duração (segundos)	CO ₂ (mg/s)	Combustível (ml/s)
SUMO	9616.00	31671179.73	13611.98
Sistema Virtual	8937.00	25217755.41	10835.47

Utilizando mais rotas foi possível gerar mais conflitos, aumentando assim a necessidade do gerenciamento para evitar possíveis colisões. Assim como a simulação anterior, em algumas rotas não foi possível o agendamento com os tempos iniciais em zero, fazendo-se necessário o acréscimo de alguns segundos para os tempos obedecerem as restrições impostas. Como pode ser visto na Tabela 5.5, os tempos das rotas obtiveram uma vantagem quando o controle do tráfego é feito pela heurística, desse modo evitando paradas desnecessárias, diminuindo o tempo gasto no tráfego e em congestionamentos, além do estresse gerado no motorista. Além de otimizar os tempos de chegada dos veículos autônomos com uma melhora em cerca de 7%, também foi possível reduzir 10% dos gases CO₂ emitidos, e 11% no combustível gasto nas rotas otimizadas.

5.2.4 Estudo de caso 5

Para testar a capacidade de resolução da heurística, foi gerado um mapa com 1024 interseções, e cerca de quase 4 mil ruas. Cada rua foi deformada a fim de garantir a aleatoriedade de tamanho entre ruas que é encontrada no mundo real. A simulação contou com 256 rotas, porém, apesar do tamanho do mapa imposto e a quantidade de rotas presentes, o tempo de solução foi cerca de 300 ms, provando assim ser capaz de resolver problemas de pequenas e grandes áreas, com poucas e muitas rotas. A instância gerada demonstra a dificuldade que pode ser imposta nos mapas reais, pois nesse caso, cada rota possuía mais de 60 cruzamentos a serem gerenciados, com diversos tempos possíveis a serem agendados devido a heterogeneidade das ruas. A Figura 5.5 demonstra a região utilizada na simulação. Como pode ser observado na Tabela 5.6 a heurística ganha em relação ao tempo das rotas, do consumo de combustível e emissão dos gases.

Figura 5.5: Super Região: 1024 cruzamentos

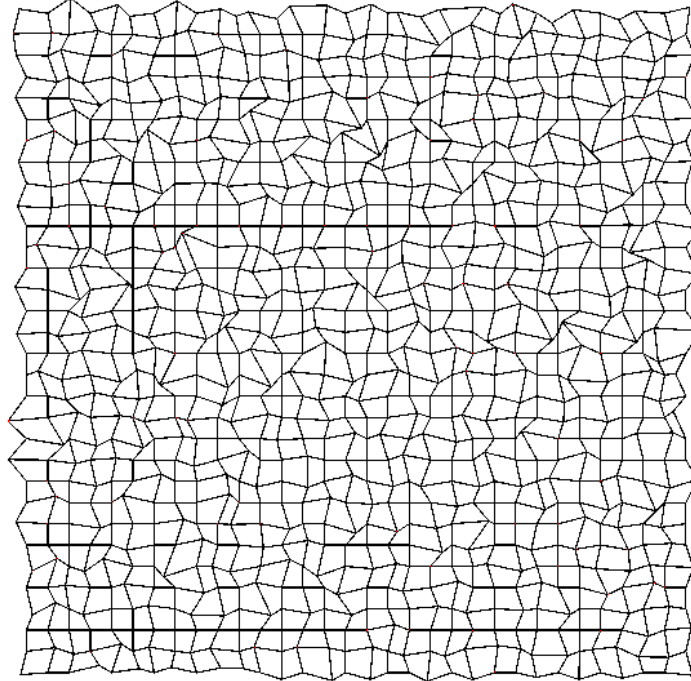


Tabela 5.6: Descrição dos valores da região com 1024 cruzamentos

	Duração (segundos)	CO ₂ (mg/s)	Combustível (ml/s)
SUMO	140163.00	450411233.93	194003.43
Sistema Virtual	130491.00	332133716.66	142772.07

Capítulo 6

Conclusão

6.1 Contribuições

A presente dissertação apresentou uma proposta para o gerenciamento e otimização do tráfego utilizando a resolução de conflitos nos cruzamentos em cenários que possuam apenas carros autônomos. Foi apresentado um modelo matemático simples capaz de resolver os conflitos nas interseções através do gerenciamento dos tempos de chegada nos cruzamentos. Foram impostas restrições para garantir um intervalo de segurança no cruzamento quando houvessem conflitos, de modo que ao atravessar a interseção conflitante, nenhum outro carro estaria presente ao mesmo tempo. Através das restrições, foi possível também garantir que os veículos autônomos obedecessem as regras do trânsito, não excedendo os limites de velocidades impostos nas ruas.

O modelo conseguiu resolver os conflitos para as instâncias dos testes que não possuíam uma grande quantidade de rotas, assim como uma grande quantidade de cruzamentos. Conforme esses números aumentaram, o tempo de solução do problema também aumentou, tornando inviável a utilização do modelo para a obtenção dos tempos que necessitavam uma resposta rápida, e em alguns casos, até minutos foram levados para retornar a solução. Porém, através do modelo foi possível mostrar que o gerenciamento do tráfego apenas com veículos autônomos para otimizar o tempo de chegada é possível.

Para os casos em que o modelo não foi capaz de retornar a solução de forma rápida, foi utilizada uma heurística desenvolvida baseada nas restrições impostas pelo modelo. A heurística seguiu os preceitos de segurança nas interseções ao atravessar um cruzamento

conflitante, assim como as restrições de tráfego com relação ao limite de velocidade. Conforme a quantidade de cruzamentos aumentou, assim como a de veículos, ficou demonstrado através dos testes que a heurística foi capaz de resolver esses problemas com o tempo em milissegundos, o que torna viável a utilização da mesma para casos que o fator tempo real possa ser utilizado.

A heurística demonstrou ser uma solução ótima para os casos em que vários conflitos estão presentes perante todo o mapa utilizado para ser gerenciado, de forma a garantir a segurança de todos, e a otimização do tempo de chegada no destino final. Nos casos em que não há muitos conflitos, a heurística obteve pouco ganho em relação ao SUMO, devendo-se ao fato de como é feito o gerenciamento da velocidade quando há um conflito presente na interseção. Ao ser identificado um conflito, a heurística por usar velocidade constante, na rua antecessora ao cruzamento, modifica a velocidade do carro para a especificada pelo modelo, dessa forma, o conflito é resolvido, e não há paradas desnecessárias no cruzamento. O SUMO por sua vez, garante a aceleração máxima do carro até poucos metros antes do cruzamento, fazendo com que nesses casos, o tempo para percorrer a mesma rua seja menor, porém, ainda fazendo-se necessário parar no cruzamento em alguns casos, diferentemente da heurística.

A heurística para os casos que apresentam muitos conflitos, garante que não haja paradas desnecessárias, e otimiza o tempo de chegada no destino final. Com esse gerenciamento, além de conseguir o objetivo imposto no presente trabalho, introduz a diminuição no consumo de combustível, e na emissão de CO₂, reduzindo o impacto ambiental na área em que a heurística gerencia.

6.2 Limitações

A presente dissertação apesar de introduzir um modelo matemático para o gerenciamento do tráfego de veículos autônomos no mundo real, utilizou apenas simulações estáticas, dessa forma, não utilizando o fator tempo real nos casos apresentados anteriormente.

A heurística demonstrou ser uma saída para os casos que apresentam uma grande quantidade de ruas e veículos, porém, a heurística não utiliza nenhum processo de otimização sobre a solução encontrada, em outras palavras, a heurística pode não retornar como solução

o ótimo do problema proposto.

Em casos que são apresentados poucas rotas, assim como a quantidade de ruas, a heurística apresentou pouca melhora no ganho de tempo em relação ao simulador, demonstrando não ser tão eficiente em comparação aos casos em que são utilizados uma grande quantidade de ruas e rotas.

6.3 Trabalhos Futuros

A heurística por possuir um tempo de resolução dos problemas independente do tamanho da instância gerenciada na casa dos milissegundos, pode ser utilizada para trabalhos em que o fator tempo real seja considerado, desse modo, tornando ainda mais realista o problema enfrentado ao gerenciar o tráfego dos veículos atualmente. Para a simulação não ser estática, a utilização do TraCi (Traffic Control Interface) através da modificação do código utilizando Python é sugerida para ser integrada ao SUMO, podendo ser adaptada a solução rápida da heurística, com o fator de simulação da ferramenta.

É sugerido que haja um pré-processamento dos tempos a serem utilizados de forma que não aconteça uma grande diferença de velocidades na transição das ruas entre os cruzamentos quando os limites de velocidades possuírem uma diferença considerável, por exemplo, de uma rua paralela a uma BR. Para esse processamento, é necessário levar em consideração os tempos anteriores, assim como o tamanho da rua, dessa forma, sendo possível identificar um valor a ser utilizado nessa transição.

Um outro ponto que pode ser abordado é a modificação do cálculo utilizado para identificar o tempo da travessia de cada rua. A aceleração é um fator que deverá ser levado em conta, de tal forma que quando identificado um conflito, o carro comece a desacelerar perto da interseção, garantindo a otimização do tempo na rua em que o conflito acontece, consequentemente, diminuindo também o tempo de chegada no destino final.

Para o modelo, é sugerido a modificação para que quando necessário, e se possível, a rota possa ser alterada de forma que garanta a otimização do tempo, e a chegada no destino, evitando possíveis conflitos que pudessem existir.

Por fim, para os casos em que é necessário o gerenciamento dos tempos em um determinado local, a heurística pode ser adaptada e utilizada para a resolução do problema proposto,

não limitando-se apenas ao gerenciamento do tráfego dos veículos. Um exemplo conforme já foi citado (Seção 1.2), seria a utilização na aplicação de Elon Musk no gerenciamento das rotas subterrâneas de modo a garantir a inexistência dos conflitos ao chegar no destino desejado passando pelos cruzamentos que possam apresentar conflitos ocasionados na translocação de um ponto a outro.

Bibliografia

- [Ahn, Colombo e Vecchio 2014]AHN, H.; COLOMBO, A.; VECCHIO, D. D. Supervisory Control for Intersection Collision Avoidance in the Presence of Uncontrolled Vehicles. *2014 American Control Conference*, n. 12, 2014.
- [Alves e Delgado 1997]ALVES, R.; DELGADO, C. Programação Linear Inteira. *Faculdade de Economia do Porto*, n. 1, 1997.
- [Arenales et al. 2007]ARENALES, M. et al. *Pesquisa Operacional*. São Paulo,SP,Brasil: Elsevier, 2007. 524 p. ISBN 978-85-352-5193-7.
- [Ashtiani, Fayazi e Vahidi 2018]ASHTIANI, F.; FAYAZI, S. A.; VAHIDI, A. Multi-Intersection Traffic Management for Autonomous Vehicles via Distributed Mixed Integer Linear Programming. *2018 Annual American Control Conference (ACC)*, 2018.
- [Belkhouche 2017]BELKHOUCHE, F. Control of autonomous vehicles at an unsignalized intersection. *American Control Conference (ACC)*, n. 1, 2017.
- [Blokpoel et al. 2015]BLOKPOEL, R. et al. Emission optimized control for isolated intersections. *ITS World Congress*, n. 22, 2015.
- [Budán et al. 2018]BUDAN, G. et al. An analysis of vehicle-to-infrastructure communications for non signalised intersection control under mixed driving behaviour. *Analog Integrated Circuits and Signal Processing*, 2018.
- [Campos et al. 2014]CAMPOS, G. R. et al. Cooperative receding horizon conflict resolution at traffic intersections. *IEEE Conference on Decision and Control*, n. 53, 2014.

- [Campos et al. 2017]CAMPOS, G. R. de et al. Traffic coordination at road intersections: Autonomous decision-making algorithms using model-based heuristics. *IEEE Intelligent Transportation Systems Magazine*, n. 1, 2017.
- [Campos, Falcone e Sjöberg 2015]CAMPOS, G. R. de; FALCONE, P.; SJÖBERG, J. Traffic safety at intersections: a priority based approach for cooperative collision avoidance. *3rd International Symposium on Future Active Safety Technology Towards zero traffic accidents*, n. 3, 2015.
- [Charalampidis e Gillet 2014]CHARALAMPIDIS, A. C.; GILLET, D. Speed Profile Optimization for Vehicles Crossing an Intersection Under a Safety Constraint. *European Control Conference (ECC)*, n. 10, 2014.
- [Dai et al. 2016]DAI, P. et al. Quality-of-Experience-Oriented Autonomous Intersection Control in Vehicular Networks. *IEEE Transactions on Intelligent Transportation Systems*, n. 17, 2016.
- [Dresner e Stone 2008]DRESNER, K.; STONE, P. Multiagent Traffic Management: A Reservation-Based Intersection Control Mechanism. *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, n. 31, 2008.
- [Fayazi, Vahidi e Luckow 2017]FAYAZI, S. A.; VAHIDI, A.; LUCKOW, A. Optimal Scheduling of Autonomous Vehicle Arrivals at Intelligent Intersections via MILP. *American Control Conference (ACC)*, n. 1, 2017.
- [Funke et al. 2017]FUNKE, J. et al. Collision Avoidance and Stabilization for Autonomous Vehicles in Emergency Scenarios. *IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY*, n. 25, 2017.
- [Gong, Shen e Du 2016]GONG, S.; SHEN, J.; DU, L. Constrained optimization and distributed computation based car following control of a connected and autonomous vehicle platoon. *Transportation Research Part B*, n. 96, 2016.
- [Haklay e Weber 2008]HAKLAY, M.; WEBER, P. OpenStreetMap: User-Generated Street Maps. *IEEE Pervasive Computing*, n. 7, 2008.

- [Hallmann 2011]HALLMANN, H. V. *Comparação entre Softwares Simuladores de Trânsito*. 104 p. — Instituto de Informática, Universidade Federal do Rio Grande do Sul, 2011.
- [Hausknecht, Au e Stone 2011]HAUSKNECHT, M.; AU, T.-C.; STONE, P. Autonomous Intersection Management: Multi-Intersection Optimization. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, n. 1, 2011.
- [Hult et al. 2015]HULT, R. et al. An approximate solution to the optimal coordination problem for autonomous vehicles at intersections. *American Control Conference (ACC)*, n. 2015, 2015.
- [Isele et al. 2017]ISELE, D. et al. Navigating Intersections with Autonomous Vehicles using Deep Reinforcement Learning. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, n. 30, 2017.
- [Jin et al. 2012]JIN, Q. et al. Multi-Agent Intersection Management for Connected Vehicles using an Optimal Scheduling Approach. *International Conference on Connected Vehicles and Expo*, n. 31, 2012.
- [Jo et al. 2014]JO, K. et al. Development of Autonomous Car—Part I: Distributed System Architecture and Development Process. *IEEE Transactions on Industrial Eletronics*, n. 61, 2014.
- [Jo et al. 2015]JO, K. et al. Development of Autonomous Car—Part II: A Case Study on the Implementation of an Autonomous Driving System Based on Distributed Architecture. *IEEE Transactions on Industrial Eletronics*, n. 62, 2015.
- [Kamal et al. 2015]KAMAL, M. A. S. et al. A Vehicle-Intersection Coordination Scheme for Smooth Flows of Traffic Without Using Traffic Lights. *IEEE Transactions on Intelligent Transportation Systems*, n. 16, 2015.
- [Kamal, Imura e Ohata 2013]KAMAL, M. A. S.; IMURA, J.; OHATA, A. Coordination of Automated Vehicles at a Traffic-Lightless Intersection. *Intelligent Transportation Systems - (ITSC), 16th International IEEE Conference on*, n. 2013, 2013.

- [Lamouik, Yahyaouy e Sabri 2017]LAMOUIK, I.; YAHYAOUY, A.; SABRI, M. A. Smart multi-agent traffic coordinator for autonomous vehicles at intersections. *Advanced Technologies for Signal and Image Processing (ATSIP), International Conference on*, n. 2017, 2017.
- [Lee e Park 2012]LEE, J.; PARK, B. Development and Evaluation of a Cooperative Vehicle Intersection Control Algorithm Under the Connected Vehicles Environment. *IEEE Transactions on Intelligent Transportation Systems*, n. 13, 2012.
- [Lim et al. 2018]LIM, W. et al. Hierarchical Trajectory Planning of an Autonomous Car Based on the Integration of a Sampling and an Optimization Method. *IEEE Transactions on Intelligent Transportation*, n. 19, 2018.
- [Makarem e Gillet 2011]MAKAREM, L.; GILLET, D. Decentralized Coordination of Autonomous Vehicles at intersections. *IFAC Proceedings Volumes*, n. 44, 2011.
- [Makarem e Gillet 2012]MAKAREM, L.; GILLET, D. Fluent coordination of autonomous vehicles at intersections. *Systems, Man, and Cybernetics (SMC), IEEE International Conference on*, n. 2012, 2012.
- [Medina, Wouw e Nijmeijer 2015]MEDINA, A. I. M.; WOUW, N. van de; NIJMEIJER, H. Automation of a T-intersection using virtual platoons of cooperative autonomous vehicles. *Intelligent Transportation Systems (ITSC), IEEE 18th International Conference on*, n. 2015, 2015.
- [Milanés et al. 2012]MILANÉS, V. et al. Controller for Urban Intersections Based on Wireless Communications and Fuzzy Logic. *IEEE Transactions on Intelligent Transportation Systems*, n. 2012, 2012.
- [Neis e Zipf 2012]NEIS, P.; ZIPF, A. Analyzing the Contributor Activity of a Volunteered Geographic Information Project — The Case of OpenStreetMap. *ISPRS International Journal of Geo-Information*, n. 1, 2012.
- [Pourmehrab, Eleftheriadou e Ranka 2017]POURMEHRAB, M.; ELEFTERIADOU, L.; RANKA, S. Smart Intersection Control Algorithms for Automated Vehicles. *Proceedings of 2017 Tenth International Conference on Contemporary Computing*, n. 10, 2017.

- [Pourmehrab et al. 2017]POURMEHRAB, M. et al. Optimizing Signalized Intersections Performance under Conventional and Automated Vehicles Traffic. *Transportation Research Part C: Emerging Tech.*, n. 1, 2017.
- [Riegger et al. 2016]RIEGGER, L. et al. Centralized mpc for autonomous intersection crossing. *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, n. 19, 2016.
- [Sadigh et al. 2016]SADIGH, D. et al. Planning for Autonomous Cars that Leverage Effects on Human Actions. *Conference, Robotics: Science and Systems*, n. 12, 2016.
- [Sharon e Stone 2017]SHARON, G.; STONE, P. A Protocol for Mixed Autonomous and Human-Operated Vehicles at Intersections. *The 2nd International Workshop on Agent-based modeling of urban systems (ABMUS 2017)*, Sao Paulo, Brasil, n. 2, 2017.
- [Wu, Perronnet e Abbas-Turki 2014]WU, J.; PERRONNET, F.; ABBAS-TURKI, A. Cooperative vehicle-actuator system: a sequencebased framework of cooperative intersections management. *IET Intelligent Transport Systems*, n. 8, 2014.
- [Wu, Guizani e Hsieh 2016]WU, T.-Y.; GUIZANI, N.; HSIEH, C.-Y. An efficient adaptive intelligent routing system for multi-intersections. *Wireless Communications and Mobile Computing*, n. 16, 2016.
- [Wuthishuwong e Traechtler 2013]WUTHISHUWONG, C.; TRAECHTLER, A. Coordination of Multiple Autonomous Intersections by Using Local Neighborhood Information. *International Conference on Connected Vehicles and Expo (ICCVE)*, n. 1, 2013.
- [Zhang et al. 2013]ZHANG, K. et al. Analysis and Modeled Design of One State-Driven Autonomous Passing-through Algorithm for Driverless Vehicles at Intersections. *IEEE International Conference on Computational Science and Engineering*, n. 16, 2013.
- [Zheng et al. 2017]ZHENG, Y. et al. Research on cooperative vehicle intersection control scheme without using traffic lights under the connected vehicles environment. *Advances in Mechanical Engineering*, n. 12, 2017.

[Zhu e Ukkusuri 2015]ZHU, F.; UKKUSURI, S. V. A linear programming formulation for autonomous intersection control within a dynamic traffic assignment and connected vehicle environment. *Transportation Research Part C*, n. 2015, 2015.

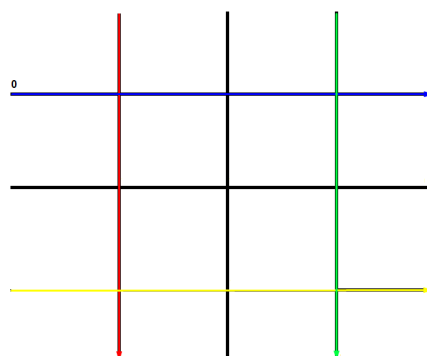
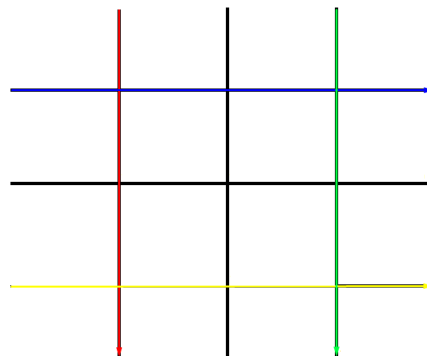
[Zohdy e Rakha 2012]ZOHDY, I.; RAKHA, H. Optimizing driverless vehicles at intersections. *ITS World Congress, Vienna, Austria*, n. 19, 2012.

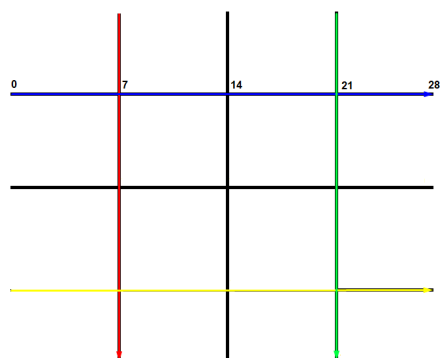
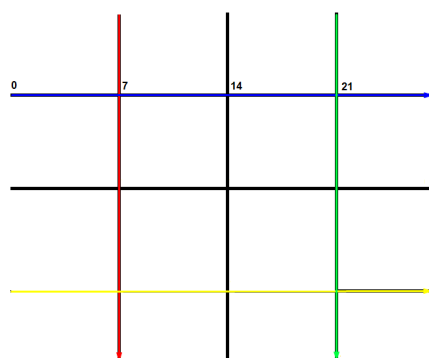
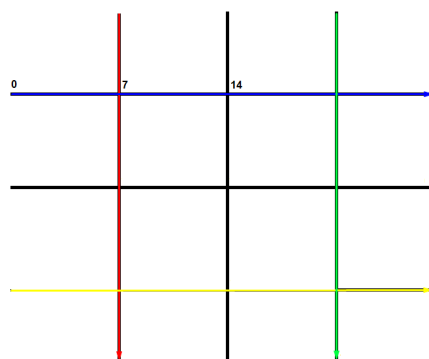
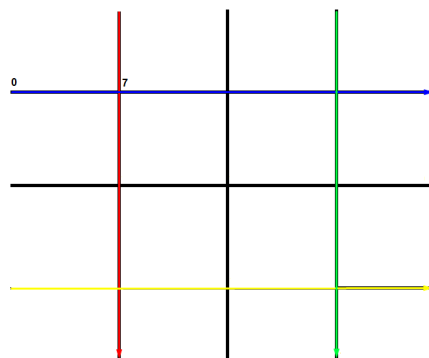
[Zohdy, Kamalanathsharma e Rakha 2012]ZOHDY, I. H.; KAMALANATHSHARMA, R. K.; RAKHA, H. Intersection Management for Autonomous Vehicles using iCACC. *International IEEE Conference on Intelligent Transportation Systems*, n. 15, 2012.

Apêndice A

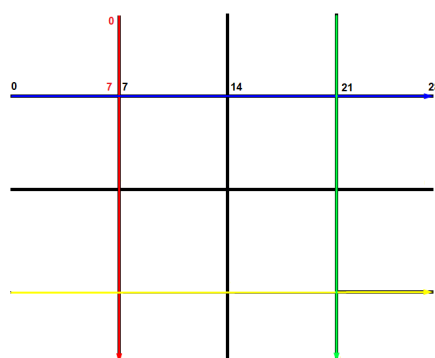
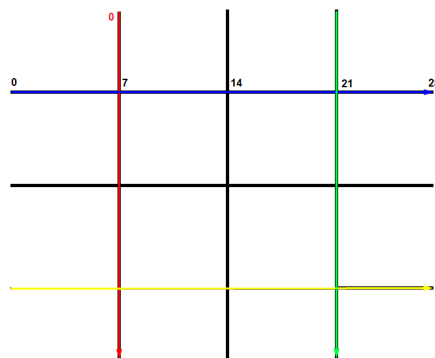
Passo a passo da resolução da Heurística

Neste apêndice encontram-se os passos utilizados pela heurística para resolução do problema apresentado para a compreensão de como é solucionado os casos de conflitos, assim como o agendamento dos tempos em cada cruzamento, conforme apresentado previamente na Seção 4.2.

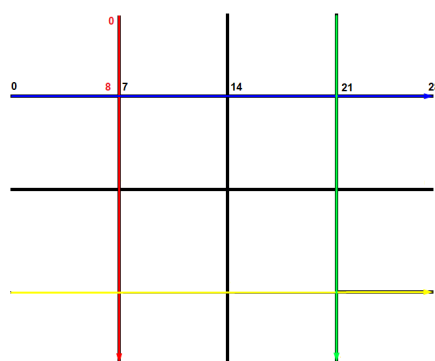


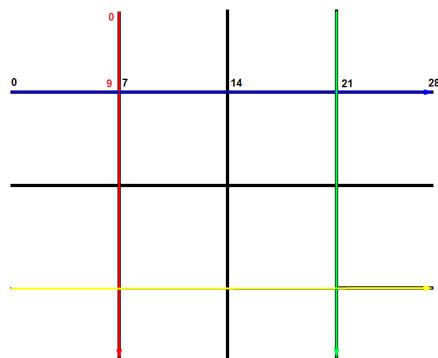


Neste ponto, todos os tempos presentes na primeira rota foram agendados.

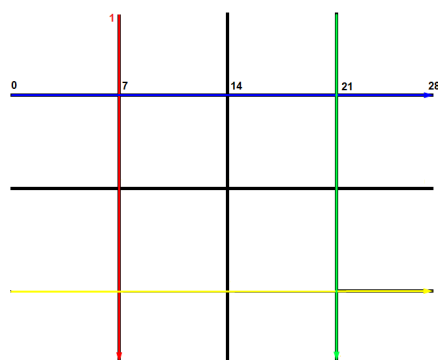
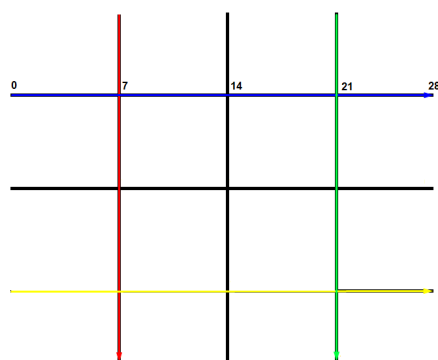


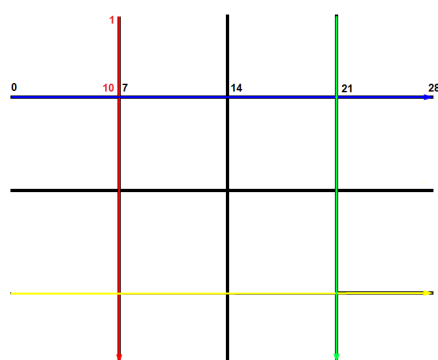
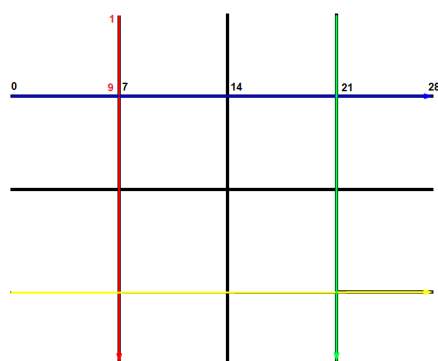
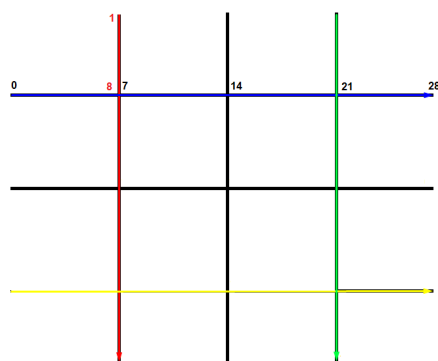
Foi identificado um conflito nesse cruzamento, a heurística utilizará os possíveis valores de tempos para verificar se é possível solucionar o conflito, dessa forma fazendo o agendamento do cruzamento na rota especificada.



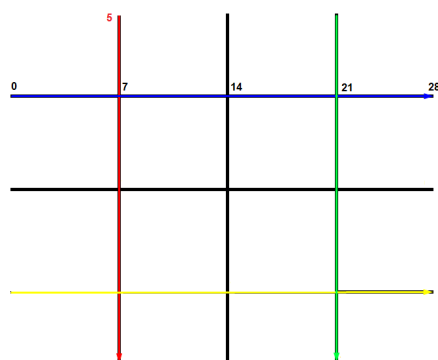
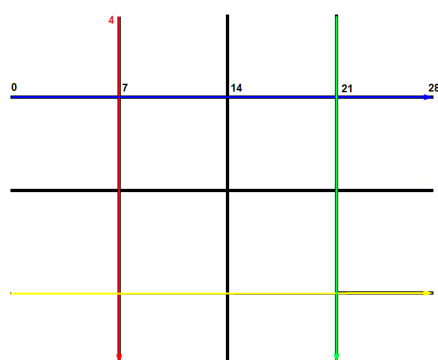
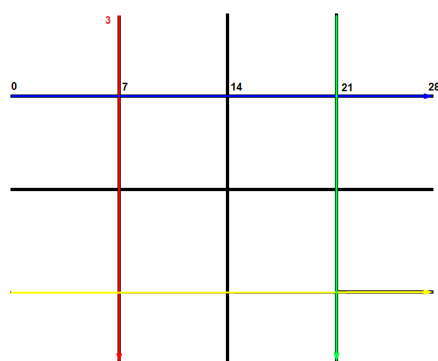
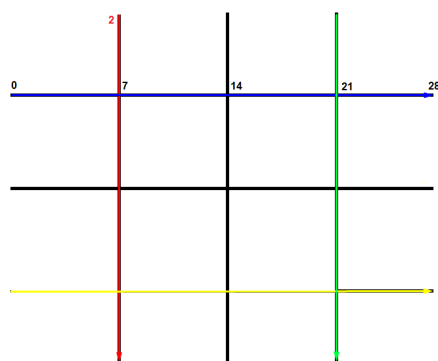


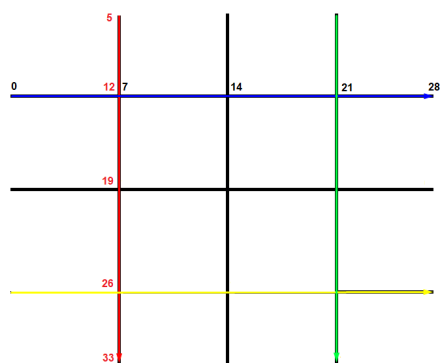
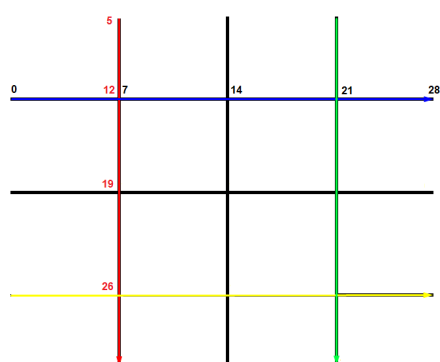
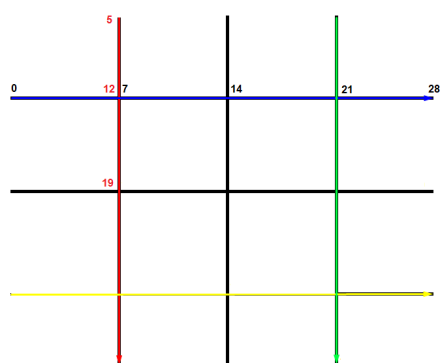
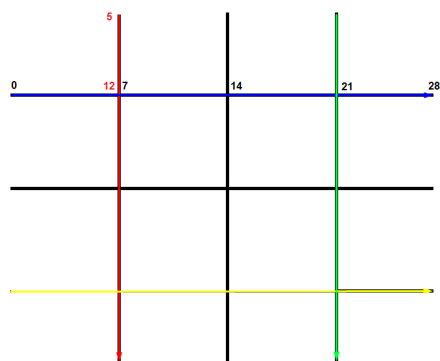
A heurística identificou que não foi possível agendar os tempos desejado para o cruzamento devido ao intervalo de segurança não ser obedecido, são desfeitos os tempos previamente agendados, acrescido 1 no tempo inicial, e então o processo de agendamento é repetido.

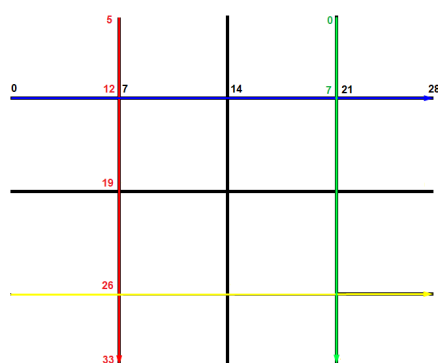
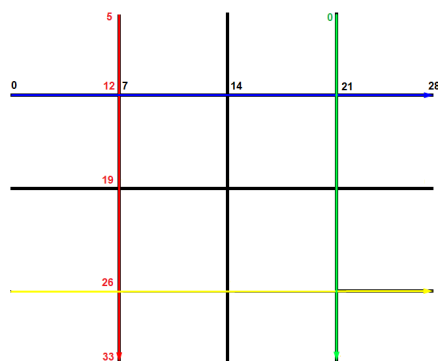




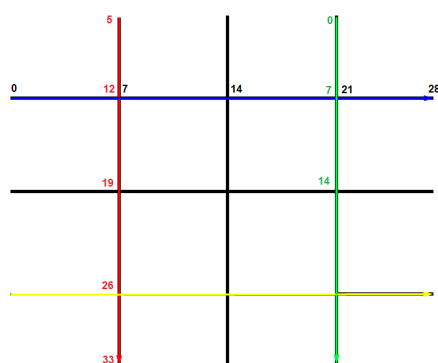
Mais uma vez, a heurística identificou que não é possível agendar os tempos especificados por não obedecer o intervalo de segurança estimado. São desfeitos os tempos previamente agendados, acrescido 1 no tempo inicial, e o processo de agendamento é refeito. Como pode ser observado nos próximos passos, o agendamento só será possível quando o tempo inicial chegar em 5.

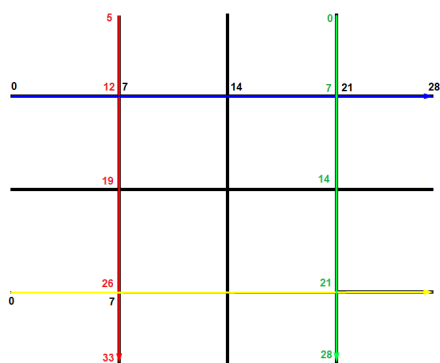
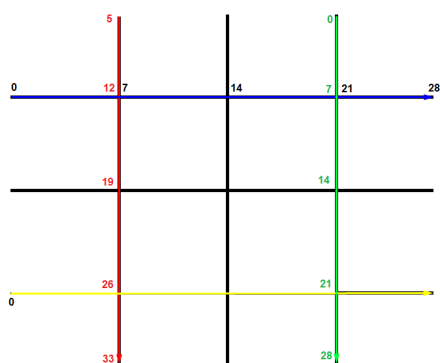
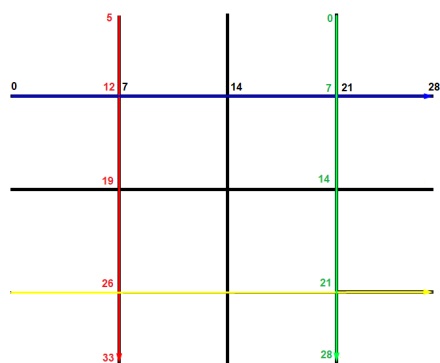
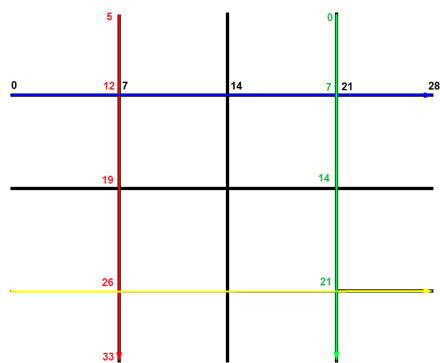


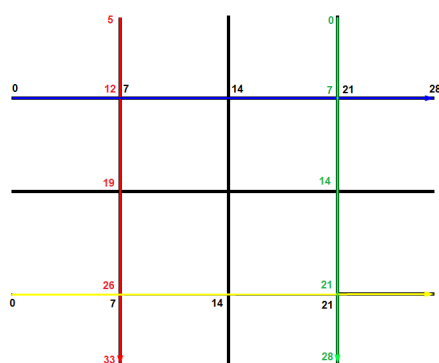
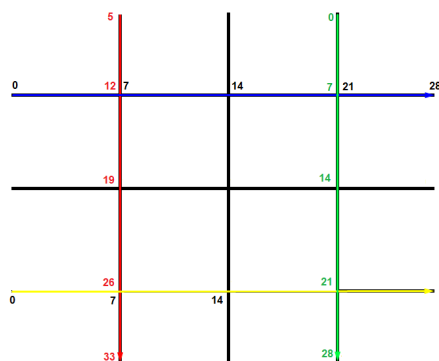




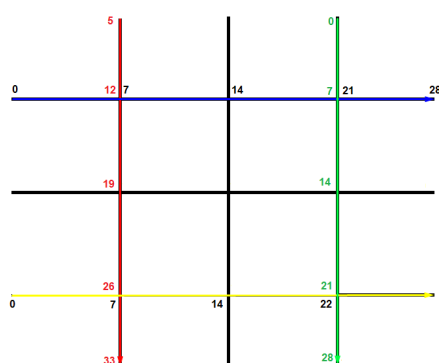
Nesse ponto, a heurística identificou que houve um conflito de cruzamento em diferentes rotas, porém, o tempo desejado para a rota 3 obedece todas as restrições impostas, dessa forma, sendo agendado sem causar conflito.

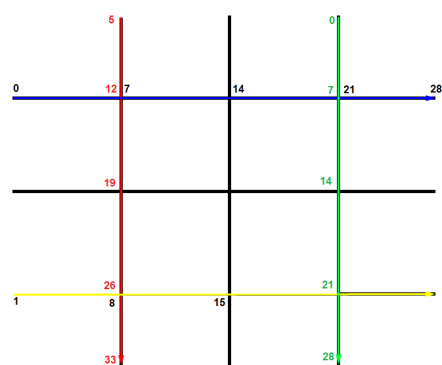
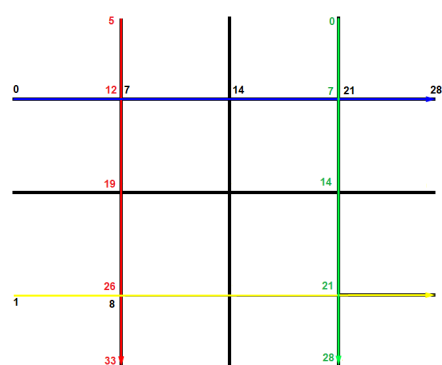
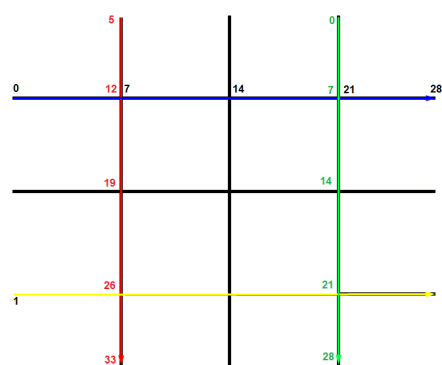
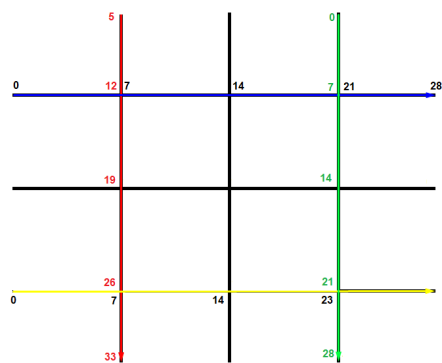


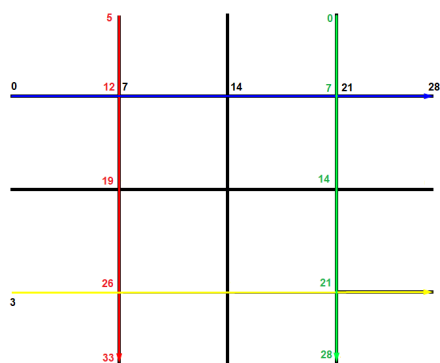
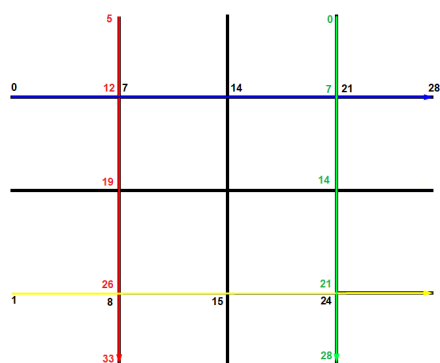
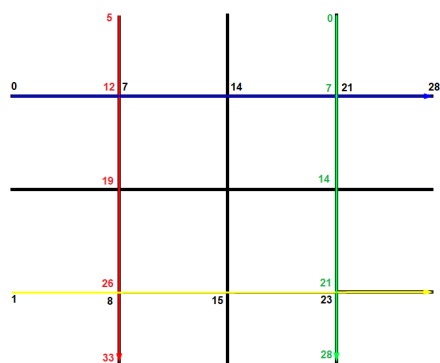
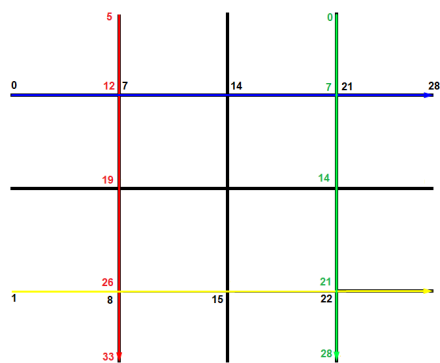


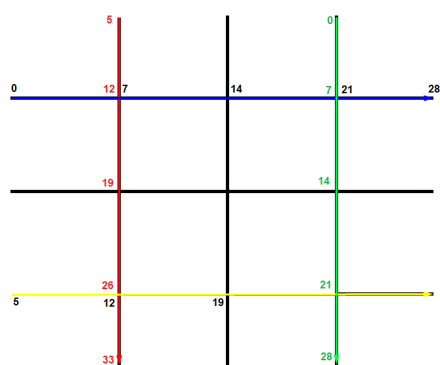
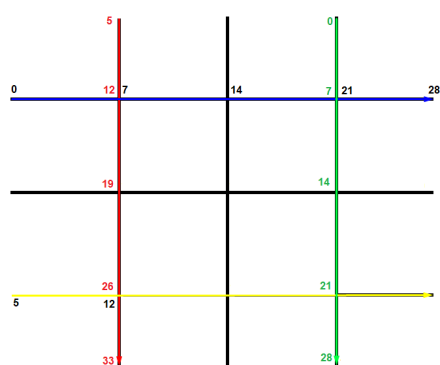
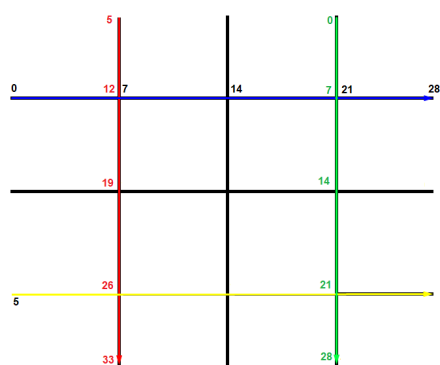
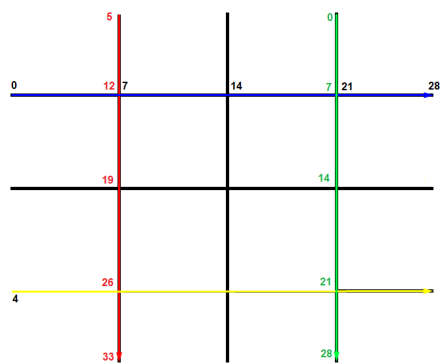


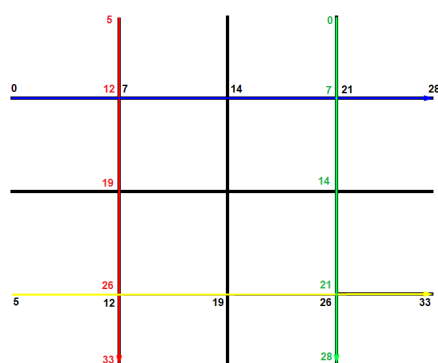
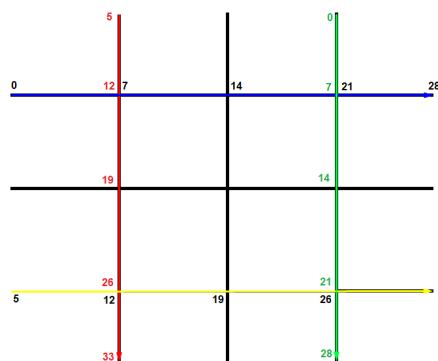
Mais uma vez, a heurística identifica um cruzamento com conflito nos tempos desejados que não obedecem o intervalo de segurança, então, a heurística irá tentar agendar com os tempos especificados, caso não seja possível, serão desfeitos os tempos, acrescido 1 no tempo inicial, e refeito o processo de agendamento até que todas as restrições tenham sido obedecidas, conforme pode ser observado nos próximos passos.











A heurística conseguiu agendar todos os tempos das rotas especificadas, resolver os conflitos que apareceram, agendar a velocidade mais rápida quando foi possível, e retornar os tempos que cada carro deve obedecer para diminuir o tempo gasto nos cruzamentos.

Catálogo na publicação
Seção de Catalogação e Classificação

S586g Silva, Victor Hugo Azevêdo da.

Gerenciamento de interseções para otimização do tráfego de carros autônomos / Victor Hugo Azevêdo da Silva. - João Pessoa, 2020.
75 f. : il.

Orientação: Claurirton de Albuquerque Siebra.
Coorientação: Anand Subramanian.
Dissertação (Mestrado) - UFPB/Informática.

1. Informática - Automóveis. 2. Tráfego - Otimização.
3. Carros autônomos. 4. Simulação computacional. I.
Siebra, Claurirton de Albuquerque. II. Subramanian,
Anand. III. Título.

UFPB/BC

CDU 004:629.3(043)