

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**INVESTIGAÇÃO DE ALGORITMOS PARA O
CONTROLE DE CRUZAMENTOS EM REDES
DE TRANSPORTE**

TRABALHO DE GRADUAÇÃO

Liza Lunardi Lemos

Santa Maria, RS, Brasil

2015

INVESTIGAÇÃO DE ALGORITMOS PARA O CONTROLE DE CRUZAMENTOS EM REDES DE TRANSPORTE

Liza Lunardi Lemos

Trabalho de Graduação apresentado ao Curso de Bacharelado em Ciência da
Computação da Universidade Federal de Santa Maria (UFSM, RS), como
requisito parcial para a obtenção do grau de

Bacharel em Ciência da Computação

Orientadora: Prof^a. Dr. Marcia Pasin

**Trabalho de Graduação N.407
Santa Maria, RS, Brasil**

2015

Lunardi Lemos, Liza

Investigação de Algoritmos para o Controle de Cruzamentos em
Redes de Transporte / por Liza Lunardi Lemos. – 2015.

38 f.: il.; 30 cm.

Orientadora: Marcia Pasin

Monografia (Graduação) - Universidade Federal de Santa Maria,
Centro de Tecnologia, Curso de Bacharelado em Ciência da Computa-
ção, RS, 2015.

1. Redes de transporte. 2. Interseção. 3. Trânsito. 4. Simulação.
I. Pasin, Marcia. II. Título.

© 2015

Todos os direitos autorais reservados a Liza Lunardi Lemos. A reprodução de partes ou do todo deste trabalho só poderá ser feita mediante a citação da fonte.

E-mail: llemos@inf.ufsm.br

**Universidade Federal de Santa Maria
Centro de Tecnologia
Curso de Bacharelado em Ciência da Computação**

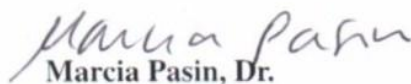
A Comissão Examinadora, abaixo assinada,
aprova o Trabalho de Graduação

**INVESTIGAÇÃO DE ALGORITMOS PARA O CONTROLE DE
CRUZAMENTOS EM REDES DE TRANSPORTE**

elaborado por
Liza Lunardi Lemos

como requisito parcial para obtenção do grau de
Bacharel em Ciência da Computação

COMISSÃO EXAMINADORA:



Marcia Pasin, Dr.
(Presidente/Orientadora)



Alice Kozakevicius, Dr. (UFSM)



João Carlos Damasceno de Lima, Dr. (UFSM)

Santa Maria, 11 de Dezembro de 2015.

AGRADECIMENTOS

Agradeço primeiramente à UFSM e ao Curso de Ciência da Computação, e seus professores pelos ensinamentos passados.

Agradeço ao Programa de Educação Tutorial Ciência da Computação (PET CC), o qual foi de grande importância para o meu crescimento acadêmico, proporcionando a oportunidade de ampliar os meus conhecimentos, e para meu crescimento pessoal.

Agradeço à professora Marcia Pasin pela orientação, ajuda e dedicação e pela troca de opiniões, de experiências e de ensinamentos que ajudaram a fundamentar este trabalho.

Agradeço aos meus colegas de curso, Giane, Jéssica, Maia, Ricardo, Vando, Tiago, Andressa e Ana (mesmo que por curto período), pela ajuda em estudos pré prova, pelas horas de sono perdidas fazendo trabalho. Mas, principalmente, pelas momentos de risadas e rodadas de truco entre uma aula e outra, pois são os momentos com os amigos que formam as lembranças e a compreensão de cada um em entender o que passamos e que juntos superamos mais uma etapa de nossas vidas, tão sonhada e idealizada.

Agradeço à minha colega de apartamento, Natalia, pelas horas de companheirismo.

Agradeço aos meus amigos de Santo Ângelo, principalmente a Nicole, Juliane, Rodrigo, Carol, Hannah e Marina, que mesmo estando longe uns dos outros, mantivemos contato, nos vendo sempre que possível e ajudando uns aos outros na realização de nossos sonhos.

Agradeço à toda a minha família que sempre me apoiou e esteve torcendo por mim para que tudo desse certo.

Por fim, agradeço aos meus pais, que desde o início me apoiaram e me incentivaram, de modo que eu chegasse até aqui. Faltam-me palavras para descrever a gratidão de tudo que fizeram por mim e de quanto eu os amo.

RESUMO

Trabalho de Graduação
Curso de Bacharelado em Ciência da Computação
Universidade Federal de Santa Maria

INVESTIGAÇÃO DE ALGORITMOS PARA O CONTROLE DE CRUZAMENTOS EM REDES DE TRANSPORTE

AUTORA: LIZA LUNARDI LEMOS

ORIENTADORA: MARCIA PASIN

Local da Defesa e Data: Santa Maria, 11 de Dezembro de 2015.

Gerenciamento do trânsito é um assunto recorrente no contexto de cidades inteligentes e um ponto importante é a gerência de interseções. Diferentes técnicas podem ser usadas para controlar interseções em redes de transporte, melhorar o serviço e minimizar a emissão de poluentes. Este trabalho avalia diferentes algoritmos de controle de interseções em uma rede de transporte com múltiplas interseções. Os algoritmos em questão pertencem a dois grupos distintos: orientados a comboios, visando eficiência em termos de vazão, e orientados a escolhas individuais, visando justiça e minimização do tempo de atendimento. Os algoritmos foram avaliados com o apoio de simulação computacional, usando a ferramenta SUMO (Simulation of Urban MObility), específica para redes de transporte. As métricas usadas para a avaliação incluem cálculo da vazão e emissão de poluentes. Foi observado que algoritmos eficientes em vazão também fornecem valores mais interessantes para a emissão de poluentes.

Palavras-chave: Redes de transporte. Interseção. Trânsito. Simulação.

ABSTRACT

Undergraduate Final Work
Graduate Program in Computer Science
Federal University of Santa Maria

EVALUATION OF ALGORITHMS TO INTERSECTION MANAGEMENT IN TRANSPORTATION NETWORKS WITH DIFFERENT INTERSECTIONS

AUTHOR: LIZA LUNARDI LEMOS

ADVISOR: MARCIA PASIN

Defense Place and Date: Santa Maria, December 11st, 2015.

Traffic management is nowadays a major relevant subject in the context of smart cities and an important issue is efficient intersection control. Different techniques can be used to deal with intersection control in transport networks, aiming to improve service and minimize the emission of pollutants. In this work, we evaluate different intersections control algorithms in a transportation network with multiple intersections. The algorithms in question are classified into two distinct groups: platoon-oriented, to provide efficiency in terms of throughput, and individual choices-oriented, aiming fairness and focused on minimizing waiting time. The algorithms were evaluated with the support of computer simulation, using the SUMO (Simulation of Urban MObility), a simulator specific to transportation networks. Metrics used for evaluation include throughput and pollutant emissions. It was observed that efficient algorithms flow also give smaller values for the emissions pollutants.

Keywords: Adaptive Control, Intersection Control, Traffic, Simulation.

LISTA DE FIGURAS

Figura 3.1 – Ferramentas utilizadas e suas ligações	20
Figura 3.2 – Arquivos de entrada e saída do SUMO	23
Figura 3.3 – Fragmento de um arquivo XML.....	24
Figura 3.4 – Elementos de um grafo	25
Figura 3.5 – Rede de transporte com quatro interseções indicando direção dos fluxos e nodos adicionais para a redução de velocidade dos veículos	26
Figura 3.6 – Captura de tela da rede de transporte no SUMO com múltiplas interseções: uma via arterial e quatro vias coletoras	26
Figura 3.7 – Exemplo do algoritmo maior fila primeiro, mostrando o tamanho das filas ...	27
Figura 3.8 – Exemplo do algoritmo <i>zipper merge</i> , o qual passa k veículos por vez.....	28
Figura 3.9 – Veículos e seus tempos calculados a partir do algoritmo FIFO, no qual o carro com o menor tempo de chegada cruza a interseção primeiro.....	29
Figura 4.1 – Execução do algoritmo <i>zipper merge</i> e as respectivas filas formadas durante a execução da simulação.....	32
Figura 4.2 – Captura de tela com a execução do algoritmo <i>fifo</i> e as respectivas filas formadas durante a execução da simulação	33
Figura 4.3 – Captura de tela com a execução do algoritmo <i>lqf</i> , o qual não formou filas nas vias durante a execução da simulação.....	33
Figura 4.4 – Vazão	34
Figura 4.5 – Emissão de dióxido de carbono (CO ₂)	34
Figura 4.6 – Consumo de combustível por veículo na simulação	35

LISTA DE TABELAS

Tabela 4.1 – Sigla dos algoritmos aplicados às métricas.....	31
--	----

LISTA DE ABREVIATURAS E SIGLAS

CET	Companhia de Engenharia de Tráfego
CO	Monóxido de Carbono
CO ₂	Dióxido de Carbono
CONTRAN	Conselho Nacional de Trânsito
CTB	Código de Trânsito Brasileiro
DENATRAN	Departamento Nacional de Trânsito
EPTC	Empresa Pública de Transporte e Circulação
FIFO	<i>First In, First Out</i>
HBEFA	<i>Handbook Emission Factors for Road Transport</i>
HC	Hidrocarbonetos
IPCC	<i>Intergovernmental Panel on Climate Change</i>
MANETs	<i>Mobile Ad Hoc NETworks</i>
NO _x	Óxidos de Nitrogênio
SUMO	<i>Simulation of Urban MObility</i>
TraCI	<i>Traffic Control Interface</i>
TraCI4J	<i>Traffic Control Interface for Java</i>
VANETs	<i>Vehicular Ad Hoc NETworks</i>
V2I	<i>Vehicle to Infrastructure</i>
V2V	<i>Vehicle to Vehicle</i>
VTL	<i>Virtual Traffic Lights</i>
XML	<i>eXtensible Markup Language</i>

SUMÁRIO

1 INTRODUÇÃO	11
1.1 Motivação	11
1.2 Objetivos	13
1.2.1 Objetivo geral	13
1.2.2 Objetivos específicos	13
1.3 Justificativa	14
1.4 Organização do texto	14
2 REVISÃO DA LITERATURA	16
3 FERRAMENTAS E MÉTODOS UTILIZADOS	20
3.1 SUMO: Simulation of Urban MObility	20
3.2 TraCI4J	22
3.3 XMLSpy	23
3.4 Gnuplot	24
3.5 Definindo uma rede de transporte com muitos cruzamentos	24
3.6 Algoritmos	25
3.6.1 Maior fila primeiro	27
3.6.2 <i>Zipper Merge</i> (intercalação de filas)	27
3.6.3 FIFO (<i>First In, First Out</i>)	28
4 AVALIAÇÃO EXPERIMENTAL	30
4.1 Métricas	30
4.1.1 Vazão	30
4.1.2 Emissão de poluentes	30
4.2 Experimentos	31
5 CONCLUSÃO E TRABALHOS FUTUROS	36
REFERÊNCIAS	37

1 INTRODUÇÃO

1.1 Motivação

Com o crescente aumento de veículos nas vias, fenômeno que é presenciado também em nossa cidade, há cada vez mais congestionamentos impactando na demora do deslocamento das pessoas (SOBRAL et al., 2014). Ao longo de cada dia acumulam-se cada vez mais veículos nas vias, sendo que esse acúmulo é influenciado por diversos fatores, como a hora do dia, a presença de semáforos e outros eventos, tais como acidentes e agentes meteorológicos (chuva, névoa, neve, gelo).

Enfrenta-se também outra dificuldade com esse crescente número de veículos automotores, que é a ampla emissão de gases poluentes e a alta demanda por combustíveis. Nas grandes cidades, os automóveis destacam-se como causadores da emissão de substâncias tóxicas, as quais têm um efeito negativo sobre o sistema respiratório e circulatório. Contudo, não são somente humanos que sofrem com a emissão das substâncias tóxicas. A emissão de poluentes tóxicos impactam também na fauna e na flora, como a acidificação de rios e florestas (AZUGA, 2000).

Os gases poluentes emitidos pelos veículos que mais se destacam por serem prejudiciais à saúde e ao meio ambiente são: monóxido de carbono (CO), dióxido de carbono (CO₂), hidrocarbonetos (HC) e óxidos de nitrogênio (NO_x). Veículos movidos a diesel (ônibus e caminhões) são responsáveis pela maior fração da emissão de óxido de nitrogênio e enxofre, enquanto veículos movidos à gasolina ou álcool são responsáveis pela maior parte da emissão de monóxido de carbono e hidrocarbonetos. Mesmo que a emissão de cada veículo não seja relevante, numa grande quantidade de veículos, como no grandes centros, a geração desses poluentes é de toneladas por dia (TEIXEIRA; FELTES; SANTANA, 2008). O dióxido de carbono é o gás produzido pela combustão de carvão, petróleo e gás. Ele atua na superfície terrestre, causando o fenômeno chamado de efeito estufa (GOLDEMBERG, 2000).

Desta forma, precisa-se otimizar o escoamento dos veículos, pois redução nos tempos de deslocamentos implica em eficiência, mas também precisa-se de soluções menos poluentes. Para melhorar o escoamento dos carros, pode-se utilizar de várias técnicas, como os semáforos inteligentes, que são calibrados de acordo com o fluxo de veículos. Sensores são instalados no asfalto para captar o fluxo de veículos. Alternativamente, câmeras podem ser usadas para

coletar informações sobre o fluxo de veículos. Semáforos inteligentes já são uma realidade nas grandes cidades como São Paulo, Curitiba, Porto Alegre, entre outras cidades. Um dos problemas dos semáforos inteligentes são os altos custos de instalação e manutenção que eles representam para os cofres públicos. Outro problema é que a decisão é tomada com base no fluxo e não em cada carro individualmente. Justiça pode ser penalizada. Justiça significa que o carro que chega antes deve passar antes. Se o tempo de chegada não é levado em conta para a tomada de decisão, a garantia de justiça pode ser comprometida.

Um passo mais adiante, mirando o futuro, seria a adoção de semáforos virtuais ou *Virtual Traffic Lights* (VTL), onde o controle semaforico é gerenciado pelos próprios carros (descentralizado), com o apoio de comunicação inter-veicular. Alternativamente, pode-se implementar soluções com o apoio da comunicação veículo-infraestrutura (um computador com suporte *wi-fi*, por exemplo, que coordena a passagem da interseção), mas sempre adotando o conceito de redes veiculares *ad-hoc* (VANETs) (SHERALI ZEADALLY RAY HUNT, 2012).

VANETs são redes móveis, do tipo das MANETs (*Mobile Ad-Hoc Networks*), formadas pelos veículos que nela circulam e pela infraestrutura de uma rede viária. As VANETs se diferenciam das MANETs, redes móveis formadas por nós que se comunicam entre si, por serem limitadas a uma malha viária, dessa forma os veículos somente podem se movimentar dentro dessa malha. Os veículos podem obedecer dois comportamentos distintos: atuam como nós que enviam e recebem mensagens ou ainda podem se comportar como um roteador que recebe e reenvia mensagens ao destinatário (SHERALI ZEADALLY RAY HUNT, 2012).

O conceito de VANETs, em contraste com as soluções baseadas em fluxo, permite avaliar as necessidades de cada carro individualmente, implementando, desta forma, justiça no atendimento a solicitações para passar interseções. Cada carro envia uma mensagem para os demais carros ou para um dispositivo de controle que toma a decisão levando em conta dados como a posição atual do veículo, velocidade desenvolvida, destino e, possivelmente, diferentes aspectos (escoamento, justiça, etc.).

Neste contexto, ambos, semáforos inteligentes e VTL, tem objetivo de aumentar a eficiência do trânsito (vazão) e, conseqüentemente, reduzir o tempo de deslocamento, mas nem sempre garantem o equilíbrio no atendimento das vias. Para garantir vazão e justiça devem ser usados algoritmos específicos, anteriormente propostos (PASIN; SCHEUERMANN; MOURA, 2015). Entretanto, falta explorar mais intensivamente esses algoritmos em redes de transporte mais complexas. Não se sabe exatamente como esses algoritmos se comportarão em VANETs

com múltiplas interseções. Dessa forma, simulação computacional pode ser usada para avaliar o comportamento de diferentes algoritmos antes da implementação real em redes mais complexas.

Neste trabalho, é proposta a implementação de algoritmos de controle de interseção para investigar o seu comportamento de acordo com o fluxo dos carros em redes de transporte com múltiplas interseções. A investigação é realizada com o apoio de simulação computacional. Métricas avaliadas são a emissão de gases poluentes, o consumo de combustível e vazão.

1.2 Objetivos

1.2.1 Objetivo geral

A principal finalidade deste trabalho é investigar algoritmos para controlar múltiplos cruzamentos em redes de transporte e, conseqüentemente, melhorar o fluxo de veículos em uma rede de transporte urbana. Para isso, é investigado o comportamento dos algoritmos, tais como, maior fila primeiro, *zipper merge* e FIFO (*first in, first out*), em redes de transporte com múltiplas interseções, através de experimentação em um ambiente de simulação computacional. Para realizar os experimentos é usado o simulador SUMO (Simulation of Urban Mobility) (BEHRISCH et al., 2011), que é uma ferramenta de código aberto específica para redes de transporte.

1.2.2 Objetivos específicos

Para atingir o objetivo do trabalho, precisa-se realizar os seguintes passos:

- Estudar e instalar o simulador SUMO e interface Java TraCI4J;
- Definir e gerar a rede de transporte;
- Estudar e implementar algoritmos para o controle da passagem de interseções;
- Definir formalmente métricas para avaliar os algoritmos;
- Implementar na simulação as métricas em questão;
- Gerar cenários de teste;

- Avaliar através de simulação, usando as métricas definidas e os cenários de teste, os algoritmos de controle da passagem de interseções.

1.3 Justificativa

Hoje em dia a população em geral perde muito tempo em deslocamentos no trânsito, afetando demasiadamente a qualidade de vida. Pessoas por muitas vezes sofrem de stress por ficar muito tempo em longos congestionamentos (F. MARTINS; LOPES; FARINNA, 2014). Neste sentido, o que impulsionou a realização deste trabalho é uma tentativa de entender melhor este problema e colaborar na solução de congestionamento das vias e a perda de tempo das pessoas no seu deslocamento. Espera-se que este trabalho auxilie na melhora da eficiência do trânsito, bem como a diminuição do tempo de deslocamento em redes de transporte, através da aplicação de algoritmos para gerenciamento de semáforos virtuais.

Atualmente, utilizam-se semáforos baseados em tempo, o que não são muito eficientes, pois o sinal verde pode ficar aberto em vias que não possuem veículos, enquanto a outra via possui veículos esperando para passar. Em contraste, semáforos adaptativos ou inteligentes são calibrados de acordo com o fluxo, fazendo com que haja um melhor escoamento dos veículos nas vias.

Além disso, há a emissão de poluentes que prejudicam a saúde humana e o meio ambiente, e o alto consumo de combustível. Desse modo, é pretendido medir essas emissões e o consumo de combustível para cada algoritmo, com a finalidade de observar o efeito causado por cada um dos algoritmos.

1.4 Organização do texto

O texto está organizado como segue. O capítulo 2 apresenta uma revisão de literatura englobando semáforos convencionais, semáforos virtuais, VANETs e o SUMO, fazendo uma análise de trabalhos publicados na área de mobilidade urbana e controle de interseções.

O capítulo 3 apresenta as ferramentas utilizadas neste trabalho, como o simulador SUMO que é uma ferramenta de simulação de mobilidade urbana, a interface para controle de tráfego, TraCI4J, além da ferramenta de manipulação de arquivos XML, XMLSpy, e um programa para gerar gráficos através de linhas de comando, o Gnuplot. Além disso, define-se uma rede de transporte com múltiplos cruzamentos e apresenta-se a rede de transporte utilizada, a qual pos-

sui 4 cruzamentos. Em seguida, explica-se a implementação dos algoritmos usados para controlar semáforos, sendo eles, maior fila primeiro, *zipper merge* e FIFO.

No capítulo 4 são explicados as métricas utilizadas, tais como vazão, emissão de poluentes e consumo de combustível. Além disso, apresentam-se os experimentos aplicados aos algoritmos e os resultados obtidos.

O capítulo 5 apresenta as conclusões de acordo com os resultados alcançados através dos experimentos e também possíveis trabalhos futuros que possam trazer contribuição à otimização de redes transporte e mobilidade urbana.

2 REVISÃO DA LITERATURA

O semáforo (também conhecido popularmente com sinaleira ou sinal) é um controlador de tráfego que utiliza cores de forma alternada a fim de gerenciar pedestres, ciclistas e motoristas. Esse tipo de sinalização tem como objetivo fornecer informações aos usuários sobre quem tem o direito de travessia da via. A sinalização pode conter também símbolos, como bicicletas, setas e boneco de pedestre, havendo assim, semáforos específicos para cada usuário.

O primeiro semáforo foi projetado em 1868 por J. P. Knight e possuía dois braços horizontais que quando girava 45° significava "siga com cuidado". Esse semáforo estava localizado em Londres e explodiu alguns dias depois de sua implantação. Anos depois, em 1912, foi projetado o primeiro semáforo elétrico por Lester Wire, o semáforo era de madeira e tinha lâmpadas nas cores verde e vermelho. Somente em 1914, um semáforo elétrico foi instalado. Esse foi projetado por James Hoge que anos depois ganhou uma patente por sistema de controle de tráfego, o semáforo foi instalado em Ohio, nos Estados Unidos (Fonte: History).

Já em 1917, Willian Potts, foi o primeiro a criar os semáforos que conhecemos hoje, com três cores, sendo elas, verde, amarelo e vermelho. O semáforo de Potts era controlado por policiais e o primeiro deles foi instalado em Detroit, nos Estados Unidos e anos mais tarde, ele criou o semáforo de quatro lados. Somente em 1922 os semáforos começaram a ser controlados através de um temporizador, pois havia a necessidade de gerenciar um cruzamento com muitas vias (MCSHANE, 1999).

Atualmente, em Porto Alegre, uma das cidades do país com o maior número de semáforos, tem 1065 deles, sendo uma média de 0,38 semáforos por quilômetro, para se ter uma comparação, em cidades como Curitiba e Rio de Janeiro, esse número é de 0,28 (Fonte: EPTC 2014). No entanto, São Paulo possui 6138 semáforos, tendo em média 0,37 semáforos por quilômetro (Fonte: CET SP 2011).

As prefeituras de grandes cidades instalam constantemente novos semáforos. São Paulo, no ano de 2013, recebeu mais de 1000 pedidos de instalação de semáforos. Entretanto, um estudo deve ser feito antes de instalá-los. Um semáforo pode ajudar na diminuição de acidentes de trânsito, mas se colocado em local inapropriado, pode gerar um aumento no tempo de espera, além de representar gasto de instalação e operação desnecessários. O custo também precisa ser avaliado. O custo de instalação de um semáforo varia entre R\$ 10.000,00 e R\$ 14.000,00 (Fonte: EPTC Março 2011).

Os critérios para instalação dos semáforos são listados pelo DENTRAN (Departamento Nacional de Trânsito) e envolvem o aumento da segurança viária, melhoria no fluxo de trânsito, controle do conflito de passagem entre pedestres e veículos, redução dos atrasos e credibilidade dos usuários em relação à sinalização.

Uma das vantagens dos semáforos é controlar o tráfego de um cruzamento que há conflitos de passagem. Com o semáforo as vias conflitantes são gerenciadas através do tempo de cada cor do semáforo. Outras vantagens incluem: permitir que vias menores cruzem vias com grande fluxo de carros com segurança e permitir que pedestres consigam atravessar vias, evitando acidentes.

Essa sinalização é classificada de acordo com sua função, podendo ser usada para regulamentar o direito de passagem numa via, tanto dos pedestres, quanto dos veículos ou para advertir condutores sobre algum obstáculo ou situação perigosa. Antes de colocar um semáforo em um cruzamento, algumas diretrizes devem ser seguidas primeiro, como o controle da via a partir das normas no CTB (Código de Trânsito Brasileiro), controle a partir de placas ou sinalização vertical (faixa de pedestres) e implantação de rotatórias (Fonte: CONTRAN, 2012).

Cidades como Porto Alegre, São Paulo e Belo Horizonte já existem semáforos inteligentes em testes. Em Porto Alegre o semáforo está localizado no cruzamento da Avenida Dr. Nilo Peçanha com a Rua Carazinho (Fonte: Digicon, 2015). Em São Paulo, semáforos inteligente foram instalados em seis cruzamentos da Avenida Padre Antônio José dos Santos (SEMÁFOROS INTELIGENTES, 2012). E em Belo Horizonte esses semáforos se encontram em 28 cruzamentos da Avenida Amazonas (Fonte: Diário do Comércio, Dezembro 2013). Os semáforos inteligentes possuem câmeras integradas ou sensores no solo que identificam os carros que estão se aproximando e calculam a melhor forma de gerenciar o trânsito, obtendo melhorias no fluxo dos veículos e pedestres (Fonte: Digicon). Outro sistema que está em teste em Curitiba é o aumento do tempo do sinal para os pedestres passarem, com isso, idosos, deficientes ou pessoas com dificuldade de locomoção usam um cartão especial no semáforo que adiciona alguns segundos para a sua travessia na faixa de segurança (Fonte: Prefeitura de Curitiba, Abril 2015).

Futuramente, espera-se que ocorra a implantação de VANETs que são redes móveis compostas por uma malha viária e pelos veículos que trafegam nela, suportada pela comunicação *wireless*. As VANETs pertencem às MANETs, que são um tipo especial de rede que possui alta mobilidade. As MANETs não possuem uma estrutura física ou centralizada e os seus nós podem ser conectar e desligar quando desejarem. Para isso, os nós precisam ser configurados

como roteadores para enviar e receber pacotes (G.SINGHA; KUMAR; VERMA, 2012).

A comunicação nas VANETs pode ser feita entre os veículos, V2V (*Vehicle to Vehicle*) ou entre a infraestrutura e os veículos V2I (*Vehicle to Infrastructure*). A comunicação feita pelos veículos nas VANETs é de pequena a média distância, sendo que alguns nós são colocados estrategicamente, por exemplo, em postos de gasolina em estradas para a comunicação ter maior alcance, esses nós são estáticos e caracterizam a comunicação V2I. Essa é outra diferença das VANETs, pois se o fluxo de carros é baixo e os veículos estão distantes, pode haver uma desconexão frequente (SHERALI ZEADALLY RAY HUNT, 2012).

As maiores motivações para o seu uso são a segurança e o melhor gerenciamento de tráfego. A segurança deve-se ao fato de os veículos estarem conectados pela comunicação V2V, por exemplo veículo *a* pode enviar mensagem ao outro *b*, sobre problemas na pista ou sobre uma parada inesperada de forma a evitar acidentes. E o melhor gerenciamento de tráfego é justificado pelo fato de que os nós estáticos conseguem enviar informações referentes à acidentes na rodovia, congestionamentos ou à presença de veículos de emergência.

Semáforos inteligentes, simulação computacional e redes de transporte são assuntos recorrentes na pesquisa acadêmica. Em (KRAJZEWICZ et al., 2005), o SUMO foi utilizado para simular semáforos virtuais com a finalidade de otimizar o fluxo, sendo o tamanho da fila utilizado como dado de entrada. Nessa simulação, obteve-se melhorias nos engarrafamentos, mas não em filas com fluxo pequeno ou com fluxos balanceados nas diferentes direções. Assim, foi constatado que o melhor lugar para aplicar o algoritmo de semáforos virtuais é em cruzamentos, onde o fluxo muda constantemente.

Segundo (DRESNER; STONE, 2004), semáforos convencionais podem não ser sempre eficientes. Em contraste, passagens elevadas, apesar de serem o ideal, possuem um custo muito alto. Então foi proposto um sistema baseado em reserva, no qual os carros são tratados como agentes autônomos e recebem e enviam informações da interseção sobre qual veículo deve passar. Foi avaliada principalmente a vazão, chegando à conclusão que o sistema baseado em reserva é mais eficiente que semáforos convencionais e se aproxima do ótimo teórico.

Outra forma de controlar o fluxo de carros seria com VANETs, na qual é estabelecida uma conexão veículo-a-veículo. Em (FERREIRA et al., 2010), os veículos mais próximos da interseção elegem um líder que é responsável por emitir mensagens sobre que via deve passar. Quando esse passa a interseção, um novo líder é eleito. Através da comunicação veicular conflitos são resolvidos de forma autônoma e podem gerar a gestão de tráfego onipresente.

Além do desempenho no escoamento eficiente de veículos em redes de transporte, outros problemas interessantes são o consumo de combustíveis e a emissão de gases poluentes. Estes itens também serão explorados neste trabalho. O SUMO é capaz de simular emissão de poluentes e ruídos por veículo (BEHRISCH et al., 2011), e ainda permite visualizar as emissões produzidas pelas arestas (KRAJZEWICZ et al., 2014). Segundo o Quinto Relatório de Avaliação do IPCC (*Intergovernmental Panel on Climate Change*), o setor de transporte é responsável por aproximadamente 23% do total de energia relacionada a emissão de CO₂. Isso torna este assunto extremamente relevante. Iniciativas no sentido de reduzir a emissão de poluentes são de fundamental importância.

3 FERRAMENTAS E MÉTODOS UTILIZADOS

Este capítulo descreve brevemente ferramentas utilizadas no contexto deste trabalho. O leitor que desejar uma informação mais completa deverá buscar a bibliografia relacionada com a ferramenta específica. Posteriormente, é descrito a definição de uma rede de transporte e o detalhamento da rede de transporte utilizada. E por último são especificados os algoritmos que foram implementados para o controle das interseções de uma rede de transporte.

Basicamente, foi utilizada a biblioteca TraCI4J para a implementação dos algoritmos de controle de interseções, o simulador para redes de transporte SUMO para gerar os fluxos sobre um mapa com avenidas e múltiplas interseções, e o XMLSpy que é uma ferramenta que auxilia no processamento de arquivos XML gerados pelo SUMO. Finalmente, foi usado o *gnuplot* que oferece *scripts* para a geração de gráficos que foram gerados com os resultados extraídos via XMLSpy. A Figura 3.1 apresenta as ferramentas utilizadas e suas conexões.

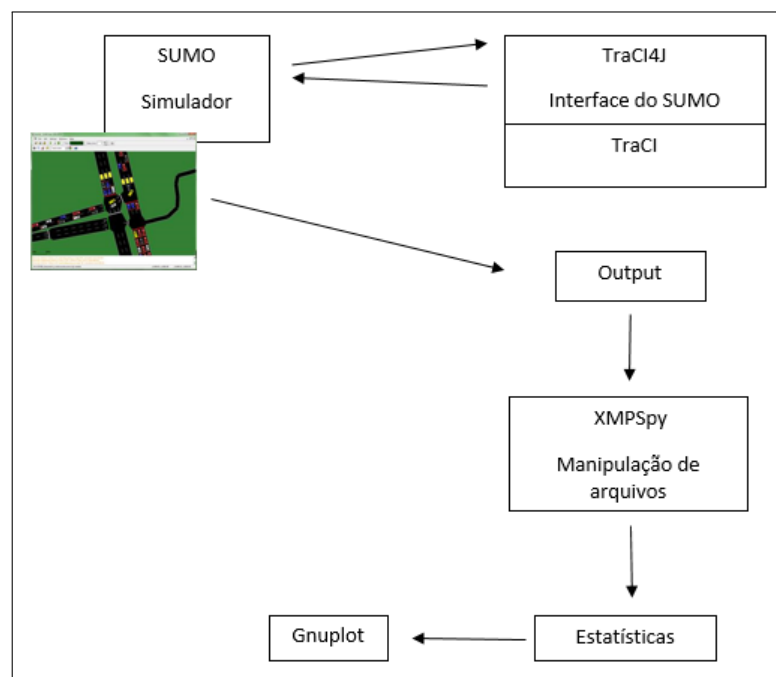


Figura 3.1 – Ferramentas utilizadas e suas ligações

3.1 SUMO: Simulation of Urban MObility

O SUMO (BEHRISCH et al., 2011) é um simulador de tráfego desenvolvido pelo Centro Aeroespacial da Alemanha e disponível desde 2001. O simulador tem como finalidade apoiar a comunidade com uma ferramenta de código aberto, na qual possam ser implementados os

seus próprios algoritmos para redes de transporte. Para mais informações basta acessar o site do SUMO disponível em www.dlr.de/ts/sumo/en/.

Para construir uma rede de transporte no SUMO é necessário descrever a rede em arquivos XML (*eXtensible Markup Language*) com configurações específicas da simulação. Essas configurações envolvem um arquivo com o nodos e um arquivo com as arestas. Os nodos são os cruzamentos e as arestas ligam um nodo a outro formando as quadras. Existe também um arquivo de rotas, no qual é especificado um conjunto de rotas e o período com que os veículos são inseridos na simulação, representando diferentes volumes de tráfego. A simulação possui um tempo discreto chamado de *step*, que por *default* representa 1s.

Os veículos do SUMO possuem atributos como identificador, tempo de partida, velocidade, entre outros. Esses atributos podem ser monitorados ao longo da execução para possibilitar a avaliação de algoritmos e respectivas métricas, como é feito neste trabalho.

No SUMO, cada veículo é associado a uma classe de emissão de poluentes. O simulador também muda a cor dos veículos de acordo com a emissão de CO₂, que pode ser visualizada pela interface gráfica. O SUMO permite a estimativa de emissões de poluentes e do gasto de combustível do veículo quando esse termina a simulação e salva em um arquivo de texto. É possível coletar os dados dos poluentes emitidos e do consumo de combustível de cada veículo ou também de cada via, a cada *step* através do TraCI (KRAJZEWICZ et al., 2012).

Para estimar essa emissão de poluentes o SUMO possui o próprio modelo baseado na base de dados HBEFA (*Handbook Emission Factors for Road Transport*) versão 2.1. O HBEFA é uma base de dados que fornece dados sobre fatores de emissão (quente, arranque a frio, evaporação) para todos os gases poluentes, para mais informações basta acessar o seu site disponível em <http://www.hbefa.net/e/index.html>. O modelo de emissão do SUMO foi implementado extraíndo os dados da HBEFA e aplicando eles em uma função contínua. Essa função contínua foi obtida simplificando a função de potência do motor do veículo que deve superar a força de resistência de condução (KRAJZEWICZ et al., 2014). Essa função é dada por:

$$c_0 + c_1va + c_2va^2 + c_3v + c_4v^2 + c_5v^3$$

O SUMO consegue simular pequenas redes, com poucos nodos e arestas, ou mais complexas, muitos nodos e arestas, e de fluxo variado, sendo que cada via pode ter um fluxo e velocidade dos veículos diferentes. Aqui serão explicadas apenas as funcionalidades que fazem

parte do escopo deste trabalho, tais como os arquivos XML necessários para descrever uma rede.

O arquivo XML que especifica os nodos, possui quatro atributos: identificador, valor x , valor y e tipo. O identificador precisa ser único, o valor x e o valor y , irão determinar a localização do nodo no cenário, lembrando que esses valores são medidos em metros. E, o atributo tipo, o qual é admitido dois valores: *priority* e *traffic_light*. *Priority* significada que o nodo possui prioridade para passar diante dos outros e *traffic_light* significa que a interseção é controlada por um semáforo. Nesse trabalho, a rede possui quatro nodos com atributos *traffic_light*, um para cada interseção.

As aretas conectam os nodos formando as quadras. No arquivo, cada aresta possui como atributos: identificador, origem (*from*), destino (*to*), prioridade (*priority*), número de faixas (*numLanes*), e velocidade (*speed*). O atributo origem identifica a partir de qual nodo a aresta irá começar e o atributo destino identifica o nodo em que a aresta irá terminar. O atributo prioridade é um inteiro que informa a prioridade da aresta. O atributo número de faixas é o número de faixas que a aresta possui, neste caso foi usado o valor um para todas. O atributo velocidade, que é a velocidade máxima permitida na aresta em questão, neste trabalho foi usado 16,667 m/s (60km/h) para via arterial e 11,111 m/s (40km/h) para via coletora (Fonte: CTB).

A partir desses arquivos, é usado o *netconvert* para gerar um arquivo *[...].net.xml*. O *netconvert* é um componente do SUMO usado para importar redes. Para utilizá-lo, primeiro deve ser setada a variável do SUMO no *classpath* e após utilizar o comando: *netconvert -edge-files [...].edg.xml -node-files [...].nod.xml -o [...].net.xml*. O SUMO requer como entrada também criar um arquivo com as rotas. Neste arquivo as arestas são ligadas formando as vias. Finalmente, o arquivo *textit[...].sumocfg*, precisa ser definido e serve de entrada para a simulação. Ele contém o caminho para o arquivo *[...].net.xml* e para o arquivo de rotas. A Figura 3.2 apresenta os arquivos de entrada e saída do SUMO explicados nesse capítulo. O arquivo *tripinfo.xml* possui os dados referentes à emissão de poluentes.

3.2 TraCI4J

O TraCI (Traffic Control Interface) que é uma interface em Python que se comunica com a simulação em tempo real. Essa interface é implementada em uma arquitetura TCP cliente/servidor, onde o SUMO faz o papel de servidor. Devido a esse fato a instância do SUMO pode estar rodando na mesma máquina ou em outra máquina da interface.

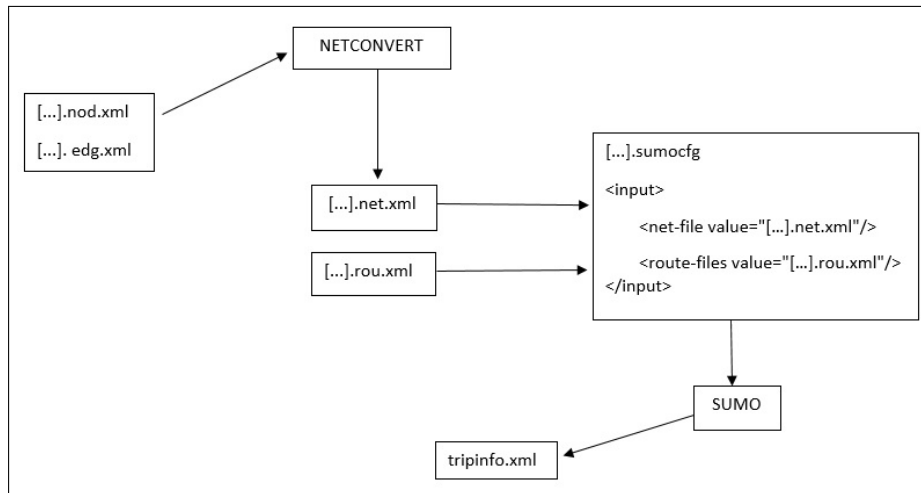


Figura 3.2 – Arquivos de entrada e saída do SUMO

Já o TraCI4J é uma biblioteca Java usada para controlar uma simulação no SUMO por meio do TraCI. A biblioteca foi desenvolvida pelos membros do grupo ApPeAL (*Applied Pervasive Architectures Lab*) do Politécnico de Torino, na Itália. O TraCI4J pode atuar como *front-end* para uma instância do SUMO e fazer com que o simulador inicie, pare ou execute passo a passo. Enquanto a simulação está sendo executada algumas informações podem ser adquiridas: estáticas, sobre a topologia da rede, ou dinâmicas, sobre a posição e velocidade dos veículos.

A biblioteca TraCI4J pode ser adquirida pelo endereço <https://github.com/egueli/TraCI4J>, onde há o código fonte de um projeto do Eclipse. Assim basta importá-la para o Eclipse ou construí-la pelo Apache Ant. O Apache Ant é uma biblioteca Java para automação do processo de compilação de *software*. Essa biblioteca está disponível em <http://ant.apache.org/>. Neste trabalho a biblioteca foi importada para o Eclipse versão 3.8.

3.3 XMLSpy

Para realizar a leitura de arquivos XML gerados pelo SUMO com dados sobre a emissão de poluentes e consumo de combustível, foi utilizado o XMLSpy que é uma ferramenta de manipulação de arquivos XML desenvolvida pela Altova, disponível desde 1999. Essa ferramenta permite não apenas criar arquivos XML, mas possibilita também o uso de outras tecnologias baseadas em XML, tais como XML Schema, XSLT, XQuery, XPath. Para obter mais informações e direções para instalar a ferramenta, basta acessar o site <http://www.altova.com/xmlspy.html>.

Neste trabalho, foi utilizado o XPath que é uma sintaxe para consulta em um arquivo XML. Ela seleciona partes em um arquivo XML, descrevendo o caminho para os elementos. Por exemplo, a Figura 3.3 mostra o trecho de um arquivo XML, para acessar o atributo *CO_abs*, da tag *emissions*. Com XPath pode-se fazer `/tripinfo/emissions/@CO_abs`. Nota-se que os atributos são acessados através do símbolo @, o resultado dessa consulta será 552.1.

```
<tripinfo id="veiculo1" >
  <emissions CO_abs="552.1" />
</tripinfo>
```

Figura 3.3 – Fragmento de um arquivo XML

3.4 Gnuplot

O Gnuplot é um programa de linha de comando extremamente popular feito para plotar gráficos 2D ou 3D. Esse programa consegue plotar gráficos de funções matemáticas ou de conjuntos de dados. Foi contruído em 1986, inicialmente, para apoiar a comunidade acadêmica a visualizar funções matemáticas e dados de forma iterativa. O projeto se expandiu e hoje é amplamente utilizado em trabalhos científicos.

O programa gera saídas em diversos tipos de arquivo, por exemplo, PNG, JPEG e EPS. Além disso, consegue gerar códigos a fim de serem inseridos no Latex. Os gráficos podem ser produzidos através de linha de comando via terminal ou através de *scripts*. A sua documentação e maiores informações sobre instalação e modo de uso são encontradas em <http://www.gnuplot.info/>.

3.5 Definindo uma rede de transporte com muitos cruzamentos

Uma rede descreve uma estrutura que pode ser física (por exemplo, vias e interseções ou linhas de telefone) ou conceitual (por exemplo, troca de informações e pessoas). Ambas possuem dois elementos, pontos e segmentos de reta, os quais conectam tais pontos. Na matemática, a rede é chamada de grafo, onde os pontos representam os vértices ou nodos, e os segmentos de reta são chamados de arestas ou arcos (SHEFFI, 1985). A Figura 3.4 apresenta um grafo $G = (V, E)$ sendo, $V = \{1, 2, 3, 4, 5\}$ e $E = \{a, b, c, d, e, f, g\}$, e seus elementos.

Uma rede de transporte é um grafo direcionado (também chamado de dígrafo), no qual

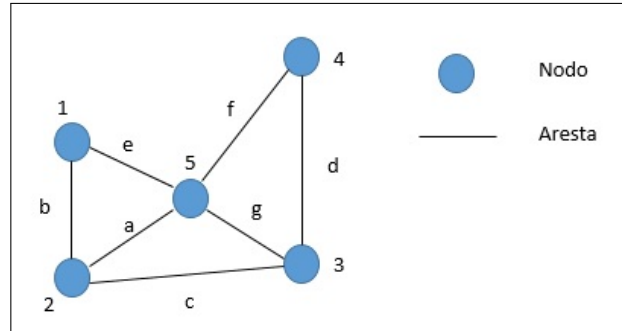


Figura 3.4 – Elementos de um grafo

cada arco tem uma capacidade e suporta um fluxo. Sendo $G = (V, E)$ um grafo finito direcionado, no qual V é um conjunto de vértices e E um conjunto de arestas. Cada aresta $(u, v) \in E$ tem um valor não negativo e capacidade de $c(u, v)$. Pode-se distinguir dois tipos de vértices: origem s e destino t . O fluxo em uma rede de transporte é dado por uma função real $f : V \times V \rightarrow \mathbb{R}$ (MALHOTRA; KUMAR; MAHESHWARI, 1978).

Para este trabalho foi construída uma rede de transporte com cinco vias, das quais, uma é arterial e quatro são coletoras. A partir dessas vias, gerou-se quatro interseções, onde cada via coletora cruza com a arterial. Na Figura 3.5, este contexto é exemplificado, onde a via arterial é a via horizontal e as coletoras estão na vertical. As vias coletoras possuem também sentidos diferentes, de acordo com as setas da Figura 3.5.

Para inserir os semáforos nas interseções, foi adicionado um nodo a mais em cada via antes de cada semáforo a fim de diminuir a velocidade do veículo entre o novo nodo e o semáforo. Este nodo encontra-se um pouco antes de cada semáforo. A inserção desse nodo é necessária, pois quando o semáforo muda de cor, para o vermelho, o veículo precisa diminuir a velocidade senão ele pode parar em cima da interseção ou até mesmo cruzá-la.

A Figura 3.6 apresenta uma captura de tela da rede de transporte utilizada no simulador SUMO. Este fragmento é usado como cenário neste trabalho.

3.6 Algoritmos

Neste trabalho, são implementados três algoritmos descritos brevemente a seguir com o intuito de gerenciar o tráfego em uma rede de transporte através de semáforos virtuais. Uma descrição mais detalhada encontra-se em (PASIN; SCHEUERMANN; MOURA, 2015).

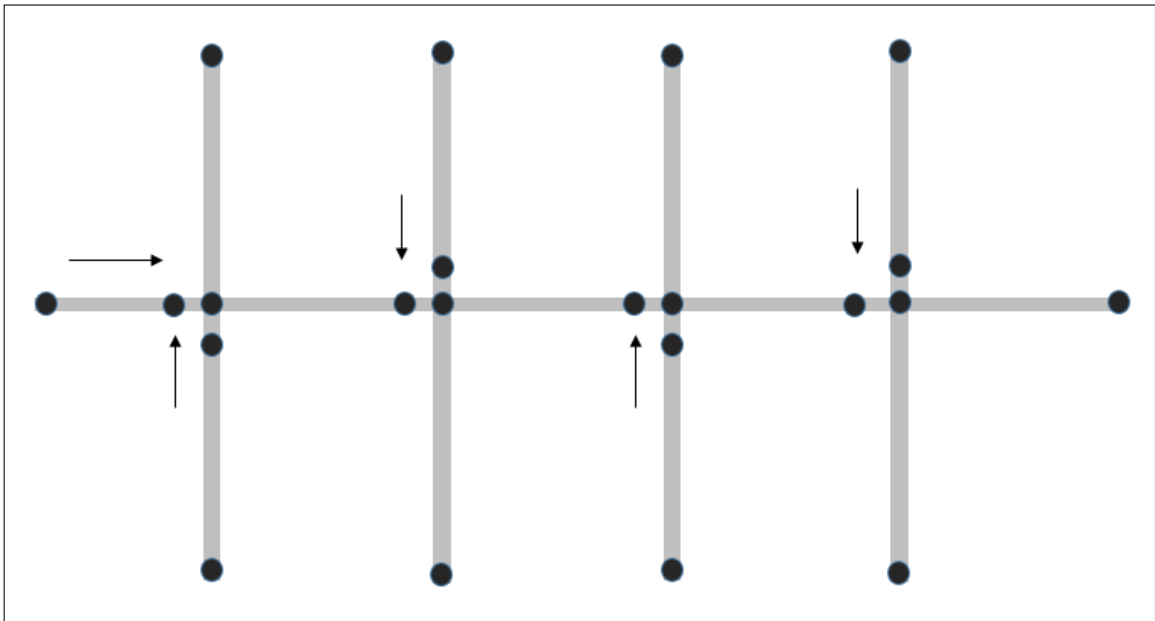


Figura 3.5 – Rede de transporte com quatro interseções indicando direção dos fluxos e nodos adicionais para a redução de velocidade dos veículos

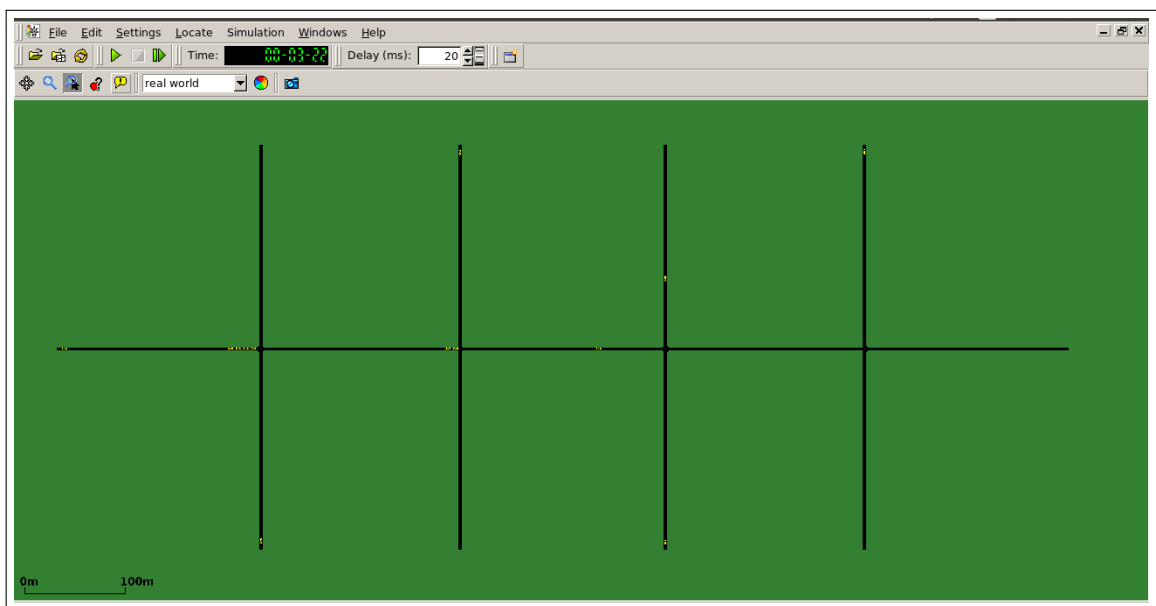


Figura 3.6 – Captura de tela da rede de transporte no SUMO com múltiplas interseções: uma via arterial e quatro vias coletoras

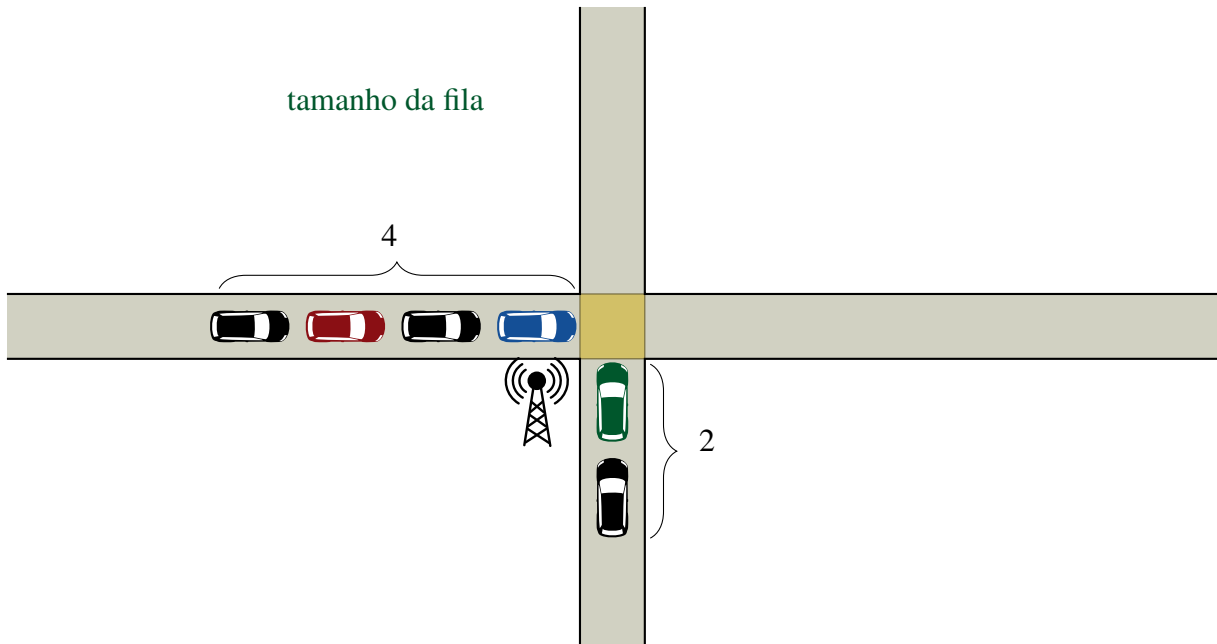


Figura 3.7 – Exemplo do algoritmo maior fila primeiro, mostrando o tamanho das filas

3.6.1 Maior fila primeiro

É uma técnica, na qual a maior fila tem prioridade. Neste caso, existem duas filas de veículos em cada aresta, por exemplo na Figura 3.6.1, uma no sentido oeste-leste e outra no sentido sul-norte, ambas com tamanhos distintos. Através de comunicação veículo-infraestrutura, ou ainda veículo-a-veículo, é calculado o tamanho das filas. Esse cálculo é feito de acordo com o número de veículos que tem esperando em cada aresta para passar a interseção. A fila com maior número de veículos, ou seja, a maior fila, irá passar primeiro. A outra fila terá a chance de ser a maior na próxima comparação.

Essa é uma técnica orientada a comboios. O número de trocas entre as filas é minimizado e este algoritmo prova valores eficientes para a vazão.

3.6.2 Zipper Merge (intercalação de filas)

Como cada interseção possui duas filas de veículos, isso é mostrado na Figura 3.6.2, esse algoritmo intercala essas duas filas passando k veículos de cada vez, na qual k é o número máximo de veículos que devem passar na interseção de cada fila. No exemplo da Figura 3.6.2, $k = 1$.

Esse algoritmo é orientado à decisão individual, portanto, quando os fluxos de entrada

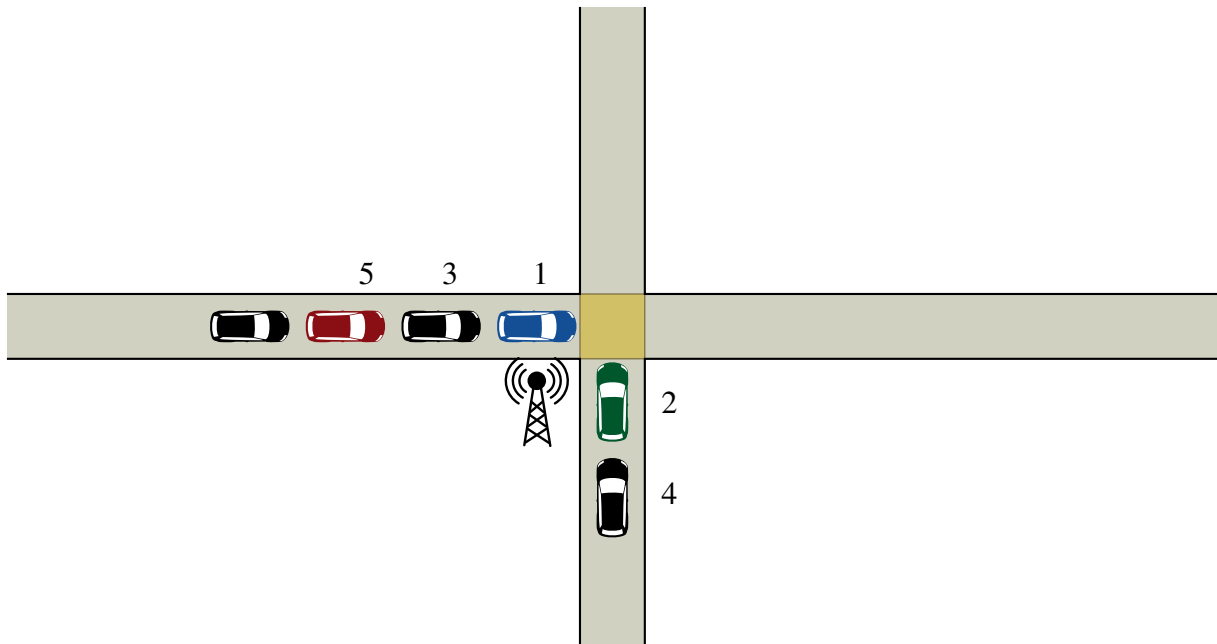


Figura 3.8 – Exemplo do algoritmo *zipper merge*, o qual passa k veículos por vez

das filas são iguais ou parecidos, este algoritmo é ótimo para a justiça no atendimento dos veículos (PASIN; SCHEUERMANN; MOURA, 2015). Mas devido ao número de trocas, não é muito adequado à vazão.

3.6.3 FIFO (*First In, First Out*)

O FIFO (*First In, First Out*), significa que o primeiro a chegar, deve ser o primeiro a passar a interseção. Nesse sentido, calcula-se qual veículo deve chegar a interseção primeiro, levando em consideração a sua distância e velocidade, então o veículo que obtiver menor valor, terá a chance de cruzar a interseção primeiro, a Figura 3.6.3 mostra os veículos e o tempo estimado de cada um para passar a interseção. Dado que esse algoritmo é orientado a decisões individuais, a vazão não é eficiente, no entanto ele é apropriado para a justiça e na minimização do tempo de atendimento das filas.

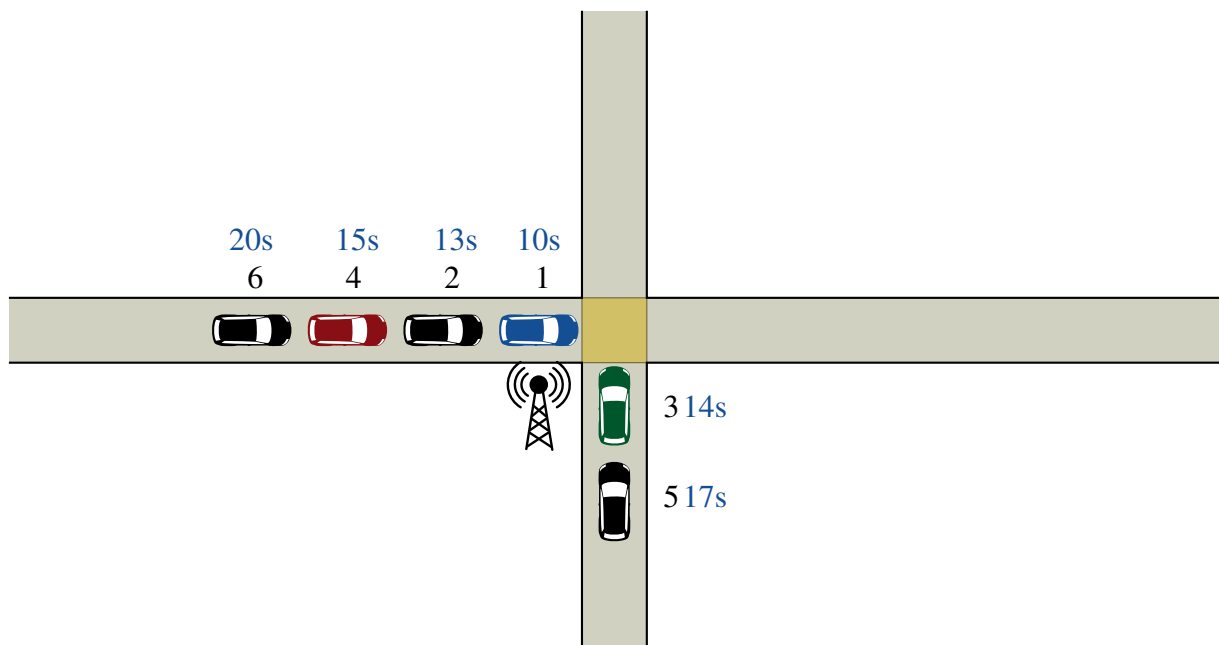


Figura 3.9 – Veículos e seus tempos calculados a partir do algoritmo FIFO, no qual o carro com o menor tempo de chegada cruza a interseção primeiro

4 AVALIAÇÃO EXPERIMENTAL

Esse capítulo apresenta conceitos sobre as métricas utilizadas na avaliação experimental. E após, são detalhados os experimentos realizados e os resultados obtidos com a execução de diferentes algoritmos em um rede de transporte com múltiplas interseções.

4.1 Métricas

Para avaliar os efeitos dos algoritmos descritos no capítulo 3 em uma rede de transporte com múltiplos cruzamentos, foram usadas as seguintes métricas: vazão e emissão de poluentes.

4.1.1 Vazão

Vazão (Q) indica o número de carros que passam por um determinado segmento de rota, delimitado por dois pontos A e B, que representam, de fato, instantes de tempo. A vazão é calculada para todos os veículos levados em conta usando-se a seguinte fórmula:

$$Q = \frac{V}{t_f - t_i}$$

na qual t_f é o tempo final da simulação (B) e t_i (A) é o tempo inicial da simulação e V representa o número total de veículos que foram inseridos na simulação e que terminaram o percurso.

4.1.2 Emissão de poluentes

A emissão de poluentes é calculada automaticamente pelo SUMO. Valores são salvos em um arquivo XML, mas precisam ser interpretados.

O automóvel corresponde a 36,1% da emissão de CO₂ por passageiro por quilômetro do total das emissões, sendo o restante das emissões feito por metrô, ônibus e motocicletas. Deve ser levado em conta que cada automóvel tem uma ocupação média de 1,5 passageiro (CARVALHO, 2011).

4.2 Experimentos

Na avaliação experimental, os algoritmos descritos no item 3.6 foram avaliados frente às métricas descritas no item 4.1.

Para realizar os experimentos foram utilizados o simulador SUMO versão 0.22 e a biblioteca Java TraCI4J. O *hardware* que suporta este *software*, SUMO, possui um processador Intel Core i7, com 6GB de memória RAM.

Para obter os valores da emissão de poluentes, foi adicionada uma opção a mais no SUMO, fazendo com que ele gerasse um arquivo XML, no qual para cada veículo há uma *tag* com diversas informações fornecidas pelo simulador, como por exemplo, o valor da emissão de emissão dióxido de carbono (CO₂) e consumo de combustível. Para os experimentos foram usados esses valores, além da vazão.

Nos experimentos, executou-se 10 vezes cada teste, a fim de ter uma melhor acuracidade dos valores, e obteve-se a média como resultado final para a geração de gráficos. Cada simulação teve duração de 7200 *steps* no SUMO, ou seja, 2 horas.

Aqui os algoritmos descritos no capítulo 3 são novamente elencados na Tabela 4.1, a coluna *Nome* representa o nome dos algoritmos e a coluna *Sigla* representa a sigla de cada algoritmo utilizada nos gráficos e no texto que segue.

Nome	Sigla
Maior fila primeiro	lqf
<i>Zipper merge</i>	zm
FIFO (<i>first in first out</i>)	fifo

Tabela 4.1 – Sigla dos algoritmos aplicados às métricas

Durante as execuções foi observado que os algoritmos *zm* e *fifo* formam longas filas, enquanto no algoritmo *lqf*, isso não acontece. Nas Figuras 4.1, 4.2 e 4.3, essa diferença das filas de veículos é mostrada com amostras retiradas da simulação quase no mesmo instante de tempo. Por exemplo, na Figura 4.1, que mostra a execução do algoritmo *zm*, observa-se veículos esperando para passar nas duas vias, isso acontece porque há muita troca entre as filas impactando no tempo de espera. Na Figura 4.2, que mostra a execução do algoritmo *fifo*, acontece o mesmo gerando uma fila de espera indefinida para cruzar a interseção. Já na Figura 4.3, que mostra a execução do algoritmo *lqf*, não há longas filas de espera, pois não muita troca entre as filas, fazendo que o trânsito flua melhor.

Referente à vazão, esta foi calculado pelo número de veículos que cruzaram a interseção

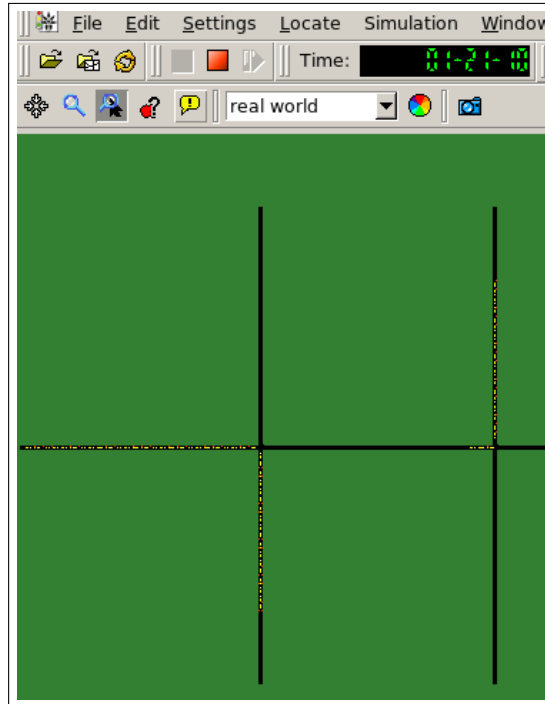


Figura 4.1 – Execução do algoritmo *zipper merge* e as respectivas filas formadas durante a execução da simulação

e terminaram a simulação dividida pelo tempo total, em minutos, da simulação. O Gráfico 4.4, apresenta os valores para cada algoritmo em relação à vazão. O algoritmo *lqf* destacou-se em relação aos outros por ter o melhor resultado, obtendo 5,89 veículos/min. Já os algoritmos *zm* e *fifo* não foram tão eficientes, resultando em valores pequenos para a vazão. Os valores pequenos encontrados são justificados pelo fato deles realizarem muitas trocas entre as vias.

A Figura 4.5 apresenta um gráfico com a emissão de dióxido de carbono. O algoritmo *lqf* possui um valor menor, 539,68 g/veículo, em relação aos algoritmos *zm* e *fifo*.

No consumo de combustível, os três algoritmos resultaram valores bem próximos. No entanto, o algoritmo *lqf* atingiu o menor valor, 0,22 l/veículo. No gráfico da Figura 4.6 é mostrada essa diferença sobre o consumo de combustível dos veículos.

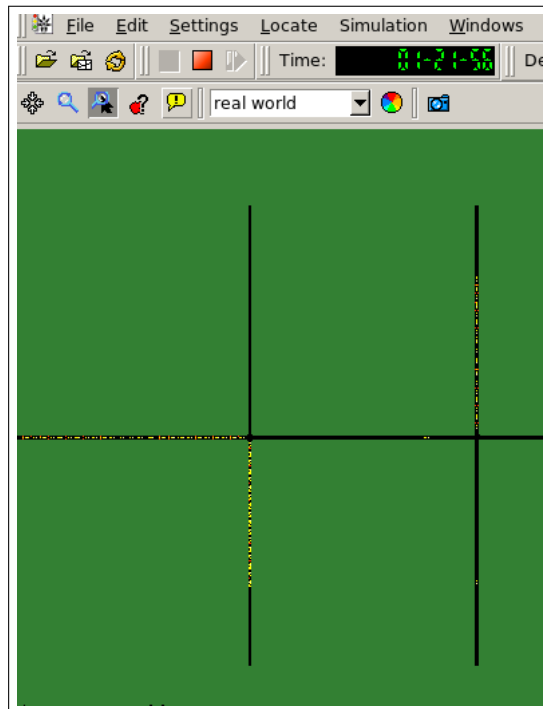


Figura 4.2 – Captura de tela com a execução do algoritmo *fifo* e as respectivas filas formadas durante a execução da simulação

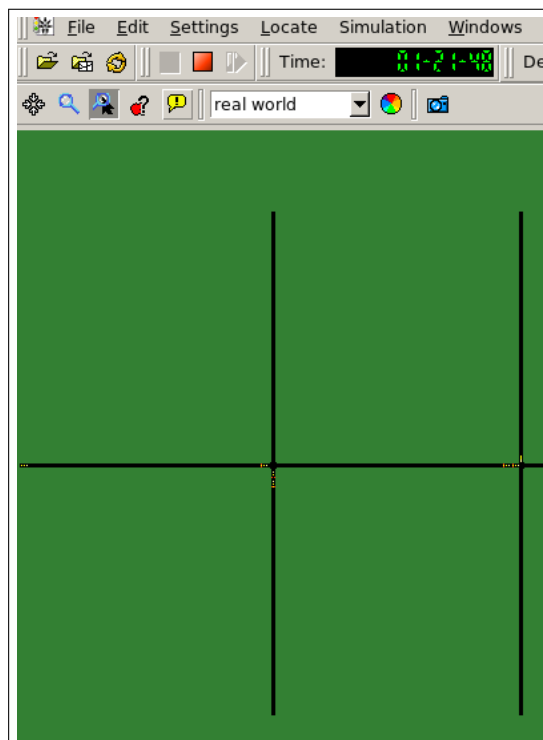


Figura 4.3 – Captura de tela com a execução do algoritmo *lqf*, o qual não formou filas nas vias durante a execução da simulação

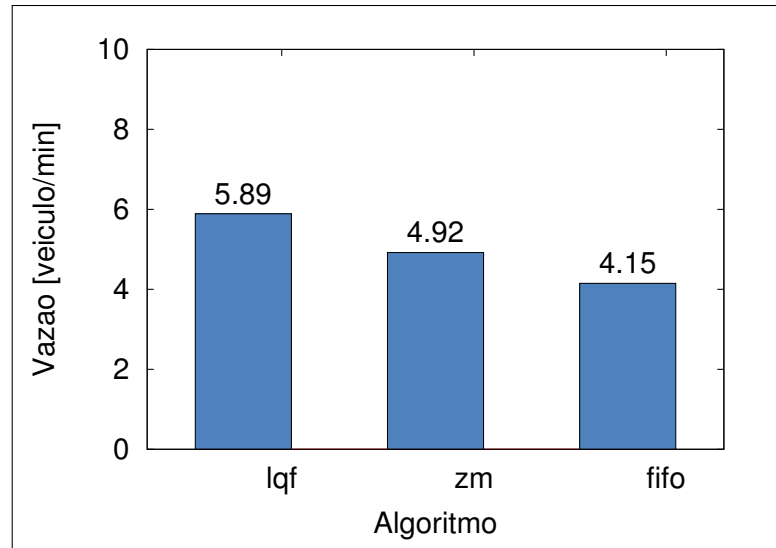


Figura 4.4 – Vazão

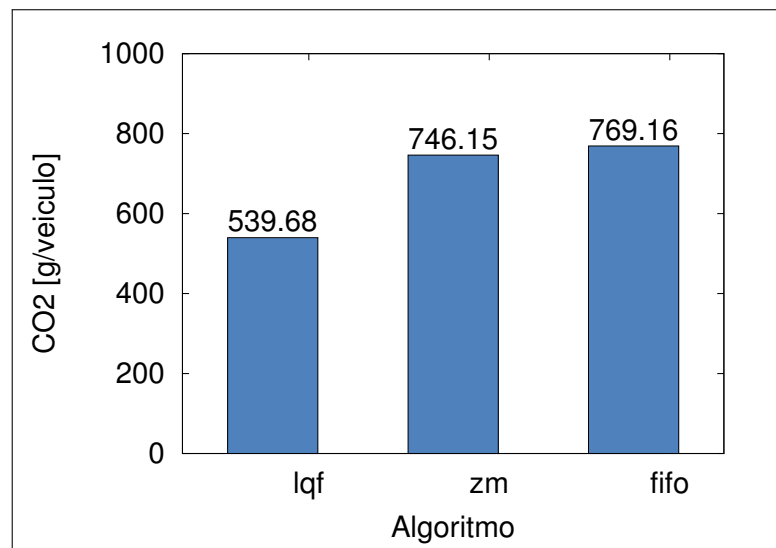


Figura 4.5 – Emissão de dióxido de carbono (CO2)

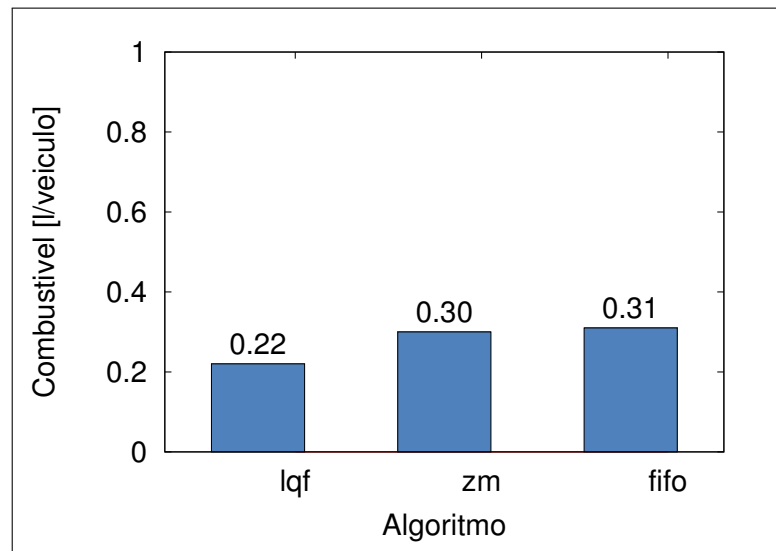


Figura 4.6 – Consumo de combustível por veículo na simulação

5 CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho foi apresentada uma implementação entre algoritmos para o controle de cruzamentos em uma rede de transporte com múltiplas vias. Os algoritmos implementados são de dois tipos: orientados a comboio e orientados a decisão individual. Para avaliar esses algoritmos foi utilizado simulação computacional, bem como o simulador de mobilidade urbana SUMO e métricas. Essas métricas são a vazão e a emissão de poluentes.

Foram instalados o simulador SUMO e a interface TraCI4J que faz uma comunicação em tempo real com o SUMO. O simulador a interface se mostraram eficientes durante a execução dos experimentos e corresponderam às expectativas.

Para gerar o cenários de teste foi necessário especificar três arquivos XML, sendo eles um arquivo de nodos, um arquivo de arestas e um arquivo de rotas. Com isso, gerou-se uma rede de transporte com uma via arterial e quatro vias coletoras. Nessa rede de transporte foi inserido quatro semáforos a fim de gerenciar o trânsito. Para fazer esse gerenciamento foram implementados os algoritmos maior fila primeiro que é orientado a comboio e *zipper merge* e FIFO que são orientados a decisão individual.

A avaliação desses algoritmos foi feita através de métricas tais como a vazão, emissão de gases poluentes e o consumo de combustível. O algoritmo orientado a comboio, maior fila primeiro, obteve os melhores resultados comparado aos outros algoritmos não apenas na vazão, mas também na emissão de gases poluentes e no consumo de combustível, se mostrando melhor que os algoritmos orientados a escolhas individuais, *zipper merge* e FIFO, os quais priorizam a minimização do tempo de espera.

Como trabalhos futuros, poderá ser realizado os mesmos experimentos com os algoritmos em outro simulador de rede de transporte a fim de comparar os resultados obtidos. Outra ideia, seria de acrescentar métricas, como a justiça e tempo de espera das filas com o objetivo de complementar o trabalho realizado. Além disso, pode-se aumentar o cenário utilizado, com mais nodos e arestas ou até mesmo um cenário real.

REFERÊNCIAS

- AZUGA, D. **Danos ambientais causados por veículos leves no Brasil**. 2000. M.sc, Programa de Planejamento Energetico — Universidade Federal do Rio de Janeiro.
- BEHRISCH, M. et al. SUMO - A Simulation of Urban MObility: an overview. In: INTERNATIONAL CONFERENCE ON ADVANCES IN SYSTEM SIMULATION (SIMUL 2011). THINKMIND. SIMUL 2011, OUTUBRO 2011., 3. **Proceedings...** [S.l.: s.n.], 2011.
- CARVALHO, C. de. Emissões relativas de poluentes do transporte motorizado de passageiros nos grandes centros urbanos brasileiros. In: IPEA. **Anais...** [S.l.: s.n.], 2011. p.39.
- DRESNER, K.; STONE, P. Multiagent Traffic Management: a reservation-based intersection control mechanism. In: THE THIRD INTERNATIONAL JOINT CONFERENCE ON AUTONOMOUS AGENTS AND MULTIAGENT SYSTEMS. **Anais...** [S.l.: s.n.], 2004. p.530–537.
- F. MARTINS, F. de; LOPES, R.; FARINNA, M. Nível de estresse e principais estressores do motorista de transporte coletivo. **Bol. Acad. Paulista de Psicologia, São Paulo, Brasil**, [S.l.], n.87, p.523–536, 2014.
- FERREIRA, M. et al. Self-organized traffic control. **ACM New York, NY, USA ©2010**, [S.l.], n.533104, p.85–90, 09 2010.
- GOLDEMBERG, J. Mudanças climáticas e desenvolvimento. **Estudos Avançados**, [S.l.], v.14, p.77 – 83, 08 2000.
- G.SINGHA; KUMAR, N.; VERMA, A. K. Ant colony algorithms in MANETs: a review. **Journal of Network and Computer Applications**, [S.l.], n.2, p.1964–1972, 11 2012.
- KRAJZEWICZ, D. et al. Simulation of modern traffic lights control systems using the open source traffic simulation SUMO. In: INDUSTRIAL SIMULATION CONFERENCE 2005, EUROSIS-ETI. 3RD INDUSTRIAL SIMULATION CONFERENCE, 3. **Proceedings...** [S.l.: s.n.], 2005. p.299–302.
- KRAJZEWICZ, D. et al. Recent Development and Applications of SUMO - Simulation of Urban MObility. In: INTERNATIONAL JOURNAL ON ADVANCES IN SYSTEMS AND MEASUREMENTS. **Anais...** [S.l.: s.n.], 2012. p.128–138.

KRAJZEWICZ, D. et al. **Second generation of pollutant Emission Models for SUMO**. [S.l.]: Springer International Publishing, 2014. 203-221p.

MALHOTRA, V. M.; KUMAR, M. P.; MAHESHWARI, S. N. An $O(|V|^3)$ algorithm for finding maximum flows in networks. **Information Processing Letters**, [S.l.], v.7, n.6, p.277–278, 1978.

MCSHANE, C. The Origins and Globalization of Traffic Control Signals. **Telecommunication Systems**, [S.l.], v.25, n.3, p.379–404, 3 1999.

PASIN, M.; SCHEUERMANN, B.; MOURA, R. F. de. VANET-based Int. Control with a Throughput/Fairness Tradeoff. In: IFIP WIRELESS AND MOBILE NETWORKING CONFERENCE (WMNC 2015), 5-7), 8., Munique, Alemanha. **Proceedings...** [S.l.: s.n.], 2015. p.208–215.

SEMÁFOROS Inteligentes. Accessed: 2015-12-14, <http://www.poli.usp.br/comunicacao/noticias/arquivo-de-noticias/1088-semaforos-inteligentes.html>.

SHEFFI, Y. **Urban Transpostation Networks Equilibrium Analysis with Mathematical Programming Methods**. Englewood, NJ, USA: Prentice Hall Inc., 1985.

SHERALI ZEADALLY RAY HUNT, Y.-S. C. A. I. a. A. H. Vehicular ad hoc networks (VANETS): status, results, and challenges. **Telecommunication Systems**, [S.l.], v.50, n.4, p.217–241, 8 2012.

SOBRAL, M. et al. Um estudo sobre o fluxo de trânsito em uma via do município de Caruaru. In: REVISTA DOS MESTRADOS PROFISSIONAIS. **Anais...** [S.l.: s.n.], 2014.

TEIXEIRA, E.; FELTES, S.; SANTANA, E. Poluição atmosférica proveniente da queima de combustíveis derivados do petróleo em veículos automotores. **Química Nova**, [S.l.], n.2, p.1678–7064, 2008.