

Gustavo Soares da Silva Oliveira, Pedro Henrique Galhardi Cazelatto, Vitor  
Fernando Tonetti

# **Rede de Controle Viário para Redução de Tráfego**

São Paulo, SP

2023



Gustavo Soares da Silva Oliveira, Pedro Henrique Galhardi Cazellatto, Vitor  
Fernando Tonetti

## **Rede de Controle Viário para Redução de Tráfego**

Trabalho de conclusão de curso apresentado  
à Escola Politécnica da Universidade de São  
Paulo para obtenção do Título de Engenheiro.

Universidade de São Paulo – USP

Escola Politécnica

Departamento de Engenharia de Computação e Sistemas Digitais (PCS)

Departamento de Engenharia de Telecomunicações e Controle (PTC)

Orientador: Profa. Dra. Tereza Cristina Melo de Brito Carvalho

São Paulo, SP

2023



# Resumo

O crescimento populacional brasileiro nas últimas décadas se deu de forma desorganizada, gerando diversos problemas nos grandes centros urbanos, como o aumento do trânsito e dos congestionamentos. Dentre os diversos fatores que contribuem para essa questão está o controle antiquado executado pelos semáforos, que atualmente não se utilizam da evolução da tecnologia para gerar respostas mais inteligentes e que beneficiem o fluxo de veículos nesse cenário. Nesse contexto, diversos algoritmos foram levantados baseando-se na utilização de dados em tempo real de forma a gerar métodos mais eficientes. Especificamente nos últimos anos, houve um aumento na literatura de implementações baseadas em procedimentos descentralizados, que possuem uma escalabilidade melhor dentro de um cenário de uma grande metrópole, como é o caso do “*Backpressure*”, derivado de mecanismos de controle de redes internet. No entanto, as especificações destes são ainda muito teóricas para o caso de tráfego de veículos, tendo o funcionamento dos algoritmos explicado, mas não especificando detalhes como o método de extração das informações no mundo real e a viabilidade dentro do contexto de um país subdesenvolvido como o Brasil. Esse trabalho analisou então as principais implementações derivadas do “*Backpressure*”, verificando suas possíveis execuções no mundo real e simulando-as dentro de cenários coerentes com a conjectura brasileira. Verificou-se ao final que as 4 implementações analisadas são passíveis de serem instaladas, tendo prós e contras a serem validados a cada situação que desejar-se aplicá-las.

**Palavras-chave:** Controle viário. Controle descentralizado. Backpressure.



# Abstract

Brazilian population growth in recent decades has occurred in a disorganized manner, generating several problems in large urban centers, including increased traffic and congestion. Among the various factors that contribute to this issue is the outdated control carried out over traffic lights, which currently do not use the evolution of technology to generate more intelligent responses that benefit the flow in order to improve this scenario. In this context, several algorithms were developed based on the use of real-time data in order to generate more efficient methods. Specifically in recent years, there has been an increase in the literature of implementations based on decentralized procedures, which have better scalability within a large metropolis scenario, as is the case with “Backpressure”, derived from internet network control mechanisms. However, the specifications of these are still very theoretical for the case of vehicle traffic, with the functioning of the algorithms explained, but not specifying details such as the method of extracting information in the real world and the viability within the context of an underdeveloped country such as Brazil. This work then analyzed the main implementations derived from “Backpressure”, verifying their possible executions in the real world and simulating them within scenarios consistent with the brazilian conjecture. In the end, it was verified that the 4 implementations analyzed are capable of being installed, with pros and cons to be analyzed in each situation in which they are desired to be applied.

**Keywords:** Road control. Decentralized control. Backpressure.





# Lista de ilustrações

Figura 1 – Links de uma junção de 4 vias . . . . .	18
Figura 2 – Fases de uma junção de 4 vias . . . . .	19
Figura 3 – Pseudocódigo do algoritmo de <i>Backpressure</i> . . . . .	20
Figura 4 – Fluxograma do método do trabalho. . . . .	26
Figura 5 – Faixas dedicadas . . . . .	34
Figura 6 – Fluxo de trabalho para simulação e avaliação de dados . . . . .	41
Figura 7 – Rede da junção simples para simulação . . . . .	43
Figura 8 – Rede da Avenida Paulista para simulação . . . . .	44
Figura 9 – Exemplo de atributo definido pelo usuário . . . . .	45
Figura 10 – <i>Vehicle attribute decision</i> . . . . .	46
Figura 11 – Modelo de gastos de combustível . . . . .	46
Figura 12 – Tempo médio de deslocamento dos veículos . . . . .	50
Figura 13 – Tempo máximo de deslocamento dos veículos . . . . .	51
Figura 14 – Média do consumo de combustível . . . . .	52



# Lista de abreviaturas e siglas

IoT	<i>Internet Of Things</i> (Internet Das Coisas)
V2I	<i>Vehicle-to-infrastructure</i> (veículo-a-infra-estrutura)
COM	<i>Component Object Model</i>



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
1.1	Motivação	14
1.2	Objetivos	15
1.3	Justificativa	15
1.4	Organização do Trabalho	16
<b>2</b>	<b>ASPECTOS CONCEITUAIS</b>	<b>17</b>
2.1	Algoritmo de <i>Backpressure</i>	17
2.1.1	Organização viária	17
2.1.2	Problema das trocas de fase e algoritmo de <i>Backpressure</i>	18
2.1.3	Variações do algoritmo	20
2.2	Características reais	21
2.2.1	Escalabilidade	21
2.2.2	Investimento e economia	21
2.2.3	Estado da tecnologia e evolução	21
2.2.4	Latência e tempo de processamento	22
2.3	Métricas de desempenho	22
2.3.1	Tempo médio de deslocamento	22
2.3.2	Tempo máximo de deslocamento	22
2.3.3	Gasto de combustíveis	22
2.4	Considerações do capítulo	22
<b>3</b>	<b>MÉTODO DO TRABALHO</b>	<b>25</b>
<b>4</b>	<b>ESPECIFICAÇÃO DE REQUISITOS</b>	<b>27</b>
4.1	Levantamento dos dispositivos reais	27
4.2	Escolhas das implementações	27
4.3	Simulador	27
4.4	Análise	27
<b>5</b>	<b>DESENVOLVIMENTO DO TRABALHO</b>	<b>29</b>
5.1	Levantamento de dispositivos reais	29
5.1.1	Tecnologias para detecção de presença de veículos	30
5.1.2	Tecnologias para caracterização de veículos	31
5.1.3	Tecnologias para troca de dados	32
5.2	Implementações do Algoritmo <i>Backpressure</i>	32

5.2.1	Taxas de Roteamento Desconhecidas . . . . .	33
5.2.2	Baseado em atraso . . . . .	35
5.2.3	<i>Backpressure</i> com conhecimento de fluxos . . . . .	37
5.3	Conjuntos de tecnologias e implementações . . . . .	38
5.4	Simulações . . . . .	40
5.4.1	Simulador e fluxo de trabalho . . . . .	40
5.4.2	Configuração das redes e propriedades . . . . .	41
5.4.2.1	Definição dos links e conectores . . . . .	42
5.4.2.2	Geração de atributos para o <i>Backpressure</i> e análise de dados . . . . .	42
5.4.2.3	Configurações gerais . . . . .	46
5.4.3	Execução dos <i>scripts</i> . . . . .	48
5.4.4	Apresentação dos resultados das simulações . . . . .	50
<b>6</b>	<b>CONSIDERAÇÕES FINAIS</b> . . . . .	<b>53</b>
6.1	Avaliação dos Resultados . . . . .	53
6.2	Contribuições do Projeto de Formatura . . . . .	53
6.3	Perspectivas de Continuidade . . . . .	54
	<b>REFERÊNCIAS</b> . . . . .	<b>55</b>

# 1 Introdução

A invenção dos automóveis representou uma mudança significativa no modo de viver do ser humano moderno. Longas distâncias deixaram de ser impeditivos para a locomoção e o paradigma de residência próxima às atividades do cotidiano modificou-se, criando-se uma malha de deslocamento mais intensa.

No entanto, com o crescimento populacional, a expansão das grandes metrópoles e a verticalização das cidades, gerou-se um efeito crescente de congestionamento de veículos, tornando as viagens urbanas muito mais lentas e limitando o efeito de mobilidade inicialmente introduzido pelos automóveis.

Diversas abordagens são feitas em conjunto para tentar modificar essa situação, seja de forma direta ou indireta, tais como:

- Introdução de radares e lombadas para minar acidentes e, conseqüentemente, diminuir focos de tráfego lento;
- Alargamento de rodovias e ruas;
- Política de rodizio de carros para diminuição da frota diária;
- Introdução de semáforos com diferentes fases e tempos para controlar o fluxo e evitar incidentes;
- Investimento em transportes individuais de menor porte, como bicicletas, motocicletas e patinetes, e nas estruturas necessárias para os mesmo, como ciclovias;
- Investimento em transportes coletivos de forma a comportar mais pessoas dentro de cada metro quadrado nas vias, como no caso de ônibus, ou no uso de espaços não ocupados pelos veículos tradicionais, como metrô e balsas.

Os dois últimos itens são extremamente relevantes e foram muito bem aplicados em outros países, como a Holanda, na qual cerca de 25% das viagens são realizadas por bicicletas. Segundo estudo do governo holandês (HARMS; KANSEN, 2018), as bicicletas são essenciais para diminuir o contingente de veículos nas ruas. No entanto, em centros urbanos mais verticalizados e em países com outras culturas, pode ser difícil uma adoção em massa desse meio de transporte.

## 1.1 Motivação

Dentre os itens apresentados na seção anterior, aquele que mais se apresenta como um elemento de controle e que poderia ter o benefício da associação com a tecnologia, mas ainda está muito engessado em moldes antigos, são os semáforos. Estes apresentam-se como elementos que ditam o comportamento dos motoristas em tempo real, em contraponto com os outros listados, podendo então ser empregados para evitar o trânsito.

A expansão do conceito IoT, microcontroladores e dispositivos de sensoriamento contrastam com o funcionamento dos semáforos até o presente momento. Normalmente estes dispositivos são apenas pré-programados, muitas vezes sem o uso de dados temporais para a definição de fases. Porém, o trânsito é influenciado por diversos fatores, como o dia da semana, tempo meteorológico, estações do ano ou ocorrência de acidentes. Dessa forma, não basta haver um planejamento prévio do que se espera acontecer, mas também deveria haver uma reação baseada na grande quantidade de dados capazes de serem produzidos e transmitidos. Essa discrepância se apresenta como o motivador central para a tentativa de estipular um sistema de controle visando diminuir o tráfego.

Essa análise da necessidade de introdução de tecnologia no processo de decisão dos semáforos não é nova. Algoritmos como SCOOTs (HUNT et al., 1981) e SCATS (LOWRIE, 1982) vêm sendo apresentados desde o início dos anos 1980. No entanto, esses dois modelos, assim como outros similares, consistem na coleta de dados em tempo real para serem processados por uma unidade centralizada, que então atribui os parâmetros a cada uma das intersecções que possuem semáforos. Este esquema é impraticável, ao menos de forma facilmente escalonável, em áreas metropolitanas modernas, nas quais a quantidade de nós, dados e decisões a serem tomadas seriam tão grandes que geraria problema de sobrecarga no processamento.

Visando apresentar alternativas a essa solução, recentemente surgiram na literatura possibilidades de algoritmos de controle descentralizados. Nestes esquemas, o problema consiste na otimização em cada uma das junções ao invés da avaliação global. Dentre esses métodos, um com a maior quantidade de referências e diferentes variações de implementações encontrado na literatura é o algoritmo de *Backpressure*, modelo que foi inspirado em um algoritmo de roteamento de pacotes para internet, primeiramente expandido para o cenário de veículos por Wongpiromsarn et al. (2012).

No entanto, a análise apresentada na literatura ainda está fortemente ligada a simulações e a premissas não necessariamente encontradas no mundo real. Por exemplo, o algoritmo original apresentado no parágrafo anterior não especifica que tipo de sensoriamento seria utilizado, muito menos o investimento necessário para implementá-lo. Já a aplicação desenvolvida por Maipradit et al. (2021), embora apresente possíveis ganhos de desempenho em relação ao tempo de deslocamento de veículos, pressupõe um cenário



apenas com veículos autônomos, onde todos possuem sensores de GPS e velocidade precisos e podem se comunicar com o sistema de controle, algo extremamente distante da realidade de um país em desenvolvimento como o Brasil. Este trabalho visa então preencher esta lacuna de análise, investigando quais métodos seriam possíveis no cenário brasileiro e quais tecnologias podem ser utilizadas.

## 1.2 Objetivos

Este trabalho tem por objetivo analisar, por meio de um software de simulação, soluções de controle de tráfego viário baseadas no algoritmo *Backpressure* que reproduzam, da maneira mais fiel possível, um sistema cabível de ser implementado dentro do cenário brasileiro atual ou num futuro de curto prazo.

Para esta análise ser possível, deve-se inserir elementos no simulador que reproduzam o comportamento real do sistema físico esperado, adotando-se premissas que estejam de acordo com tecnologias existentes ou em fase de implementação.

A partir da introdução dessas características, os modelos devem então ser testados e poderá ser avaliada a funcionalidade esperada a partir de métricas relevantes.

## 1.3 Justificativa

Há alguns pontos que são relevantes e que justificam a tentativa de introdução de um controle de tráfego.

O primeiro deles, e o mais intuitivo de todos, consiste na otimização de tempo. O ser humano tem uma vida limitada, e é natural que se prefira utilizar o tempo em atividades consideradas mais relevantes, seja no âmbito pessoal ou profissional. O deslocamento é uma tarefa apenas intermediária na maioria dos casos, que não traz nenhum benefício em si, de forma que o tempo gasto nisso poderia ser aproveitado em outras situações.

Esse intervalo perdido no tráfego também está relacionado a outro fator que justifica a importância deste estudo. Seja pela sensação de perder tempo ou seja pelo cenário agressivo como o encontrado em muitas metrópoles o trânsito gera estresse e ansiedade, atrapalhando a saúde física e psicológica dos habitantes das cidades e muitas vezes até mesmo diminuindo a produtividade no trabalho.

O último ponto é que o excesso de trânsito gera prejuízos financeiros e ambientais. O Instituto de Transporte da Texas A&M estima que cerca de 100 bilhões de dólares sejam gastos devido ao desperdício de combustíveis e devido à perda de tempo para outras atividades (SCHRANK et al., 2021). Em um cenário em que cada vez mais se fala em pagada de carbono e consumo verde, diminuir as emissões se torna essencial.

## 1.4 Organização do Trabalho

O capítulo 2 versará sobre os conceitos teóricos necessários para o entendimento do projeto, focando nas métricas usadas para avaliar um algoritmo de controle de tráfego e no funcionamento do algoritmo *Backpressure*.

O capítulo 3 traz a especificação das etapas propostas para o desenvolvimento do projeto.

No capítulo 4 são apresentados os requisitos considerados essenciais para a completa análise que o projeto se propõe a fazer, considerando aspectos já mencionados e outros desenvolvidos nos próximos capítulos.

O capítulo 5 exhibe o desenvolvimento do trabalho, explicando passo a passo as atividades realizadas a fim de se obter os objetivos desejados.

No capítulo 6 são expostas as considerações finais, com ponderações acerca dos resultados obtidos, as contribuições do trabalho e a perspectiva de continuidade.

## 2 Aspectos Conceituais

Este capítulo tem por objetivo apresentar alguns conceitos chave que serão utilizados para o entendimento deste projeto. Primeiramente, estabelece-se o funcionamento base do algoritmo de *Backpressure*. O detalhamento das diferentes implementações dar-se-á frente neste trabalho. Após isso, apresentam-se as características dos modelos reais que possivelmente devem ser consideradas tanto para a modelagem da simulação quanto para a verificação de possibilidade de execução real. Por fim são definidas métricas importantes para estabelecer o desempenho de cada método.

### 2.1 Algoritmo de *Backpressure*

O algoritmo de *Backpressure* foi derivado do roteamento em redes, primeiramente introduzido para o problema de tráfego de veículos por [Wongpiromsarn et al. \(2012\)](#). Em seu formato original, ele é esquematizado para atuar de forma descentralizada resolvendo problemas de otimização locais. Nas seções seguintes são explicados o seu funcionamento conforme estruturado pelo autor, de forma sintetizada.

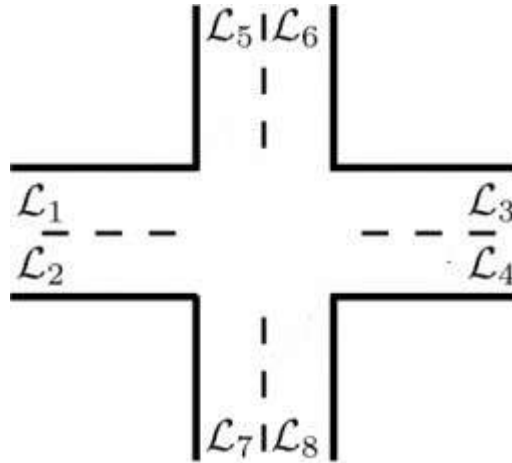
#### 2.1.1 Organização viária

A organização viária para o nosso problema se dá como um conjunto de links e junções sinalizadas. Nesse contexto, e no restante do trabalho, uma junção ou cruzamento é definido como uma intersecção na qual é necessário um controle semafórico para coordenar o movimento dos veículos. O link ou via, por sua vez, é um trecho de uma rua, avenida ou similar que conecta junções entre si ou que conecta uma junção a uma entrada ou saída da rede. Vale ressaltar que os links são orientados, dessa forma, um trecho de uma avenida em que transitam veículos nos dois sentidos constituem dois links diferentes, por exemplo.

O conjunto das junções é dado por  $\mathcal{J} = \mathcal{J}_1, \dots, \mathcal{J}_N$  e o dos links por  $\mathcal{L} = \mathcal{L}_1, \dots, \mathcal{L}_N$ . Na figura 1 podemos observar um exemplo de junção em que existem 8 links e que, considerando o lado da direção pela esquerda, possui 4 links de entrada à junção ( $\mathcal{L}_1, \mathcal{L}_4, \mathcal{L}_6$  e  $\mathcal{L}_7$ ) e 4 de saída ( $\mathcal{L}_2, \mathcal{L}_3, \mathcal{L}_5$  e  $\mathcal{L}_8$ ).

Cada junção  $\mathcal{J}_i$  pode ser descrita por uma tupla  $J_i = (\mathcal{M}_i, \mathcal{P}_i, \mathcal{Z}_i)$ . O primeiro desses elementos,  $\mathcal{M}_i$ , é um conjunto de pares ordenados  $(\mathcal{L}_a, \mathcal{L}_b)$  que representa todos os possíveis movimentos por  $\mathcal{J}_i$ , onde  $\mathcal{L}_a$  é um link de entrada e  $\mathcal{L}_b$  um link de saída. Para uma intersecção de 4 vias, ou 8 links, como a da figura 1, o conjunto  $\mathcal{M}$  é dado por:

Figura 1 – Links de uma junção de 4 vias



Fonte: modificado de [Wongpiromsarn et al. \(2012\)](#)

$$\begin{aligned} \mathcal{M} = \{ & (\mathcal{L}_1, \mathcal{L}_3), (\mathcal{L}_4, \mathcal{L}_2), (\mathcal{L}_6, \mathcal{L}_2), (\mathcal{L}_7, \mathcal{L}_2), \\ & (\mathcal{L}_1, \mathcal{L}_5), (\mathcal{L}_4, \mathcal{L}_5), (\mathcal{L}_6, \mathcal{L}_3), (\mathcal{L}_7, \mathcal{L}_3), \\ & (\mathcal{L}_1, \mathcal{L}_8), (\mathcal{L}_4, \mathcal{L}_8), (\mathcal{L}_6, \mathcal{L}_8), (\mathcal{L}_7, \mathcal{L}_5) \} \end{aligned} \quad (2.1)$$

Adicionalmente, um subconjunto de movimentos que pode ser realizado ao mesmo tempo, sem haver conflito na passagem dos veículos, é denominado de fase. O conjunto das fases de uma junção  $\mathcal{J}_i$  é representado por  $\mathcal{P}_i$ . No caso de 4 vias, apresentado na figura 1, temos um conjunto  $\mathcal{P}$  que pode ser observado no grupo de equações 2.2 e na figura 2:

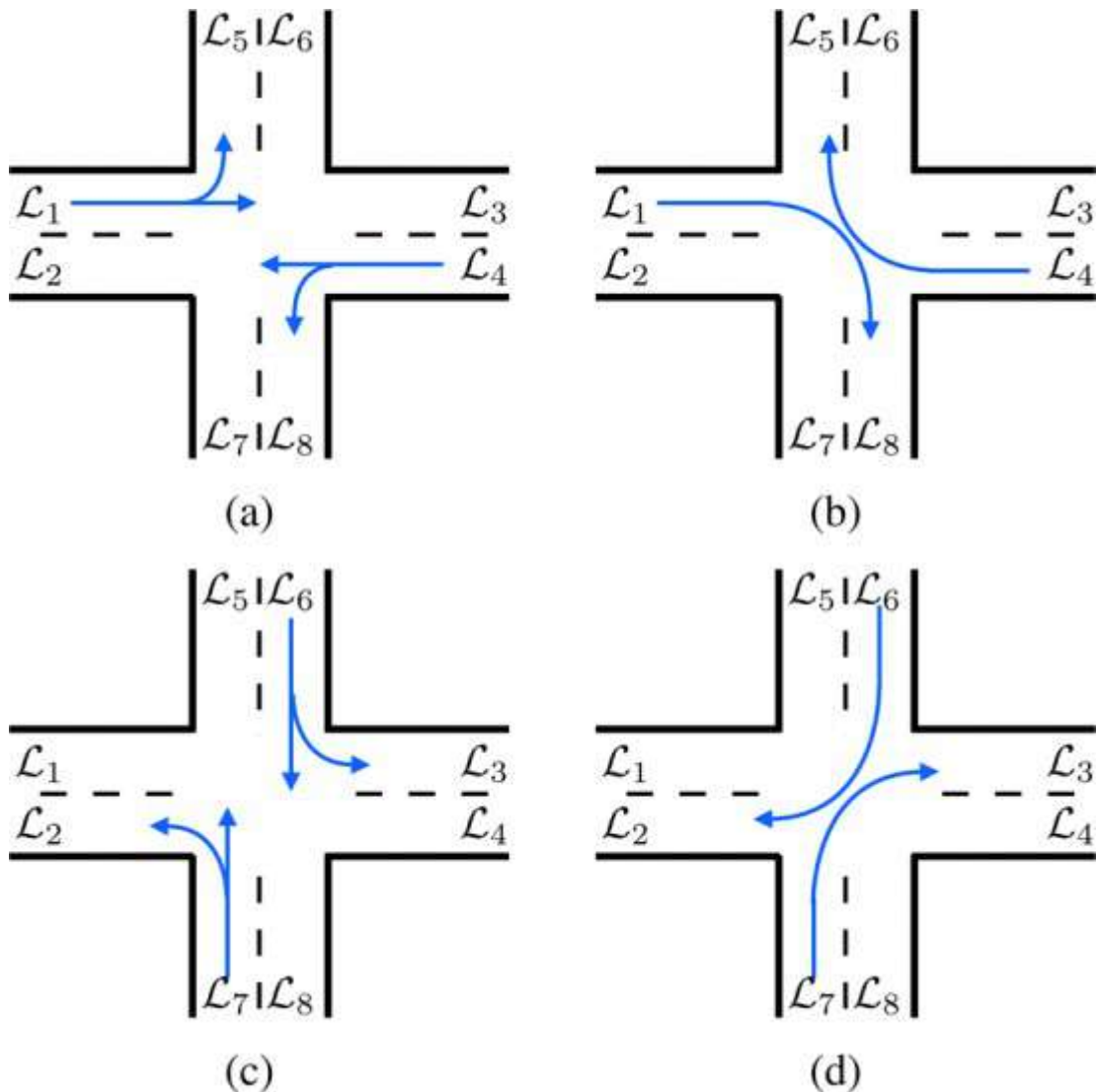
$$\begin{aligned} \mathcal{P} &= \{\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_4\} \\ \mathcal{P}_1 &= \{(\mathcal{L}_1, \mathcal{L}_3), (\mathcal{L}_1, \mathcal{L}_5), (\mathcal{L}_4, \mathcal{L}_2), (\mathcal{L}_4, \mathcal{L}_8)\} \\ \mathcal{P}_2 &= \{(\mathcal{L}_1, \mathcal{L}_8), (\mathcal{L}_4, \mathcal{L}_5)\} \\ \mathcal{P}_3 &= \{(\mathcal{L}_7, \mathcal{L}_5), (\mathcal{L}_7, \mathcal{L}_2), (\mathcal{L}_6, \mathcal{L}_8), (\mathcal{L}_6, \mathcal{L}_3)\} \\ \mathcal{P}_4 &= \{(\mathcal{L}_7, \mathcal{L}_3), (\mathcal{L}_6, \mathcal{L}_2)\} \end{aligned} \quad (2.2)$$

Por fim,  $Z_i$  representa um conjunto finito de estados de tráfego que capturam fatores que influenciam o fluxo na junção  $J_i$  como congestionamentos e condições climáticas.

### 2.1.2 Problema das trocas de fase e algoritmo de *Backpressure*

Considerando a modelagem realizada na seção anterior, o problema consiste em, para cada junção, escolher no início de cada intervalo de tempo  $t$  qual fase  $p$  deve-se ativar, ou seja para qual delas deve-se liberar o fluxo de veículos, tal que o fluxo da malha seja maximizado.

Figura 2 – Fases de uma junção de 4 vias



Fonte: [Wongpiromsarn et al. \(2012\)](#). As fases de  $\mathcal{P}_1$  a  $\mathcal{P}_4$  são representadas, respectivamente, de (a) a (d)

Esse método apresenta-se da seguinte forma: considere que exista um sistema de monitoramento capaz de fornecer ao controlador as filas dos links  $Q_w(t)$  e os estados de tráfego  $Z_i(t)$ . Considerando uma junção  $\mathcal{J}_i$  qualquer, o algoritmo consiste em três passos principais, sendo o primeiro deles calcular o peso associado a cada movimento dentro de  $\mathcal{M}_i$ :

$$W_{ab}(t) = Q_a(t) - Q_b(t) \quad (2.3)$$

Após isso o controlador deve computar, para cada uma das fases da junção, a pressão associada a cada uma delas, sendo  $\xi_i$  a taxa na qual os veículos podem se locomover de  $\mathcal{L}_a$  para  $\mathcal{L}_b$ , calculado por parâmetros já especificados:

$$S_p(t) = \sum_{(\mathcal{L}_a, \mathcal{L}_b) \in p} W_{ab}(t) \xi_i(p, \mathcal{L}_a, \mathcal{L}_b, z_i(t)) \quad (2.4)$$

Por fim, seleciona-se para ativar durante o intervalo de tempo  $t$  a fase que, naquele momento, maximiza a pressão da malha:

$$p^* = \arg \max_{p \in \mathcal{P}_i} S_p \quad (2.5)$$

Uma possível implementação desses passos é dada em pseudocódigo na figura 3.

Figura 3 – Pseudocódigo do algoritmo de *Backpressure*

**Algorithm 1:** Computation of phase  $p^*$  to be activated during time slot  $t$  at junction  $\mathcal{J}_i$ .

---

**Input:**  $z_i(t)$  and  $Q_a(t)$  for all  $a \in \{1, \dots, N\}$  such that  $(\mathcal{L}_a, \mathcal{L}_b) \in \mathcal{M}_i$  or  $(\mathcal{L}_b, \mathcal{L}_a) \in \mathcal{M}_i$  for some  $\mathcal{L}_b \in \mathcal{L}$

**Output:**  $p^* \in \mathcal{P}_i$  to be activated during time slot  $t$

- 1  $S_p^* \leftarrow -\infty$ ;
- 2  $p^* \leftarrow \emptyset$ ;
- 3 **foreach**  $(\mathcal{L}_a, \mathcal{L}_b) \in \mathcal{M}_i$  **do**
- 4    $W_{ab} \leftarrow Q_a(t) - Q_b(t)$ ;
- 5 **foreach**  $p \in \mathcal{P}_i$  **do**
- 6    $S_p \leftarrow \sum_{(\mathcal{L}_a, \mathcal{L}_b) \in p} W_{ab} \xi_i(p, \mathcal{L}_a, \mathcal{L}_b, z_i(t))$ ;
- 7   **if**  $S_p > S_p^*$  **then**
- 8      $p^* \leftarrow p$ ;
- 9      $S_p^* = S_p$ ;

Fonte: [Wongpiromsarn et al. \(2012\)](#)

### 2.1.3 Variações do algoritmo

A implementação especificada acima corresponde ao modelo original do algoritmo. Ao longo do tempo a literatura produziu modificações que trouxeram melhorias para o funcionamento, mas levaram a outros tipos de restrições ou desvantagens. Uma implementação dada por [Maipradit et al. \(2021\)](#) por exemplo assume carros conectados à internet e que seguiriam o caminho escolhido pelo algoritmo de controle. Os resultados

obtidos por ele são promissores, porém este modelo pode estar muito mais longe de uma implementação real do que o modelo original. É necessário então verificar o funcionamento de cada uma dessas variantes de forma individual.

## 2.2 Características reais

Há uma série de características que servem para designar se um modelo de fato pode ser implementado no mundo real. Algumas das mais importantes, e a razão dessa importância, estão listadas nas subseções abaixo.

### 2.2.1 Escalabilidade

A escalabilidade é uma peça chave no problema observado. Como os locais onde se apresentam os maiores índices de atraso no trânsito são os grandes centros urbanos, e estes muitas vezes possuem milhares de junções sinalizadas, é necessário analisar se o esquema escolhido pode de fato se refletir em uma malha desse porte sem perda de eficiência.

### 2.2.2 Investimento e economia

O investimento disponível em países desenvolvidos e em desenvolvimento é bem diferente. Uma implementação que visa se estabelecer em um local como o Brasil precisa em geral ser o mais barata possível, visto que, mesmo que se comprove um benefício social, o valor inicial pode ser um impeditivo para o andamento do projeto. Logo, o uso de sensores e dos demais elementos de construção do sistema devem prezar a acessibilidade, mesmo que aconteça alguma perda de precisão.

Um outro ponto a ser levantado é o quanto de economia cada tipo de implementação diferente traria. Esta poderia ser tanto referente ao tempo, permitindo que se utilize o mesmo para outras atividades, quanto à redução do uso de combustíveis, que implica na redução da emissão de poluentes e do gasto com os mesmos por parte dos usuários finais.

### 2.2.3 Estado da tecnologia e evolução

Um ponto a ser discutido é o estado da tecnologia e a durabilidade do sistema. Como apresentado anteriormente, há soluções que consideram carros automatizados. Esse cenário, embora plausível num futuro já observável nos dias de hoje, ainda é muito restrito para um país como o Brasil. Dessa forma, embora a introdução desse tipo de veículo fosse interessante para o controle, ela não pode ser considerada essencial para o projeto. No entanto, ela pode, e deve, ser considerada como um acréscimo possível, devendo ser viável adicionar elementos ao sistema que introduzam essas novas tecnologias sem a necessidade da troca dele como um todo.

## 2.2.4 Latência e tempo de processamento

Dois elementos importantíssimos que não tem um enfoque grande na literatura são a latência e o tempo de processamento. Esses dois são possíveis de serem aferidos no simulador de forma simples e permitem verificar os requisitos mínimos de transmissão e capacidade de processamento necessários.

## 2.3 Métricas de desempenho

Várias métricas diferentes podem ser implementadas para caracterizar o desempenho do sistema simulado. Para efeitos de simplificação, optou-se por priorizar aquelas consideradas mais relevantes dentro da motivação para implementar o sistema. Estas estão listadas nas subseções a seguir.

### 2.3.1 Tempo médio de deslocamento

O tempo médio de deslocamento é o principal foco do problema e de abordagem do algoritmo já especificado em [2.1](#).

### 2.3.2 Tempo máximo de deslocamento

Embora o tempo médio seja a principal variável, é importante a preocupação com um segundo ponto que têm relação com a equidade do sistema selecionado.

Caso os fluxos dentro de cada um dos links não sejam semelhantes, podem haver variações grandes o suficiente para gerar situações em que uma das fases pouco ou nunca é selecionada, impedindo assim que os veículos se desloquem. Isso pode gerar uma assimetria não desejada e que deve ser evitada.

### 2.3.3 Gasto de combustíveis

Embora a diminuição do tempo médio de deslocamento indique uma tendência na redução do gasto de combustíveis, estas duas variáveis não estão ligadas de forma direta. O consumo é mais complexo, com influência da velocidade de deslocamento, distância percorrida e tempo parado. O levantamento desse fator é importante para considerar uma implementação mais eficiente que as demais.

## 2.4 Considerações do capítulo

Nesse capítulo foram apresentados pontos essenciais para o entendimento do projeto. Em resumo, os tópicos mais importantes desta parte da obra consistem em:



- O funcionamento base do algoritmo de *Backpressure* e como o mesmo se propõe selecionar a fase com o maior peso associado;
- A existência de variantes ao método original e como suas premissas podem melhorar a eficiência, apesar de acrescentar outros fatores a serem analisados;
- As características reais que devem ser observadas para corretamente permitir avaliar a viabilidade de uma implementação prática do algoritmo;
- As métricas que permitem comparar, de forma quantitativa, qual modelo simulado apresenta respostas mais vantajosas para os casos testados.

Todos esses itens formam a base para as atividades realizadas nos próximos capítulos.



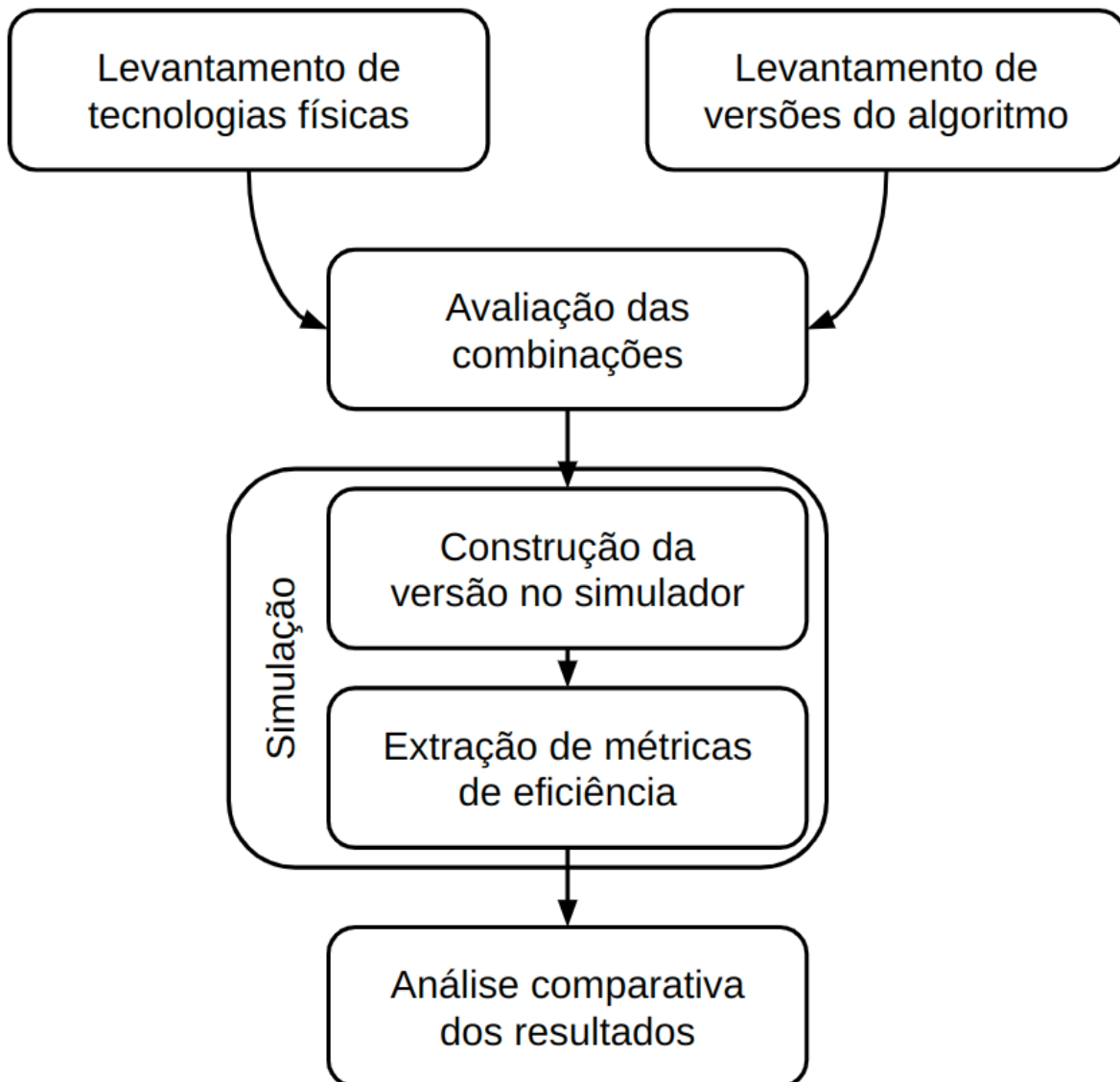
## 3 Método do trabalho

No intuito de desenvolvimento do projeto, definiu-se o seguinte conjunto de passos a serem realizados para alcançar o objetivo desejado.

- **Levantamento das tecnologias consideradas passíveis de serem utilizadas dentro do contexto do cenário brasileiro**, levando em conta fatores como estado da tecnologia atual e expectativas de curto prazo, investimento necessário, funcionalidade de cada um dos fatores, entre outros. Esta análise deve incluir, por exemplo, tipos de controladores disponíveis, tecnologias e topologias de rede e possibilidades de sensoriamento.
- **Levantamento das principais implementações de algoritmos de *Backpressure*** para controle de tráfego presentes na literatura atual, listando vantagens e desvantagens de cada uma delas, assim como as premissas assumidas pelos autores.
- **Avaliação das alternativas:** Baseado nas alternativas levantadas no item anterior, será feita uma avaliação sobre a possibilidade de implementar cada uma delas e quais componentes dentre os descritos no primeiro item fariam parte de cada uma das alternativas.
- **Teste de procedimentos:** Tendo uma lista de cada um desses diferentes procedimentos, será implementada no simulador o algoritmo que testaria o funcionamento desses elementos em conjunto com suas diversas características. Serão produzidas métricas capazes de estimar a eficiência de cada um deles, comparando-os entre si e com um esquema de funcionamento de semáforos tradicional com tempo e fase fixos.
- **Análise da escolha dos métodos:** Ao final será feita uma a comparação dos diversos métodos analisando qual seria a melhor escolha, observando-se o investimento de cada um deles, as métricas de retorno e a possibilidade real do sistema ser implementado.

Organizando essas ações de forma temporal, pode-se criar um fluxograma tal qual o apresentando na figura 4. Os dois primeiros passos podem ser feitos em paralelo, sem dependência entre resultados. Após estes, todo o trabalho desenvolve-se de forma sequencial.

Figura 4 – Fluxograma do método do trabalho.



Fonte: Autoria própria.

## 4 Especificação de Requisitos

Partindo dos objetivos do projeto, definem-se requisitos para guiar o desenvolvimento do projeto de análise.

### 4.1 Levantamento dos dispositivos reais

O levantamento dos dispositivos reais deve levar em conta as características identificadas no capítulo 2. Deve-se, portanto, ser feito um comparativo técnico entre as diferentes possibilidades levantando prós e contras de cada uma delas.

### 4.2 Escolhas das implementações

Deseja-se obter, pelo menos, 3 implementações que difiram em algum ponto relevante, seja pelo uso de uma variação de otimização no algoritmo apenas por programação, nas premissas de sensoriamento ou com adição de tecnologias.

### 4.3 Simulador

O simulador deve representar de forma precisa um cenário de tráfego de veículos, lidando com as variáveis decorrentes de cenário complexo. Para tal será utilizado um software já específico para esse propósito, como por exemplo o PTV Vissim ou o pacote de código aberto SUMO, trazendo uma maior veracidade para a simulação.

Deve ser possível também inserir os elementos que representem o comportamento do sistema real, conforme já foi elucidado nos capítulos 1 e 3. Além disso, deve ser possível extrair as métricas estabelecidas de forma a realizar as comparações necessárias.

Tendo essas características, o simulador deve ser capaz de reproduzir as implementações escolhidas assim como uma opção de controle estático, com tempo e fase fixos, para comparação de funcionalidade.

### 4.4 Análise

Com os resultados das diferentes implementações simuladas será realizada uma análise buscando compreender a viabilidade das opções disponíveis. A viabilidade dos sensores físicos será avaliada de acordo com as características definidas na seção 2.2, e

a viabilidade dos algoritmos será avaliada de acordo com os resultados das simulações considerando as métricas definidas na seção 2.3.

A comparação técnica elencará tanto as soluções de *software* como de *hardware*, indicando qual combinação das duas gerará mais benefícios à sociedade, como economia de tempo e redução de gasto de combustíveis, com o melhor custo-benefício associado, de forma que facilmente escalável pela cidade e não necessite de investimentos pesados por quem for instalar o sistema.

## 5 Desenvolvimento do Trabalho

Neste capítulo será apresentado o desenvolvimento das atividades conforme os itens apontados no método de trabalho, explicitado no capítulo 3. Têm-se por objetivo levantar, até o final deste fragmento, as variantes dos algoritmos de *Backpressure*, as tecnologias associadas para serem de fato executadas no mundo real e as métricas resultantes dos dados da simulação para efeitos de comparação.

### 5.1 Levantamento de dispositivos reais

Nesta seção são levantadas as considerações sobre as tecnologias possíveis de serem implementadas fisicamente em um cenário de controle de tráfego viário.

Um ponto a ser levantado antes do prosseguimento desta parte do documento é que, na seção 2.2, foram definidas as características reais que seriam relevantes de serem obtidas, refletidas nas simulações e consideradas para análise da melhor implementação. No entanto, com o desenvolvimento das pesquisas deste capítulo, observou-se que documentos técnicos detalhados sobre essas tecnologias não são acessíveis de maneira simples, e na grande maioria das vezes sendo necessário um contato com departamentos de vendas de empresas. Efetuaram-se múltiplas tentativas para obtermos essas informações, contudo não obteve-se sucesso.

Portanto, foi possível apenas acesso a algumas referências encontradas de maneira esparsa e sem uma documentação adequada que asseguravam a sua veracidade. Dessa forma, uma vez que implementar esses poucos dados nos modelos de simulação não trariam um retorno relevante, por não permitir uma comparação fiel que retratasse a realidade, optou-se por não considerá-los, ao menos não de forma quantitativa.

Desse modo, a análise das tecnologias que serão apresentadas nos itens 5.1.1 à 5.1.3 é uma versão bem simplificada do seu formato previamente pensado, já que boa parte das informações técnicas não pôde ser obtida. No entanto, considera-se mesmo assim que ela apresenta uma base relevante para verificar as possibilidades de execução dos algoritmos em um caso do mundo real. Com isso, observa-se a necessidade de estudos adicionais a este documento de forma a complementar a ausência desses dados e trazer mais profundidade a alguns elementos. Para cada um dos tópicos levantados no item 2.2 definiu-se então o que seria analisado:

- Escalabilidade: Foram consideradas características em relação à instalação física dos componentes por trás de cada tecnologia. Quanto mais rápido e barato essa for, mais

escalável é o processo;

- Investimento e economia: Não foi viável obter dados quantitativos em relação aos preços associados a cada tecnologia, mas ainda sim foi possível inferir diferenças qualitativas. Por exemplo, uma opção mais nova e pouco utilizada tende a ser mais onerosa que outra já inserida no mercado há muito tempo;
- Estado da tecnologia e evolução: Neste tópico optou-se pelo foco no atual uso de cada uma das tecnologias dentro do cenário brasileiro;
- Latência e tempo de processamento: Esses dados são os mais dependentes de especificações técnicas das quais não se teve acesso. Desta forma, optou-se por não caracterizá-los, e por consequência, também não os simular na seção 5.4

Além disso, como não foi possível obter documentações dos sensores, não há modos de estabelecer quais protocolos e tipos de comunicação eles utilizam. Isso torna impraticável escolher um microprocessador associado pela falta de caracterização de seus requisitos, tornando qualquer escolha feita um palpite sem suporte técnico.

### 5.1.1 Tecnologias para detecção de presença de veículos

Um dos sensoriamentos mais básicos que se pode realizar no contexto de redes viárias é a detecção de um veículo em uma determinada posição da via. As principais soluções são listadas abaixo:

- Sensor de Laço Indutivo
  - Funcionamento: A partir de uma bobina instalada sob o asfalto, o controlador é capaz de detectar mudanças na indutância com a passagem de veículos. É possível associar os dados ao longo do tempo para realizar a contagem dos automóveis que passaram sobre o sensor em um determinado período.
  - Estado da tecnologia: Extremamente estabelecida, utilizada por exemplo em lombadas eletrônicas.
  - Escalabilidade: Para se instalar o sensor é necessário colocá-lo debaixo da via, o que é um processo que demanda mais tempo, tornando-o menos escalável.
  - Investimento: Baixo por se tratar de uma tecnologia já bem definida.
- Câmeras de detecção de presença
  - Funcionamento: É realizada a captura de imagens, que são então processadas para detectar a presença de um veículo em uma determinada posição da imagem. É possível contar os veículos somando-se todas as regiões que o controlador detecta que há um automóvel.



- Estado da tecnologia: Mais recente e ainda pouco utilizada, não depende de outras tecnologias associadas.
- Escalabilidade: Instalação depende apenas da colocação da câmera no local desejado.
- Investimento: Por ser uma tecnologia recente exige um aporte financeiro maior.

### 5.1.2 Tecnologias para caracterização de veículos

Além de detectar um possível veículo, pode ser necessário caracterizá-lo, ou seja, diferenciá-lo dos demais como um objeto único. As principais soluções para o problema são listadas a baixo:

- Câmeras com reconhecimento de placa
  - Funcionamento: Processa as imagens obtidas rodando algoritmos, que muitas vezes utilizam técnicas de inteligência artificial, para identificar a placa de cada veículo e caracterizá-lo.
  - Estado da tecnologia: Ainda pouco utilizado por ser um produto mais recente, mas não possui limitações em relação à necessidade de outras tecnologias.
  - Escalabilidade: Instalação é feita de maneira simples, sendo necessário apenas instalar a câmera posicionada para o local desejado.
  - Investimento: Por ser derivado de tecnologias recentes necessita de um capital mais alto para ser instalado.
- Comunicação entre veículos e infraestrutura (V2I)
  - Funcionamento: Em locais específicos posicionam-se infraestruturas capazes de se comunicar com os veículos. Esses podem passar informações, se identificado por exemplo.
  - Estado da tecnologia: Pouco utilizado por ser recente e depender da adição de módulos de comunicação em cada veículo presente na rede. Em geral, não se apresenta viável para o cenário brasileiro por necessitar da troca da frota inteira para seu pleno funcionamento, o que é improvável de acontecer pelo alto investimento necessário. Em um futuro distante no entanto, esse sistema pode ser eficiente para maximizar o controle sobre veículos autônomos, informando a eles acerca de condições das vias, permitindo realizar, por exemplo, o roteamento automático.
  - Escalabilidade: Instalação das infraestruturas é realizada de maneira simples em locais específicos. Já a dos veículos é complexa pela magnitude da quantidade de elementos.

- Investimento: Altíssimo por exigir a adaptação da frota inteira de veículos e por ser uma tecnologia mais recente.

### 5.1.3 Tecnologias para troca de dados

Na implementação original, assim como se espera para as demais que serão escolhidas, os dados de uma via podem ser utilizados por duas junções diferentes: a que está localizada no início da via e a que está localizada no fim da via. Dessa forma, deve ser necessário a troca de informações entre elas ou entre um elemento comum de acesso centralizado em cada via. Independente do modo, é necessário definir que tipo de tecnologia será utilizada para transmitir esses dados. Os dois principais são listados abaixo:

- Rede cabeada
  - Funcionamento: Troca de dados é feita a partir de cabos, que necessitam ser instalados junto ao restante do sistema.
  - Estado da tecnologia: Modo mais comum nos dias atuais de se trocar informações em sistemas que exigem maior confiabilidade.
  - Escalabilidade: Instalação dos cabos deve ser feita de forma a conectar cada uma das junções, o que é um processo lento, por consequência menos escalável.
  - Investimento: Baixo por se tratar de uma tecnologia já bem definida.
- Rede 5G
  - Funcionamento: Dados são trocados usando transmissão por rede celular, suportada pela tecnologia 5G, que traz garantia de serviços como baixo tempo de latência e maior banda.
  - Estado da tecnologia: As redes 5G no Brasil começaram a ser instaladas recentemente, mas ainda estão limitadas. Há perspectiva de uma cobertura alta em grandes capitais dentro de alguns anos, porém em regiões mais afastadas esse processo pode demorar muito décadas.
  - Escalabilidade: Extremamente escalável do ponto de vista dos aparelhos que utilizam a rede 5G, mas quando se avalia as estruturas de comunicação, como por exemplo antenas, é necessário maior tempo de instalação.
  - Investimento: Alto por se tratar de uma tecnologia recente, tanto em aparelhos que utilizam a rede quanto para a instalação da infraestrutura da mesma.

## 5.2 Implementações do Algoritmo *Backpressure*

A partir do algoritmo original apresentado por [Wongpiromsarn et al. \(2012\)](#), diversos outros autores apresentaram modificações com a pretensão de aprimorar o funcionamento

do controle e/ou permitir premissas que facilitassem a sua execução fora do mundo digital das simulações.

São apresentadas nas subseções seguintes algumas dessas implementações, exprimindo suas especificidades, vantagens e desvantagens, em alguns casos específicos a dificuldade de simulação, discussões sobre o que diferencia elas do algoritmo original e uma análise acerca da viabilidade no cenário brasileiro. Ponderando os aspectos anteriores, para cada uma delas é feito um parecer sobre a decisão de realizar a simulação ou não dentro do escopo deste documento.

### 5.2.1 Taxas de Roteamento Desconhecidas

A primeira das implementações foi elucidada por [Gregoire et al. \(2014\)](#) e descreve o problema de conhecer a taxa de roteamento dos veículos que estão cruzando cada uma das vias. Em outras palavras, é estabelecida a problemática de saber qual movimento  $\mathcal{M}_i$  um determinado automóvel irá realizar dado que o mesmo está no link  $\mathcal{L}_i$ .

O artigo discute que essas taxas de roteamento estavam implícitas no algoritmo criado por [Wongpiromsarn et al. \(2012\)](#), escondidas pela parte temporal associada à taxa de locomoção  $\xi_i(p, \mathcal{L}_a, \mathcal{L}_b, Z_i(t))$  por  $Z_i(t)$ . Como esse componente é representativo dos estados de tráfego, que podem variar de maneiras muito complexas, pode ser necessário, no caso do mundo real, a inserção de uma quantidade muito grande de sensores ou agregadores de informação além dos necessários para se calcular a fila de cada uma das vias. Dessa forma, pode não ser tão simples realizar a execução correta do mesmo.

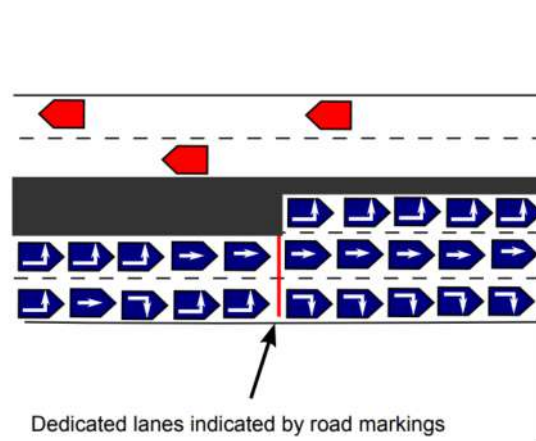
Uma abordagem para este problema é apresentada por [Gregoire et al. \(2014\)](#), com a aplicação de faixas dedicadas sinalizadas em cada via das junções. A utilização dessas já é comum hoje em dia em algumas vias públicas, com a pintura de setas orientadoras no pavimento. Tais direções servem para que motoristas possam prever o comportamento alheio olhando apenas a faixa que seus veículos ocupam. Uma seta à direita no chão indica que aquela faixa pode ser usada apenas para virar naquela direção, por exemplo.

O emprego de destas faixas dedicadas permite que um sensor externo identifique o movimento que os veículos farão, conforme observado na figura 5. A partir da linha vermelha, as faixas tornam-se dedicadas, e apenas automóveis que seguirão naquela direção podem permanecer na faixa.

Com essa disposição, o cálculo da fase a ser escolhida é feito de maneira um pouco diferente. Comparando-se com o algoritmo presente na figura 3,  $S_p$  agora é calculado conforme apresentado na equação 5.1.

$$S_p(t) = \sum_{(\mathcal{L}_a, \mathcal{L}_b) \in p} W_{ab}(t) \mu_{ab}(p) \quad (5.1)$$

Figura 5 – Faixas dedicadas



Fonte: [Gregoire et al. \(2014\)](#)

No qual  $\mu_{ab}(p)$  representa a taxa de locomoção em termos de quantidade de veículos na saturação designada pelo artigo como  $s_{ab}$ . Essa define o montante de automóveis que são transferidos de a até b durante o tempo da fase t caso esse movimento esteja liberado. A definição de  $W_{ab}$  também é diferente, apresentado na equação 5.2.

$$W_{ab} = \min(Q_{ab}(t)/s_{ab}, 1) \max(Q_a - Q_b, 0) \quad (5.2)$$

Nessa definição  $Q_{ab}(t)$  é a fila, localizada após a faixa exclusiva, de veículos desejando se locomover da via  $\mathcal{L}_a$  até  $\mathcal{L}_b$ , e  $Q_i$  é a somatória de todas as filas da via. Isso se apresenta como um outro benefício da parte de sensoriamento, uma vez que no algoritmo original é necessário avaliar a fila de toda a via, as possibilidades se resumem a captar “instantaneamente” esse dado ou realizar um processo integrativo para obtê-lo. No primeiro caso, deveria-se inserir um conjunto de sensores que pudessem, independentemente de dados passados, calcular a fila em toda a extensão do link, o que pode ser extremamente custoso a depender do tamanho do mesmo. Já no segundo, o princípio consistiria em calcular a fila pela entrada e saída de veículos em cada via das junções, realizando um processo integrativo, o que se tornaria mais barato que a opção anterior, mas exigiria elementos extras para realizar essa contagem. Além disso, esse método também traria susceptibilidade a erros maiores nos dados utilizados no algoritmo, já que automóveis podem sair ou entrar nos links fora das junções, por exemplo em caso de estacionamentos, influenciando no desempenho da implementação.

Outra característica da implementação definida por [Gregoire et al. \(2014\)](#) é de que, durante o intervalo de tempo t após a seleção de uma fase, apenas veículos dentro de alguma faixa exclusiva que teve o sentido liberado podem se locomover até um próximo link. Como as faixas exclusivas são pequenas, o período entre duas execuções da escolha

das fases é menor. A principal implicação disso seria um padrão diferente de condução pelos motoristas, que poderia gerar descontentamento pela maior probabilidade de parar após um curto intervalo e tendo como efeito colateral um possível aumento no gasto de combustíveis pela necessidade extra de acelerar e desacelerar.

De forma geral, as premissas dessa implementação apresentam-se muito interessantes para o cenário brasileiro, principalmente pela economia de instalação inicial que a mesma traria. No contexto do país, onde muitos projetos, apesar de viáveis a longo prazo, acabam não sendo executados pela falta de orçamento, isso seria um bom diferencial. Alguns pontos negativos, além dos já comentados, são a necessidade de mudança das vias para sinalizar as faixas exclusivas acrescido da dificuldade cultural de habituar o motorista a se posicionar corretamente na faixa exclusiva.

Considerando todos os aspectos levantados anteriormente decidiu-se por realizar a simulação da implementação especificada.

### 5.2.2 Baseado em atraso

A próxima implementação abrange uma preocupação já apresentada em [Wongpiromsarn et al. \(2012\)](#), que se expõe quantitativamente pelo conceito de *fairness* (que será aqui traduzido livremente como equidade), que consiste em utilizar métricas para avaliar o quão justo está sendo o tratamento entre os diferentes entes da rede. No caso de algoritmos de *Backpressure*, o cuidado se dá em verificar a diferença entre os atrasos dos veículos, ou seja a quantidade de tempo desde que o veículo entra em um link  $\mathcal{L}_a$  até que se desloque para o link  $\mathcal{L}_b$  desejado, em cada uma das junções. Isso é um problema relevante no esquema original pois como podem haver taxas de chegada muito diferentes entre os diversos movimentos, é possível que algumas fases sejam pouco selecionadas, já que no instante de avaliação do algoritmo elas possuem uma fila pequena em comparação às demais. Isso pode ocasionar efeitos indesejados na rede, como um veículo esperando vários minutos para poder prosseguir o seu caminho, o que seria inconcebível em um cenário no mundo real. Nesse contexto, dois algoritmos apresentados por [Wu et al. \(2018\)](#) apresentam variações que visam resolver essa problemática.

Iniciando da premissa de aumentar a equidade, o primeiro arranjo modificado apresenta a ideia de calcular a pressão associada a cada movimento do algoritmo *Backpressure* não partindo mais de métricas relacionadas ao tamanho das filas, como as utilizadas na figura 3, mas sim utilizando o tempo de permanência dos veículos dentro de cada junção. Dessa forma haveria uma tendência de equalizar a espera dos vários automóveis que desejam seguir caminhos diferentes, tornando o esquema mais justo.

Para implementar esse esquema, é estabelecido, em cada junção, o que é denominado por “*Sojourn time*”, ou tempo de estadia, representado por  $T_{ab,k}(t)$ . Este é determinado

para cada veículo  $k$  esperando na fila  $Q_{ab}$  como o intervalo desde a entrada dele no link  $\mathcal{L}_a$  até o tempo  $t$ .

Com essa definição modifica-se o cálculo do peso associado a cada movimento  $W_{ab}$ . Agora ele é determinado como o tempo de estadia do veículo no início da fila  $Q_{ab}$ . A taxa de locomoção utilizada na aferição de  $S_P$  também é simplificada, sem variar com as condições de tráfego como no algoritmo original, vide 2.4, mas sendo sempre numericamente igual ao fluxo de veículos que realizam o movimento num cenário saturado.

Partindo dessas premissas, observa-se uma diferença significativa em relação ao algoritmo original na questão de sensoriamento. Na versão primitiva não era necessário diferenciar entre diferentes veículos, bastava contá-los. Nessa nova implementação, é indispensável identificar cada automóvel que entra em um link, armazenar a informação sobre o tempo da sua entrada e depois comparar com aquele mais a frente da fila. Essa distinção gera uma limitação no tipo de tecnologia que pode ser utilizada, já que caracterizar cada veículo é um processo mais específico do que simplesmente verificar sua presença, podendo ser muitas vezes mais caro também. Além disso, assim como a técnica integrativa de contagem de veículos explicada na subseção 5.2.1, esse método está sujeito à automóveis entrando e saindo no meio do link. Neste caso, deve ser pensado qual seria o tempo de estadia atribuído a um veículo que adentrou uma via deste modo, já que o mesmo estaria indefinido.

Além dessa primeira implementação, Wu et al. (2018) define uma segunda que consiste em realizar o cálculo de  $S_p$  utilizando somas ponderadas entre aquelas que seriam realizados no algoritmo original e na versão apenas com uso de atraso detalhada nos últimos parágrafos. Para cada movimento é definido  $\eta_{ab}^Q$  e  $\eta_{ab}^D$  que expressam, respectivamente, os pesos da ponderação da variante baseada em filas e da baseada em atraso. Dessa forma, é possível priorizar de maneira distinta movimentos em que o tempo de espera é mais importante que o tamanho da fila e vice versa. É também estabelecido  $r_{ab} = \eta_{ab}^Q / \eta_{ab}^D$ , que apresenta uma proporção da prioridade do algoritmo original em relação ao de atraso.

Considerando os aspectos levantados nessa subseção, as duas implementações levantadas podem exigir um sensoriamento mais robusto por padrão que a original, o que poderia encarecer o produto. No entanto, a questão da equidade é extremamente importante, sendo necessário a sua compreensão e a elaboração de formas para impedir atrasos muito elevados em um cenário que se pretende colocar o sistema para rodar no mundo real. Não há também grandes limitações com relação às premissas que impediriam a execução do projeto.

Optou-se por realizar a simulação de ambas as implementações, pois a importância do problema da equidade se demonstra maior que as possíveis dificuldades que poderiam ser observadas. Além disso, como a segunda implementação possui um parâmetro relevante,  $r_{ab}$ , decidiu-se simula-la duas vezes, variando o valor desse.

### 5.2.3 Backpressure com conhecimento de fluxos

O último conjunto de implementações é baseado no fato de que é possível um melhor controle caso se possuam dados sobre a trajetória global dos veículos que adentram a rede. As discussões levantadas em [Zaidi, Kulcsár e Wymeersch \(2016\)](#) abrangem então dois controles do tipo *Backpressure* que presumem a associação de cada veículo entrando na estrutura viária a um fluxo, ou seja, um par ordenado de origem e destino.

O artigo explora em detalhes como são realizadas ambas as implementações especificadas, mas para o escopo deste projeto, será apresentado apenas superficialmente o funcionamento, visto que a especificação detalhada não traz discussões relevantes.

São detalhados dois modelos em [Zaidi, Kulcsár e Wymeersch \(2016\)](#). Um deles se baseia em um esquema de roteamento fixo e o outro adaptativo. No primeiro deles, considera-se que todos os veículos em um mesmo fluxo seguem caminho idênticos, já na segunda opção os automóveis são roteados ao entrar em cada link, ou seja, quando um deles entra em uma nova via  $\mathcal{L}_a$  o controlador decide dinamicamente qual dos movimentos o automóvel irá realizar, e por consequência em qual fila  $Q_{ab}$  ele entrará.

Ambos os modelos se utilizam do conceito de fila virtual, dessa forma ao invés de realizar a contagem do número de veículos  $Q_a(t)$ , em cada link é armazenado  $\tilde{Q}_a^f(t)$  para o caso de roteamento fixo e  $\tilde{Q}_a^d(t)$  para o adaptativo. Esses parâmetros representam os números de automóveis virtuais de cada fluxo  $f$  e de cada destino  $d$  esperando na via  $\mathcal{L}_a$ .

De forma mais detalhada, o que acontece é que quando um automóvel entra na rede física a partir de um  $\mathcal{L}_c$ , é adicionado um valor a  $\tilde{Q}_c^x(t)$  (onde  $x$  aqui é um termo genérico para representar fase ou destino, a depender da implementação sendo analisada) para exprimir a sua entrada. Com a liberação de um movimento  $(\mathcal{L}_c, \mathcal{L}_d)$ ,  $\tilde{Q}_c^x(t)$  é reduzido de  $s_{cd}(t)$  enquanto que  $\tilde{Q}_d^x(t)$  é acrescido dessa mesma quantidade. Esse cálculo de entrada e saída é realizado para todos os links e  $x$  diferentes.

Tendo essas filas virtuais, o algoritmo ocorre de maneira semelhante ao já apresentado na figura 3, com a diferença que o peso associado a cada movimento agora é dado pela equação 5.3. Dessa forma, é escolhido para cada  $W_{ab}$  apenas o fluxo ou destino que apresenta um maior peso para ser utilizado no cálculo.

$$W_{ab}(t) = \arg \max_{x \in x_i} (\max(\tilde{Q}_a^x(t) - \tilde{Q}_b^x(t), 0)) \quad (5.3)$$

A diferenciação entre as duas implementações se dá pelo fato de que, além de aplicar o *Backpressure* da forma explicada acima, a versão com roteamento calcula probabilidades para que os veículos, ao adentrarem um link  $\mathcal{L}_a$ , escolham um dos movimentos possíveis para seguir. De forma muito simplificada, isso permite com que automóveis seguindo o mesmo

fluxo sigam caminhos diferentes, distribuindo o trânsito e evitando congestionamentos localizados.

No entanto, apesar do artigo demonstrar benefícios das duas implementações em relação ao algoritmo original, especialmente em situações de alto tráfego, as suas premissas são bem fortes considerando o cenário brasileiro. Ambas consideram que, ao entrar na rede, o veículo informa o destino que está almejando, o que já é algo complexo de ser implementado, tanto por questões tecnológicas como de privacidade dos dados do motorista. Além disso, a primeira delas leva em conta que todos os veículos em um mesmo fluxo vão seguir caminhos idênticos, o que é improvável em um cenário em que as pessoas tomam decisões por si só, tentando naturalmente “evitar o trânsito”, podendo mudar de escolhas sobre o destino final a qualquer momento. A segunda é ainda mais restritiva, onde é necessário não só uma comunicação entre os automóveis e a infraestrutura de controle (V2I), mas também que os primeiros aceitem uma rota definida. Essas proposições só seriam plausíveis em contexto de carros autônomos e conectados à rede, algo totalmente desconexo com a realidade atual, não exigindo somente investimentos de ordem pública, mas que também a troca de toda a frota de veículos, algo que seria inconcebível.

Diante desses pontos levantados, foi considerado que as premissas invalidam a aplicação dessas implementações no Brasil, então não seria produtivo simular nenhuma das duas, já que isso não geraria um produto passível de ser realizado no mundo real (ao menos em um cenário de curto a médio prazo).

### 5.3 Conjuntos de tecnologias e implementações

Cada uma das implementações escolhidas em 5.2 exige um conjunto de informações e troca de dados diferente. Considerando as tecnologias levantadas em 5.1, é feita abaixo então uma correspondência do que poderia ser utilizado em cada um dos casos para a questão do sensoriamento:

- Original
  - Informações necessárias para rodar o algoritmo: tamanho das filas em cada uma das vias a todo instante
  - Primeira opção: Utilização de sensores de laço indutivo na entrada e na saída de cada uma das vias de forma a obter, por um processo integrativo, a quantidade de carros em cada uma delas.
    - \* Vantagens: baixo custo
    - \* Desvantagens: baixa escalabilidade



- \* Cenário mais provável de uso: aplicação em sistemas urbanos menos complexos no qual não é um problema fechar uma via para instalar os laços indutivos.
- Segunda opção: Utilização de câmeras de detecção de presença na entrada e saída de cada uma das vias de forma a obter, por um processo integrativo, a quantidade de carros em cada uma delas.
  - \* Vantagens: alta escalabilidade
  - \* Desvantagens: alto custo
  - \* Cenário mais provável de uso: aplicação em sistemas urbanos complexos no qual é um problema fechar uma via para instalação do sistema.
- Atraso
  - Informações necessárias para rodar o algoritmo: tempo de estadia do primeiro veículo na fila de cada um dos movimentos
  - Utilização de câmeras com detecção de placas no início e final de cada link. Ao adentrar um novo link, o tempo é anotado e depois comparado com o veículo mais a frente da fila para obter o tempo de estadia
    - \* Vantagens: alta escalabilidade
    - \* Desvantagens: alto custo
    - \* Cenário mais provável de uso: aplicação em sistemas urbanos complexos, no qual é um problema fechar uma via para instalação do sistema.
- Atraso e fila
  - Informações necessárias para rodar o algoritmo: tempo de estadia do primeiro veículo na fila de cada um dos movimentos, assim como o tamanho da fila de cada um dos links
  - Utilização de câmeras com detecção de placas e de veículos no início e final de cada link. Ao adentrar uma nova via, o tempo é anotado e depois comparado com o veículo mais a frente da fila para obter o tempo de estadia. Além disso é realizado um processo integrativo da diferença dos automóveis sendo detectados na entrada e na saída para se obter a quantidade de veículos em cada link
    - \* Vantagens: alta escalabilidade
    - \* Desvantagens: alto custo
    - \* Cenário mais provável de uso: aplicação em sistemas urbanos complexos no qual é um problema fechar uma via para instalação do sistema.
- Taxas de roteamento desconhecidas

- Informações necessárias para rodar o algoritmo: Fila dentro das faixas exclusivas de cada link
- Primeira opção: Utilização de sensores de laço indutivo na entrada e na saída de cada faixa exclusiva, obtendo de forma integrativa o tamanho das filas.
  - \* Vantagens: baixo custo
  - \* Desvantagens: baixa escalabilidade
  - \* Cenário mais provável de uso: aplicação em sistemas urbanos menos complexos no qual não é um problema fechar uma via para instalar os laços indutivos.
- Segunda opção: Utilização de câmeras de detecção de presença apenas apontadas para as faixas exclusivas. É possível obter a quantidade de carros diretamente pelos algoritmos da câmera sem um processo integrativo
  - \* Vantagens: alta escalabilidade
  - \* Desvantagens: alto custo
  - \* Cenário mais provável de uso: aplicação em sistemas urbanos complexos no qual é um problema fechar uma via para instalação do sistema.

Além disso, de forma geral para todas as implementações, caso seja desejado priorizar um investimento menor é preferível utilizar uma rede cabeada, mas caso a escalabilidade seja o fator principal, é melhor utilizar a tecnologia 5G, caso a mesma já esteja bem desenvolvida no local.

## 5.4 Simulações

Nesta seção serão apresentados todos os elementos referentes as simulações das implementações descritas anteriormente, desde a escolha de software, passando pela configuração dos arquivos até a apresentação dos resultados.

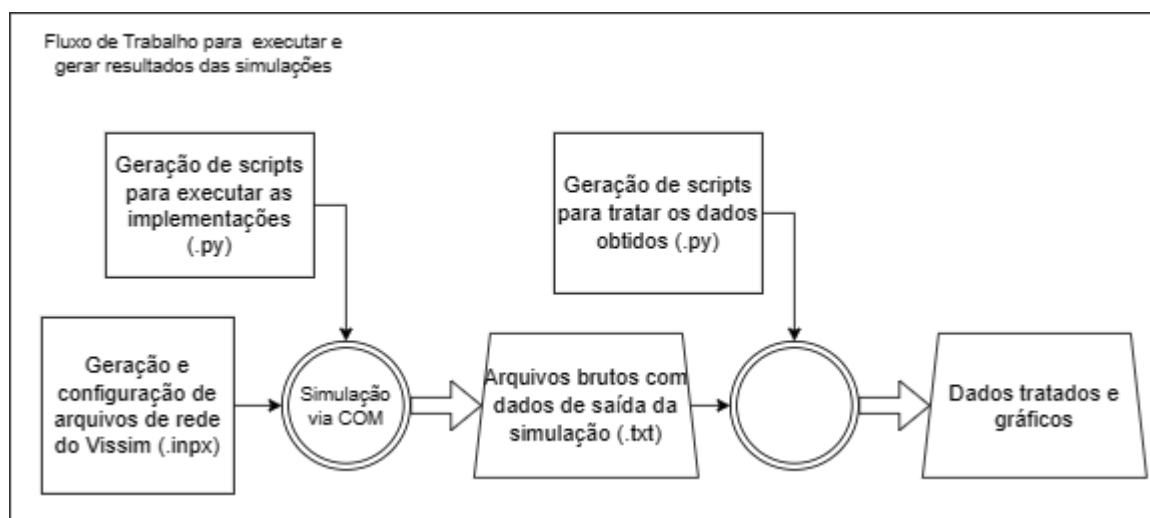
### 5.4.1 Simulador e fluxo de trabalho

Devido a possibilidade de acesso a uma licença associada à Universidade, optou-se pela utilização do aplicativo PTV Vissim, um dos principais *softwares* no mercado de microssimulação de tráfego. O *software* já possui todo o aparato necessário para calcular a dinâmica de evolução da rede baseada na interação individual entre cada um dos veículos com seus pares e com os demais componentes, como os próprios links, semáforos, áreas com maior ou menor velocidade máxima, entre outros. Além disso, o *software* também é capaz de introduzir, de forma automatizada, elementos estocásticos, como por exemplo a entrada de automóveis na rede, de forma a melhor caracterizar um cenário

real de deslocamento urbano, no qual é muito complexo observar padrões consistentes, se apresentando basicamente como variáveis aleatórias.

Para de fato simular as implementações escolhidas, não basta apenas a utilização do Vissim, visto que as suas opções de geração de algoritmos são limitadas. Dessa forma, uma solução apresentada para casos como esse é a interface COM, que permite a comunicação de outros processos ao elementos da simulação. Por ela é possível acessar métodos e componentes que permitem verificar e alterar o estado da rede e do próprio ensaio. No caso da proposta deste documento, o que se é desejado é, nos momentos de escolha de fase para cada implementação sendo simulada, a aplicação conectado ao simulador recebe os dados pertinentes para avaliar a fase a ser acionada durante o próximo intervalo de tempo. Essa é então transmitida para o software que prossegue a simulação a partir da modificação, e esse processo se repete até o fim do tempo estipulado para os testes. Além disso, enquanto se executam esses passos, os dados necessários para se avaliar cada cenário devem ser registrados para depois serem analisados. Esse fluxo de trabalho é demonstrado na figura 6, e é detalhado nas seções a seguir.

Figura 6 – Fluxo de trabalho para simulação e avaliação de dados



Fonte: Autoria própria. As caixas retangulares representam arquivos gerados pelos autores, os círculos são processos automatizados que utilizam os arquivos de entrada para gerar uma saída, marcadas pela caixas trapezoidais.

#### 5.4.2 Configuração das redes e propriedades

Como demonstrado na figura 6, a primeira atividade a ser realizada é a criação dos arquivos “.inpx”, que servem como base de funcionamento das simulações a serem realizadas. Subdividiu-se a tarefa em algumas etapas que estão detalhadas dentre 5.4.2.1 à 5.4.2.3.

#### 5.4.2.1 Definição dos links e conectores

O primeiro passo para gerar os arquivos e poder estabelecer nossa rede é realizar a criação dos links, utilizando tanto os critérios espaciais de posicionamento de cada um deles quanto gerando configurações secundárias como a velocidade que se espera que os veículos desejem atingir. Decidiu-se por criar dois cenários diferentes para avaliar as implementações sobre perspectivas distintas. O primeiro deles constitui uma junção isolada com quatro fases nos moldes da figura 2, e seria uma base simplificada para permitir testes iniciais e familiarização com o *software*, mas que ao mesmo tempo permite um controle mais específico sobre alguns parâmetros do que em uma rede maior. O segundo cenário é a virtualização de parte da Avenida Paulista e arredores, localizado na cidade de São Paulo. É um modelo com mais junções, proporcionando ver o deslocamento dos fluxos a partir do funcionamento do algoritmo, e também é representativo de um caso real, possibilitando avaliar como as implementações de fato seriam úteis.

Além dos links tradicionais também são definidos os conectores, que são estruturas básicas no modelo de funcionamento do Vissim que funcionam como “links especiais”. Eles permitem o deslocamentos dos veículos que estão sendo simulados de um link convencional para o outro. Nos casos em que esses estão localizados em junções sinalizadas eles representam os movimentos caracterizados na seção 2.1.

As figuras 7 e 8 representam, respectivamente, a estrutura básica das redes do cruzamento simples e da Avenida Paulista. Na primeira delas é possível observar de maneira mais detalhada os conectores que ligam os 8 links presentes no modelo a partir dos movimentos permitidos. Já na segunda, fica evidenciado o mapa do mundo real utilizado de plano de fundo para gerar a rede representativa da região.

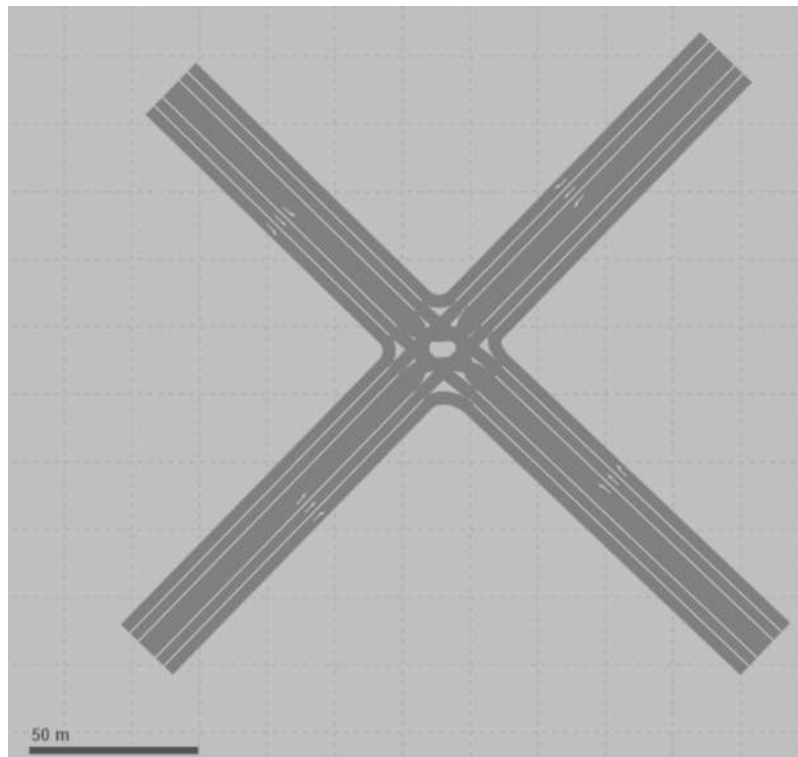
Vale ressaltar que, apesar das duas redes estarem sendo aqui mostradas de forma conjunta, isso se deu por motivos organizacionais e para facilitar o entendimento ao longo do documento. No desenvolvimento do projeto, primeiro foram realizadas as simulações para a junção simples e após isso, baseado nas características aprendidas, foi desenvolvida a rede da Avenida Paulista e replicadas as ações sobre ela.

#### 5.4.2.2 Geração de atributos para o *Backpressure* e análise de dados

O segundo trabalho dentro das configurações do arquivo está em esquematizar o acesso das informações necessárias para implementar os algoritmos escolhidos e colher os dados necessários para análise ao final das simulações.

O Vissim está organizado de tal forma que cada objeto dentro da rede de simulação possui atributos pré-definidos, que podem ser acessados para cada componente existente. Estes podem ser fixos ou variáveis no tempo simulado, modificando-se com base em fórmulas. Links, por exemplo, possuem um atributo fixo associado que expressa o comprimento de

Figura 7 – Rede da junção simples para simulação



Fonte: Autoria própria.

cada um deles. Por outro lado, também dispõem de um outro que varia com o tempo de simulação que representa a quantidade de carros presentes naquele momento. Esses atributos são todos acessíveis pela interface COM, permitindo então que os *scripts* acessem essas informações para rodar os algoritmos.

No entanto, observou-se que, para as implementações desejadas, seria necessário realizar muitas transformações nos dados que o Vissim oferece por padrão, o que tornaria os *scripts* a serem realizados posteriormente muito longos. Dessa forma, optou-se por uma funcionalidade disponível no *software* que é a criação de atributos definidos pelo usuário. Esses podem ser determinados também por fórmulas, permitindo uma gama de opções que funciona muito bem para o resultado que se deseja obter. Um exemplo é apresentado na figura 9, onde é definido um atributo para um conector que representa  $W_{ab}$  utilizado no algoritmo da implementação original, tal como se observa na equação 2.3. Como todas as variantes se baseiam em somar valores atrelados a cada um dos movimentos para obter o  $S_p$  associado as fases, foi realizado para os demais casos de algoritmos um processo análogo, criando fórmulas que resultassem nos dados adequados para cada um deles.

Dois dos atributos criados no entanto se diferem dos demais por não serem derivados diretos de uma fórmula, o “ATIME” e o “ $s_{ab}$ ”.

O primeiro deles representa o tempo de chegada no link em que ele está aguardando,

Figura 8 – Rede da Avenida Paulista para simulação



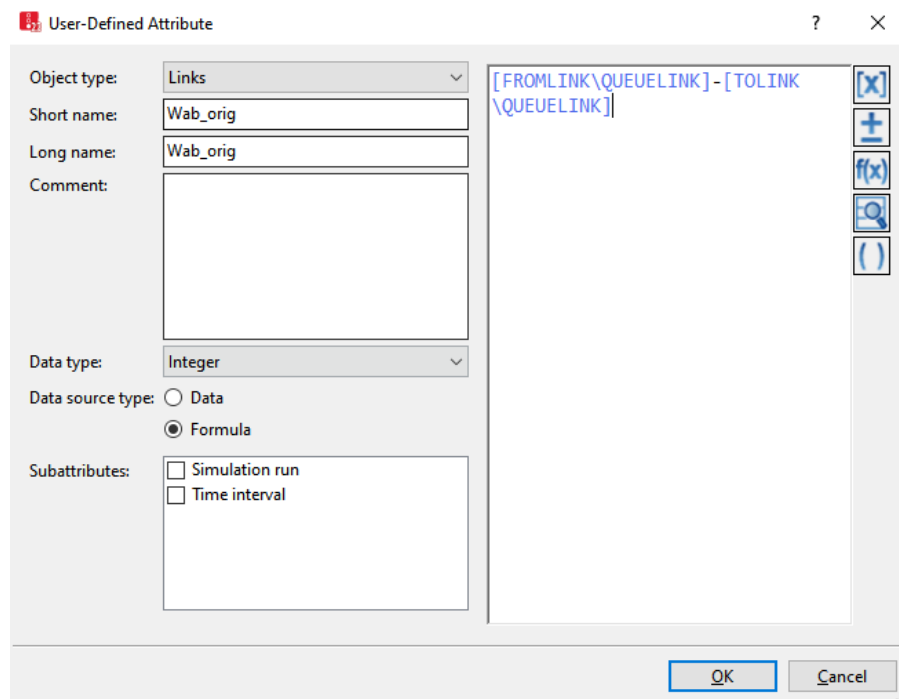
Fonte: Autoria própria.

utilizado para calcular o tempo de estadia definido na subseção 5.2.2. Como não foi possível estabelecer uma fórmula que representasse esse parâmetro, o atributo foi estabelecido como sendo do tipo “data” e o seu valor é modificado por um elemento de rede disponível no Vissim chamado de “*Vehicle attribute decision*”. Dessa forma, quando um automóvel passa por esse objeto, “ATIME” é atualizado como sendo o tempo da simulação naquele instante e só é alterado no momento que atravessa um outro componente igual. Com o posicionamento desses no início de cada link é possível então obter o comportamento desejado. Isso é demonstrado na figura 10.

Já o segundo é o valor do fluxo de transferência de veículos do link  $\mathcal{L}_a$  para  $\mathcal{L}_b$  em um cenário de tráfego saturado, associado a cada um dos conectores. Como isso não é um dado estabelecido pelo Vissim por uma fórmula, também optou-se pela utilização do tipo de atributo “data”, mas dessa vez esse foi configurado manualmente para cada um dos movimentos e permanece sendo o mesmo em toda a simulação. Para se obter esse valor realizou-se pequenos testes com veículos direcionados apenas ao movimento analisado e, com o uso de um semáforo, verificou-se quantos automóveis atravessam esse durante a abertura de fase. A partir disso calculou-se  $s_{ab}$  para cada um dos movimentos.

Além disso, há um último atributo que não é utilizado diretamente no controle dos algoritmos, mas que é necessário para analisar o resultado das simulações conforme especificado na seção 2.3. O uso de combustíveis é um dado essencial para comparar as implementações, porém esse não é implementado pelo Vissim na licença que obteve-se acesso, mas é possível calcular baseado nas informações da simulação.

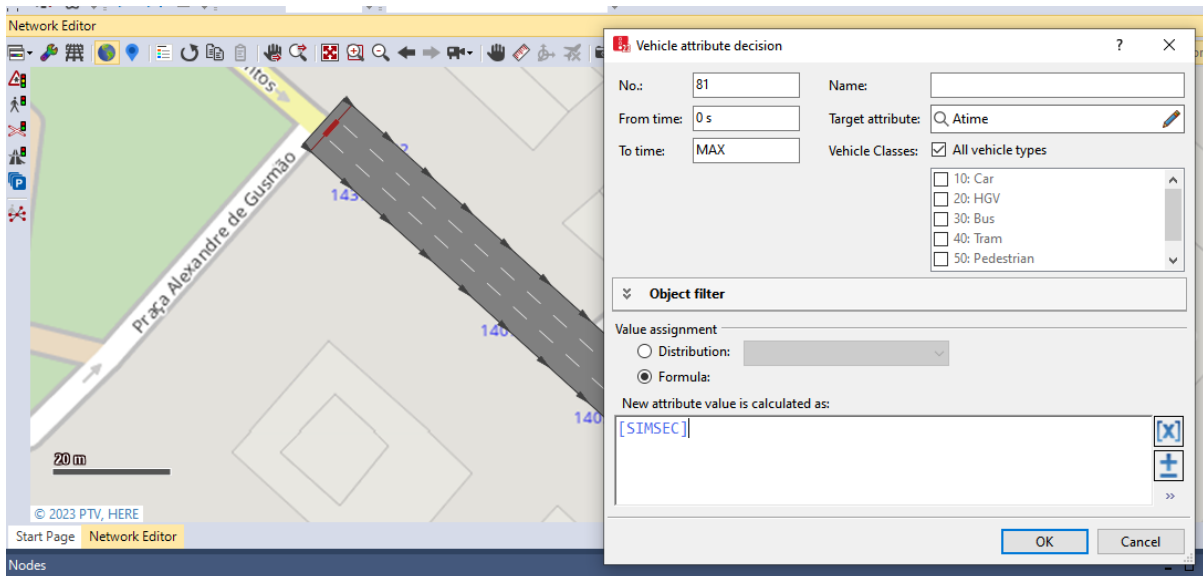
Figura 9 – Exemplo de atributo definido pelo usuário



Fonte: Autoria própria. Na imagem, é possível observar a definição do atributo  $W_{ab}$ , que é definido para um conector como sendo a diferença entre a fila do link de origem com a fila do link de destino

No manual de outro simulador de microtráfego muito utilizado no mercado, o “Aimsun”, é dado em [Aimsun-Manual \(2023\)](#) o modelo de gasto de combustíveis implementado por eles. Ele se baseia no fato de que, para cada estado que um veículo pode possuir, aceleração, desaceleração, velocidade contínua e inatividade, há uma fórmula associada que representa o gasto de combustível (em ml/s) baseado em variáveis como a velocidade do automóvel ( $v$ ), sua aceleração ( $a$ ) e uma série de constantes que dependem de cada veículo. Essas fórmulas são apresentadas na figura 11, e os valores das constantes são detalhados no manual da “Aimsun”. Vale ressaltar que o manual apresenta esses valores pra diversos tipos de veículos, porém para fins de simplificação e por haver uma preocupação maior com fins comparativos do que os valores exatos dos gastos de combustíveis, decidiu-se utilizar para todos os veículos na rede as constantes que o “Aimsun” determina para carros do tipo “*Medium/Large*”.

Por fim, vale ressaltar que optou-se por coletar os dados da para análise posterior utilizando os veículos presentes na rede. Dessa maneira, para cada passo da simulação, são guardadas informações pré-selecionadas (como velocidade, tempo desde o ingresso, gasto de combustível durante o último intervalo de tempo, entre outras) para cada automóvel presente naquele momento simulado. Preferiu-se gerar dados brutos ao invés de utilizar algumas ferramentas já presentes no Vissim para melhor analisar as métricas definidas na seção 2.3 e por haver maior controle para se obter gráficos considerados interessantes de

Figura 10 – *Vehicle attribute decision*

Fonte: Autoria própria. À esquerda da imagem é possível observar o elemento que realiza a modificação no atributo posicionado no início da via, e à direita o terminal que determina a configuração desse elemento que está atribuindo o tempo atual da simulação ao atributo “ATIME”.

Figura 11 – Modelo de gastos de combustível

Vehicle state	fuel consumed (ml) during t
Idling	$F_i \Delta t$
Accelerating with acceleration of $a$ (m/s/s) and speed $v$ (m/s)	$(c_1 + c_2 a v) \Delta t$
Cruising at speed $v$ (m/s)	$\left( k_1 \left( \frac{v^3}{2V_m^3} \right) + k_2 v \right) \Delta t$
Decelerating	$F_d \Delta t$

Fonte: [Aimsun-Manual \(2023\)](#). Os valores das constantes apresentadas nas fórmulas estão detalhadas no manual do Aimsun.

serem levantados.

#### 5.4.2.3 Configurações gerais

Para além das construções dos links, é necessário outras configurações para o funcionamento da simulação, com ou sem a utilização de *scripts* associados. Dentre esses estão:

- Definição da composição dos veículos na rede.



- Posicionamento dos semáforos e definição do controle semaforico para o caso de tempo fixo.
- Configuração do fluxo de entrada de veículos em cada um dos links de acesso, aqueles em que veículos são adicionados à rede de simulação.
- Determinação das taxas de roteamento para cada link.

A configuração desses elementos é essencial para um funcionamento adequado de toda a simulação e alguns cuidados devem ser tomados para se obter um cenário que minimamente reflita algo no mundo real.

A título de exemplo, em testes iniciais para o caso da junção simples sem uso de *script*, foi observado que ao realizar pequenas modificações em alguns parâmetros havia desestabilização da rede. Essa é caracterizada por veículos “presos”, que não conseguem se movimentar por realizar deslocamentos discordantes. Um possível caso em que isso ocorre é dado por um veículo tentando virar a esquerda estando na faixa do meio enquanto outros se deslocam pela lateral, impedindo que o automóvel entra nela. Nessa situação esse pode ficar parado de maneira indefinida, travando todos os veículos que estão atrás. Quando há esse acontecimento gera-se filas maiores, sobrecarregando os links e aumentando a probabilidade disso acontecer novamente. O Vissim pode até realizar a remoção desse automóvel depois de um tempo, mas mesmo isso não ajuda a representar um cenário real. Dessa forma deve-se configurar a rede de modo a evitar a ocorrência desse fenômeno.

Essa configuração é regulada em conjunto com os diversos parâmetros já expressados nessa seção, não bastando apenas observar o efeito individual de cada um deles, mas como se comportam juntos. Como há muitas variáveis possíveis, foi necessário fixar algumas delas para tornar a análise possível. Essas escolhas são explicadas ao longo dos próximos parágrafos.

A primeira decisão feita foi definir que na composição de veículos na rede haveriam apenas carros, eliminando motocicletas e ônibus, por exemplo. Essa definição foi feita visando dois pontos: simplificar a configuração dos outros elementos, diminuindo os tipos de interações possíveis, e também pela questão de que, nas implementações escolhidas, não há referências sobre outros tipos de veículos, nem como a introdução desses pode levar a alguma perda de desempenho, sendo inadequado acrescentar esses outros meios de locomoção sem uma análise mais profunda que fugiria do contexto da proposta desse projeto, que é analisar as variantes já existentes.

Para a configuração dos outros três elementos os métodos realizados foram um pouco diferentes para o caso da junção simples e da Avenida Paulista. No entanto, para ambos os casos assumiu-se uma premissa em comum, desejaria-se que, pelo menos para os links de acesso com maior volume de entrada de veículos, o ajuste fosse feito de forma a

estar no limite da desestabilização, obtendo o maior número de automóveis possível, de modo a criar um trânsito maior e ampliar a diferença entre o caso de tempo fixo e o das implementações do *Backpressure*.

Para a junção simples não havia nenhuma informação prévia, já que se tratava de um cenário experimental, dessa forma todos os parâmetros precisaram ser definidos do zero. A fim de simplificar a configuração, assumiu-se a primeira hipótese de realizar o roteamento em cada link de modo igualitário, ou seja, um carro entrando teria chances iguais de virar à esquerda, à direita ou seguir em frente. Definiu-se volumes iniciais desiguais para os links de acesso, com esses possuindo fluxos bem diferentes entre si, de forma a analisar como os algoritmos lidam com esse cenário, que seria mais adverso do que com fluxos semelhantes. Para atingir o patamar pré-desestabilização citado anteriormente e calibrar o controle semafórico foi utilizado uma ferramenta que permite otimizar os sinais dentro de uma área, que o próprio Vissim oferece. Aumenta-se o fluxo dos links a partir dos valores inicialmente estabelecidos e realiza-se a otimização de maneira iterativa, até observar que a a rede começava a desestabilizar e então fixou-se os parâmetros semafóricos obtidos.

Já para o caso da Avenida Paulista, foi possível realizar a aferição pessoalmente dos tempos de fase da via principal e das transversais, e a partir dessa configurar também as junções laterais com intervalos semelhantes. Dessa forma, pode-se partir de um cenário mais próximo ao do mundo real e com menos indefinições se comparado com o anterior. O roteamento também presume ser semelhante ao que se espera da atividade na região no horário de pico, tendo dentre as características por exemplo a maioria dos carros seguindo na avenida principal enquanto apenas alguns viram para as ruas laterais. Por fim, a entrada de veículos na rede foi baseado num estudo feito pela [CET \(2018\)](#), que demonstra que em horário de pico cerca de 2500 automóveis atravessam a Avenida Paulista por hora. Esse número então foi usado como base e alterado para calibrar a rede conforme o quesito de estar no limiar da desestabilização, já explicado anteriormente. Os acessos secundários foram configurados de forma a possuir volumes inferiores em comparação com a Avenida para condizer também com a realidade.

### 5.4.3 Execução dos *scripts*

Com os arquivos de simulação configurados, o passo seguinte é gerar e executar os *scripts* representativos dos cenários de tempo fixo e das implementações escolhidas.

Esse processo foi extremamente simplificado devido ao trabalho realizado na subseção 5.4.2 de facilitar o acesso às informações necessárias em cada uma das implementações. Dessa forma, a composição dos *scripts* se torna algo simples, tendo os principais elementos como:

- A definição, no início do arquivo e de forma manual, da composição de cada uma das

junções, ou seja, seus movimentos (que no caso da simulação se apresentam pelos conectores) e as fases associadas a cada movimento;

- O cálculo, a cada intervalo de tempo  $t$  da simulação, da fase a ser escolhida para cada uma das junções baseada no algoritmo da implementação sendo analisada e dos dados coletados dos conectores via interface COM;
- A definição dos atributos semafóricos de cada uma das fases de forma a liberar apenas a selecionada;
- O ajuste, via interface COM, dos volumes de entrada de veículos de forma a zerar a entrada de novos automóveis na rede depois de uma hora de simulação;
- O fim da simulação após um tempo simulado de uma hora e dez minutos.

Esse fluxo foi seguido, de forma geral, para todas as implementações, baseado nas informações que cada uma requiriria em seus algoritmos conforme elucidado anteriormente nesse documento. A variação se deu pela execução no caso de tempo fixo, que apenas aplica os dois últimos itens apresentados acima.

Vale ressaltar que, durante os testes realizados para o algoritmo original, em uma primeira rodada as simulações foram realizadas aplicando  $\xi_i(p, \mathcal{L}_a, \mathcal{L}_b, Z_i(t))$  como sendo a taxa de locomoção durante a saturação, de forma a simplificar a execução e representar um modelo mais simples de ser executado no mundo real. No entanto, observou-se que para a rede composta por apenas uma junção não havia estabilização com esse método, o que diferiu do cenário da Avenida Paulista. Isso ocorreu pois, no caso do cruzamento simples, haviam movimentos de saída de um link associados a fases diferentes, de forma que uma das fases era priorizada em relação a outra, ocasionando, por exemplo, uma fileira de carros desejando virar a direita esperando indefinitivamente enquanto a fase que permitia seguir em frente e virar a esquerda era sempre a escolhida. Para resolver esse problema foram feitos alguns testes com as informações disponíveis nos outros algoritmos de forma a poder determinar a taxa de locomoção de forma dinâmica. Ao final, uma opção encontrada foi utilizar os dados da implementação 5.2.1, considerando faixas dedicadas virtuais de forma a modificar a taxa de locomoção caso um movimento tivesse poucos veículos em comparação aos demais.

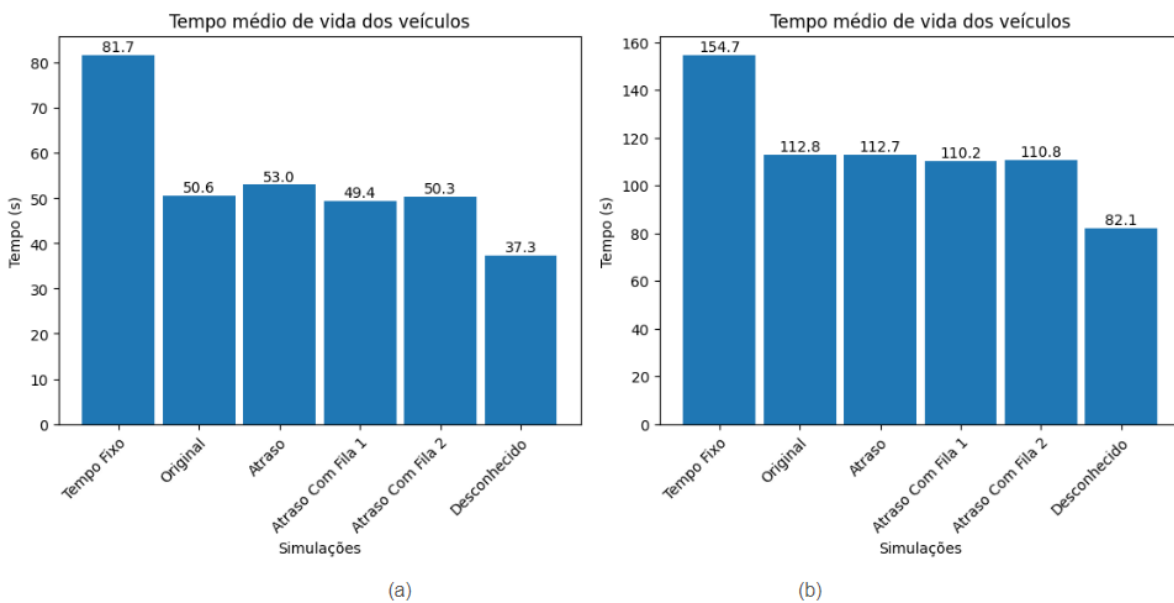
Com as considerações anteriores levantadas nessa subseção, cada um dos arquivos foi executado gerando os dados brutos para permitir a análise do desempenho em cada um dos cenários levantados. Outros *scripts* foram então produzidos para levantar as informações tratadas e gerar gráficos que serão apresentados na próxima seção.

#### 5.4.4 Apresentação dos resultados das simulações

Nesta subseção serão apresentados os resultados gerados a partir dos *scripts* descritos anteriormente. Esses consistem nas métricas definidas na seção 2.3.

O primeiro deles é o tempo médio de deslocamento, apresentado na figura 12. Conforme pode ser observado em ambos os casos, Av. Paulista e junção simples, todas as implementações se demonstraram eficazes em diminuir esse parâmetro quando comparados com o modelo de tempo fixo. A menor diminuição ocorreu no caso da Avenida Paulista com o uso do algoritmo original, com os veículos saindo da rede 27% mais rápido. Observa-se que a implementação baseada em taxas de roteamento desconhecida apresentou um tempo significativamente menor em relação as demais variações do *Backpressure*, provavelmente devendo-se ao fato de possuir um tempo de fase menor que os demais, fazendo o controle reagir mais rapidamente as variações de trânsito.

Figura 12 – Tempo médio de deslocamento dos veículos

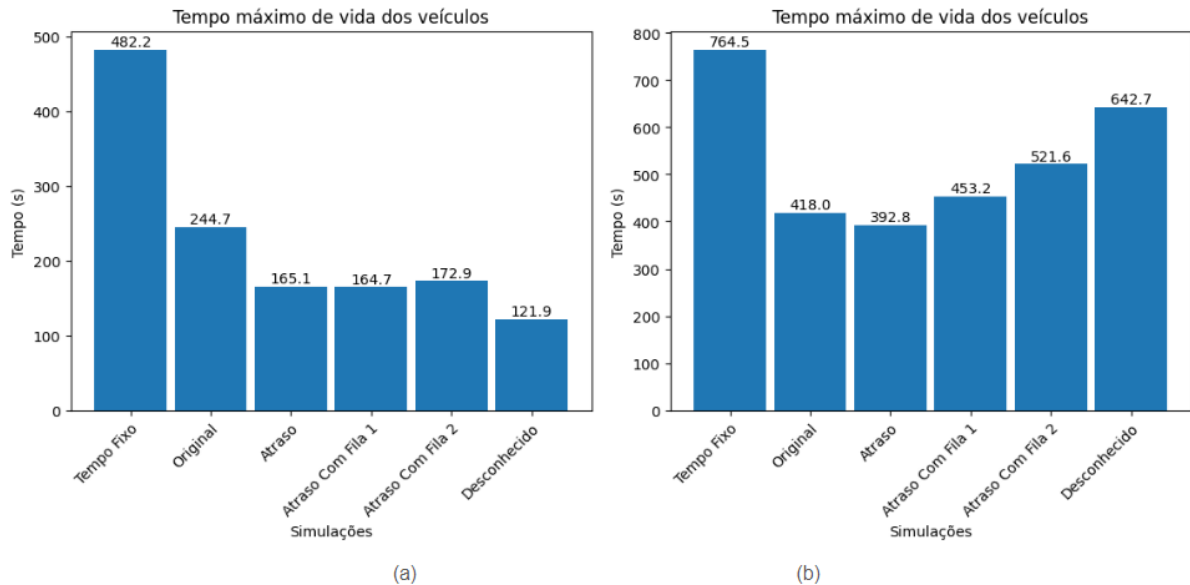


Fonte: Autoria própria.

O segundo aspecto a ser analisado está relacionado ao tempo máximo de deslocamento, apresentado na figura 13, e auxilia mais a realizar a uma comparação entre as implementações de *Backpressure* em si, já que consiste em avaliar a equidade na escolha de fases. Conforme pode ser observado, os dados apresentados para as duas redes diferem muito entre si. Isso pode ocorrer uma vez que há mais variáveis no caso da Paulista, com mais cruzamentos e fluxos que podem afetar de maneira complexa o roteamento e o tempo de espera de um veículo, há uma suscetibilidade maior de esse dado não trazer um aspecto comparativo relevante, já que se torna muito aleatório. Dessa forma, considerou-se que a informação do cruzamento simples é a que tem um peso maior para avaliar a equidade, e

nesse caso a informação relevante que se destaca desse gráfico é o baixo desempenho do algoritmo original em comparação com as demais implementações.

Figura 13 – Tempo máximo de deslocamento dos veículos

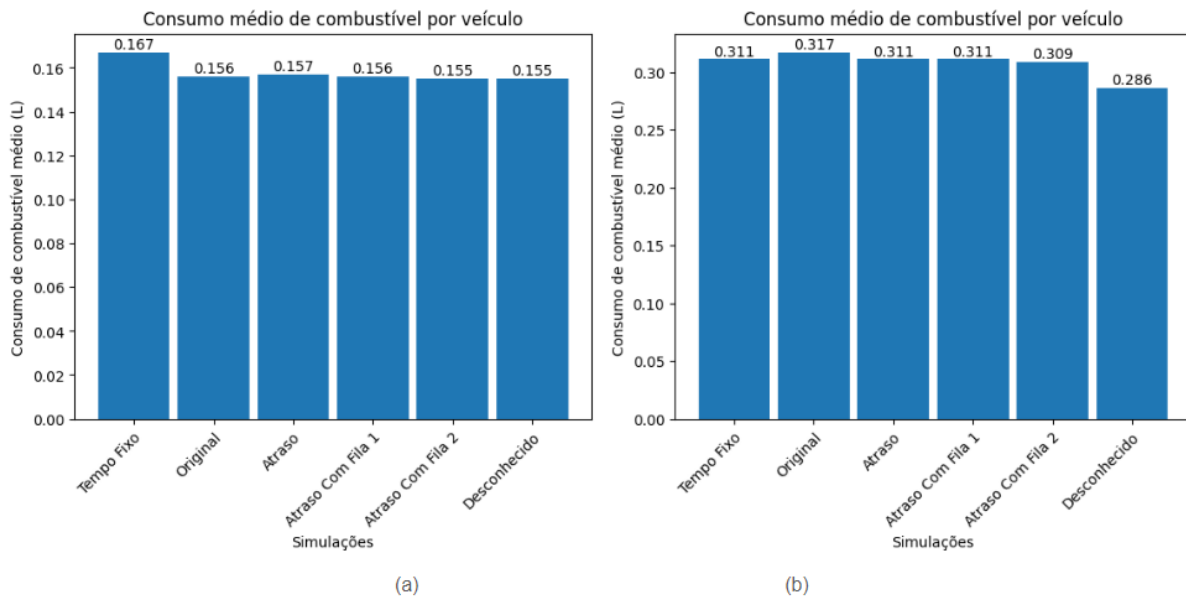


Fonte: Autoria própria.

A última análise se deu em cima do consumo de combustíveis dos veículos, que é apresentado na figura 14. Nesse caso não observou-se uma diferença realmente significativa em relação as implementações do *Backpressure* e o caso de tempo fixo, havendo uma variação menor que 10% para todas as situações. Dessa forma considera-se que não é possível inferir que os algoritmos trouxeram algum benefício ou prejuízo ao gasto de combustível, apenas que houve manutenção do status anterior. Essa tendência talvez possa ser explicada pelo fato de que nas aplicações do *Backpressure* há uma predisposição de economia de combustível pelo veículo permanecer menos tempo na rede, mas que pode ser minimizada pela característica dos tempos de fase serem menores que no tempo fixo, ocasionando dos automóveis acelerarem e desacelerarem mais, aumentando a tendência de consumo por outro lado.

Considerando esses resultados, todas as implementações apresentam vantagens em relação ao algoritmo de tempo fixo, sendo, em um primeiro momento, quaisquer uma delas possíveis de serem executadas. Há, no entanto, um maior incentivo para o uso da variante com taxas de roteamento desconhecidas se consideramos o tempo de deslocamento e há um estímulo menor em utilizar o algoritmo original pela questão da equidade.

Figura 14 – Média do consumo de combustível



Fonte: Autoria própria.

## 6 Considerações Finais

### 6.1 Avaliação dos Resultados

Pelos resultados obtidos na subseção 5.4.4 e as características levantadas na seção 5.2, são feitas algumas conclusões:

- Todas as implementações simuladas têm condições de serem executadas no mundo real, por trazerem benefícios em relação ao cenário de tempo fixo.
- A melhor implementação em relação à tempo de deslocamento é a de taxas de roteamento desconhecidas. No entanto não definiu-se que ela seria a melhor em todos os cenários por apresentar um tempo de fase muito curto, no qual os motoristas podem não se adaptar a depender das diferentes condições do local implementado.
- Os casos em que se utilizou-se o conjunto de *Backpressure* com uso de atraso mais fila não tiveram variações consideráveis em relação aos de uso de apenas fila ou atraso. Como esse cenário exigiria a combinação do sensoriamento dos dois casos, sendo mais caro e complexo, não é recomendado a aplicação do mesmo.

Com essa avaliação, considera-se que os requisitos especificados no capítulo 4 foram cumpridos, pois obteve-se dados de 4 implementações diferentes e foi possível compará-las a partir de informações extraídas de um *software* específico para simulação de fluxo de tráfego. Cabe observar que houveram mudanças ocasionadas pela falta de informações de *hardware* elucidada na seção 5.1, mas isso não impediu o trabalho como um todo apesar das limitações impostas à análise proposta.

### 6.2 Contribuições do Projeto de Formatura

As principais contribuições desse projeto residem na análise das implementações do algoritmo de *Backpressure* dentro do cenário brasileiro, incluindo o levantamento dos diferentes métodos, a atribuição das tecnologias a serem utilizadas no mundo real para cada um deles, a criação dos documentos para as simulações e a avaliação dos resultados destas. Além disso, o detalhamento dado dentro deste texto para os *scripts* gerados permite que futuros trabalhos consigam desenvolver pesquisas com base nesse de forma mais rápida e simples.

### 6.3 Perspectivas de Continuidade

Como esse trabalho constituiu uma análise de viabilidade, espera-se que a partir dos resultados considerados promissores apresentados e das soluções de tecnologia levantadas, ele possa servir de base para estimular a implementação desses algoritmos no mundo real, visando melhorar a qualidade de vida dos habitantes dos centros urbanos. Cabe ressaltar que, como o fragmento da análise que correspondia as tecnologias a serem utilizadas foi prejudicado, uma atividade complementar deve ser realizada para verificar como a introdução desses elementos pode modificar o sistema, dando sequência a novas simulações, além de um comparativo de preços para estabelecer custos gerais para o projeto. Todo esse processo no entanto é simplificado pela documentação dessa obra, o que deve tornar o estudo muito mais simples caso uma empresa ou órgão governamental deseje executar estas implementações no mundo real.



# Referências

- AIMSUN-MANUAL. *Aimsun Next Users Manual-Environmental Models*. 2023. Acessado em: 30-10-2023. Disponível em: <<https://docs.aimsun.com/next/22.0.2/UsersManual/EnvironmentalModels.html>>. Citado 2 vezes nas páginas 45 e 46.
- CET. *Pesquisa de Monitoração da Mobilidade*". [S.l.], 2018. Citado na página 48.
- GREGOIRE, J. et al. Back-pressure traffic signal control with unknown routing rates. *IFAC Proceedings Volumes*, Elsevier, v. 47, n. 3, p. 11332–11337, 2014. Citado 2 vezes nas páginas 33 e 34.
- HARMS, L.; KANSEN, M. *Kim | Cycling facts - government.nl*. Netherland's Ministry of Infrastructure and Water Management, 2018. Disponível em: <<https://www.government.nl/binaries/government/documenten/reports/2018/04/01/cycling-facts-2018/Cycling%20facts%202018.pdf>>. Citado na página 13.
- HUNT, P. B. et al. Scoot-a traffic responsive method of coordinating signals. In: . [S.l.: s.n.], 1981. Citado na página 14.
- LOWRIE, P. The sydney coordinated adaptive traffic system - principles, methodology, algorithms. In: . [S.l.: s.n.], 1982. Citado na página 14.
- MAIPRADIT, A. et al. An adaptive traffic signal control scheme based on back-pressure with global information. *Journal of Information Processing*, v. 29, p. 124–131, 01 2021. Citado 2 vezes nas páginas 14 e 20.
- SCHRANK, D. et al. *2021 urban mobility report*. 2021. Disponível em: <<https://trid.trb.org/view/1862637>>. Citado na página 15.
- WONGPIROMSARN, T. et al. Distributed traffic signal control for maximum network throughput. In: *2012 15th International IEEE Conference on Intelligent Transportation Systems*. [S.l.: s.n.], 2012. p. 588–595. Citado 8 vezes nas páginas 14, 17, 18, 19, 20, 32, 33 e 35.
- WU, J. et al. Delay-based traffic signal control for throughput optimality and fairness at an isolated intersection. *IEEE Transactions on Vehicular Technology*, v. 67, n. 2, p. 896–909, 2018. Citado 2 vezes nas páginas 35 e 36.
- ZAIDI, A. A.; KULCSÁR, B.; WYMEERSCH, H. Back-pressure traffic signal control with fixed and adaptive routing for urban vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, v. 17, n. 8, p. 2134–2143, 2016. Citado na página 37.