



**Universidade Federal de Santa Catarina
Centro Tecnológico
Departamento de Engenharia Civil**

Fabiana Weimann Araujo

**Experimentação com simulação de tráfego mesoscópica utilizando o *software* SUMO
aplicada à Região Metropolitana da Grande Florianópolis**

22 de março de 2022

Fabiana Weimann Araujo

Experimentação com simulação de tráfego mesoscópica utilizando o *software* SUMO aplicada à Região Metropolitana da Grande Florianópolis

Trabalho de Conclusão de Curso apresentado pela acadêmica Fabiana Weimann Araujo à banca examinadora do Curso de Graduação em Engenharia Civil da Universidade Federal de Santa Catarina como requisito parcial para obtenção do título de Engenheiro Civil.

Professor orientador: Alexandre Hering Coelho, Dr.

22 de março de 2022

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Araujo, Fabiana Weimann

Experimentação com simulação de tráfego mesoscópica
utilizando o software SUMO aplicada à Região Metropolitana
da Grande Florianópolis / Fabiana Weimann Araujo ;
orientador, Alexandre Hering Coelho, 2022.

112 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, , Graduação em ,
Florianópolis, 2022.

Inclui referências.

1. . 2. Simulação de tráfego. 3. Mesoscópica. 4. SUMO. I.
Coelho, Alexandre Hering. II. Universidade Federal de
Santa Catarina. Graduação em . III. Título.

Fabiana Weimann Araujo

**Experimentação com simulação de tráfego mesoscópica utilizando o *software* SUMO
aplicada à Região Metropolitana da Grande Florianópolis**

Este Trabalho de Conclusão de Curso foi julgado adequado para a obtenção do Título de Engenheira Civil e aprovado em sua forma final pelo Departamento de Engenharia Civil da Universidade Federal de Santa Catarina.

Florianópolis, 22 de março de 2022

Prof. Lisiane Ramos da Silva, Dra.
Coordenadora do Curso

Banca examinadora:

Prof. Alexandre Hering Coelho, Dr.
Orientador

Prof. Luciana Rohde, Dra.
Membro

Eng. Fernanda Beckhauser Mallon
Membro

Agradecimentos

À minha família, em especial meus pais Lucyane e Ricardo, pelo carinho e por todo o suporte durante a minha vida.

Às minhas amigas Lídia e Nayara pelo companheirismo, amizade e a cumplicidade.

Ao Leonardo por todo apoio e carinho.

Aos amigos que fiz durante a graduação, em especial Francielle e Valéria, por tornar toda essa trajetória mais alegre.

Ao meu orientador Alexandre por todo o suporte e orientação durante a execução do trabalho.

RESUMO

Simulação de transportes é uma poderosa ferramenta de análise de tráfego que permite avaliar uma determinada área de estudo. Ela tem como objetivo a otimização de operações, modelagem de redes e a avaliação de resultados para cenários alternativos. A elaboração de modelos de simulação envolve o uso de dados de infraestrutura e de demanda, recursos computacionais e treinamento. Este trabalho visa o desenvolvimento e avaliação de modelos de simulação de tráfego em nível mesoscópico da Região Metropolitana da Grande Florianópolis utilizando o software SUMO e dados abertos. Os dados de demanda são obtidos por meio do Plano de Mobilidade Urbana Sustentável da Grande Florianópolis (PLAMUS) e os dados de infraestrutura do *Openstreetmap* (OSM).

Os modelos de simulação elaborados no trabalho contêm o processo de calibração utilizando diferentes métodos de alocação de tráfego, troca de parâmetros e edições na infraestrutura. A avaliação da simulação é feita apenas de forma visual comparando a imagens geradas no *Google Maps*, não são utilizados dados numéricos.

No trabalho é possível notar a sensibilidade da simulação a diferentes parâmetros. Entre os modelos elaborados, o modelo de simulação com semáforos ativados e com método de alocação incremental com 5 iterações apresentou resultados satisfatórios, não há trancamento da rede e todos os veículos finalizam suas viagens em tempo representativo. Os dados abertos utilizados no trabalho foram suficientes a caráter de experimentação.

Palavras-chave: simulação de tráfego; mesoscópico; SUMO

ABSTRACT

Traffic simulation is a powerful traffic analysis tool that allows the evaluation of a given study area. The simulation aims at optimizing operations, modeling networks and predicting results for alternative scenarios. The elaboration of simulation models requires infrastructure and demand data, equipment and training. This work aims at the development and evaluation of traffic simulation models at a mesoscopic level in the metropolitan region of Florianópolis using the SUMO software and open data. The demand data is obtained through the Plano de Mobilidade Urbana Sustentável da Grande Florianópolis (PLAMUS) and infrastructure data from the Openstreetmap (OSM).

The simulation models developed in the work contain the calibration process using different methods of traffic allocation, parameter change and infrastructure edits. The evaluation of the simulation is done only in a visual way comparing to images generated in Google Maps, numerical data are not used.

In the work it is possible to notice the sensitivity of the simulation to different parameters. Among the models developed, the simulation model with traffic lights activated and with an incremental allocation method with 5 iterations showed satisfactory results, there is no network locking and all vehicles finish their trips in representative time. The open data used in the work were sufficient for experimentation.

Key-words: traffic simulation; mesoscopic; SUMO

Lista de Figuras

| | | |
|----|--|----|
| 1 | Evolução da frota de veículos no Brasil | 1 |
| 2 | Estrutura para tomada de decisão racional com modelos | 5 |
| 3 | Representação de zona e centroide | 11 |
| 4 | Movimentos na área de estudo | 13 |
| 5 | Representação de matriz OD | 14 |
| 6 | Representação de <i>links</i> e nós | 15 |
| 7 | Relação entre os níveis de acessibilidade e mobilidade | 17 |
| 8 | Tempo de percurso como impedância em <i>links</i> para alocação de tráfego | 20 |
| 9 | Modelagem do fluxo de tráfego | 24 |
| 10 | Tempo de <i>Warm-up</i> | 25 |
| 11 | Rede de tráfego do <i>LuST</i> | 28 |
| 12 | Comparação de tráfego do modelo Lust | 31 |
| 13 | Rede de tráfego <i>TuST</i> | 33 |
| 14 | Alocação Incremental - TuST | 35 |
| 15 | Comparação de alocação de tráfego Tudo ou Nada e Incremental do modelo TuST | 36 |
| 16 | Fluxograma do método | 38 |
| 17 | Teste para confirmação da impedância do <i>duarouter</i> | 46 |
| 18 | Localização da área de estudo | 54 |
| 19 | Cidades da área de estudo | 55 |
| 20 | Zonas de tráfego da área de estudo | 56 |
| 21 | Rede de tráfego da área de estudo | 60 |
| 22 | Comparação de velocidades permitidas antes e após ajuste | 64 |
| 23 | Gráfico do método de alocação Tudo ou Nada sem teletransporte | 67 |
| 24 | <i>Edges</i> com congestionamento na rede, método de alocação aplicado: Tudo ou Nada | 68 |
| 25 | Gráfico do método de alocação Tudo ou Nada com teletransporte | 68 |
| 26 | Gráfico da simulação com método Incremental sem teletransporte | 70 |
| 27 | Gráfico da simulação com método de alocação Incremental com teletransporte | 72 |
| 28 | Gráfico da simulação com método de alocação Variação Incremental | 73 |
| 29 | Gráfico da simulação com o método de alocação SUE | 74 |
| 30 | Gráfico da simulação com método de alocação UE | 75 |
| 31 | Gráfico de comparação: UE, SUE, Incremental e Variação Incremental | 76 |
| 32 | Simulação no tempo de <i>warm-up</i> (960 segundos) | 77 |
| 33 | Gráfico da simulação com método de alocação Incremental com semáforos | 78 |
| 34 | Gráfico simulação Incremental com semáforos e coordenação de fases | 79 |

| | | |
|----|--|----|
| 35 | Semáforos sem sincronismo | 80 |
| 36 | Semáforos antes do agrupamento | 81 |
| 37 | Semáforos agrupados | 81 |
| 38 | Gráfico da simulação com alocação Incremental com semáforos após a correção semafórica | 82 |
| 39 | Gráfico da simulação com método Incremental com e sem semáforos | 83 |
| 40 | Comparação da velocidade da simulação com a do trânsito típico do <i>Google Maps</i> , Governador Celso Ramos (latitude: -27.388, longitude: -48.590) | 84 |
| 41 | Comparação da velocidade da simulação com a do trânsito típico do <i>Google Maps</i> , BR-101, Biguaçu (latitude:-27.506 longitude: -48.648) | 85 |
| 42 | Comparação da velocidade da simulação com a do trânsito típico do <i>Google Maps</i> , BR-101, São José (latitude: -27.548, longitude: -48.619) | 86 |
| 43 | Comparação da velocidade da simulação com a do trânsito típico do <i>Google Maps</i> , São José (latitude: -27.570, longitude:-48.607) | 87 |
| 44 | Comparação da velocidade da simulação com a do trânsito típico do <i>Google Maps</i> , Florianópolis (latitude: -27.603, longitude: -48.591) | 88 |
| 45 | Comparação da velocidade da simulação com a do trânsito típico do <i>Google Maps</i> , Florianópolis (latitude: -27.593, longitude: -48.551) | 89 |
| 46 | Comparação da velocidade da simulação com a do trânsito típico do <i>Google Maps</i> , Florianópolis (latitude: -27.579, longitude: -48.525) | 90 |
| 47 | Comparação da velocidade da simulação com a do trânsito típico do <i>Google Maps</i> , Florianópolis (latitude: -27.543, longitude: -48.503) | 91 |
| 48 | Comparação da velocidade da simulação com a do trânsito típico do <i>Google Maps</i> , Florianópolis (latitude: -27.650, longitude: -48.518) , | 92 |
| 49 | Comparação da velocidade da simulação com a do trânsito típico do <i>Google Maps</i> , bairro Trindade, Florianópolis (latitude: -27.595, longitude: -48.517) , | 93 |

Lista de Tabelas

| | | |
|----|---|----|
| 1 | Classificação das vias conforme CTB | 18 |
| 2 | Velocidades máximas segundo o CTB | 19 |
| 3 | Dados da rede de tráfego LuST | 29 |
| 4 | Dados da simulação LuST | 29 |
| 5 | Dados da rede de tráfego TuST | 32 |
| 6 | Configuração do equipamento utilizado | 39 |
| 7 | Classificação das vias para o OSM | 57 |
| 8 | Síntese de dados de infraestrutura da rede de tráfego da área de estudo | 60 |
| 9 | Distribuição de <i>edges</i> por classes | 62 |
| 10 | Correlação de classificações | 62 |
| 11 | Definição do veículo | 66 |
| 12 | Dados da simulação utilizando o método Tudo ou Nada com diferentes tempos de teletransporte | 69 |
| 13 | Dados da simulação utilizando o método Incremental sem teletransporte | 71 |
| 14 | Dados da simulação utilizando o método Incremental com teletransporte | 71 |
| 15 | Dados de veículos que recalcularam rotas com o método de alocação Variação Incremental | 73 |
| 16 | Comparação dos dados de infraestrutura dos modelos Lust, TuST e da Região Metropolitana da Grande Florianópolis | 96 |

Lista de Quadros

| | | |
|----|---|----|
| 1 | Função <i>netconvert</i> | 40 |
| 2 | Execução <i>tlsCycleAdaptation.py</i> | 41 |
| 3 | Execução <i>tlsCoordinator.py</i> | 41 |
| 4 | Definição do veículo | 42 |
| 5 | Função <i>polyconvert</i> | 43 |
| 6 | Função <i>edgesInDistricts.py</i> | 43 |
| 7 | Configuração <i>od2trips</i> | 44 |
| 8 | Estrutura do arquivo com os dados de matriz OD | 44 |
| 9 | Execução do <i>od2trips</i> | 45 |
| 10 | Configuração do <i>duarouter</i> | 47 |
| 11 | Execução do <i>duarouter</i> | 47 |
| 12 | Execução <i>route2trips.py</i> | 47 |
| 13 | Execução <i>duaiterate.py</i> | 48 |
| 14 | Configuração Incremental | 49 |
| 15 | Execução Incremental | 49 |
| 16 | Configuração SUE | 50 |
| 17 | Definição transporte público | 51 |
| 18 | Configuração de execução | 53 |
| 19 | <i>Script</i> para obter informação de quantidade de <i>edges</i> por classe de via | 61 |
| 20 | <i>Script</i> para ajuste de velocidade | 63 |

Sumário

| | | |
|----------|---|-----------|
| 1 | Introdução | 1 |
| 1.1 | Objetivos | 2 |
| 1.1.1 | Objetivo geral | 2 |
| 1.1.2 | Objetivos específicos | 3 |
| 1.2 | Limitações do trabalho | 3 |
| 2 | Revisão bibliográfica | 4 |
| 2.1 | Planejamento de transportes | 4 |
| 2.2 | Ferramentas de análise de tráfego | 6 |
| 2.3 | Simulação de tráfego | 7 |
| 2.4 | Levantamento de dados para modelagem de transportes urbanos | 9 |
| 2.4.1 | Delimitação da área de estudo e zoneamento | 9 |
| 2.4.2 | Dados de demanda | 11 |
| 2.4.3 | Dados de infraestrutura | 14 |
| 2.4.4 | Classificação funcional das vias urbanas | 16 |
| 2.5 | Alocação de tráfego | 19 |
| 2.5.1 | Tudo ou Nada | 21 |
| 2.5.2 | Incremental | 21 |
| 2.5.3 | Equilíbrio do Usuário | 22 |
| 2.5.4 | Equilíbrio Estocástico do Usuário | 22 |
| 2.6 | Calibração e validação de modelos de simulação de tráfego | 23 |
| 2.7 | <i>Warm-up</i> | 24 |
| 2.8 | Semáforos | 25 |
| 2.8.1 | Controle semafórico | 25 |
| 2.8.2 | Tempo de ciclo | 27 |
| 2.9 | Exemplos de aplicação do SUMO em grandes áreas urbanas | 27 |
| 2.9.1 | <i>Luxembourg SUMO Traffic (LuST) Scenario</i> | 27 |
| 2.9.2 | <i>Turin SUMO Traffic</i> | 32 |
| 3 | Método | 38 |
| 3.1 | Recursos de <i>softwares</i> utilizados | 38 |
| 3.2 | Modelagem da infraestrutura viária | 39 |
| 3.2.1 | Rede de tráfego | 39 |
| 3.2.2 | Semáforos | 41 |

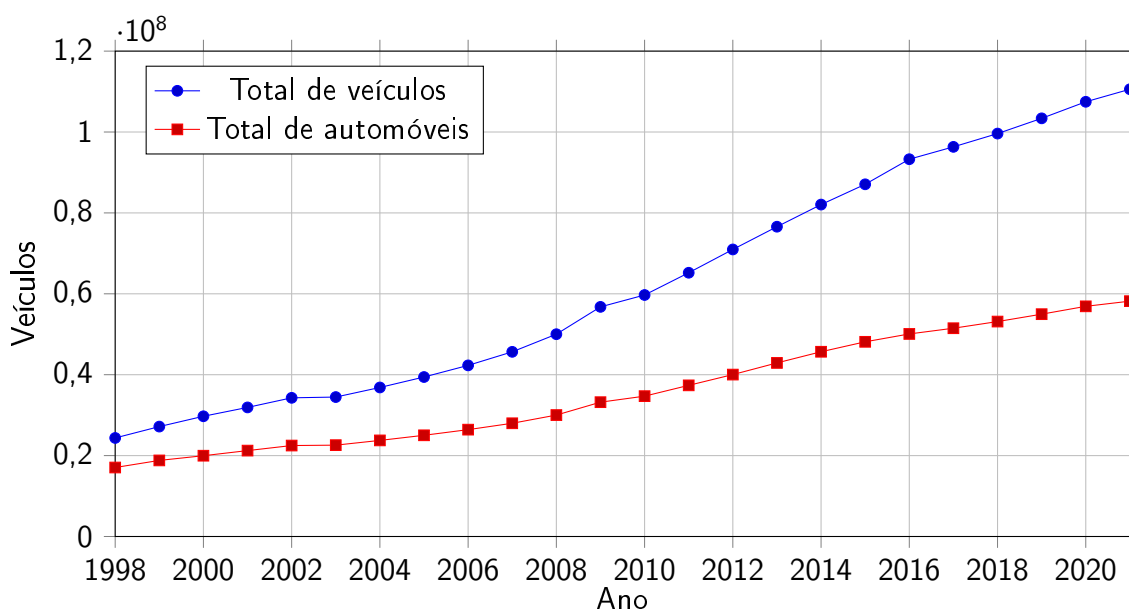
| | | |
|----------|--|-----------|
| 3.2.3 | Veículos | 41 |
| 3.3 | Modelagem da demanda | 42 |
| 3.3.1 | Definição das zonas de tráfego | 42 |
| 3.3.2 | Definição da demanda | 43 |
| 3.4 | Alocação de tráfego | 45 |
| 3.4.1 | Tudo ou Nada | 45 |
| 3.4.2 | Equilíbrio do Usuário | 47 |
| 3.4.3 | Incremental | 48 |
| 3.4.4 | Variação do método Incremental | 49 |
| 3.4.5 | Equilíbrio Estocástico do Usuário | 50 |
| 3.5 | Modelagem da oferta de transporte público coletivo | 51 |
| 3.6 | Execução da simulação | 51 |
| 4 | Área de estudo | 54 |
| 4.1 | Zonas de tráfego e matriz OD | 55 |
| 4.2 | Infraestrutura | 56 |
| 5 | Aplicações na área de estudo | 59 |
| 5.1 | Modelagem da infraestrutura | 59 |
| 5.2 | Transporte público | 64 |
| 5.3 | Definição de veículo | 65 |
| 5.4 | Demanda | 66 |
| 5.5 | Escolha do método de alocação | 66 |
| 5.5.1 | Tudo ou nada | 66 |
| 5.5.2 | Incremental | 69 |
| 5.5.3 | Variação do método Incremental | 72 |
| 5.5.4 | SUE | 73 |
| 5.5.5 | UE | 74 |
| 5.5.6 | Comparação dos modelos | 75 |
| 5.6 | Tempo de <i>warm-up</i> | 76 |
| 5.7 | Simulação com semáforos ativados | 77 |
| 5.7.1 | Modificações semaforicas | 79 |
| 5.8 | Avaliação dos resultados | 83 |
| 6 | Conclusões | 95 |
| 6.1 | Recomendações para trabalhos futuros | 97 |

1 Introdução

O Brasil, nas últimas décadas, sofreu intensa urbanização. Segundo Santos (2013), em 1940, o nível de urbanização era de 26,35%, em 1980 alcançou 68,86%, os dados mais recentes fornecidos pela PNAD (2015) indicam que 84,72% dos brasileiros vivem em áreas urbanas. Ainda de acordo com Santos (2013), de 1940 a 1980 a população total do Brasil triplicou e a população urbana se multiplicou por sete vezes e meia.

Paralelo ao crescimento urbano, acontece o aumento da frota de automóveis no país. A Figura 1 apresenta o gráfico do número total de veículos e de automóveis no Brasil. No período de 2011 a 2021 a frota total de veículos teve aumento de 70% e a frota de automóvel aumento de 56%, sendo o principal tipo de veículo utilizado no Brasil (DENATRAN, 2021).

Figura 1: Evolução da frota de veículos no Brasil



Fonte: Elaborado com dados do DENATRAN (2021)

Para conciliar o aumento da população concentrada nas cidades e o uso do transporte individual exigindo da infraestrutura da rede viária é necessário planejamento de transportes. De acordo com Bruton (1979), realizar propostas para investimentos em construções de nova infraestrutura e projetos de melhorias nas existentes, melhorarão as condições operacionais dos fluxos nas redes. Os projetos para melhoria de tráfego exigem estudos para identificar os problemas e encontrar soluções eficientes e viáveis.

Para realizar projetos de melhorias é necessário a avaliação do tráfego que pode ser feita com o auxílio de diferentes ferramentas. Cada ferramenta é apropriada para diferentes objetivos e necessita

de dados específicos. Conforme colocam Ortúzar e Willumsen (2011), também são contabilizados na escolha da ferramenta os recursos disponíveis para a análise, a habilidade e o treinamento da equipe que irá realizar o estudo e a acurácia necessária dos resultados.

As ferramentas de simulação são comumente utilizadas em tarefas de avaliação, otimização de operações de transporte, modelagem de redes existentes e previsão de resultados prováveis para projetos de alternativas. Os simuladores de tráfego possuem a capacidade de representar a realidade de forma simplificada através de modelos computacionais. Os gráficos e as animações fornecidas como forma de resultado conseguem passar para o público e/ou tomadores de decisão de forma fácil e ilustrativa os resultados de cenários alternativos.

No entanto, para gerar simulações de tráfego condizentes à realidade, é necessário empenhar tempo de modelagem, recursos computacionais e amplo número de dados de entrada. A base de dados necessita de constante manutenção para que os resultados sejam representativos.

Atualmente diversos *softwares* de simulação estão disponíveis no mercado. Uma opção de *software* de simulação de tráfego é o *Simulation of Urban MObility* (SUMO)¹. O desenvolvimento do SUMO começou no ano 2000, com a primeira implementação sendo iniciada no ano de 2001. Conforme colocam Barceló et al. (2010), no início, o SUMO foi desenvolvido em colaboração entre o *Center for Applied Informatics Cologne* (ZAIK) e o *Institute of Transportation Systems* (ITS), no *German Aerospace Center* (DLR). Desde 2004, o SUMO continua apenas no DLR, com contribuição de organizações externas.

O SUMO é um *software* de código aberto permite a modelagem de sistemas de tráfego intermodal, incluindo transporte privado, transporte público e de pedestres. O *software* SUMO também traz uma série de ferramentas para auxiliar na modelagem dos dados que são utilizados por ele, para infraestrutura, para demanda e para transporte coletivo. O SUMO conta com ferramentas que automatizam e facilitam a criação, execução e avaliação de simulações de tráfego. O *software*, inicialmente apenas microssimulador, foi estendido para modelo mesoscópico, sem alterações nas interfaces (BARCELÓ et al., 2010). O SUMO é utilizado para simular desde conjunto de interseções, como também para grandes áreas de estudo como cidades e bairros, pois possui os requisitos para operar com grande quantidade de dados.

1.1 Objetivos

1.1.1 Objetivo geral

O objetivo geral deste Trabalho de Conclusão de Curso é o desenvolvimento e avaliação de modelos de simulação de tráfego da Região Metropolitana da Grande Florianópolis utilizando o *software* SUMO e dados abertos.

¹<<https://www.eclipse.org/sumo/>> Acesso em 10/12/2021

1.1.2 Objetivos específicos

Os objetivos específicos são colocados a seguir.

- a) Avaliar a disponibilidade de dados abertos para apoiar a modelagem da infraestrutura e da demanda da área de estudo.
- b) Avaliar o grau de realismo possível de ser alcançado nos modelos pela utilização dos dados abertos disponíveis.
- c) Avaliar os recursos disponíveis, o grau de desenvolvimento e a facilidade de uso do *software* SUMO.
- d) Explorar recursos computacionais disponíveis capazes de apoiar o trabalho de modelagem no SUMO.

1.2 Limitações do trabalho

Este Trabalho de Conclusão de Curso conta com as seguintes limitações:

- 1. a validação da simulação é realizada apenas de forma visual.
- 2. as soluções aos problemas encontrados são propostas, em geral, a nível sistemático, ou seja, a realização de qualquer processo de edição manual nos dados é evitada.

2 Revisão bibliográfica

A revisão bibliográfica é composta por definições na área de planejamento de transportes e em simulação de tráfego.

2.1 Planejamento de transportes

As ferramentas de análise de tráfego são utilizadas como apoio para o planejamento de transportes. É necessário entender a lógica e as fases do processo de planejamento para decidir de forma adequada a respeito da ferramenta. O planejamento de transportes é estruturado em etapas e fases para organizar e preparar os passos a serem cumpridos. Para planejar é necessário identificar o problema, definir objetivos, prazos e recursos.

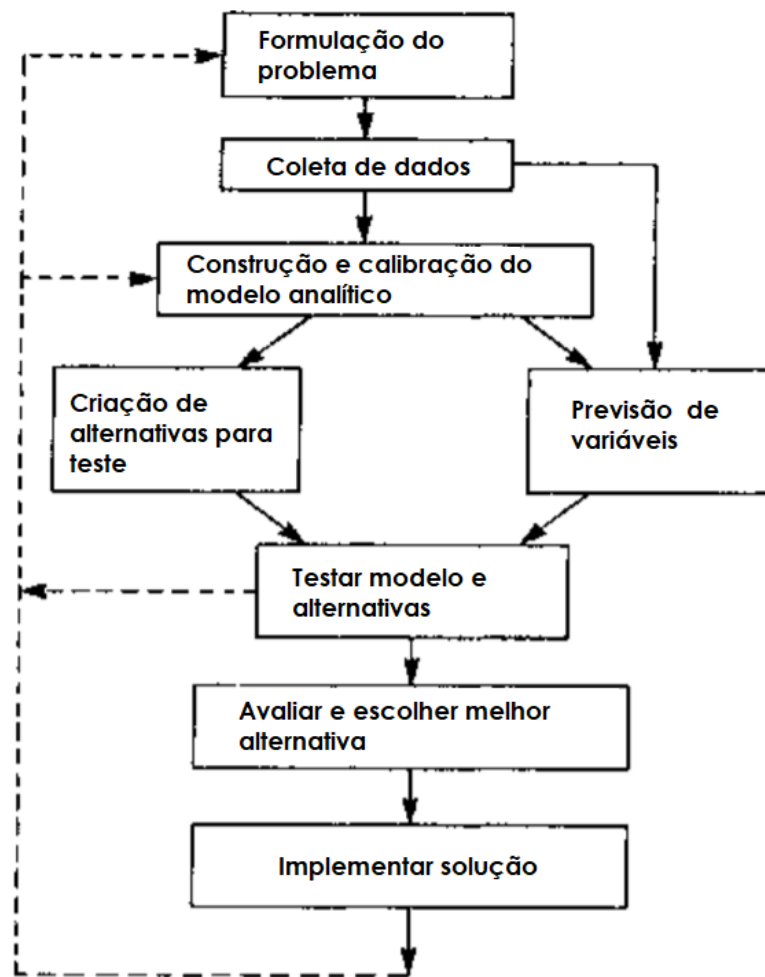
De acordo com Bruton (1979), o planejamento de transportes apresenta uma estrutura dividida em quatro etapas principais colocadas a seguir.

1. Etapa para estimativa da demanda futura por viagens e para a análise das falhas nos sistemas de transportes existentes, com objetivo de adequá-los à demanda futura.
2. Etapa de elaboração de uma série de planos alternativos de transporte, estabelecido conforme os objetivos da comunidade e a satisfação das viagens estimadas.
3. Etapa de avaliação dos planos alternativos em termos econômicos e sociais, a fim de obter uma solução ótima.
4. Etapa de financiamento e implementação do plano alternativo escolhido.

Segundo Campos (2013), o plano pode ser de longo, médio ou curto prazo. Sua duração depende dos recursos disponíveis e dos objetivos definidos. O processo deve ser dinâmico: é indicado voltar e verificar quais os impactos que as soluções propostas podem trazer à demanda com a implantação das mesmas.

A Figura 2 apresenta fluxograma que mostra as oito fases do processo de planejamento segundo Ortúzar e Willumsen (2011).

Figura 2: Estrutura para tomada de decisão racional com modelos



Fonte: Adaptado de Ortúzar e Willumsen (2011)

Este trabalho se encaixa na etapa de coleta de dados e de construção e calibração do modelo analítico. Segundo Ortúzar e Willumsen (2011) estas duas fases são assim definidas:

1. coleta de dados sobre o estado atual do sistema de interesse para apoiar o desenvolvimento do modelo analítico.
2. construção de um modelo analítico do sistema de interesse, através do uso de ferramentas técnicas, incluindo procedimentos de análise de demanda e de verificação de desempenho do sistema, de uma perspectiva tática e estratégica. A construção de um modelo analítico envolve especificá-lo, estimar ou calibrar seus parâmetros e validar seu desempenho.

2.2 Ferramentas de análise de tráfego

Diversas ferramentas de análise de tráfego estão disponíveis para auxiliar a avaliação. Segundo FHWA (2004a) as ferramentas de análise são classificadas da seguinte forma.

- a) Ferramentas de planejamento a nível de esboço.
- b) Modelos de demanda de viagens.
- c) Ferramentas analíticas/determinísticas (baseadas em HCM).
- d) Ferramentas de otimização de sinais de tráfego.
- e) Modelos de simulação macroscópica.
- f) Modelos de simulação mesoscópica.
- g) Modelos de simulação microscópica.

Para realizar análise de tráfego é necessário escolher a ferramenta mais adequada ao objetivo proposto. Segundo FHWA (2004a) os critérios para a escolha da ferramenta de análise de tráfego são:

1. capacidade de analisar o escopo geográfico ou área de estudo para análise, incluindo interseções isoladas, rodovias, corredores ou redes viárias de tráfego;
2. capacidade de modelagem de diferentes tipos de vias e conexões;
3. possibilidade de analisar diferentes modos de viagem;
4. capacidade de analisar estratégias e ferramentas de gerenciamento de tráfego e aplicações como rampas de acesso controlado, coordenação semafórica, gerenciamento de acidentes;
5. capacidade de estimar as respostas dos viajantes às estratégias de gerenciamento de tráfego, incluindo desvio de rota, escolha de horário de partida, mudança de modo e escolha de destino;
6. capacidade de produzir diretamente medidas de desempenho, como segurança (acidentes, fatalidades), eficiência (rendimento e volumes), mobilidade (tempo de viagem, velocidade, veículo-horas de viagem), produtividade (economia) e medidas ambientais (emissões, consumo de combustível, ruído);

7. custo-benefício da ferramenta de acordo com a tarefa. Os parâmetros que influenciam o custo-benefício incluem custo para utilização da ferramenta, nível de esforço necessário, facilidade de uso, requisitos de *hardware*, requisitos de dados e animação.

Com base nos requisitos necessários e na oferta de ferramenta deve ser escolhida a melhor a ser empregada na área de estudo. Com os critérios definidos por FHWA (2004a), são indicados três passos para a escolha adequada da ferramenta:

1. identificar o contexto analítico do projeto;
2. filtrar pelos critérios de 1 a 6 para limitar à ferramenta apropriada em uma ou duas opções;
3. por fim, o critério número 7, de custo-benefício deve ser usado para selecionar a ferramenta final baseado em parâmetros fora do âmbito da análise técnica.

2.3 Simulação de tráfego

FHWA (2004a) coloca que as ferramentas de simulação são eficientes em avaliar a evolução dinâmica do congestionamento em sistemas de transporte. A simulação ao dividir o período em fatias de tempo pode avaliar a construção, dissipação e a duração do congestionamento. Os simuladores de tráfego podem modelar a variabilidade de características de veículos e motoristas.

A simulação pode ser de nível microscópico, mesoscópico ou macroscópico. O nível de simulação é definido conforme o tratamento da demanda na simulação.

Macrossimulação

A simulação macroscópica, de acordo com FHWA (2004a), tem como base a relação determinística de volume, velocidade e densidade do fluxo de tráfego. Ela ocorre seção por seção e não pelo rastreamento individual de veículos.

Hensher (2004) coloca que a simulação macroscópica possui maior unidade de análise, tipicamente agrega valores de zonas geográficas. Esse modelo é bastante usado pois necessita de baixo poder computacional. O autor ainda coloca que o modelo precisa de menos quantidade de dados e que eles são mais facilmente encontrados em censos.

Microsimulação

Spiegelman, Park e Rilett (2011) colocam modelos de microsimulação de tráfego buscam representar a interação do sistema físico (estradas, cruzamentos, controle de tráfego, etc.) e os usuários (a demanda: rotas, características do motorista, etc). Os autores ainda colocam que estes modelos são referidos como micro por operarem a um nível de indivíduo (veículos, pessoas) e são

referidos como simulação porque procuram modelar os processos internos do sistema (decisões dos motoristas, características dos veículos, semaforização, etc.) e não simplesmente a saída do sistema.

De acordo com FHWA (2004a), modelos microscópicos simulam o movimento individual dos veículos baseado em teorias de perseguição de carros e na troca de faixas. Geralmente os veículos entram na rede de transporte utilizando distribuição estatística de chegadas e são rastreados na rede em curtos intervalos de tempo. O tempo de processamento no computador e os requisitos de armazenamento para modelos microscópicos são significativos, geralmente limitando o tamanho da rede e o número de execuções de simulação que podem ser concluídas.

O fluxo de tráfego, em nível microscópico, é modelado por meio de algoritmos específicos que descrevem a interação entre os veículos. Ortúzar e Willumsen (2011) colocam que o modelo microscópico é baseado em perseguição (*car-following*), mudanças de faixas (*lane changing*), aceitação de brechas (*gap acceptance*) e fusões dos algoritmos. Em síntese, para Eissfeldt (2004), os algoritmos tem as seguintes finalidades:

1. perseguição (*car-following*): descrever como um veículo em posição de seguidor faz a manutenção do distanciamento do veículo posterior *headway*;
2. mudanças de faixas (*lane changing*): descrever as condições nas quais os veículos realizam as mudanças de faixa. Essa situação ocorre na realização de ultrapassagens ou na preparação para alguma conversão em interseção;
3. aceitação de brechas (*gap acceptance*): descrever como é o comportamento dos veículos em relação à brecha aceitável para mudanças de faixas ou cruzamentos de correntes de tráfego.

A necessidade de grande quantidade de dado e os requisitos computacionais geralmente limitam o tamanho da área de estudo. No entanto, há estudos de simulação de tráfego em nível microscópico de grandes áreas (ver Seção 2.9).

Mesossimulação

Segundo FHWA (2004a), modelos de simulação mesoscópicos combinam propriedades dos modelos de microssimulação e de macrossimulação. A unidade de tráfego são os pelotões de veículo. O movimento segue a abordagem da macrossimulação, ele é governado pela velocidade média de viagem do *link*. O modelo mesoscópico de previsão de viagens ocorre em um nível agregado e não considera relações dinâmicas de velocidade / volume.

Segundo Hensher (2004) o modelo mesoscópico usa dados agregados em alguns aspectos do modelo e onde for viável dados mais detalhados. Eissfeldt (2004) coloca que o modelo mesoscópico aborda que carros individuais movem de acordo com a média densidade das vias ou em modelos de

fila (*queueing model*). O autor coloca que o modelo mesoscópico mais simples é modelar cada *link* da rede como uma fila com certa capacidade. A ideia base do modelo é que o tempo de viagem em um *link* é a soma do tempo necessário para viajar em seu comprimento e o tempo gasto esperando em fila. As características do *link* podem ser descritas por comprimento, capacidade e o número de veículos que cabem quando há congestionamento.

2.4 Levantamento de dados para modelagem de transportes urbanos

De acordo com o plano a ser desenvolvido, os planejadores devem definir, primeiramente, os dados que devem ser coletados. Para realizar o planejamento, são necessários dados de infraestrutura e de demanda que envolvem a área de estudo delimitada.

De acordo com Bruton (1979), as diferentes pesquisas necessárias para coletar os dados básicos consomem tempo e exigem expressivos recursos humanos e financeiros. Portanto, a fase de coleta de dados requer programação cuidadosa, sendo prática corrente analisar cada elemento no processo e preparar cronograma para garantir que os dados sejam obtidos de forma rápida e eficiente.

2.4.1 Delimitação da área de estudo e zoneamento

A área de estudo deve ser delimitada para que os dados necessários e condizentes com o local sejam coletados.

Segundo Bruton (1979), os limites geralmente escolhidos para definir a área de estudo se aproximam da vertente de viagens pendulares associados com o centro. Esse limite é chamado de cordão externo (*cordon line*). O cordão externo deve englobar todos os movimentos importantes da área de estudo (especialmente no caso urbano, as viagens casa-trabalho) e deve cruzar as principais vias e corredores, quanto menor o número melhor (CAMPOS, 2013).

A linha que passa cortando a área de estudo é chamada de (*screen line*). Pignataro et al. (1973) define *screen line* como uma linha que divide a área de estudo com o objetivo de verificar a precisão dos dados de pesquisa. Segundo Campos (2013) a *screen line* pode ser definida por uma via urbana ou um acidente geográfico que apresenta interseção com um número reduzido de vias que têm diariamente uma grande quantidade de viagens de interligação entre sub-regiões.

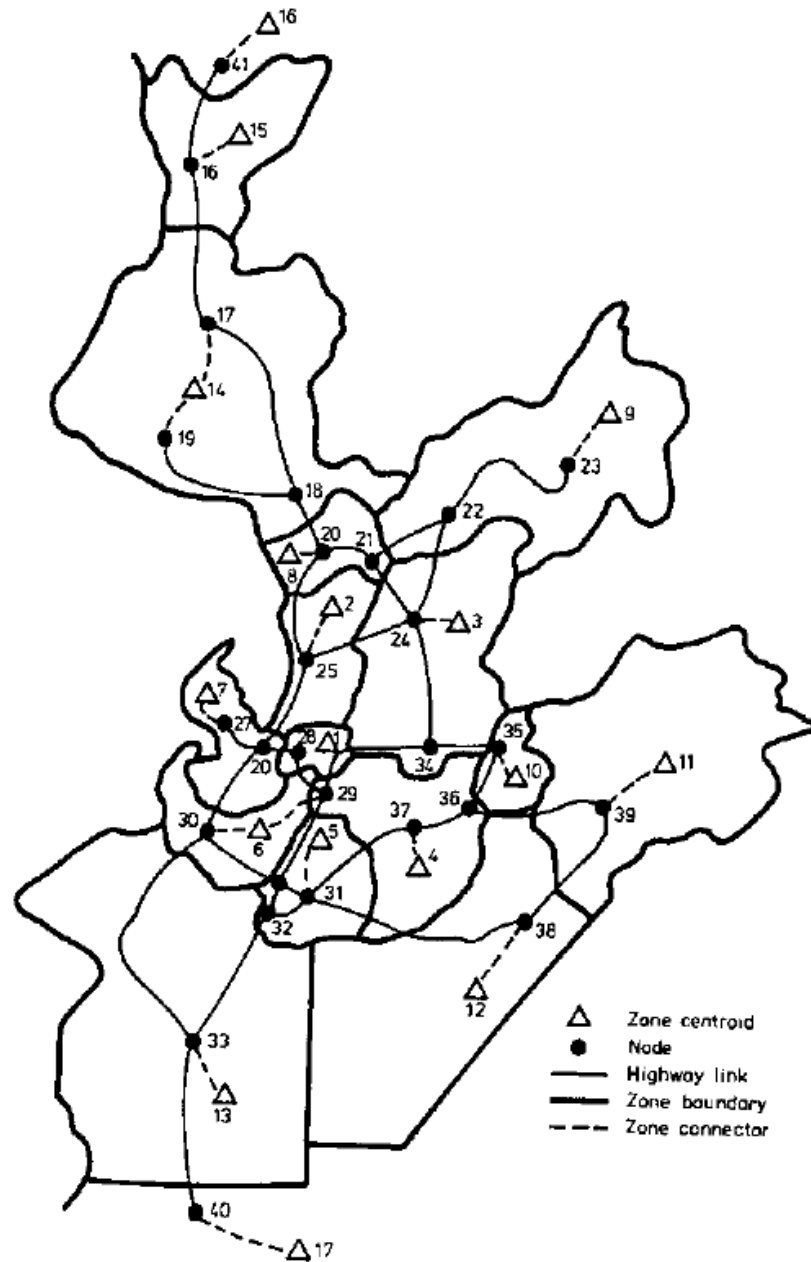
A área de estudo é dividida em zonas de tráfego, Bruton (1979) coloca que o propósito é aglomerar as origens e destinos de movimentos. Para Garber e Hoel (2009) a divisão das zonas deve seguir as seguintes considerações.

1. As características socioeconômicas devem ser homogêneas.
2. As viagens intrazonais devem ser minimizadas.

3. Os limites físicos, políticos e históricos devem ser utilizados sempre que possível.
4. As zonas não devem ser criadas dentro de outras zonas.
5. O sistema de zonas deve gerar e atrair viagens, domicílios, população ou área aproximadamente iguais.
6. As zonas devem usar os limites dos setores censitários sempre que possível.
7. O número total de zonas não deve ser tão grande a ponto de sobrecarregar o computador.

Ainda segundo Garber e Hoel (2009), o centroide é o ponto da zona onde as viagens possuem origem ou destino. A Figura 3 apresenta uma representação de zonas e de centroides.

Figura 3: Representação de zona e centroide



Fonte: Ortúzar e Willumsen (2011)

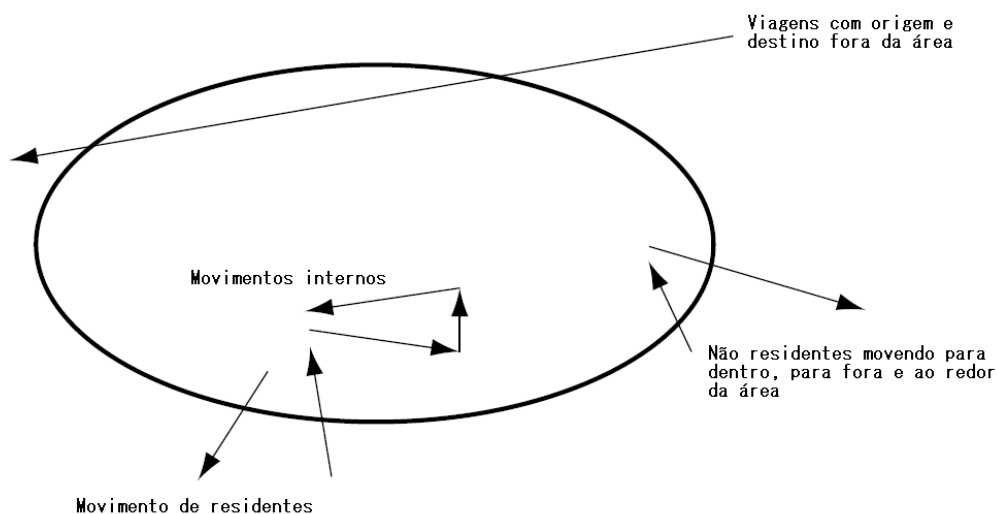
2.4.2 Dados de demanda

O levantamento de dados de demanda variam em função do objetivo, do horizonte de projeto e do tamanho da área de estudo. Campos (2013) coloca os levantamentos realizados para extração de dados em áreas urbanas:

1. entrevistas domiciliares: mais abrangente, pode identificar tanto as necessidades de deslocamento como o perfil socioeconômico da população. É indispensável em caso de planos de longo prazo.
2. contagem de tráfego: realizada em locais críticos onde existe grande movimento de veículos, nos principais corredores de transporte.
3. pesquisa no tráfego: permite identificar a origem e o destino da viagem a partir de uma entrevista com os motoristas.
4. pesquisa de embarque e desembarque em ônibus: identifica as origens e os destinos dos usuários das linhas de ônibus em todo o seu percurso, verificando o ponto em que o usuário pega o ônibus e onde desce.
5. pesquisa no cordão externo e na *screen line*: realizada nos principais corredores que cortam estas linhas com o objetivo de checar os movimentos observados através da pesquisa domiciliar.

Quatro tipos de movimentos ocorrem em relação a área de estudo. Movimentos com origem e destino internos a *cordon line*; movimentos externo-externo que possuem origem e destino fora da *cordon line* e passam pela área de estudo; movimentos interno-externo com origem dentro e destino fora da *cordon line*; e movimentos externo-interno com origem fora e destino dentro da *cordon line*. A Figura 4 apresenta os movimentos em relação a *cordon line*. Ainda segundo o autor, contagens volumétricas do tráfego cruzando um cordão interno são feitas para comparar-se o número de viagens estimadas a partir dos resultados das entrevistas com o número de viagens realmente observado na via.

Figura 4: Movimentos na área de estudo

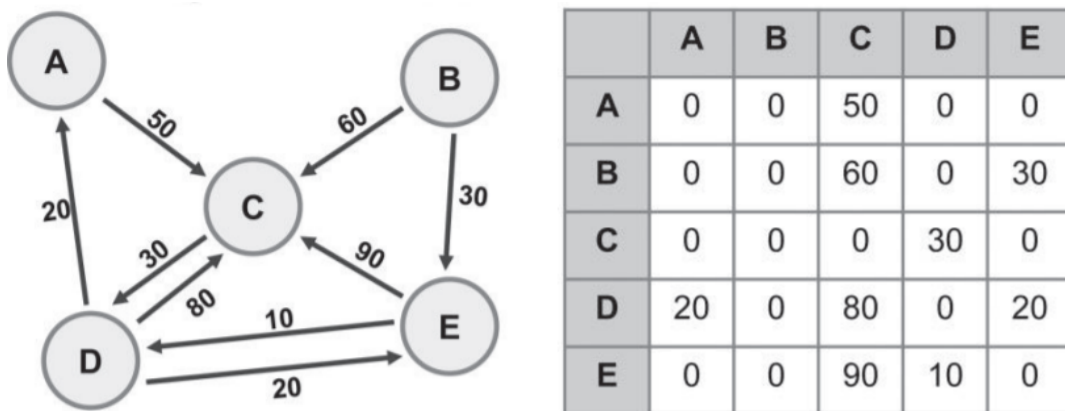


Fonte: Traduzido de Ortúzar e Willumsen (2011)

Segundo Bruton (1979), o movimento externo-externo e o externo-interno são pesquisados necessariamente no *cordon line*. O movimento interno-externo é captado com pesquisa domiciliar e podem também ser detectados no cordão externo. Os movimentos internos são obtidos através da pesquisa domiciliar.

As pesquisas de origem destino são realizadas para estabelecer uma compreensão completa dos padrões de viagem dentro da área de estudo. Segundo Garber e Hoel (2009) a pesquisa OD identifica sobre cada viagem que é feita em um dia específico, a origem e o destino da viagem, o seu motivo, a hora do dia e o veículo envolvido. Também são coletados dados sobre a pessoa que fez a viagem, idade, sexo, renda, proprietário do veículo, entre outros para procedimento de projeção de demanda. As viagens com seus respectivos pares origem e destino são armazenadas em uma matriz OD, como exemplifica a Figura 5.

Figura 5: Representação de matriz OD



Fonte: Adaptado de Rodrigue, Comtois e Slack (2016)

2.4.3 Dados de infraestrutura

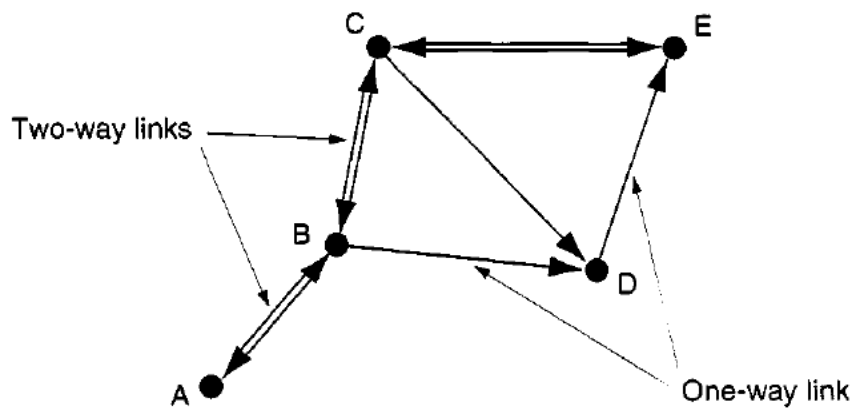
A rede de transporte (*network*), para Ortúzar e Willumsen (2011) tem como finalidade representar o que o sistema de transporte tem a oferecer para comportar as viagens dentro da área de estudo. Rodrigue, Comtois e Slack (2016) colocam que o arranjo e a conectividade de uma rede é conhecido como sua topologia, e cada modo de transporte possui o seu. Para Hensher (2004) a rede é uma construção matemática definida por um sistema interconectado de nós e ligações (*links*)². Rodrigue, Comtois e Slack (2016) definem *link*, nó e grafo:

1. nó: é um ponto terminal ou uma interseção no grafo;
2. *link*: é uma ligação entre dois nós. O link é uma abstração da infraestrutura de transporte que comporta os movimentos entre os nós;
3. grafo: é um conjunto de nós conectados por *links*.

A Figura 6 apresenta uma representação de *links* e nós em uma rede.

²O *software* SUMO, assim como outros autores, utiliza o termo *edge* para se referir a *link*.

Figura 6: Representação de *links* e nós



Fonte: Hensher (2004)

Ortúzar e Willumsen (2011) colocam que a descrição da rede em um modelo computacional pode conter diferentes níveis de detalhamento e requer especificações da infraestrutura, seus atributos e o relacionamento com o tráfego. Para efeito de simulação do tráfego são necessárias informações que forneçam descrição das vias que irão compor a referida rede. Quanto mais dados relacionados a infraestrutura da rede da área de estudo, mais próximo à realidade chega a simulação.

De acordo com DNIT (2006), para uma simulação, cada via deve possuir descrição completa dos seguintes dados:

1. local: situação na área de estudo, ressaltando sua importância, categoria e principais interseções;
2. dimensões físicas: comprimento, largura, número de faixas de tráfego, tipo de via, controles de acesso, etc;
3. características do tráfego: velocidade média de viagem nas horas de pico e fora dela, volumes de tráfego existentes, composição modal do tráfego, etc;
4. regulamentação do tráfego: sinalização das vias, mãos de direção, possibilidades de estacionamento, conversões proibidas, etc;

No que se refere à representação gráfica de uma rede de transportes coletivos, cabem as seguintes convenções (DNIT, 2006):

1. rota: caminho físico percorrido pelo modo de transporte considerado;

2. linha: representação da rota na rede.

DNIT (2006) também coloca que para a rede de transportes coletivos é necessário além dos dados levantados para a rede viária, um inventário geral do sistema de transportes coletivos:

1. mapa das rotas;
2. intervalos médios entre veículos consecutivos de cada rota (*headway*);
3. comprimento e tempo médio de percurso de cada rota;
4. períodos de operação;
5. frotas de veículos;
6. custos de operação.

2.4.4 Classificação funcional das vias urbanas

As vias urbanas podem ser classificadas de acordo com sua funcionalidade. O uso da classificação funcional é importante para criar padrões entre as vias, agrupando as por semelhança.

Segundo BRASIL (2010) a hierarquia das vias permite diferentes graus de mobilidade e acessibilidade. Ainda de acordo com o autor, a mobilidade é o grau de facilidade para deslocar-se e a acessibilidade é o grau de facilidade que oferece uma via para conectar a origem de uma viagem com seu destino.

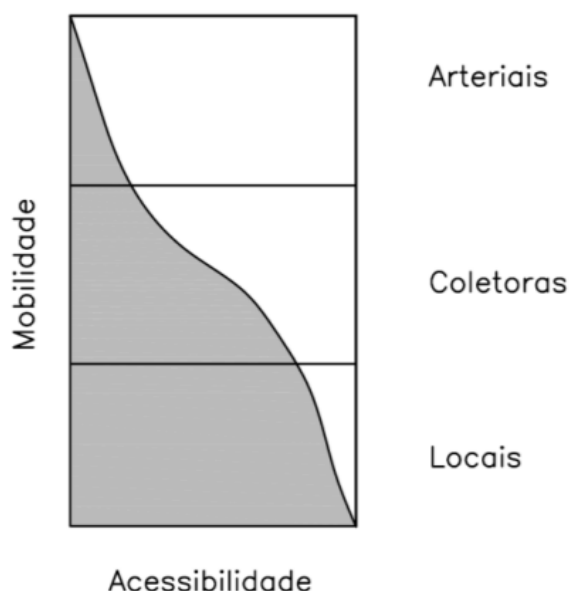
Ainda de acordo com BRASIL (2010), a classificação funcional agrupa as vias de forma hierárquica em subsistemas, conforme o tipo de serviço que oferecem e a função que exercem. Esta classificação separa as vias urbanas em quatro sistemas, com características e funções distintas.

- a) Sistema arterial principal: este sistema serve os principais centros de atividade das áreas urbanas, os corredores de maior volume de tráfego e as viagens mais longas, transporta grande parte do tráfego urbano. O sistema deve incluir conexões e extensões das rodovias arteriais e coletoras interurbanas. Fazem parte desse sistema: vias expressas primárias, vias expressas secundárias, vias arteriais primárias.
- b) Sistema arterial secundário: este sistema se interconecta com o sistema arterial principal de forma complementar. Atende aos percursos de viagens com extensões intermediárias, em níveis de serviço inferiores àqueles que são típicos das vias arteriais primárias. As viagens possuem mobilidade inferior ao do sistema arterial principal.

- c) Sistema coletor: tem como função conectar as ruas locais com as vias arteriais. O sistema proporciona continuidade ao nível das comunidades locais ou subdivisões urbanas. O sistema coletor coleta o tráfego das vias locais das áreas e o conduz ao sistema arterial. Possuem velocidade e distâncias inferiores ao sistema arterial.
- d) Sistema local: possui como função primária permitir o acesso às propriedades que lhe são adjacentes às vias de ordem superior. Oferece o menor nível de mobilidade e normalmente não contêm rotas de ônibus.

As relações entre os níveis de mobilidade e acessibilidade proporcionados pelas diferentes categorias funcionais podem ser visualizadas na Figura 7.

Figura 7: Relação entre os níveis de acessibilidade e mobilidade



Fonte: BRASIL (2010)

O Código de Trânsito Brasileiro (CTB), segundo Brasil (2003), Anexo I, traz definição de diferentes vias. São definidas seis classes, quatro para vias urbanas e duas para vias rurais conforme a Tabela 1.

Tabela 1: Classificação das vias conforme CTB

| Classificação | Característica |
|------------------------|---|
| Via de trânsito rápido | Aquela caracterizada por acessos especiais com trânsito livre, sem interseções em nível, sem acessibilidade direta aos lotes lindeiros e sem travessia de pedestres em nível. |
| Via arterial | Aquela caracterizada por interseções em nível, geralmente controlada por semáforo, com acessibilidade aos lotes lindeiros e às vias secundárias e locais, possibilitando o trânsito entre as regiões da cidade. |
| Via coletora | Aquela destinada a coletar e distribuir o trânsito que tenha necessidade de entrar ou sair das vias de trânsito rápido ou arteriais, possibilitando o trânsito dentro das regiões da cidade. |
| Via local | Aquela caracterizada por interseções em nível não semaforizadas, destinada apenas ao acesso local ou a áreas restritas. |
| Rodovia | Via rural pavimentada. |
| Estrada | Via rural não pavimentada. |

Fonte: Brasil (2003)

A partir de Brasil (2003), Artigo 61, é possível encontrar velocidades máximas permitidas das vias de acordo com a classificação. A Tabela 2 apresenta as velocidades máximas para locais sem sinalização regulamentadora conforme o CTB.

Tabela 2: Velocidades máximas segundo o CTB

| Classificação | Velocidade máxima permitida (km/h) |
|--|------------------------------------|
| Via de trânsito rápido | 80 |
| Via arterial | 60 |
| Coletora | 40 |
| Local | 30 |
| Rodovias de pista dupla | 110* |
| | 90** |
| Rodovias de pista simples | 100* |
| | 90** |
| Estradas | 60 |
| *para automóveis, camionetas e motocicletas; | |
| **para os demais veículos. | |

Fonte: Brasil (2003)

2.5 Alocação de tráfego

Esta seção aborda o conceito de alocação de tráfego e apresenta seus métodos.

Wolshon, Pande et al. (2016) definem que a alocação de tráfego (*trip assignment*) é o carregamento dos modos na rede. Segundo Garber e Hoel (2009) a alocação determina qual rota na rede de transporte será utilizada para realizar a viagem considerando que cada usuário deseja ter um tempo de viagem reduzido. A alocação de tráfego é a última etapa do Modelo 4 Etapas, ou Modelo Sequencial, utilizado para a elaboração de planos de médio e longo prazo no planejamento de transportes (CAMPOS, 2013).

Hutchinson (1974) coloca que o fundamental elemento da alocação de tráfego é selecionar um critério que explique a escolha do motorista por um rota entre um par origem e destino entre todas as outras disponíveis. Ainda segundo o autor, os dois princípios de Wardrop podem ser utilizados para prever caminhos entre pares de origem e destino:

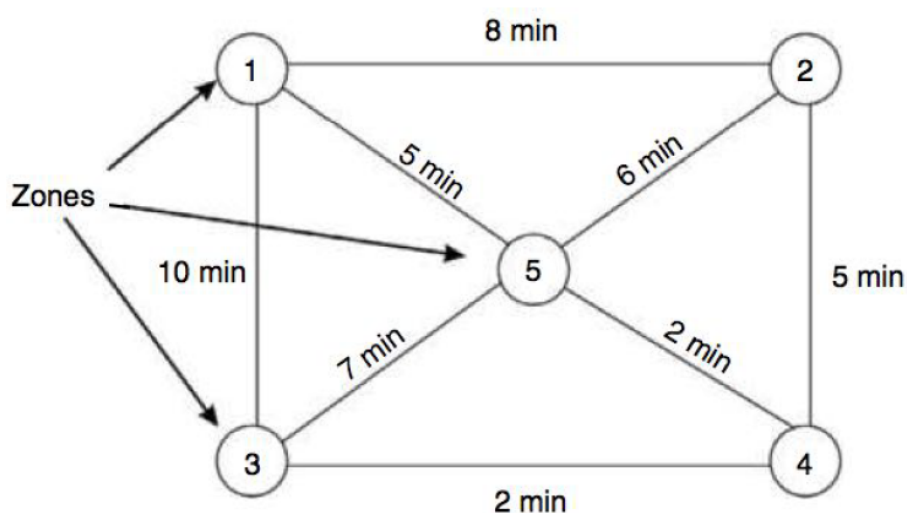
1. o tempo de viagem de todas as rotas usadas são iguais ou inferiores do que aqueles que seriam experimentados por um único veículo em qualquer rota não utilizada;
2. o tempo médio de viagem de todos os motoristas é mínimo o que significa que o tempo total gasto em viagem é mínimo.

Há diferentes modelos de alocação de tráfego na rede. Segundo (CAMPOS, 2013) os modelos têm por objetivo avaliar a distribuição do fluxo numa rede, a partir do cálculo da demanda de tráfego

entre as várias origens e destinos desta rede, tomando-se como base caminhos mínimos entre estes pares.

Os algoritmos de roteamento, conforme coloca Hensher (2004), utilizam a impedância dos *links* para calcular o caminho mínimo. O autor traz que a impedância pode ser distância, tempo de viagem ou algum tipo de custo. A Figura 8 apresenta um exemplo de rede viária com os valores de tempo de percurso assinalados em cada *link*, a serem utilizados como impedância para a alocação.

Figura 8: Tempo de percurso como impedância em *links* para alocação de tráfego



Fonte: Adaptado de Hoel, Garber e Sadek (2012)

Os modelos de alocação podem ser classificados em estático ou dinâmico. Campos (2013) coloca que os modelos dinâmicos surgiram como um aprimoramento dos modelos estáticos, considerando a dinâmica do fluxo, ou seja, o fato de que a demanda não é constante todo o tempo. A autora ainda coloca que os modelos de equilíbrio estático consideram que o fluxo é uniformemente distribuído para um determinado intervalo de tempo, não avaliando o efeito de congestionamento.

Os métodos de alocação disponíveis para utilização no *software* SUMO são:

1. Tudo ou Nada;
2. Incremental;
3. Equilíbrio do Usuário;
4. Equilíbrio Estocástico do Usuário.

2.5.1 Tudo ou Nada

Garber e Hoel (2009) colocam que o método Tudo ou Nada (*All-or-Nothing*) ou Caminho Mínimo (*Minimum Path*) atribui as viagens às rotas de menor impedância. Segundo Saw, Katti e Joshi (2015) o Tudo ou Nada é a alocação mais simples, baseada na escolha de rota com menor resistência. Segundo Rapelli, Casetti e Gagliardi (2019) as rotas dos veículos são definidas no tempo zero da simulação, e por consequência todos os veículos, de mesma origem e destino, irão escolher o mesmo caminho.

Conforme colocam Ortúzar e Willumsen (2011), este método assume que não haja efeitos de congestionamento, que todos os motoristas considerem os mesmos custos para escolha de rota e que os percebam e os pesem da mesma maneira. A ausência de efeitos de congestionamento significa que os custos são fixos; a suposição de que todos os motoristas percebem os mesmos custos significa que todos os motoristas da mesma origem e destino devem escolher a mesma rota. O método Tudo ou Nada pode ser aplicado com resultados razoáveis, segundo Ortúzar e Willumsen (2011), em redes descongestionadas, com poucas alternativas de rotas e que são muito diferentes em custo.

2.5.2 Incremental

Rapelli, Casetti e Gagliardi (2019) colocam que o modelo Incremental não tem como objetivo atingir o equilíbrio. É um algoritmo mais simples que aloca frações de volume de tráfego baseado no método de alocação Tudo ou Nada.

Ortúzar e Willumsen (2011) explicam que no modelo incremental a matriz OD é dividida em um número fracional de matrizes. Para isso, são aplicados fatores de proporção p_n em que $\sum_n p_n = 1$. As matrizes são carregadas na rede de forma incremental. Cada fração calcula a sua rota com base nos custos dos fluxos que se acumulam a cada incremento. Os autores ainda colocam que os valores p_n comumente aplicados são 0,4, 0,3, 0,2, e 0,1. Os autores apresentam o passo a passo do método Incremental:

1. selecionar um custo inicial para os *links*, usualmente é utilizado o tempo de fluxo livre. Inicializar todos os fluxos $V_a = 0$. Selecionar as frações p_n da matriz OD T em que $\sum_n p_n = 1$. Faz que $n = 0$.
2. construir o conjunto de árvores de custo mínimo para cada origem utilizando os custos correntes. Faz que $n = n + 1$.
3. carregar $T_n = p_n T$ com tudo ou nada nestas árvores. É obtido fluxos auxiliares F_a ; os fluxos são acumulados em cada *link*: $V_a^n = V_a^{n-1} + F_a$

4. calcular os novos custos dos links baseados nos fluxos V_a^n . Retornar para o passo 2 até ter terminado todas as frações da matriz OD T.

Segundo Ortúzar e Willumsen (2011) o método incremental sofre da limitação que uma vez que o fluxo foi alocado a um *link* ele não é mais removido e alocado em outro. Portanto se em uma iteração inicial for alocado fluxo demais em um *link* o modelo não irá convergir ao equilíbrio de Wardrop. Os autores também colocam que o método possui duas vantagens:

1. é de fácil programação;
2. os resultados podem ser interpretados como os acúmulos de tráfego que ocorrem no horário de pico.

2.5.3 Equilíbrio do Usuário

O método de alocação por Equilíbrio de Usuário (UE) é computacionalmente mais trabalhoso, uma vez que pode não convergir em um número limitado de iterações. Sua utilização é importante no contexto de transporte urbano e metropolitano, para representar os efeitos decorrentes do congestionamento, com base no primeiro princípio de Wardrop (SENNA, 2014). Ortúzar e Willumsen (2011) trazem que, considerando o primeiro princípio de Wardrop, todas as rotas usadas do mesmo par origem e destino devem ter o mesmo custo, e que todas as rotas não utilizadas devem ter custo maior ou igual as utilizadas.

Hensher (2004) coloca que o UE significa que o equilíbrio das rotas conectando cada par de origem destino pode ser dividido em dois grupos. O primeiro grupo inclui todas as rotas que carregam fluxo de veículos e possuem tempos de viagem iguais. O segundo grupo inclui todas as rotas sem fluxo de veículos e que possuem tempo de viagem igual ou superior ao primeiro grupo.

Mannering e Washburn (2020) colocam que os motoristas irão escolher as rotas com base apenas no menor tempo de viagem, e que não irão considerar fatores secundários como cenário e condição de pavimento por exemplo. Segundo os autores a segunda presunção do método é que os motoristas sabem de todos os tempos levados para cada opção de rota, e que os percebem da mesma forma.

Segundo Rapelli, Casetti e Gagliardi (2019) o algoritmo leva a uma solução de equilíbrio, porém em simulações de larga escala não é praticável pois a cada novo veículo inserido na rede, todos os veículos devem recomputar seus tempos de viagem para todos os caminhos da rede.

2.5.4 Equilíbrio Estocástico do Usuário

O UE acontece se cada usuário tem perfeito conhecimento das condições da rede e todos os usuários possuem idênticas percepções do tempo de viagem, o que é uma idealização. Na realidade, coloca Hensher (2004), usuários podem ter uma certa variação na percepção do tempo ou erros no tempo

de viagem. Portanto, os usuários selecionam rotas para melhorar o tempo de viagem notado ao invés do tempo de viagem verdadeiro que não é do conhecimento deles. O princípio da escolha de rota que contém variações na percepção é chamado de Equilíbrio Estocástico do Usuário (SUE).

Segundo Ortúzar e Willumsen (2011), o método de alocação do SUE dá ênfase a variabilidade de percepção dos usuários ao custo e a composição de medidas que querem minimizar (distância, tempo de viagem, custos generalizados). Hensher (2004) coloca que para cada par origem destino, no SUE, o tempo de viagem notado por todos os usuários é igual e igual ou inferior as rotas não utilizadas. Quando um sistema está em SUE, os usuários não conseguem melhorar o tempo percebido de viagem mudando de rota.

Hensher (2004) apresenta Equação 1 para representar o SUE:

$$C_k = c_k + \xi_k \quad (1)$$

onde C_k é o tempo percebido, c_k é o tempo real e ξ_k é a variável randômica de percepção que possui diferentes modos de determinação conforme o algoritmo SUE utilizado.

2.6 Calibração e validação de modelos de simulação de tráfego

Um modelo de simulação deve ser calibrado e validado para que possam então ser elaborados, a partir dele, cenários alternativos.

Ortúzar e Willumsen (2011) colocam que calibrar um modelo requer escolher os parâmetros para otimizar uma ou mais medidas em função dos dados observados. Segundo os autores o modelo pode ser representado de forma simplificada por uma função matemática de variáveis X e parâmetros θ conforme a Equação 2.

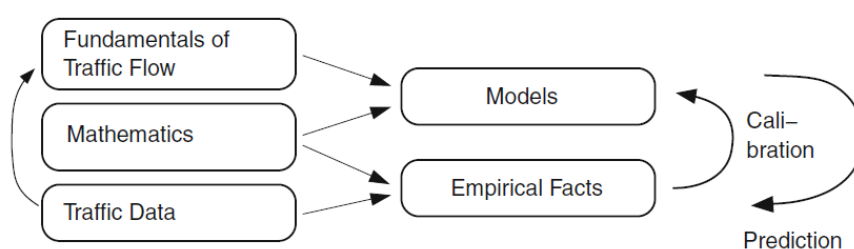
$$Y = f(X, \theta) \quad (2)$$

Para Spiegelman, Park e Rilett (2011), calibração consiste em modificar os parâmetros *default* do programa para que o modelo replique as condições observadas de tráfego de forma tão acurada como possível.

Ortúzar e Willumsen (2011) colocam que estimar os parâmetros, envolve encontrar os valores para fazer com que os dados observados se apresentem de acordo no modelo. Portanto, na prática, a simulação é executada diversas vezes testando troca de variáveis com o objetivo de obter o resultado mais próximo ao observado. Os parâmetros também podem ser definidos por meio de pesquisa de campo.

Treiber e Kesting (2013) colocam claramente a posição da calibração no processo da simulação. Os modelos de fluxo de tráfego descrevem a dinâmica de veículos e motoristas; previsões são obtidas executando a simulação do modelo; os parâmetros do modelo são escolhidos para que a simulação produza o melhor resultado com os dados (calibração do modelo). E uma vez calibrado, o modelo pode ser usado para previsão de fluxo de tráfego e outras aplicações. A Figura 9 sintetiza o que o autor coloca.

Figura 9: Modelagem do fluxo de tráfego



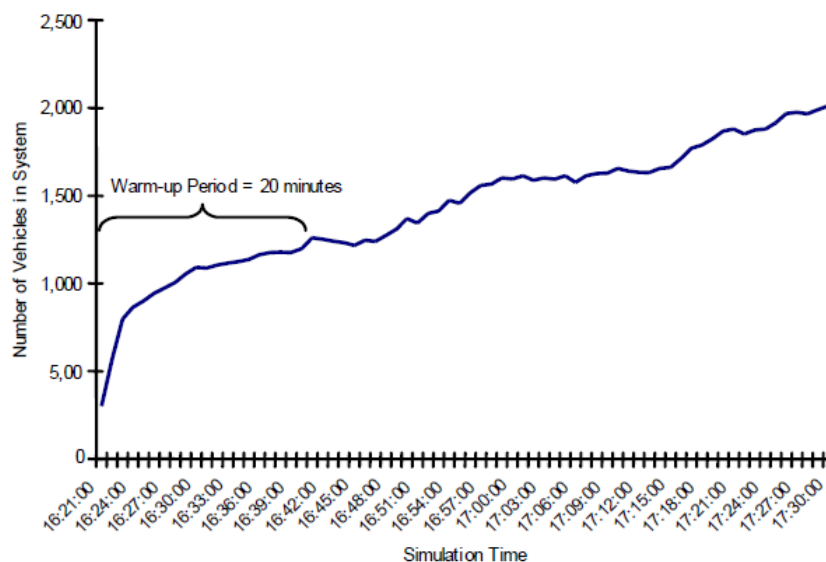
Fonte: Treiber e Kesting (2013)

A validação é o processo de afirmação da calibração. FHWA (2004a) coloca que é o processo no qual o analista checa a performance geral do tráfego previsto para um sistema de rodovias em comparação com medições de campo da performance da simulação, com volumes, tempos de viagem, velocidades médias e atrasos e deve ser feita baseada em dados de campo não usados no processo de calibração.

2.7 *Warm-up*

Modelos de simulação microscópico e mesoscópico, geralmente, iniciam com a rede vazia, sem veículos. Segundo Elefteriadou et al. (2014), o tempo necessário para que a rede receba veículos é chamado de período de *warm-up*. Os autores colocam que durante o período de *warm-up* dados não devem ser contabilizados pois as condições de simulação não estão representativas. Segundo FHWA (2004b), o tempo de *warm-up* finaliza quando número de veículos presentes na rede deixa de aumentar em um valor mínimo especificado, como mostra a Figura 10.

Figura 10: Tempo de *Warm-up*



Fonte: Dowling et al. (2004)

FHWA (2004b) coloca ainda que o período de *warm-up* deve ser no mínimo duas vezes o tempo de viagem em fluxo livre para atravessar a rede da viagem mais longa. Os autores citam esse tempo mínimo quando estão se referindo a uma microssimulação.

2.8 Semáforos

Em áreas urbanas é comum o controle de interseções com o uso de semáforos. Eles são uma importante forma de controle de interseção que possibilitam o aumento da segurança viária. Denatran (1984) define semáforo como um dispositivo de controle de tráfego que, através de indicações luminosas transmitidas para os motoristas, alterna o direito de passagem de veículos e/ou pedestres em interseções de duas ou mais vias.

2.8.1 Controle semaforico

Os semáforos podem ser controlados de diferentes formas para definição do plano semaforico. CONTRAN (2014) coloca que as formas de controle possuem níveis distintos de resposta a variações observadas nos movimentos controlados. O autor define duas maneiras de realizar o controle.

1. Tempo fixo: o controle em tempo fixo utiliza planos semaforicos calculados com base em dados de tráfego disponíveis, obtidos por contagens volumétricas e outros levantamentos de

campo. O controle pode ser efetuado com base em um único plano semafórico, ou na adoção de planos específicos para atender a demanda de tráfego histórica de períodos distintos do dia e de diferentes dias da semana. Ao longo do período de ação de um plano semafórico, o tempo de ciclo e a sequência de estágios são mantidos constantes.

2. **Atuado:** O controle atuado pode ser ou semiatuado ou totalmente atuado. O controle semiatuado é, geralmente, empregado em cruzamentos de vias de grande volume (vias principais) com vias de baixo volume de tráfego (vias secundárias). A indicação verde é dada continuamente para os veículos da via principal, sendo interrompida quando detectores implantados nas aproximações da via secundária indicarem a presença de veículos motorizados. O controle totalmente atuado decorre do monitoramento da demanda de tráfego na interseção, mediante a implantação de detectores de tráfego em todas as suas aproximações, permitindo alterações nos tempos dos estágios. O princípio básico do funcionamento em modo totalmente atuado é o da determinação do tempo de verde associado a cada estágio de sinalização, variando entre um valor mínimo e um valor máximo pré-estabelecidos. Esse tipo de controle pode permitir o ajuste em tempo real dos valores de alguns dos parâmetros de programação, como por exemplo, a prioridade a ser dada para uma aproximação congestionada.

O *software* SUMO contém o modo de controle totalmente atuado, o qual coloca nas interseções semafóricas detectores de tráfego que proporcionam os dados da condição em tempo real do tráfego da simulação. O controle atuado no SUMO funciona apenas no modo microscópico, para o mesoscópico ainda não está disponível. O *software* quando não utiliza o controle atuado e não recebe programação semafórica específica, coloca tempos de ciclo padrão para todos os semáforos, adotando tempo de ciclo de 90 segundos, o tempo de verde é dividido igualmente para todas as fases.

CONTRAN (2014) também coloca sobre as estratégias de controle semafórico. Há duas formas básicas de controle: o controle isolado e o controle em rede. O autor traz que no controle semafórico isolado, cada interseção é controlada independentemente das demais, ou seja, não ocorre nenhum tipo de coordenação semafórica. Nesse caso, a definição da programação semafórica leva em conta apenas a demanda (histórica ou atual) do tráfego em todas as aproximações. Essa estratégia de controle pode comprometer o desempenho da circulação do tráfego em situações onde as interseções controladas por sinalização semafórica estiverem muito próximas entre si. Já o controle em rede, pode visar o aumento do desempenho da circulação do tráfego ao longo de uma rede aberta (uma via ou em um percurso pré-estabelecido) ou de uma rede fechada (em uma determinada região). A estratégia de controle em rede permite a programação da sinalização semafórica visando seu desempenho global ao longo do conjunto de cruzamentos.

O *software* SUMO possui um *script* chamado *tlscoordinator.py* que tem como objetivo modificar os ciclos para por em prática a estratégia de controle em rede (ver Seção 3.2.2).

2.8.2 Tempo de ciclo

CONTRAN (2014) define o tempo de ciclo como o tempo necessário para que um ciclo inteiro seja cumprido em uma interseção, ou seção de via, e é determinado pela soma dos tempos de todos os estágios programados para o controle do tráfego no local. Valores altos para o tempo de ciclo implicam em tempos de espera muito elevados, nas situações comuns de controle esse valor não deve superar 120 segundos. Há diferentes métodos de cálculo para definição do tempo de ciclo. O método disponível no *software* SUMO é o método de Webster (ver Seção 3.2.2) com o *script* *tlsCycleAdaptation.py*.

Segundo Denatran (1984) o método de Webster possibilita calcular o tempo de verde e o comprimento de ciclo dos semáforos causando menor atraso geral possível na interseção. O método utiliza o fluxo de saturação que é o fluxo que seria obtido se houvesse uma fila de veículos na aproximação e a ela fossem dados 100% de tempo de verde no cruzamento. O outro fator do método é o tempo perdido (tempo morto) que é o tempo das reações dos motoristas no início e no fim da fase verde. Denatran (1984) ainda coloca que os valores de fluxo de saturação e tempo perdido devem ser obtidos, preferencialmente, no local por meio de medidas feitas em campo.

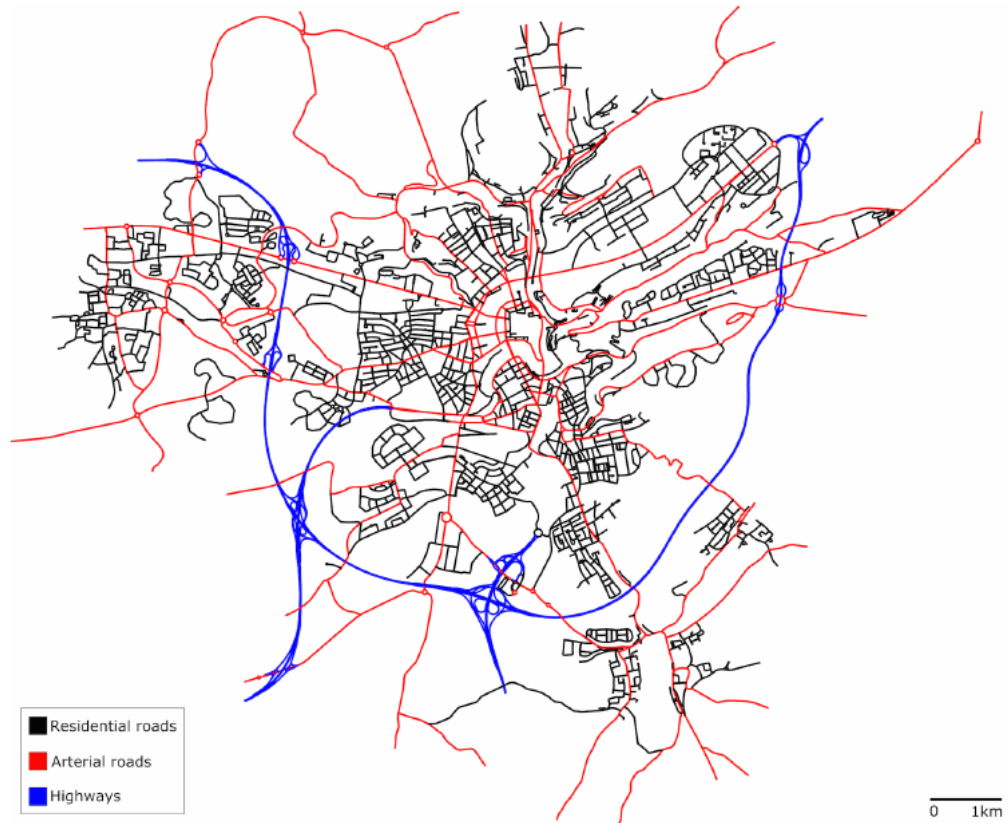
2.9 Exemplos de aplicação do SUMO em grandes áreas urbanas

Esta seção apresenta modelos de simulação que utilizaram o *software* SUMO com o objetivo de aplicar em grandes áreas de estudo.

2.9.1 *Luxembourg SUMO Traffic (LuST) Scenario*

O *Luxembourg SUMO Traffic (LuST) Scenario* é um modelo de microsimulação elaborado no SUMO por Codeca, Frank e Engel (2015). O objetivo dessa simulação é prover um cenário capaz de reproduzir com realismo o tráfego de uma cidade de tamanho médio europeia por 24 horas. O local escolhido foi a área central da cidade de Luxemburgo devido a sua topologia, Figura 11. A cidade possui um bairro central circundado por diferentes bairros conectados por rodovias arteriais e no perímetro da cidade há uma *highway*.

Figura 11: Rede de tráfego do LuST



Fonte: Codeca, Frank e Engel (2015)

Foi utilizado o OSM (ver Seção 4.2) para a obtenção dos dados de infraestrutura da rede. O JOSM³ foi utilizado para modificação de pontos e segmentos de rodovias, não foram especificados detalhes sobre o procedimento. A área de estudo possui 156km^2 . Os dados da rede de tráfego estão na Tabela 3.

³<<https://josm.openstreetmap.de/>> Acesso em 05/03/2022

Tabela 3: Dados da rede de tráfego LuST

| | |
|----------------------------------|----------|
| Total de nós | 2.372,00 |
| Total de links | 5.969,00 |
| Total comprimento de links (km) | 931,12 |
| Total comprimento de faixas (km) | 1.571,40 |
| <i>Links</i> com 1 faixa | 3.944,00 |
| <i>Links</i> com 2 faixas | 1.188,00 |
| Links com 3 faixas | 764,00 |
| <i>Links</i> com 4 faixas | 78,00 |

Fonte: Traduzido de Codeca, Frank e Engel (2015)

O cenário conta com 563 pontos de ônibus e 38 rotas de ônibus. Os dados de demanda de tráfego do cenário foram gerados com a ferramenta do SUMO chamada *ACTIVITYGEN* que necessita informações de topologia, dados demográficos da população, áreas escolares, residenciais e de trabalho. A demanda de transporte privado gerada é alocada na rede com a ferramenta *duarouter* do SUMO. O cenário é composto pela demanda de transporte público, tráfego local com movimento interno, interno-externo e externo-interno e pelo tráfego externo com movimento externo-externo.

Os relatórios emitidos após rodar a simulação apontam erros de simulação que não ultrapassam 0,2% conforme apresenta a Tabela 4. Teletransporte, faixa errada, *yield* e *jam* estão definidos na Seção 3.6.

Tabela 4: Dados da simulação LuST

| | Tráfego completo | | Tráfego local e transporte público | |
|--------------------|------------------|-------------|------------------------------------|-------------|
| | Total | Porcentagem | Total | Porcentagem |
| Veículos inseridos | 295.979,00 | | 218.938,00 | |
| Teletransportes | 479,00 | 0,162% | 183,00 | 0,084% |
| <i>Yield</i> | 98,00 | 0,033% | 37,00 | 0,0017% |
| Faixa errada | 229,00 | 0,077% | 83,00 | 0,038% |
| <i>Jam</i> | 142,00 | 0,048% | 55,00 | 0,025% |

Fonte: Traduzido de Codeca, Frank e Engel (2015)

Os autores colocam que as interseções com semáforos receberam controle totalmente atuado. O LuST aplicou o modelo de alocação que é uma variação do Incremental (ver Seção 3.6) com a opção de que 70% dos veículos podem trocar de rota durante a simulação a cada 5 minutos como

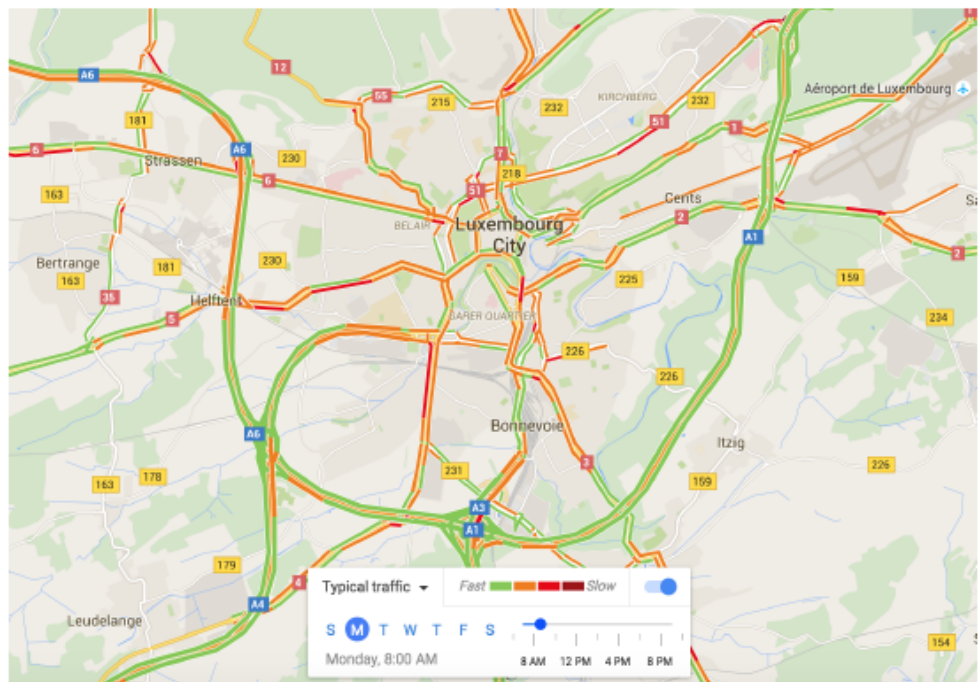
forma de reação a bloqueios no trânsito. Os dados da simulação indicam que 50% dos veículos não trocaram de rota, 30% trocaram apenas uma vez, 15% trocaram duas vezes e 5% mais de duas vezes.

O estudo realizou uma comparação entre o tráfego típico disponibilizado pelo Google Maps e o tráfego obtido pela simulação. A imagem gerada no SUMO utilizou o dado de velocidade relativa na via. A Figura 12 apresenta o tráfego na cidade às 8 horas, em verde tráfego leve e em vermelho tráfego pesado. O estudo comparou as demais horas e concluiu que a simulação é realista.

Figura 12: Comparação de tráfego do modelo Lust



(a) Simulação *LuST*



(b) *Google Maps*

Fonte: Codeca, Frank e Engel (2015)

2.9.2 *Turin SUMO Traffic*

O objetivo do modelo *Turin SUMO Traffic* (TuST) criado por Rapelli, Casetti e Gagliardi (2019) é contribuir no estudo de simulação de tráfego de grandes áreas de código aberto. A área de estudo modelada possui 400 km^2 , engloba a cidade italiana *Turin* e parte de cidades adjacentes. O dado de demanda, matriz OD, foi fornecido por uma empresa de transportes que atua na região. O total de zonas de tráfego utilizada foi de 257. A microssimulação é realizada para o período de 24 horas, com aproximadamente 2,2 milhões de viagens privadas.

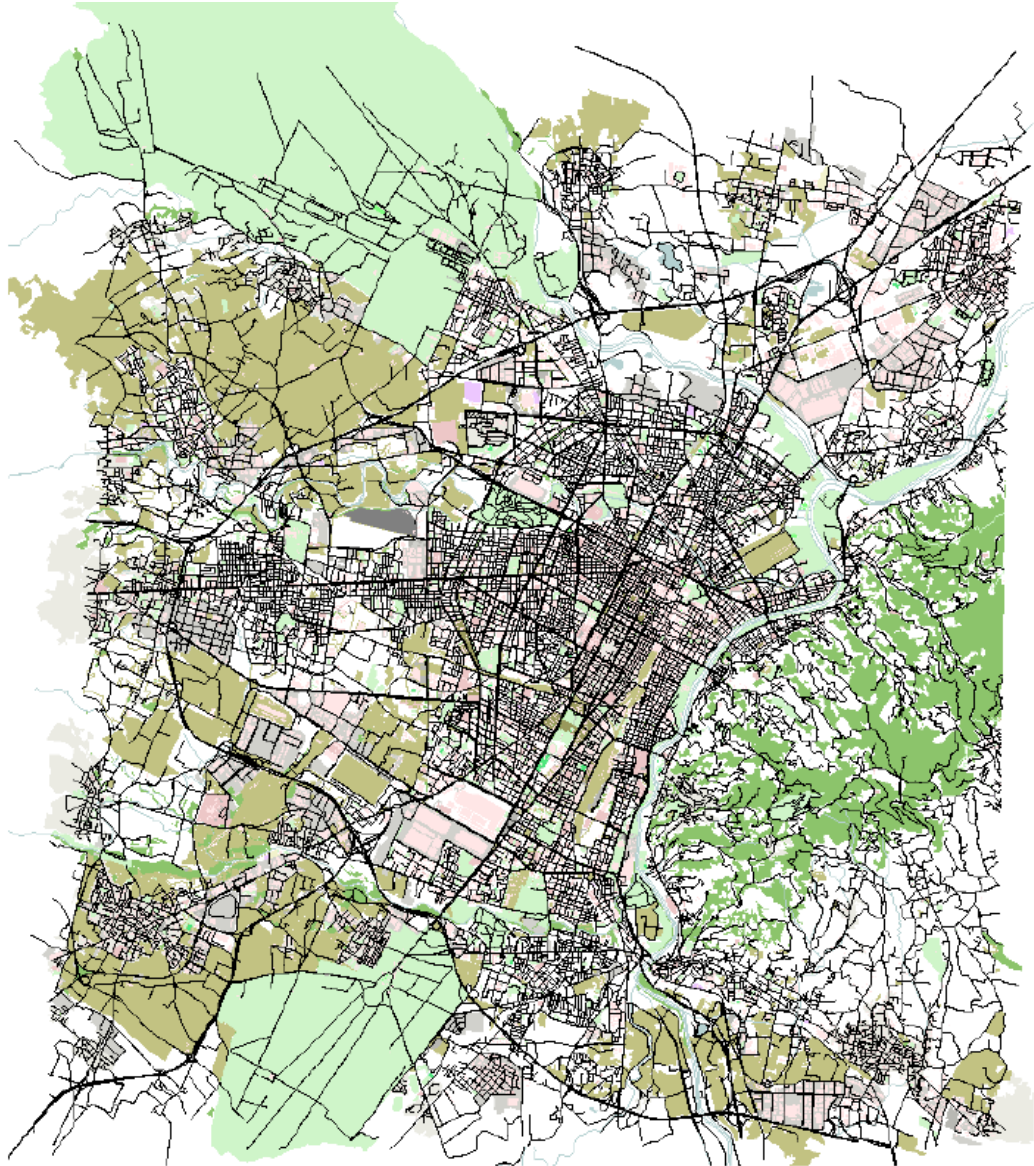
Para obter a rede viária da área de estudo, foi utilizado o OSM e o JOSM para edição e filtragem de informação. O arquivo foi compatibilizado para o SUMO com a ferramenta *netconvert*. A síntese dos dados da rede está na Tabela 5. A Figura 13 apresenta a rede de tráfego da área de estudo.

Tabela 5: Dados da rede de tráfego TuST

| | |
|---------------------------------|----------|
| Área (km^2) | 602,61 |
| Zonas de tráfego | 257 |
| Nós | 32.936 |
| <i>Links</i> | 66.296 |
| Semáforos | 856 |
| Comprimento de <i>link</i> (km) | 6.570,28 |
| Comprimento de faixa (km) | 7.723,40 |

Fonte: Rapelli, Casetti e Gagliardi (2019)

Figura 13: Rede de tráfego *TuST*



Fonte: Rapelli, Casetti e Gagliardi (2019)

Os autores afirmam que a malha possui grande quantidade de erros de infraestrutura devido as imprecisões do *OSM* e a conversão realizada pelo *netconvert* (ver Seção 3.2.1). Os autores também colocam que todos os modelos de simulação carregam erros e para arrumar em uma grande área de estudo podem demandar muito esforço. As correções, no *TuST*, só foram realizadas em pontos que comprometiam em excesso a simulação. A primeira edição foi no limite de velocidade de cada *link*. Os autores afirmam que os veículos trafegam normalmente acima da velocidade permitida. Portanto, decidiram utilizar a velocidade de fluxo livre, situação em que há apenas um veículo na via. Para obter essa velocidade foi utilizado o *Google API Distance Matrix*, que fornece o tempo necessário

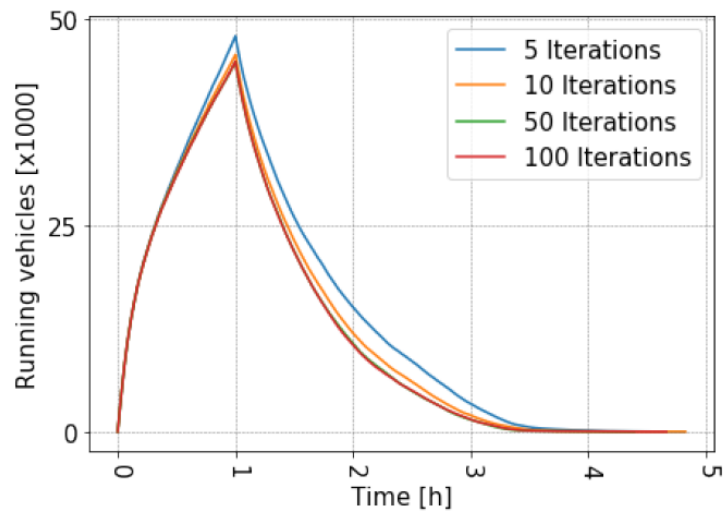
dado uma origem e destino, um horário e a condição de tráfego (otimista, médio e pessimista) para efetuar a rota. Um *script* em *Python* foi utilizado para obter o tempo em todos os segmentos para as 4:30 da manhã na condição otimista, e assim obter a velocidade de fluxo livre.

Outra edição necessária foi no número de faixas em vias que possuíam número inferior ao real. A correção foi feita de forma manual, comparando o modelo com o *Google Maps Street View*. As conexões em mais de 500 rotatórias precisaram ser manualmente verificadas devido a prioridade. As correções nas fases dos semáforos continuam em estudo. A simulação gerou os resultados com todos desligados. Todas as interseções são com base em prioridade.

Os autores colocam que no SUMO há duas formas de alocação iterativa, o método utilizando alocação microscópica e outro utilizando macroscópica. O UE utiliza a forma microscópica por meio da ferramenta *dualterate.py*. O método apresenta grandes intervalos de tempo para rodar cada iteração, e se torna inviável em grande redes. No entanto, o SUMO possui o *marouter*, uma ferramenta que gera de forma macroscópica a alocação de tráfego para os métodos SUE e Incremental. A simulação realizada no experimento utiliza diferentes métodos de alocação de tráfego para analisar os resultados obtidos com cada uma para o período de 24 horas. Os autores afirmam que não existe melhor modelo de alocação, a performance depende do caso em estudo. Os métodos de alocação testados foram: Tudo ou Nada, SUE-Gawron, SUE-Logit e Incremental.

O SUE-Gawron, coloca Rapelli, Casetti e Gagliardi (2019), computa a probabilidade de escolher a rota baseado no tempo usado para a realizar a rota na iteração anterior, na soma dos tempos de viagem para cada *link* e na probabilidade anterior de escolher a rota. O SUE-Logit, coloca Rapelli, Casetti e Gagliardi (2019), aplica uma fórmula fixa para cada rota para calcular a nova probabilidade. Ignora custos e probabilidades antigos. O SUE-Logit calcula a probabilidade com uma função exponencial e os custos são as somas dos tempos de viagem de cada *link* da última iteração. Os resultados tanto para o SUE-Logit quanto para o SUE-Gawron não foram adequados no modelo pois para a hora de pico da simulação foram necessárias mais de 9 horas para esvaziar a rede. Ou seja, as viagens apresentaram tempo elevado demais para serem finalizadas e que não é representativo para a área de estudo. A alocação incremental se mostrou eficiente com 50 iterações para a hora de pico, levando 3,5 horas para que a malha esvazie. A Figura 14 apresenta os resultados obtidos para a alocação Incremental com diferentes números de iteração. A curva de 50 iterações fica bem próxima a de 100 iterações portanto os autores colocam que 50 iterações já são suficientes.

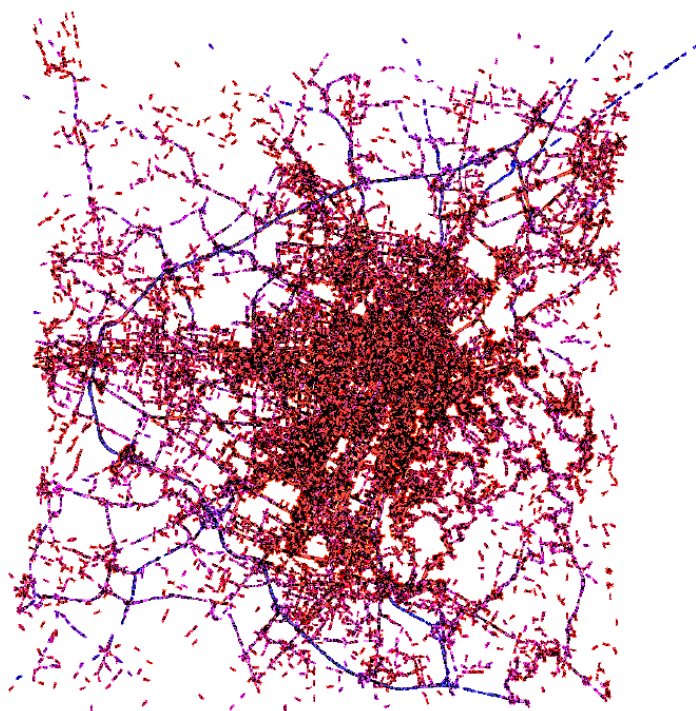
Figura 14: Alocação Incremental - TuST



Fonte: Rapelli, Casetti e Gagliardi (2019)

Segundo os autores, o resultado para hora de pico com a alocação incremental poderia simular um cenário pessimista da cidade, com congestionamento. A Figura 15 compara a alocação Incremental com a alocação Tudo ou Nada. É possível observar que na simulação com Tudo ou Nada ocorre maior concentração de veículos.

Figura 15: Comparação de alocação de tráfego Tudo ou Nada e Incremental do modelo TuST



(a) Alocação Tudo ou Nada



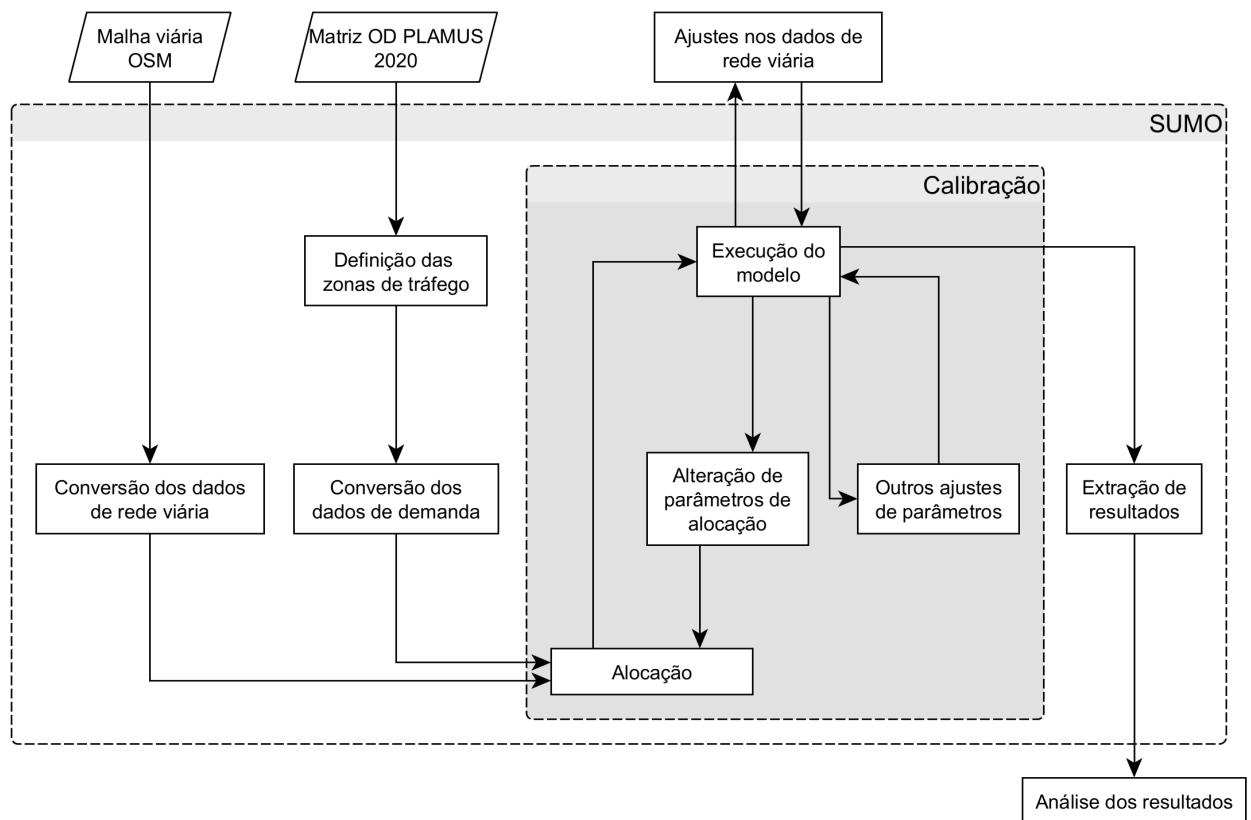
(b) Alocação Incremental com 50 iterações
Fonte: Rapelli, Casetti e Gagliardi (2019)

Para 24 horas de simulação nenhum dos métodos de alocação foi eficiente, indicando longos períodos de tempo para que a simulação ocorresse.

3 Método

O trabalho consiste em uma simulação mesoscópica englobando treze municípios de Santa Catarina utilizando o *software* SUMO e dados abertos de demanda e de malha viária. A Figura 16 apresenta o fluxograma do método adotado.

Figura 16: Fluxograma do método



3.1 Recursos de *softwares* utilizados

Os recursos de *softwares* utilizados neste trabalho estão listados a seguir.

1. SUMO

O simulador SUMO executa simulações de tráfego como microssimulador e mesossimulador. Possui interface de edição de malha viária, o *Netedit* e interface para visualizar a simulação, o *sumo-gui*. O SUMO utiliza arquivos declarados pelo usuário com extensão *.xml* como base para as operações que são executadas por linha de comando. O SUMO também possui *scripts* abertos em *Python* disponíveis para os usuários. A versão do SUMO utilizada neste trabalho é a 1.12.0 para *Windows*.

2. Quantum GIS

O Quantum GIS (QGIS)⁴ é um *software* de SIG de código aberto. Neste trabalho, o QGIS, na sua versão 3.20, foi utilizado para verificar os *shapefiles* disponíveis do PLAMUS e trocar o sistema de coordenadas do *shapefile* de zonas de tráfego.

3. Anaconda

Anaconda⁵ é uma plataforma de ciência de dados. A ferramenta computacional possibilita gerir distribuições de *Python*, ambientes de trabalho e módulos. O *Anaconda Prompt* é utilizado para executar os programas em *Python* do *SUMO*.

O equipamento utilizado neste trabalho para rodar os recursos de *software* é um *notebook* cujas configurações são apresentadas na Tabela 6.

Tabela 6: Configuração do equipamento utilizado

| Configuração | Modelo |
|----------------------------|--------------------------------------|
| <i>Sistema operacional</i> | Windows 10 |
| Processador | Intel Core i7-7700HQ 2.80GHz 2.80GHz |
| Memória RAM | 8.0 GB |
| Armazenamento | SSD: 500 GB |
| | Disco rígido: 1 TB |

Nesta seção são apresentados apenas os procedimentos do SUMO que foram utilizados para desenvolvimento deste trabalho. Quando for utilizado <arquivo> em quadros, significa que deve ser inserida uma nomeação própria.

3.2 Modelagem da infraestrutura viária

Nesta seção, são apresentadas a configuração e a metodologia para obter e editar a rede viária no SUMO, como configurar as fases semafóricas e a definição de veículos da simulação.

3.2.1 Rede de tráfego

O arquivo de rede do SUMO trabalha com os seguintes elementos:

1. *junction*: possui identificador, posição, os *lanes* em que está conectada e a geometria.

⁴<https://qgis.org/pt_BR/site/> Acesso em 05/01/2022.

⁵<<https://anaconda.org/>> Acesso em 05/01/2022.

2. *edges*: cada *edge* possui um identificador e conta com seus respectivos *lanes*, a classificação do *OSM* e a prioridade. Também possui a identificação das duas *junctions* em que está conectado.
3. *lane*: cada *lane* possui um identificador e definições da geometria, da velocidade permitida, do comprimento e da prioridade;
4. *tls*: os semáforos com a definição das fases e dos *offsets*;

O arquivo *.osm* da área de estudo deve ser convertido para formato compatível com o SUMO. O importador que exerce essa função é o *netconvert*, que é capaz de transformar arquivos de diversas fontes para o formato *.net.xml* o qual é reconhecido pelo *software* SUMO. Por meio de linha de comando essa operação é realizada. A conversão permite adicionar *flags* para ajustar a malha viária. O Quadro 1 apresenta a linha de comando para gerar o arquivo *.net.xml* e algumas *flags* de edição.

Quadro 1: Função *netconvert*

```
netconvert --osm-files <arquivo>.osm -o <arquivo>.net.xml --ptstop-output <arquivo>.add.xml --ptlines-output <arquivo>.add.xml --no-turnarounds.except-deadend --remove-edges.by-type --tls.join
```

O comando *--osm-files <arquivo>.osm* lê o arquivo do OSM, e *-o <arquivo>.net.xml* é o arquivo resultante deste processo. As seguintes *flags* possuem as funções:

1. *--ptstop-output <arquivo>.add.xml*: escrever um arquivo *.add.xml* com a localização de todos os pontos de ônibus localizados dentro da rede;
2. *--ptlines-output <arquivo>.add.xml*: escrever um arquivo *.add.xml* com as linhas de ônibus da área de estudo;
3. *--no-turnarounds.except-deadend*: remover o retorno na própria via, e manter o retorno em vias sem saída;
4. *--remove-edges.by-type*: remove os *edges* com base na classificação. A rede importada do OSM pode ser filtrada para conter apenas os *edges* de interesse para a simulação;
5. *--tls.join*: agrupa semáforos com até 20 metros de distanciamento, com objetivo de criar um plano semaforico.

3.2.2 Semáforos

O *script* do SUMO chamado *tlsCycleAdaptation.py* tem como objetivo otimizar o tempo de ciclo e o tempo de verde com base na fórmula de Webster. São utilizados o arquivo de rede *.net.xml* e o arquivo de rotas *.rou.xml* (ver Seção 3.4) como dados de entrada. Com base no arquivo de rotas, o número de carros de passeio é contabilizado para cada semáforo. Semáforos sem fluxo de veículos não são otimizados. Pode ser descontado o tempo de *warm-up* colocando a *flag* *--begin* e o tempo em que começa a simulação. Entre as definições possíveis do *script*, existe a opção de colocar o tempo de verde mínimo com *--min-green* e o tempo de amarelo com *--yellow-time*. O arquivo *output* possui a extensão *.add.xml*. O Quadro 2 apresenta uma forma de utilizar o *tlsCycleAdaptation.py*.

Quadro 2: Execução *tlsCycleAdaptation.py*

```
python tlsCycleAdaptation.py -n <arquivo>.net.xml -r <arquivo>.odtrips.rou.xml --min-green  
--yellow-time -o <arquivo>.add.xml
```

Outro *script* do SUMO, *tlsCoordinator.py*, cria *offsets* nos semáforos com objetivo de criar "ondas verdes". Os arquivos de entrada são a rede *.net.xml*, o arquivo de rotas *.rou.xml* e pode ser adicionado também um arquivo de programação semaforica *.add.xml*. O *output* do *script* também é um arquivo com extensão *.add.xml*. O Quadro 3 apresenta como rodar o *script*.

Quadro 3: Execução *tlsCoordinator.py*

```
python tlsCoordinator.py -n <arquivo>.net.xml -r <arquivo>.rou.xml -o <arquivo>.add.xml
```

3.2.3 Veículos

Os veículos utilizados na simulação devem ser definidos em um arquivo adicional *.add.xml*. Para simulação em nível mesoscópico os fatores que afetam a simulação são:

1. *vClass*: é a classe do veículo. É utilizado para o veículo não acessar *edges* que não tem permissão.
2. *length*: é o comprimento em metros do veículo;
3. *minGap*: distância mínima em metros livre entre os veículos;
4. *maxSpeed*: é a velocidade máxima que o veículo consegue atingir;

5. *speedFactor*: é um fator multiplicador da velocidade máxima que o veículo irá utilizar em relação a velocidade permitida do *edge*;
6. *speedDev*: é o desvio de velocidade do *speedFactor*;
7. *accel,decel*: a aceleração e desaceleração do veículo em m/s^2 ;
8. *impatience*: é um valor de 0 a 1 que determina como os veículos em *edges* de prioridade inferior agem sobre veículos de *edges* de prioridade superior. Com 0, o fator é desligado, os veículos não realizam movimentos de entrada em vias que façam os outros desacelerarem. Com 1, o veículo entra no *edge* se houver espaço mesmo forçando veículos a desacelerarem bruscamente. O valor de *impatience* inicia em 0 e aumenta até o valor definido, conforme a indicação de tempo —*time-to-impatience* no arquivo *.sumocfg* (ver Seção 3.6).

O arquivo *.add.xml* de declaração de veículo está no Quadro 4.

Quadro 4: Definição do veículo

```
<additional>
  <vType id=" <identificador do tipo de veículo>" vClass= leght= minGap= maxSpeed=
  speedFactor= speedDev= accel= decel= impatience= />
</additional>
```

Outra forma de definição de veículo é inserir entre as chaves `<route>` `</route>` o *vtype* como no Quadro 4 nos arquivos gerados pelo *od2trips* (ver Seção 3.3.2) e *routes2trips* (ver Seção 3.4.2). Esse modo de definir é essencial para os modelos de alocação Incremental, SUE e UE, que não aceitam bem o arquivo com definição de veículos *.add.xml*. As duas formas de definição não podem ser usadas simultaneamente.

3.3 Modelagem da demanda

Nesta seção, são apresentados como definir as zonas de tráfego, como importar a matriz OD para o SUMO e como definir pares de viagem origem e destino com *edges* para cada veículo.

3.3.1 Definição das zonas de tráfego

No SUMO, é necessário criar um arquivo que define as zonas de tráfego e os *edges* contidos nessas zonas. Esse processo tem início com a utilização do importador de geometrias do SUMO, o *poli-convert*. O *poli-convert* é capaz de importar geometrias de diversas fontes para o SUMO. O arquivo

shapefile da zona de tráfego é o dado de entrada na linha de comando. A coluna com o nome identificador de cada zona deve ser fornecido para posterior correlacionamento com a matriz OD. É necessário informar o arquivo *net.xml* na linha de comando. O Quadro 5 apresenta a linha de comando.

Quadro 5: Função *polyconvert*

```
polyconvert -n <arquivo>.net.xml --shapefile <arquivo> --shapefile.id-column  
"<identificador da coluna>" -o <arquivo>.poi.xml
```

O resultado dessa linha de comando é um arquivo no formato *.poi.xml* com a geometria de cada zona no formato do SUMO.

O próximo passo consiste em gerar um arquivo que possui as zonas de tráfego e os *edges* presentes em cada uma. Para realizar esse procedimento é necessário o *edgesInDistricts.py*, o *script* do SUMO gera o arquivo com extensão *.taz.xml*. Como entrada é fornecido o arquivo da malha *.net.xml* e o arquivo gerado pelo *polyconvert* *.poi.xml*. O arquivo de saída, *.taz.xml* conta com todas as zonas e os respectivos *edges* pertencentes. O Quadro 6 apresenta a linha de comando no *Anaconda Prompt*.

Quadro 6: Função *edgesInDistricts.py*

```
python edgesInDistricts.py -n <arquivo>.net.xml -t <arquivo>.poi.xml -o <arquivo>.taz.xml
```

3.3.2 Definição da demanda

Os dados da matriz OD são transformados em viagens pelo SUMO. Para esse procedimento é utilizado o *od2trips*, um importador de dados de matriz OD de diversos formatos para um compatível com o SUMO. Podem ser colocados em um arquivo de configuração *.config.xml* as definições de entrada (*input*) e arquivo de saída (*output*) ou pode ser indicado por linha de comando. Os arquivos de entrada são o *.taz.xml* gerado na seção 3.3.1 e o arquivo com a matriz OD *.od*. O Quadro 7 apresenta o arquivo de configuração do *od2trips*.

Quadro 7: Configuração *od2trips*

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>

  <input>
    <taz-files value=" <arquivo>.taz.xml" />
    <od-matrix-files value=" <arquivo>.od" />
  </input>

  <output>
    <output-file value=" <arquivo>.odtrips.xml" />
  </output>

</configuration>
```

O arquivo .od deve ser formatado conforme o Quadro 8. No arquivo .od, são indicados os valores de início e fim do tempo da matriz OD em horas, por exemplo 0.00 e 1.00 definem uma matriz OD de uma hora. O *Factor* do arquivo indica a escala global para o número de veículos, se utilizado 1.00 todos os veículos da matriz OD serão utilizados, ou dada uma matriz OD em número de viagens pode ser utilizado o fator para converter em número de veículos. Por fim são listados os pares de origem e destino e a respectiva quantidade de veículos.

Quadro 8: Estrutura do arquivo com os dados de matriz OD

```
$OR;D2
* From-Time To-Time
<início> <fim>
* Factor
<fator de escala>
*
* some
* additional
* comments
<origem> <destido> <quantidade de veículos>
```

A execução do arquivo `.odtrips.xml` é finalizada por linha de comando conforme o Quadro 9. O tipo de veículo de cada viagem pode ser definido com a *flag* `--vtype` com o mesmo nome do identificador utilizado no arquivo `.add.xml` de definição de veículo.

Quadro 9: Execução do *od2trips*

```
od2trips -c <arquivo>.config.xml --vtype=" <identificador do tipo de veículo> "
```

O arquivo gerado `.odtrips.xml` contém um identificador da viagem, o tempo da partida, o *edge* de origem e o *edge* de destino, a zona de origem e a zona de destino e o tipo de veículo para cada viagem.

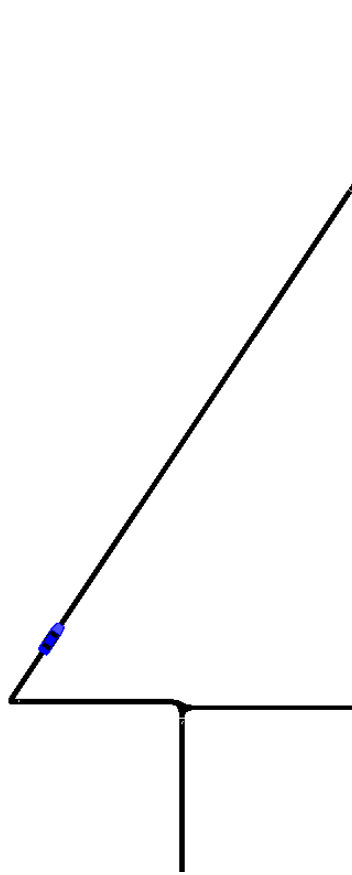
3.4 Alocação de tráfego

Os procedimentos para cada método de alocação geram um arquivo de rotas `.rou.xml` para ser utilizado ao rodar o SUMO.

3.4.1 Tudo ou Nada

O método de alocação Tudo ou Nada utiliza a ferramenta *duarouter* do SUMO. Essa ferramenta utiliza como padrão de impedância o tempo de viagem do *link*. No entanto, não está definido no arquivo da rede `.net.xml` o tempo de viagem de cada *link*. Para confirmar que o processo de cálculo de tempo de viagem de cada *link* ocorre internamente dentro do *duarouter*, foi realizado um simples teste. Uma rede foi criada no *netedit* como mostra a Figura 17.

Figura 17: Teste para confirmação da impedância do *duarouter*



O lado direito da rede apresenta menor distância, porém a velocidade permitida é baixa o que torna o tempo de viagem maior. O lado esquerdo possui distância maior, no entanto a velocidade permitida é maior o que leva a um tempo de viagem inferior ao do lado direito. O teste confirmou que o *duarouter* utiliza o tempo como impedância pois o veículo escolheu o trajeto pela esquerda.

Para gerar o arquivo *.rou.xml* pode ser criado um arquivo configuração *.duarcfg* ou utilizar a linha de comando. Os arquivos de entrada são o arquivo da rede *.net.xml* e o arquivo gerado pelo *od2trips* *.od2trips.xml*. Diversas opções de processamento podem ser definidas no arquivo *.duarcfg*. A opção *repair.from* e *repair.to* troca para o *edge* mais próximo a origem e o destino caso não haja rota válida entre origem e destino, caso também não haja rota válida a viagem é excluída. O Quadro 10 apresenta o arquivo *.duarcfg*.

Quadro 10: Configuração do *duarouter*

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>

  <input>
    <net-file value=" <arquivo>.net.xml" />
    <route-files value=" <arquivo>.odtrips.xml" />
  </input>

  <output>
    <output-file value=" <arquivo>.rou.xml" />
  </output>

  <processing>
    <repair.from value="true" />
    <repair.to value="true" />
  </processing>

</configuration>
```

Para executar o *duarouter* deve ser utilizada a linha de comando como mostra o Quadro 11.

Quadro 11: Execução do *duarouter*

```
duarouter -c <arquivo>.duarcfg
```

3.4.2 Equilíbrio do Usuário

O *script* do *python duaiterate.py* gera o arquivo de rotas *.rou.xml* com base no método do Equilíbrio do Usuário. Primeiramente, é necessário utilizar o *script route2trips.py* que transforma o arquivo de rotas gerado pelo *duarouter* em um arquivo de *trips*, como o gerado pelo *od2trips*. Esse processo de conversão nem sempre é necessário. É utilizado para caso um par OD não possua rota válida seja eliminado e não cause erro no *duaiterate.py*. O Quadro 12 apresenta como utilizar o *route2trips.py*, o arquivo de entrada é o de rotas *.rou.xml* e o *output* é um arquivo *.trips.xml*.

Quadro 12: Execução *route2trips.py*

```
python route2trips.py <arquivo>.odtrips.rou.xml ><arquivo>.trips.xml
```

Para executar o *duaiterate.py* devem ser fornecidos como entrada o arquivo de rede *.net.xml* e o arquivo *.trips.xml*. Pode ser definido o número máximo de iterações com *--last-step*, para que mesmo não convergindo para o equilíbrio o *script* pare. Arquivos adicionais *.add.xml* e outras *flags* também podem ser fornecidas para definir como a simulação irá acontecer. O Quadro 13 apresenta como executar o *duaiterate.py*.

Quadro 13: Execução *duaiterate.py*

```
python dualterate.py -n <arquivo>.net.xml -t <arquivo>.trips.xml --last-step "<número de iterações>"
```

3.4.3 Incremental

Para gerar o arquivo de rotas pelo método incremental é necessário utilizar a ferramenta *marouter*. O *marouter* roda a simulação de forma macroscópica, com o número de iterações definidas no arquivo de configuração. No entanto, não consta na documentação do SUMO sobre como é feita a divisão da demanda entre as iterações. Os arquivos de entrada necessários são o arquivo da rede *.net.xml*, o arquivo de zonas *.taz.xml* e o arquivo gerado pelo *od2trips* *.odtrips.xml*. O arquivo de configuração do Quadro 14 também contém a declaração do método *assignment-method*, o número de iterações desejadas e o número máximo de rotas alternativas para cada veículo *max-iterations*.

Quadro 14: Configuração Incremental

```
<?xml version="1.0"encoding="UTF-8"?>
<configuration>

  <input>
    <net-file value=" <arquivo>.net.xml"/>
    <additional value=" <arquivo>.xml"/>
    <routes value=" <arquivo>.odtrips.xml"/>
  </input>

  <output>
    <output-file value=" <arquivo>.rou.xml"/>
  </output>

  <processing>
    <assignment-method value="incremental"/>
    <max-iterations value = />
    <max-alternatives value = />
  </processing>

</configuration>
```

O marouter é executado por linha de comando como mostra o Quadro 15.

Quadro 15: Execução Incremental

```
marouter -c <arquivo>.config.xml
```

3.4.4 Variação do método Incremental

A variação do método Incremental utiliza o arquivo de rotas gerado pelo *duarouter*. Por meio de *flags* no arquivo de execução (seção 3.6) dentro de *processing* pode ser definido a percentagem de veículos que possuem um dispositivo para recalcular a rota. O tempo em que estes veículos recalculam a rota também é definido. As *flags* utilizadas são:

1. <device.rerouting.probability value=/>
2. <device.rerouting.period value=/>

Os veículos que estão inseridos na rede e possuem o dispositivo recalculam suas rotas a cada intervalo de tempo com base nos novos custos dos *links*. Os custos dos *links* são custos médios atualizados a cada intervalo.

3.4.5 Equilíbrio Estocástico do Usuário

Para gerar o arquivo de rotas *.rou.xml* pelo método do Equilíbrio Estocástico do Usuário (SUE) também é utilizada a ferramenta *marouter*. Os arquivos de entrada são os mesmos utilizados em 3.4.3. O Quadro 16 apresenta a configuração do método SUE. Em *assignment-method* é necessário declarar o método como SUE. Dentro da chave *processig* é possível escolher o algoritmo de roteamento em *route-choice-method* entre Logit ou Gawron.

Quadro 16: Configuração SUE

```
<?xml version="1.0"encoding="UTF-8"?>
<configuration>

  <input>
    <net-file value=" <arquivo>.net.xml"/>
    <additional value=" <arquivo>.taz.xml"/>
    <routes value=" <arquivo>.odtrips.xml"/>
  </input>

  <output>
    <output-file value=" <arquivo>.rou.xml"/>
  </output>

  <processing>
    <assignment-method value="SUE"/>
    <max-iterations value = />
    <max-alternative value = />
    <route-choice-method value = />
  </processing>

</configuration>
```

A execução do SUE se dá da mesma forma do método incremental presente no Quadro 15.

3.5 Modelagem da oferta de transporte público coletivo

Para inserir o transporte público no SUMO é necessário definir o veículo ônibus no arquivo adicional .add.xml. As linhas de ônibus são rotas que devem ser definidas por pontos de ônibus em que o veículo deve parar por *edges*. Detalhes como tempo em que o veículo deve ficar parado em cada ponto de ônibus podem ser especificados. O número de vezes e o tempo de ciclo de cada rota devem ser fornecidos. Após criar as rotas, são criados veículos do tipo ônibus com o identificador da rota desejada e o tempo de partida desse veículo. O Quadro 17 apresenta como deve ser configurado o arquivo com extensão .rou.xml de transporte público.

Quadro 17: Definição transporte público

```
<route id=" <nome da rota>" edges= repeat= cycleTime= >
  <stop busStop= duration= until= />
  .
  .
  .
  <stop busStop= duration= until= />
</route>

<vehicle id="bus" type="bus" line= route=" <nome da rota>" />
```

O modo de alocação da demanda em transporte público no SUMO é definindo fluxo de pessoas ou pessoas individuais. Cada pessoa deve ser definida com as especificações de qual linha de ônibus pegar, ponto de ônibus inicial e final. O arquivo de rotas de pessoas ou fluxo é com extensão .rou.xml.

3.6 Execução da simulação

Para rodar a simulação no SUMO é feito um arquivo de configuração com extensão .sumocfg. Os arquivos de entrada são o arquivo rede .net.xml, o arquivo de rotas .rou.xml e arquivos adicionais .add.xml. Neste arquivo é onde ocorre a definição de simulação mesoscópica com *mesosim*, por padrão o SUMO utiliza o modo microscópico. Entre as opções do modo mesoscópico, há a opção *meso-junction-control* que ativa os semáforos da simulação, por padrão eles estão desligados.

A simulação pode receber tempo de início e fim, caso não seja declarado a simulação começa no tempo 0 e finaliza quando todos os veículos deixarem a rede. Neste arquivo pode ser definido tempo de teletransporte (*teleport-time*). Caso o veículo fique com velocidade entre 0 e 0,1 m/s

pelo tempo definido ele será retirado e teletransportado para o próximo *edge* da sua rota. Caso não haja espaço o veículo aguarda fora da rede. Os veículos são teletransportados por três motivos no SUMO:

1. faixa errada (*wrong lane*): o veículo fica preso na faixa que não tem conexão com o próximo *edge* da sua rota. Esse problema é atribuído ao algoritmo *lane changing*, veículos são bloqueados e não conseguem trocar de faixa.
2. *yield*: o veículo está preso em um *edge* de prioridade inferior e não consegue espaço para acessar o *edge* de prioridade maior;
3. *jam*: o veículo está preso em um *edge* prioritário e não tem acesso ao próximo *edge* devido ao congestionamento.

No arquivo `.sumocfg` são especificados os arquivos *outputs* de relatórios sobre a simulação. Esses arquivos são gerados durante a simulação, fornecendo dados para as análises. O arquivo *tripinfo* apresenta detalhes de cada viagem realizada e o arquivo *summary* apresenta os dados por segundo de simulação como número de veículos que finalizaram suas rotas, que foram inseridos ou que estão rodando. Há diferentes arquivos *outputs* disponíveis para uso, porém não foram utilizados neste trabalho.

O Quadro 18 apresenta a configuração de execução do SUMO.

Quadro 18: Configuração de execução

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>

  <input>
    <net-file value=" <arquivo>.net.xml"/>
    <route-files value= " <arquivo>.rou.xml"/>
    <additional-files value= " <arquivo>.add.xml"/>
  </input>

  <time>
    <begin value=/>
    <end value=/>
  </time>

  <mesoscopic>
    <mesosim value=/>
    <meso-junction-control value=/>
  </mesoscopic>

  <processing>
    <time-to-teleport value=/>
  </processing>

  <output>
    <tripinfo-output value=" <arquivo>.xml"/>
    <summary value=" <arquivo>.xml"/>
  </output>

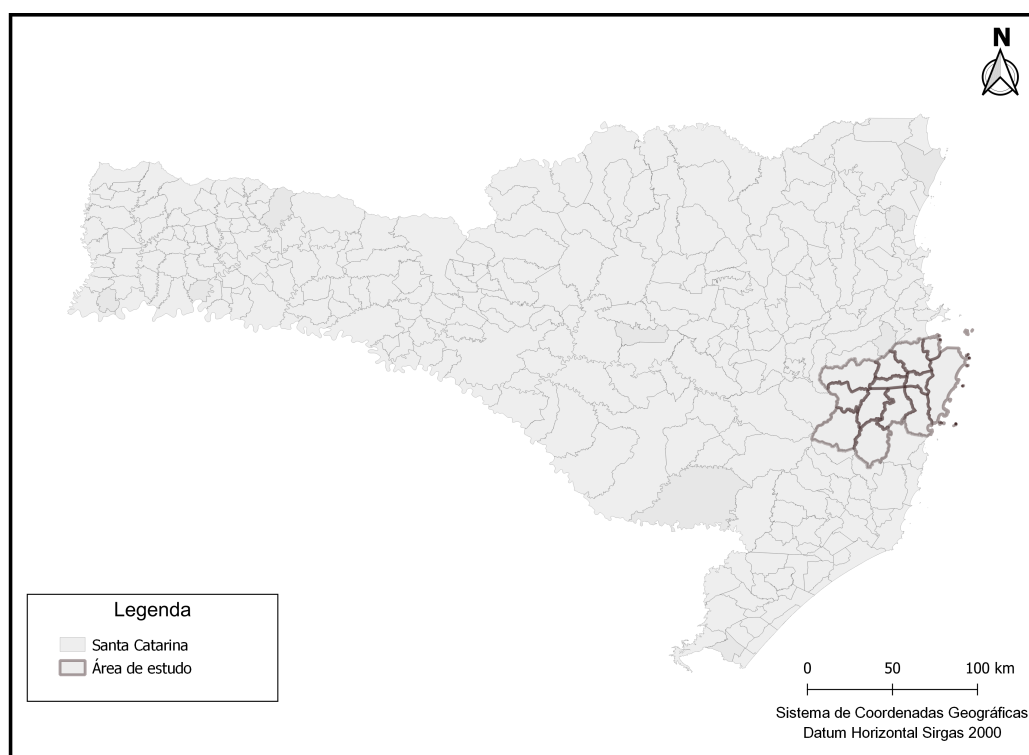
</configuration>
```

4 Área de estudo

Esta seção apresenta os dados utilizados neste trabalho, os quais são divididos entre dados de infraestrutura, composta pela rede viária da área de estudo, e dados de demanda, composta pela matriz OD.

Segundo o Fórum Nacional de Entidades Metropolitanas (FNEM)⁶ a região metropolitana da Grande Florianópolis é composta por nove municípios: Águas Mornas, Antônio Carlos, Biguaçu, Florianópolis, Governador Celso Ramos, Palhoça, Santo Amaro da Imperatriz, São José e São Pedro de Alcântara. A área de estudo também contempla mais quatro municípios que fazem parte da área de expansão metropolitana, segundo FNEM: Angelina, Anitápolis, Rancho Queimado e São Bonifácio. A área de estudo possui 1288,49 km² e está apresentada nas Figuras 18 e 19.

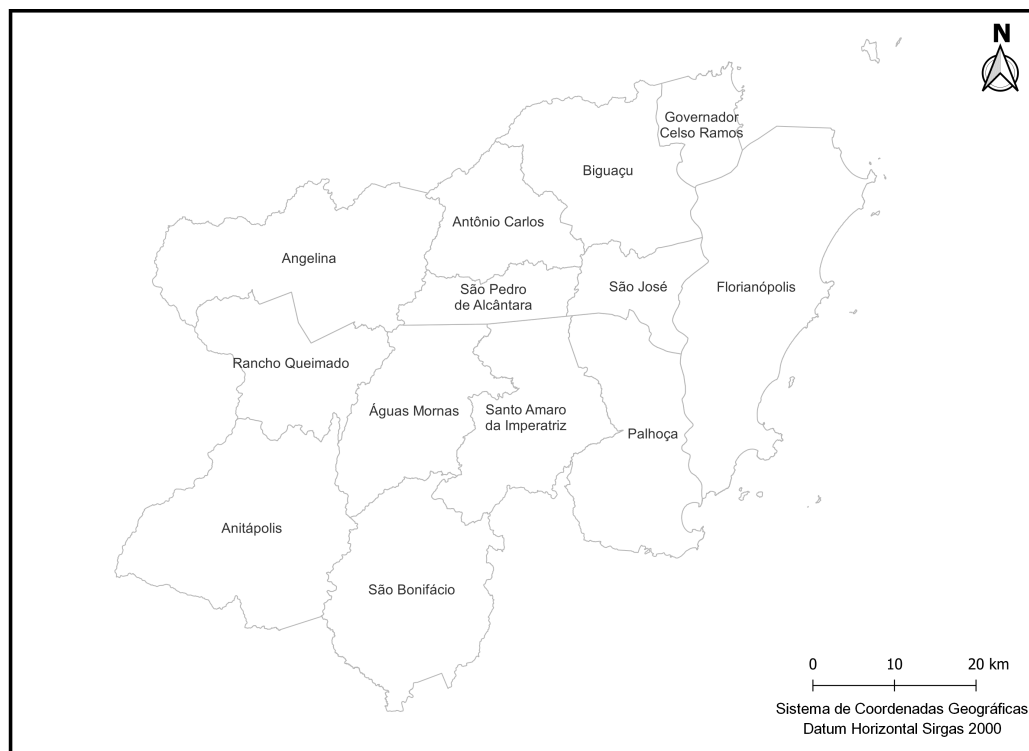
Figura 18: Localização da área de estudo



Fonte de dados: Malha Municipal (IBGE, 2020)

⁶ <<https://fnembrasil.org/regiao-metropolitana-de-florianopolis-sc/>> Acesso em 10/12/2021

Figura 19: Cidades da área de estudo



Fonte de dados: Malha Municipal (IBGE, 2020)

4.1 Zonas de tráfego e matriz OD

Os dados de demanda, matriz OD, foram obtidos no Plano de Mobilidade Urbana Sustentável da Grande Florianópolis (PLAMUS)⁷ resultante das pesquisas domiciliares de origem destino que abrange os treze municípios da área de estudo. A pesquisa de entrevista domiciliar foi realizada em 2014.

O PLAMUS disponibiliza duas opções de matriz OD para os cenários futuros, orientado e tendencial. O cenário orientado considera que o desenvolvimento urbano se concentrou no continente com a ocupação planejada da área entre a BR-101 e a Via de Contorno e que ocorreu a promoção de políticas públicas de desconcentração de atividades. O cenário com desenvolvimento urbano tendencial tem a projeção de que a área urbana continua a se expandir como ocorria em 2014, com as tendências imobiliárias existentes. No desenvolvimento tendencial, a relação continua sendo de cidade central, com todos os movimentos convergindo para o centro de Florianópolis na Ilha. Os dois cenários consideram como prioritária a implantação de corredores de transporte coletivo de média capacidade tendo os modos BRT, VLT ou Monotrilho como alternativas para avaliação.

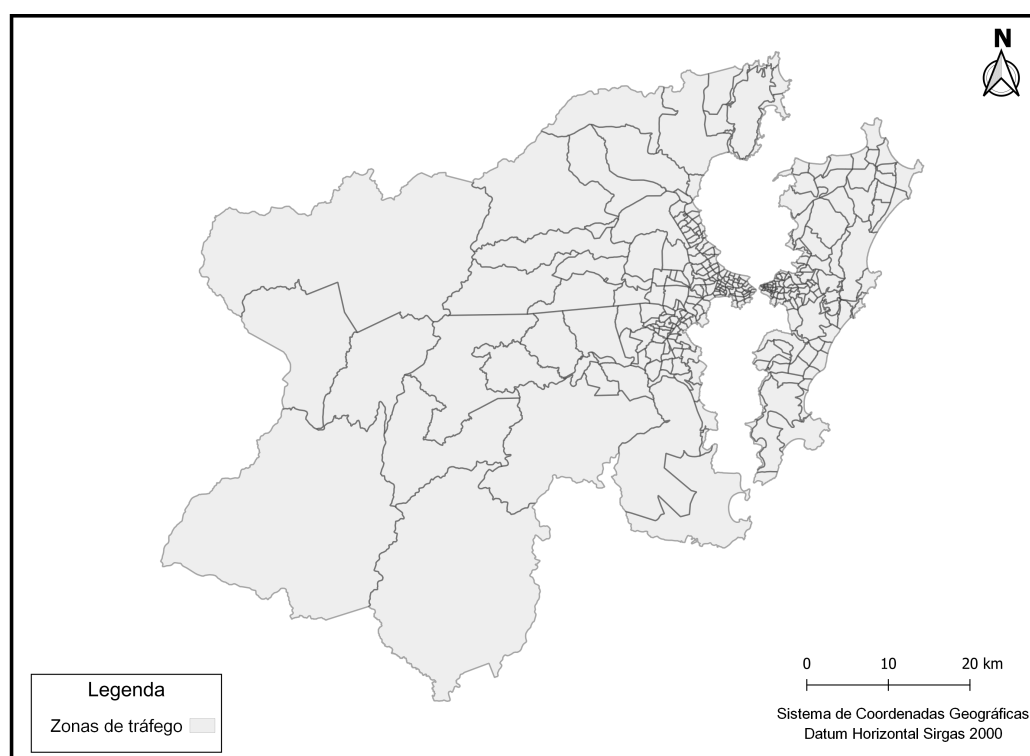
⁷ <<https://observatoriodamobilidadeurbana.ufsc.br/wp-content/uploads/2019/03/Informac%CC%A7a%CC%83o-Tendencial-v02.xlsx>> Acesso em 10/12/2021

O arquivo tendencial possui matriz OD com cenário base para 2014 e projeções para 2020, 2030 e 2040 com e sem BRT. O arquivo orientado possui projeções para 2020, 2030 e 2040 somente com BRT.

O arquivo de matriz OD utilizado neste trabalho é o tendencial sem BRT de 2020. A planilha disponibiliza para a hora de pico, 8 horas da manhã, a matriz OD para transporte público e para transporte privado. A matriz OD de transporte privado é fornecida em número de veículos, já a matriz de transporte público é fornecida em número de viagens.

A delimitação das zonas de tráfego está disponibilizada pelo PLAMUS⁸ por meio de *shapefile*. A área de estudo é dividida em 327 zonas de tráfego, Figura 20.

Figura 20: Zonas de tráfego da área de estudo



Fonte de dados: PLAMUS, 2014

4.2 Infraestrutura

O *OpenStreetMap* (OSM)⁹ é um projeto de mapeamento colaborativo livre. O OSM possui em sua base de dados a geometria das vias, localização de pontos de ônibus e trajeto de linhas de ônibus. No site do OSM é possível exportar em arquivo com extensão .osm os dados da área desejada.

⁸< <https://observatoriodamobilidadeurbana.ufsc.br/wp-content/uploads/2019/03/Zonas-de-Tra%CC%81fego.zip>> Acesso em 06/08/2021

⁹< <https://www.openstreetmap.org/>> Acesso em 10/12/2021

O OSM utiliza uma classificação própria para as vias com base em critérios estabelecidos. Na importação do arquivo .osm todas as classes são carregadas. As classes e as respectivas descrições de interesse para a simulação estão na Tabela 7.

Tabela 7: Classificação das vias para o OSM

| Classificação | Descrição |
|---------------------|---|
| <i>motorway</i> | Rodovia pavimentada duplicada, com separador físico entre as pistas em sentido contrário, sem cruzamentos em nível e pelo menos 10 km de extensão. |
| <i>trunk</i> | Rodovia pavimentada duplicada rural com cruzamentos em nível e pelo menos 10 km de extensão; Principal rota rodoviária pavimentada ligando 2 centros urbanos com mais de 200 mil habitantes, ou cidades-pólos de região geográfica intermediária; Via urbana pavimentada ligando rodovias <i>trunk</i> ou <i>motorway</i> entre si; Via estrutural urbana principal, em cidades com mais de 200 mil habitantes. |
| <i>primary</i> | Rodovia pavimentada que é rota ideal entre centros urbanos com mais de 20 mil habitantes, ou cidades-pólos de região geográfica imediata; Via urbana pavimentada ligando rodovias <i>primary</i> entre si ou com rodovias de nível superior; Via estrutural urbana, em cidades com mais de 20 mil habitantes (atravessa a cidade). |
| <i>secondary</i> | Rodovia/estrada que é rota ideal entre centros urbanos com mais de 2 mil habitantes ou sedes de municípios; Via urbana ligando rodovias/estradas <i>secondary</i> entre si ou com rodovias de nível superior; Via arterial urbana (ligando vários bairros). |
| <i>tertiary</i> | Rodovia/estrada que é rota ideal entre os demais núcleos urbanos; Via urbana ligando rodovias/estradas <i>tertiary</i> entre si ou com rodovias de nível superior; Via coletora (e/ou distribuidora) urbana (principais vias que conduzem às vias de maior nível). |
| <i>unclassified</i> | Via em zona não residencial urbana, que corresponde à classe "via local" no CTB; Demais vias públicas rurais. |
| <i>residential</i> | Via em zona residencial urbana, que corresponde à classe "via local" no CTB. |

Fonte: *OpenStreetMap*¹⁰

O OSM não possui os dados de programação semafórica. O arquivo importado do OSM contém:

1. os nós e suas conexões;
2. as classes das vias e a prioridade;

¹⁰<https://wiki.openstreetmap.org/wiki/Brazil/Classifica%C3%A7%C3%A3o_das_rodovias_do_Brasil>
Acesso em 12/01/2022

3. o limite de velocidade;
4. o número de faixas de acordo com a classificação ou por uma determinação do OSM;
5. a posição de todos os semáforos.

O PLAMUS disponibiliza dados de velocidade máxima permitida nos *links* em arquivo *shapefile*. No entanto, foi optado por utilizar os dados de velocidade máxima permitida disponibilizados pelo OSM pois seria necessário realizar compatibilização das informações dos *links* entre as duas fontes. Outro dados disponibilizado pelo PLAMUS são os tempos semafóricos por *links*, esse dado também não foi utilizado para elaboração dos modelos neste trabalho.

5 Aplicações na área de estudo

Esta seção apresenta os resultados ao aplicar o método na área de estudo. Foi optado por escolher o modelo mesoscópico de simulação pois é mais tolerante a erros de infraestrutura, roda mais rápido que o SUMO microscópico e fornece conteúdo adequado para análise final.

5.1 Modelagem da infraestrutura

Ao converter o arquivo .osm para .net.xml com o *netconvert*, Seção 3.2.1, foi utilizada a *flag remove-edges-by-type* para remover *edges* que não são de interesse para a simulação. Os *edges* removidos com base na classificação do OSM são: *highway track*, *highway service*, *highway path*, *highway cycleway*, *highway footway*, *highway steps* e *highway pedestrian*. Permaneceram *edges* do tipo *highway primary*, *highway secondary*, *highway tertiary*, *highway residential*, *highway trunk*, *highway unclassified* e *highway motorway*.

A rede de tráfego não recebeu nenhuma conferência de infraestrutura nos links e nos nós. Também não houve conferência quanto aos movimentos permitidos nos nós, a prioridade e a classificação das vias.

A Figura 21 apresenta a rede de tráfego resultante do processo de importação da rede no *netedit*.

Figura 21: Rede de tráfego da área de estudo



A Tabela 8 apresenta síntese dos dados referente a infraestrutura da rede.

Tabela 8: Síntese de dados de infraestrutura da rede de tráfego da área de estudo

| | |
|--|-----------|
| Total de nós | 43.778 |
| Total de <i>links</i> | 93.096 |
| Total comprimento de <i>links</i> (km) | 12.554,48 |
| Total comprimento de faixas (km) | 13.240,14 |

Para obter a quantidade de *edges* de cada classe que integra a rede de tráfego foi utilizado um *script* em *python* utilizando o módulo *ElementTree*¹¹. O Quadro 19 apresenta o *script* criado.

¹¹< <https://docs.python.org/3/library/xml.etree.elementtree.html> > Acesso em 06/03/2022

Quadro 19: *Script* para obter informação de quantidade de *edges* por classe de via

```
1 import xml.etree.ElementTree as ET
2 tree = ET.parse('<arquivo.net.xml')
3 root = tree.getroot()
4
5 primary=0
6 secondary=0
7 tertiary=0
8 residential=0
9 trunk=0
10 unclassified=0
11 motorway=0
12
13 for edge in root.findall(".//edge[@type='highway.primary']"):
14     primary+=1
15
16 for edge in root.findall(".//edge[@type='highway.secondary']"):
17     secondary+=1
18
19 for edge in root.findall(".//edge[@type='highway.tertiary']"):
20     tertiary+=1
21
22 for edge in root.findall(".//edge[@type='highway.residential']"):
23     residential+=1
24
25 for edge in root.findall(".//edge[@type='highway.trunk']"):
26     trunk+=1
27
28 for edge in root.findall(".//edge[@type='highway.unclassified']"):
29     unclassified+=1
30
31 for edge in root.findall(".//edge[@type='highway.motorway']"):
32     motorway+=1
33
34 print(f'primary={primary}, secondary={secondary}, tertiary={tertiary}, residential={
    residential}, trunk={trunk}, unclassified={unclassified}, motorway={motorway}')
```

Utilizando o *script* é possível detalhar mais a rede de tráfego em estudo. A Tabela 9 apresenta os dados obtidos de número de *edges* por classe.

Tabela 9: Distribuição de *edges* por classes

| Classe | Número de <i>edges</i> | Porcentagem |
|---------------------|------------------------|-------------|
| <i>motorway</i> | 592 | 0,64% |
| <i>trunk</i> | 1310 | 1,41% |
| <i>primary</i> | 4632 | 4,98% |
| <i>secondary</i> | 12404 | 13,32% |
| <i>tertiary</i> | 8317 | 8,93% |
| <i>unclassified</i> | 9121 | 9,80% |
| <i>residential</i> | 56720 | 60,93% |

Há o predomínio de vias residenciais que são de pouca mobilidade e de grande acessibilidade na área de estudo. A composição de uma área urbana, geralmente, possui a configuração de maior quantidade de vias urbanas de maior acessibilidade e quantidade inferior de vias de alta mobilidade.

No arquivo de rede *.net.xml*, a velocidade dos *lanes* com a classificação do OSM dos respectivos *edges* não seguiram padrão de velocidade permitida de acordo com sua classe. Para realizar ajuste geral no arquivo de rede *.net.xml* foi criado um *script* em *python* utilizando, novamente, o módulo *ElementTree* para edição de arquivo *.xml*. A classificação do OSM foi correlacionada com a classificação do CTB conforme a Tabela 10 para definir as velocidades para cada classe.

Tabela 10: Correlação de classificações

| Classificação OSM | Classificação CTB | Velocidade (km/h) | Velocidade (m/s) |
|---------------------|------------------------|-------------------|------------------|
| <i>motorway</i> | Rodovia | 110 | 30,55 |
| <i>trunk</i> | Via de trânsito rápido | 80 | 22,22 |
| <i>primary</i> | Via de trânsito rápido | 80 | 22,22 |
| <i>secondary</i> | Arterial | 60 | 16,66 |
| <i>tertiary</i> | Coletora | 40 | 11,11 |
| <i>unclassified</i> | Local | 30 | 8,33 |
| <i>residential</i> | Local | 30 | 8,33 |

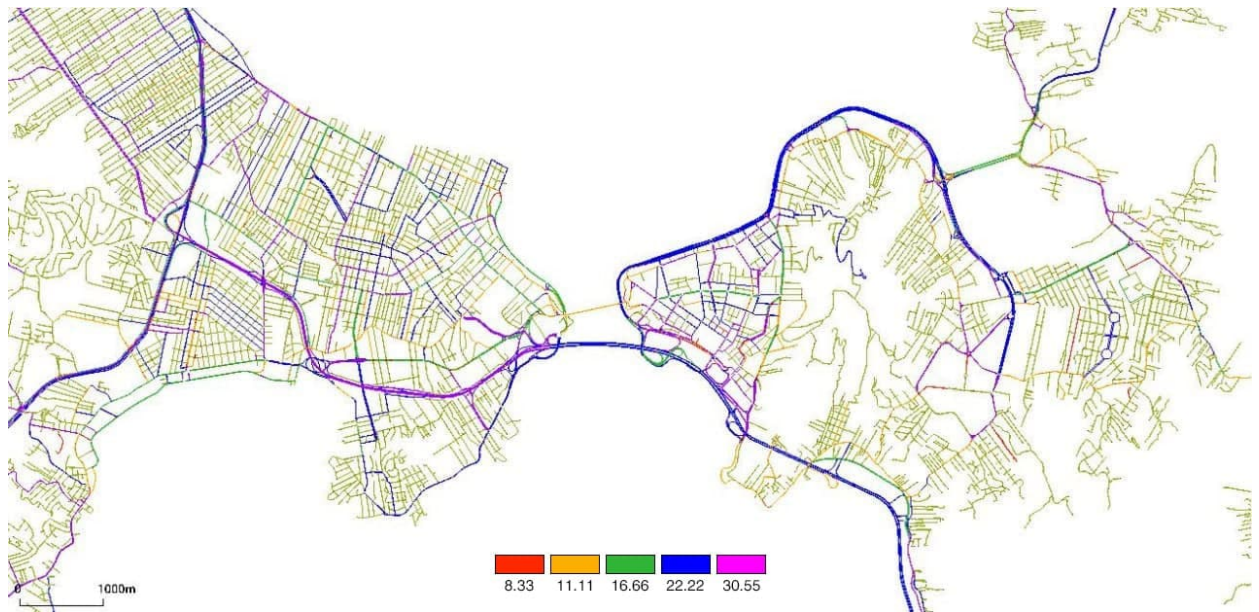
O Quadro 20 apresenta o *script* utilizado para alterar as velocidades. O arquivo de entrada é o da rede *.net.xml* e o de saída é o arquivo de rede com as velocidades alteradas.

Quadro 20: *Script* para ajuste de velocidade

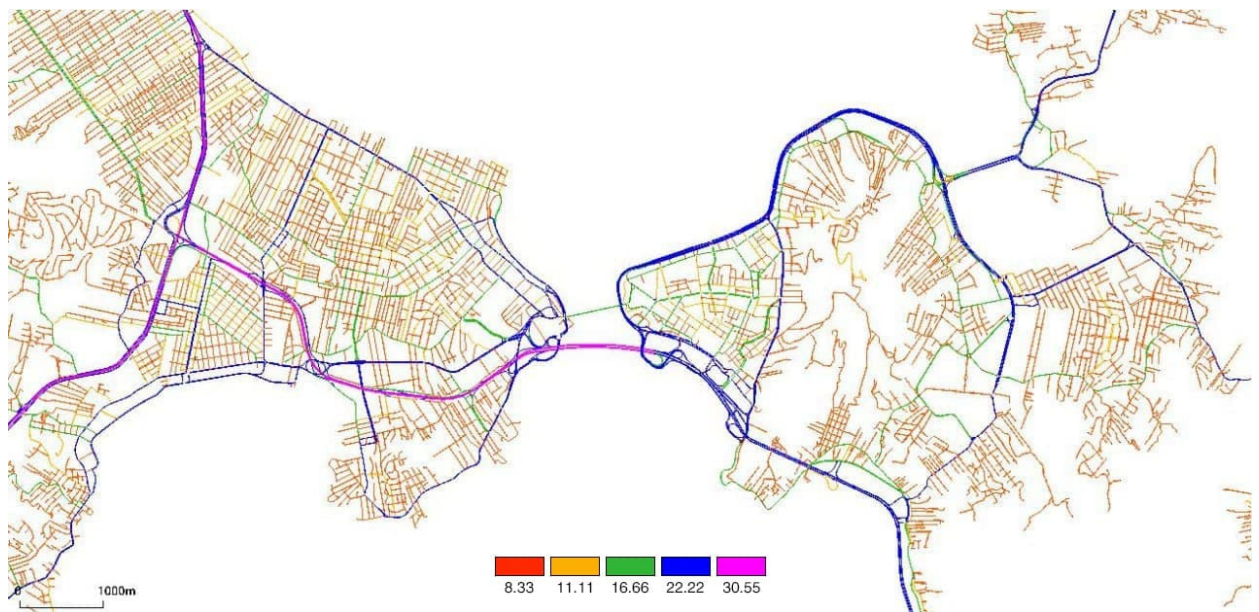
```
1 import xml.etree.ElementTree as ET
2 tree = ET.parse('<arquivo>.net.xml')
3 root = tree.getroot()
4
5 for lane in root.findall(".//edge[@type='highway.primary']/lane"):
6     lane.set('speed','22.22')
7
8 for lane in root.findall(".//edge[@type='highway.secondary']/lane"):
9     lane.set('speed','16.66')
10
11 for lane in root.findall(".//edge[@type='highway.tertiary']/lane"):
12     lane.set('speed','11.11')
13
14 for lane in root.findall(".//edge[@type='highway.residential']/lane"):
15     lane.set('speed','8.33')
16
17 for lane in root.findall(".//edge[@type='highway.trunk']/lane"):
18     lane.set('speed','22.22')
19
20 for lane in root.findall(".//edge[@type='highway.unclassified']/lane"):
21     lane.set('speed','11.11')
22
23 for lane in root.findall(".//edge[@type='highway.motorway']/lane"):
24     lane.set('speed','30.55')
25
26 tree.write('<arquivo>.net.xml')
27 print('Success.')
```

A Figura 22 apresenta parte da rede antes e após o ajuste das velocidades. É possível notar que vias locais e coletoras estavam com velocidade acima do padrão o que levaria a atração de rotas.

Figura 22: Comparação de velocidades permitidas antes e após ajuste



(a) Antes do ajuste



(b) Após ajuste

5.2 Transporte público

Para a simulação rodar com o transporte público são necessárias as linhas dos ônibus da área de estudo, com dados de frequência e rota. Os dados importados pelo OSM de localização de pontos de ônibus e de linhas de ônibus estão incompletos e os dados de frequência não são fornecidos.

Portanto seria necessário realizar um levantamento de localização de pontos de ônibus, e inseri-los no *netedit*. Outro levantamento necessário seria nos sites de transporte público de cada região a oferta de ônibus e o levantamento da rota associando a *edges* e a pontos de ônibus da rede. Após esses levantamentos é possível definir fluxos de pessoas com origem e destino da matriz OD de transporte público do PLAMUS e alocar no transporte público.

A alocação de fluxos de pessoas em redes de grande porte não foi encontrada em nenhuma referência. O transporte público não foi utilizado neste trabalho pois os levantamentos de dados de linhas, pontos de ônibus e associação das informações com a codificação dos *edges* da rede despenderiam um tempo não disponível para a realização deste trabalho.

5.3 Definição de veículo

A metodologia apresentada na Seção 3.2.3 foi utilizada para definição do veículo. As características do veículo utilizado para a simulação estão na Tabela 11. O parâmetro *speedFactor* foi estimado, considerando que os veículos ultrapassam a velocidade permitida dos *edges* em 20%, e o *speedDev* igual a 0,3 para que os veículos possam variar a velocidade em até 30% abaixo da velocidade máxima do *edge* com o *speedFactor*, estes valores foram determinados de forma arbitrária. O valor de *minGap* deve ser um intervalo seguro entre os veículos, geralmente 3 segundos, variando de acordo com a velocidade permitida nas vias, no entanto o *minGap* utilizado no SUMO é uma distância fixa para todos os veículos do mesmo tipo. O valor de *minGap* adotado foi de 2,5 m. Estes valores foram também determinados de forma arbitrária. Para os valores de aceleração, desaceleração, velocidade máxima e comprimento foram adotados os valores sugeridos pelo SUMO para um carro de passeio. A classe de veículos declarada é do tipo *passenger*, que seria a classe referente a veículos de passeio. O *impatience* foi definido com o máximo, 1.

Tabela 11: Definição do veículo

| Característica | Valor |
|--------------------------|-----------|
| <i>vClass</i> | passenger |
| <i>length</i> (m) | 5 |
| <i>minGap</i> (m) | 2,5 |
| <i>maxSpeed</i> (km/h) | 200 |
| <i>speedFactor</i> | 1,2 |
| <i>speedDev</i> | 0,3 |
| <i>impatience</i> | 1 |
| <i>accel</i> (m/s^2) | 2,6 |
| <i>decel</i> (m/s^2) | 4,5 |

5.4 Demanda

A matriz OD obtida no PLAMUS para transporte privado para o ano de 2020 possui 88.640 veículos, referente às 8 horas da manhã de um dia de semana. As viagens são distribuídas pelas 327 zonas de tráfego. O procedimento é descrito na Seção 3.3.

5.5 Escolha do método de alocação

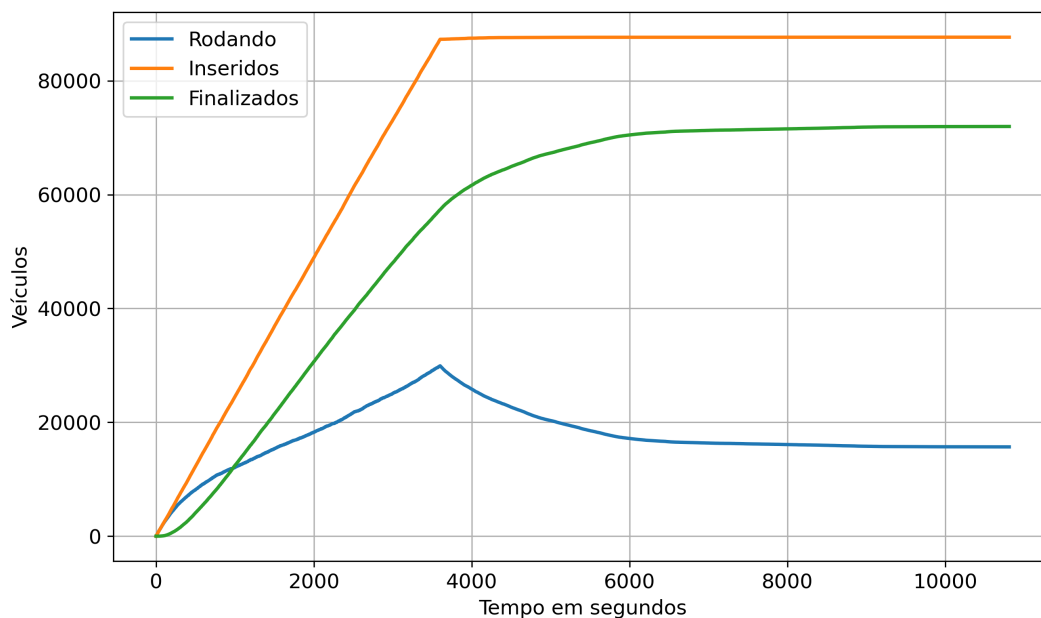
Os modelos de alocação apresentados na Seção 3.4 foram aplicados em simulações com objetivo de analisar o comportamento de cada método para a área de estudo. Os métodos foram testados com diferentes tempos de teletransporte para verificar o comportamento e então a necessidade e o tempo a ser empregado.

Todas as simulações realizadas foram rodadas, arbitrariamente, até o tempo de 10.800 segundos (3 horas). Esse tempo foi definido para que as simulações rodassem todas o mesmo período, padronizadas, e finalizassem de forma mais rápida. Todos os semáforos foram desligados durante as aplicações dos métodos. Para todas as simulações o tempo de *impatience* foi o *default* do SUMO, 300 segundos.

5.5.1 Tudo ou nada

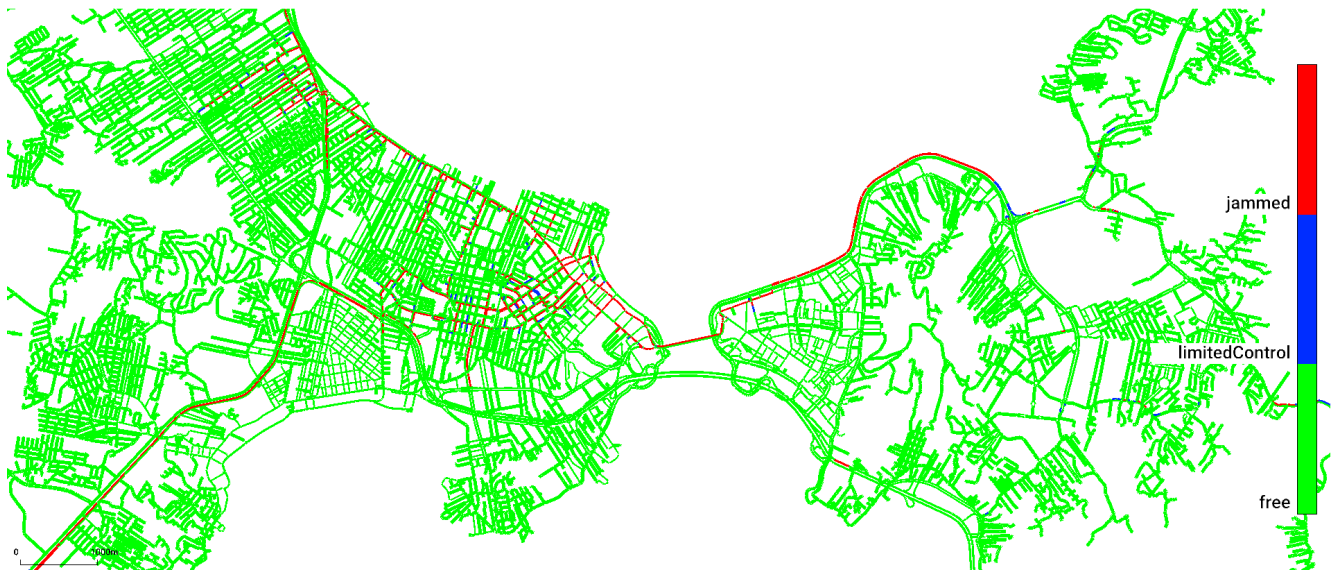
O método Tudo ou Nada foi programado para utilizar, primeiramente, tempo de teletransporte nulo. A Figura 23 apresenta os dados da simulação. Para o tempo de 3600 segundos a simulação atinge o máximo de veículos rodando, 29.924.

Figura 23: Gráfico do método de alocação Tudo ou Nada sem teletransporte



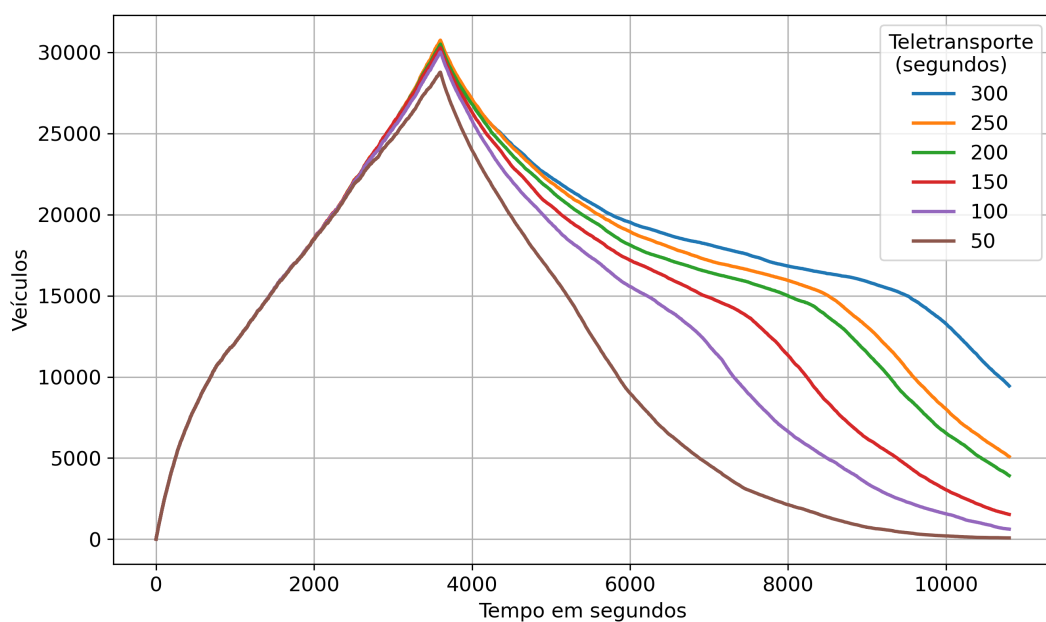
Depois do pico há uma estabilização no número de veículos rodando o que representa neste caso que houve trancamento da rede, pois há estabilização também nas curvas de veículos inseridos e finalizados na rede por um longo período de tempo. O trancamento da rede pode ter sido ocasionado por *yield*, *jam* ou faixa errada pois não há tempo de teletransporte para efetuar correção. Foi verificado na animação da simulação para o tempo de 6000 segundos o estado da rede, como mostra a Figura 24. É possível notar diversos pontos de congestionamento.

Figura 24: Edges com congestionamento na rede, método de alocação aplicado: Tudo ou Nada



Foi aplicado novamente o método Tudo ou Nada com tempo de teletransporte. Os tempos utilizados foram 300, 250, 200, 100, 150 e 50 segundos. A Figura 25 apresenta os dados de veículos rodando para os diferentes tempos de teletransporte. É possível notar que para 150, 100 e 50 segundos o número de viagens finalizadas aumenta de forma considerável.

Figura 25: Gráfico do método de alocação Tudo ou Nada com teletransporte



A Tabela 12 apresenta o número de teletransportes, veículos inseridos e finalizados para os diferentes tempos de teletransporte. A taxa de teletransporte em relação aos veículos inseridos é alta, abaixo de 250 segundos de tempo de teletransporte o valor passa de 5%.

Tabela 12: Dados da simulação utilizando o método Tudo ou Nada com diferentes tempos de teletransporte

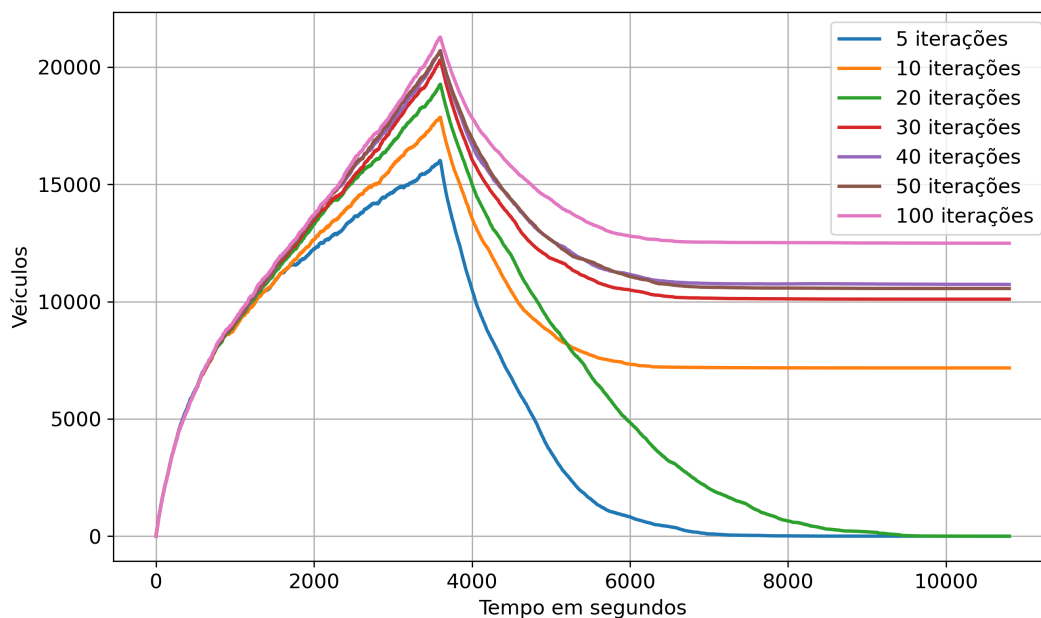
| Tempo de teletransporte (segundos) | Número de teletransportes | Veículos inseridos | Veículos finalizados | Porcentagem de teletransportes |
|------------------------------------|---------------------------|--------------------|----------------------|--------------------------------|
| 300 | 4.233 | 88.252 | 78.809 | 4,8% |
| 250 | 4.382 | 88.138 | 83.118 | 5,0% |
| 200 | 4.781 | 88.343 | 84.424 | 5,4% |
| 150 | 5.168 | 88.344 | 86.818 | 5,8% |
| 100 | 5.865 | 88.344 | 87.722 | 6,6% |
| 50 | 7.546 | 88.344 | 88.266 | 8,5% |

O método Tudo ou Nada não se apresentou adequado para a simulação. Os veículos concentrados na rota de menor custo geraram trancamento irreversível da rede. Valores de tempo baixo de teletransporte tornam a simulação não representativa, assim como altos índices de número de teletransportes. Em nenhuma das tentativas de utilização do método Tudo ou Nada todos os veículos foram inseridos.

5.5.2 Incremental

O método Incremental foi programado para gerar o arquivo de rotas utilizando o valor de máximo de alternativas por veículo igual a 5. O número de iterações utilizadas foi 10, 20, 30, 40, 50 e 100. Inicialmente, não foi aplicado tempo de teletransporte. A Figura 26 apresenta os veículos rodando para os diferentes números de iterações.

Figura 26: Gráfico da simulação com método Incremental sem teletransporte



Com 5 e com 20 iterações todos os veículos são inseridos e finalizados na rede. Com 5 iterações o pico de veículos rodando é inferior ao de 20 iterações, a quantidade de veículos ao mesmo tempo na rede é inferior. É possível notar que quanto mais iterações são realizadas o pico de veículos rodando aumenta. Mais iterações resulta em mais veículos utilizando rotas secundárias e o tráfego de veículo é mais espalhado pela rede. Foi concluído que para esta rede de tráfego erros de *jam* e *yield* são mais fáceis de acontecer com o espalhamento do tráfego, o que ocasiona o trancamento da rede. A Tabela 13 apresenta os valores de veículos inseridos e finalizados para os respectivos números de iterações.

Tabela 13: Dados da simulação utilizando o método Incremental sem teletransporte

| Número de iterações | Veículos inseridos | Veículos finalizados |
|---------------------|--------------------|----------------------|
| 5 | 88.640 | 88.640 |
| 10 | 87.395 | 80.222 |
| 20 | 88.640 | 88.640 |
| 30 | 85.956 | 75.851 |
| 40 | 86.101 | 75.366 |
| 50 | 86.010 | 75.450 |
| 100 | 85.818 | 73.327 |

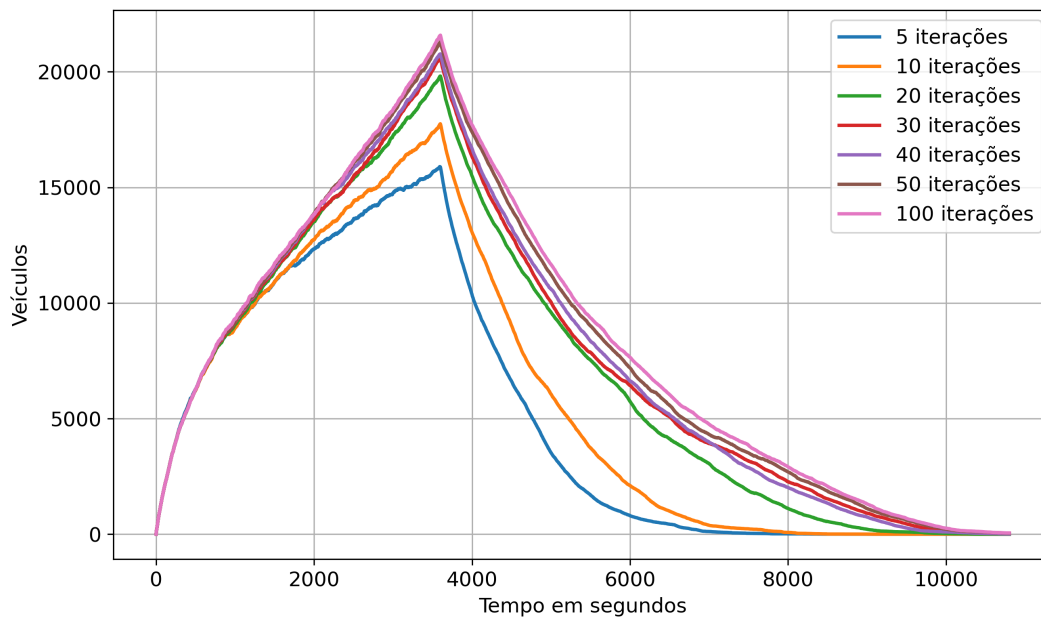
O método foi aplicado novamente com tempo de teletransporte de 300 segundos com o objetivo de verificar o desempenho das simulações, principalmente, aquelas que apresentaram trancamento da rede. Para todas as simulações 100% dos veículos foram inseridos na rede. Os dados de veículos com viagens finalizadas e o número de teletransportes está na Tabela 14.

Tabela 14: Dados da simulação utilizando o método Incremental com teletransporte

| Número de iterações | Veículos finalizados | Número de teletransportes | Porcentagem de teletransportes |
|---------------------|----------------------|---------------------------|--------------------------------|
| 5 | 88.640 | 2 | 0,002% |
| 10 | 88.640 | 93 | 0,105% |
| 20 | 88.640 | 289 | 0,326% |
| 30 | 88.614 | 357 | 0,403% |
| 40 | 88.630 | 324 | 0,366% |
| 50 | 88.595 | 377 | 0,425% |
| 100 | 88.586 | 374 | 0,422% |

É possível notar que o total de teletransportes que acontecem utilizando o método Incremental é consideravelmente inferior aos obtidos com o método Tudo ou Nada. O gráfico das simulações com diferentes números de iterações para o método incremental com tempo de teletransporte está na Figura 27. A simulação com 5 iterações continua apresentando melhor desempenho que as demais, há menor número de veículos rodando ao mesmo tempo.

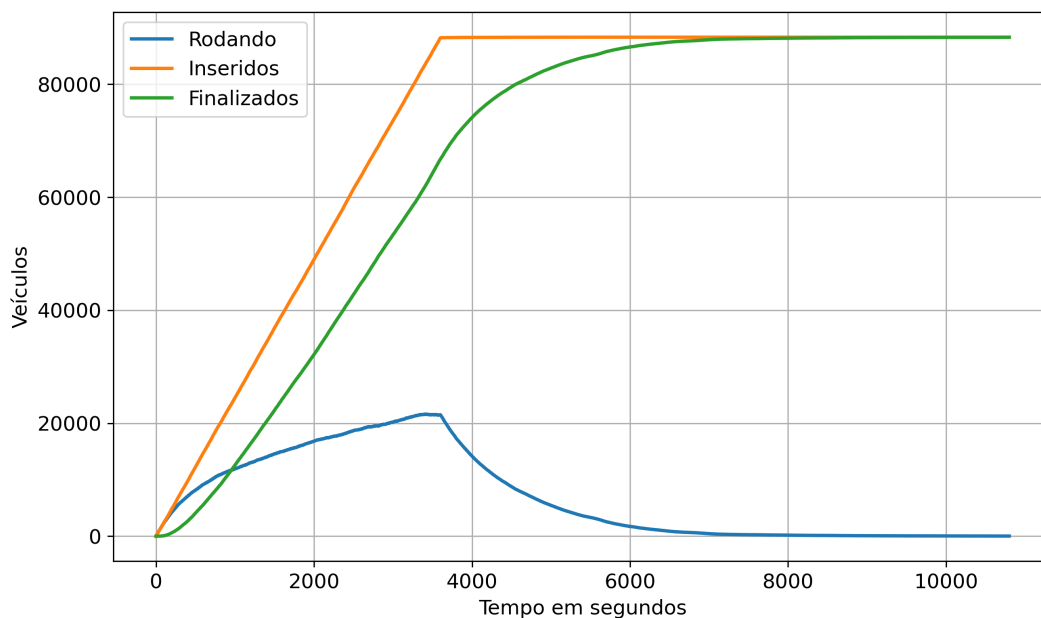
Figura 27: Gráfico da simulação com método de alocação Incremental com teletransporte



5.5.3 Variação do método Incremental

A variação do método Incremental utilizou 70% dos veículos com o dispositivo de alterar rotas, e o intervalo de 300 segundos (5 min) para analisar a troca de rota, do mesmo modo que foi utilizado na simulação LuST. O resultado da simulação está na Figura 28. Foram inseridos 88.344 veículos e 88.323 veículos finalizaram suas viagens.

Figura 28: Gráfico da simulação com método de alocação Variação Incremental



A Tabela 15 apresenta a quantidade de veículos que recalcularam suas rotas e a frequência.

Tabela 15: Dados de veículos que recalcularam rotas com o método de alocação Variação Incremental

| Número de vezes que veículo alterou a rota | Veículos | Porcentagem |
|--|----------|-------------|
| Não trocaram de rota | 51.216 | 57,97% |
| 1 a 5 vezes | 32.080 | 36,31% |
| 6 a 10 vezes | 4.217 | 4,77% |
| 11 a 15 vezes | 737 | 0,83% |
| 16 a 20 vezes | 66 | 0,07% |
| 21 a 25 vezes | 7 | 0,01% |

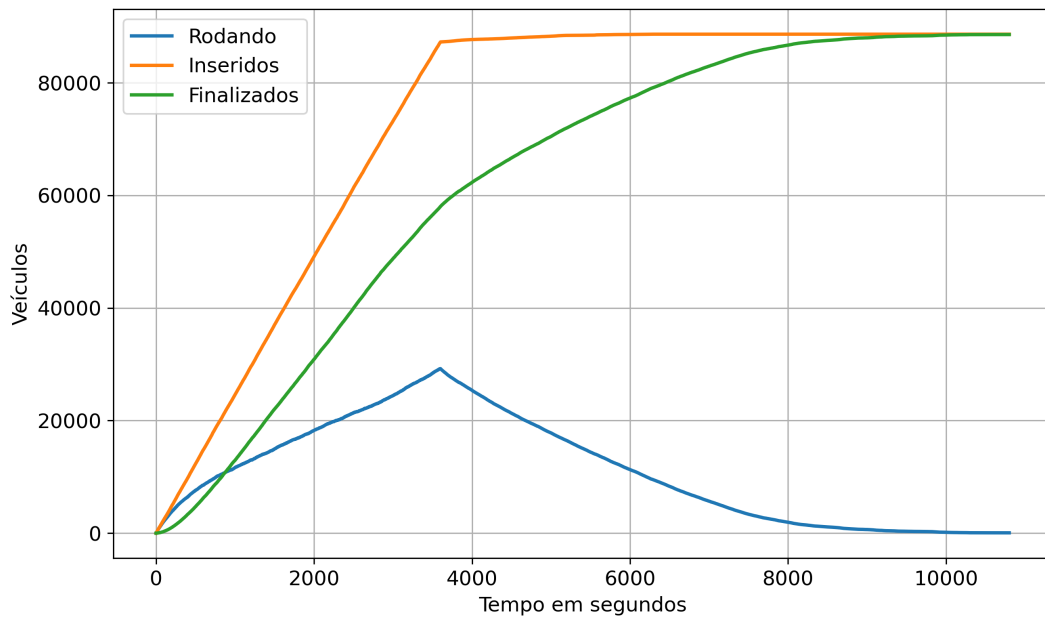
A simulação foi programada novamente adotando tempo de teletransporte de 300 segundos. Ocorreram apenas 5 teletransportes e não houve variação no número de rotas recalculadas.

5.5.4 SUE

O método SUE foi aplicado com tempo de teletransporte de 300 segundos para o algoritmo Logit. O método foi utilizado com apenas 10 iterações devido ao tempo de processamento exigido. A Figura

29 apresenta o gráfico obtido ao aplicar o método. O método não atinge o equilíbrio, é necessário maior número de iterações para que se possa ter a chance de alcançá-lo. Foram inseridos 88.640 veículos e 88.577 veículos finalizaram suas viagens. Foram registrados 221 teletransportes.

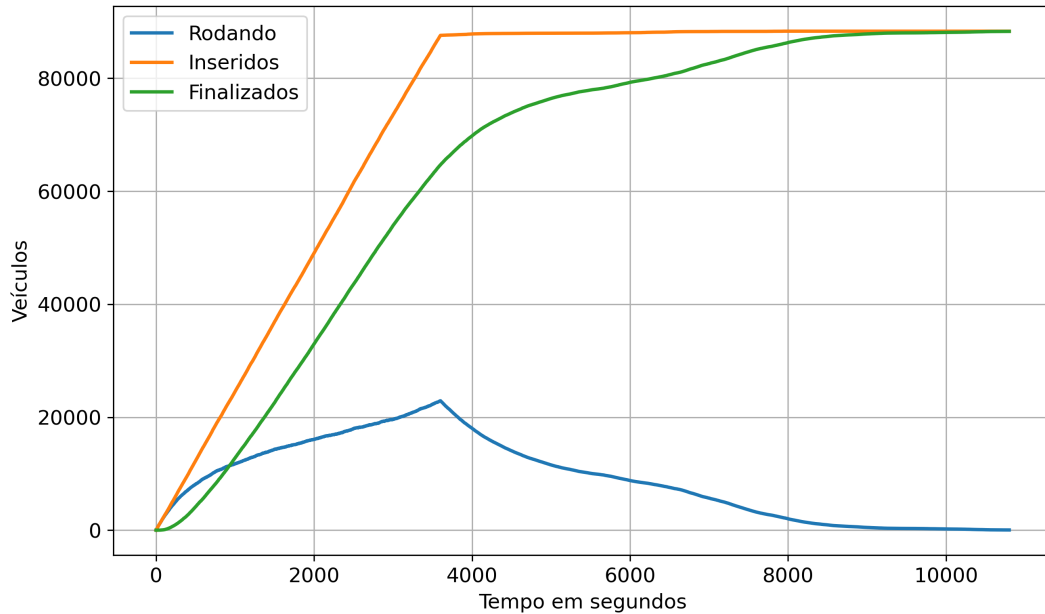
Figura 29: Gráfico da simulação com o método de alocação SUE



5.5.5 UE

O método UE foi utilizado com número de iterações definida. Para executar 20 iterações foi preciso 13 horas e 40 min de processamento. O UE rodou com 300 segundos para tempo de teletransporte. O método se torna inviável para simulações de grandes áreas pois são necessárias mais iterações para chegar ao equilíbrio. A Figura 30 apresenta os dados da simulação com o UE. Para essa simulação 88.344 veículos foram inseridos, 88.307 chegaram e ocorreram 823 teletransportes.

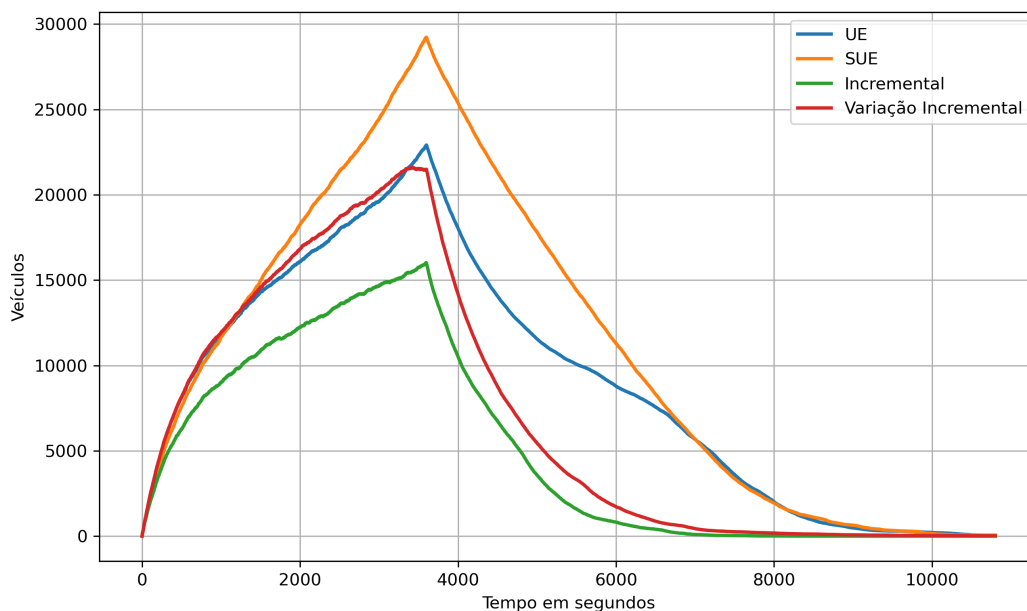
Figura 30: Gráfico da simulação com método de alocação UE



5.5.6 Comparação dos modelos

Os dados de veículos rodando das simulações Incremental com 5 iterações, variação Incremental, SUE e UE foram colocados no mesmo gráfico, todos com tempo de teletransporte de 300 segundos, com o intuito de verificar a melhor distribuição de veículos. A Figura 31 mostra o gráfico de comparação de veículos rodando dos modelos. O método incremental possui o menor pico de veículos rodando ao mesmo tempo, e durante todo o tempo de simulação. É possível concluir que o tempo geral de viagens do método Incremental, neste caso, é inferior aos outros modelos.

Figura 31: Gráfico de comparação: UE, SUE, Incremental e Variação Incremental



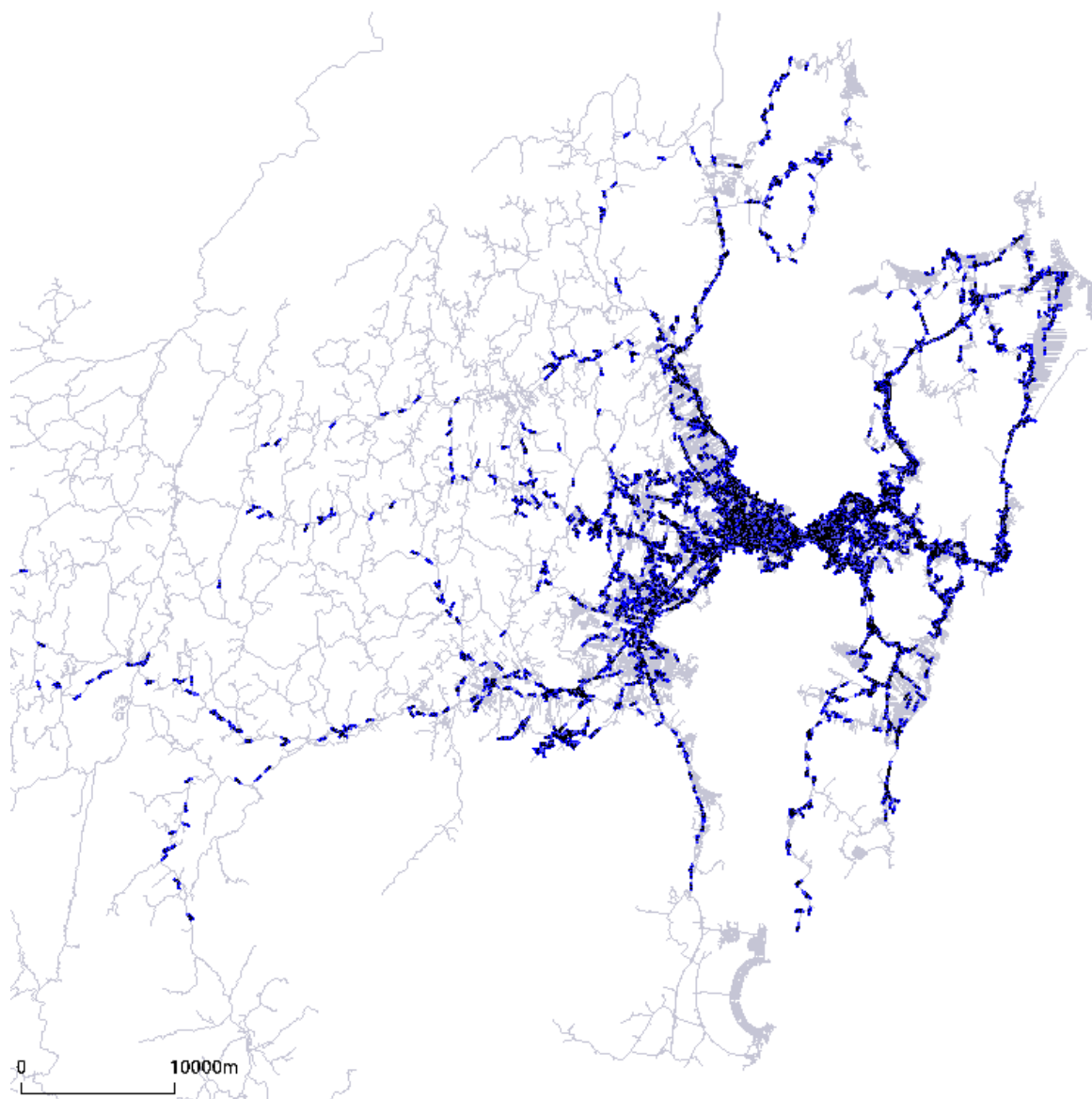
O método Incremental com 5 iterações foi o escolhido como o método utilizado para dar prosseguimento a análise da simulação.

5.6 Tempo de *warm-up*

Não consta na documentação do SUMO procedimento de *warm-up*. Com objetivo de aplicar o método de tempo de *warm-up* de duas vezes o tempo da maior viagem foi procurado no arquivo *tripinfo* a viagem mais longa da simulação Incremental com 5 iterações. Essa viagem possui 97,8 km que vai de uma zona localizada em Governador Celso Ramos para uma zona localizada em São Bonifácio. A rota dessa viagem foi isolada em um arquivo *.rou.xml* e rodada na rede em condição de fluxo livre para retirar o tempo de viagem. O tempo dessa viagem durou 4.262 segundos. Utilizando o método proposto de tempo de *warm-up* igual a duas vezes a duração da viagem de maior distância o período seria igual a 8.524 segundos. Esse valor é inviável para a simulação pois a proporção de viagens com durações inferiores é maior. 80% das viagens já finalizaram no tempo 3500 segundos.

O tempo de *warm-up* da rede foi definido, arbitrariamente, com 960 segundos (15 minutos). Neste intervalo, um quarto da demanda foi inserida na rede. Visualmente, a rede no tempo de 960 segundos está bem preenchida como apresenta a Figura 32, onde as vias estão representadas em cinza e os veículos em azul.

Figura 32: Simulação no tempo de *warm-up* (960 segundos)

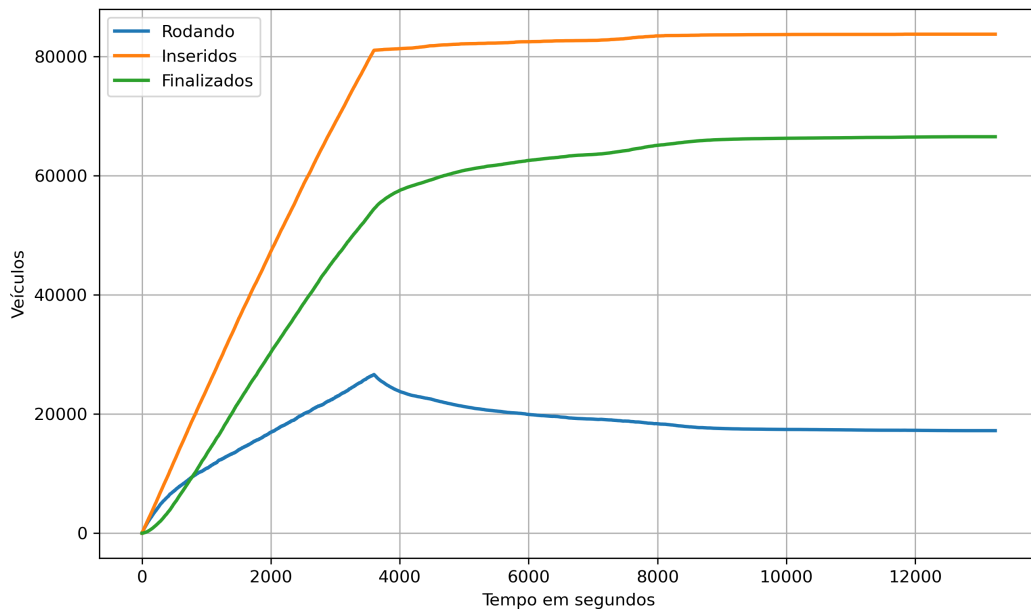


5.7 Simulação com semáforos ativados

A rede possui 256 semáforos com base no dado do OSM. Os semáforos foram programados para utilizar o *tlsCycleAdaptation.py* para modificar o tempo de ciclo. Foi definido, arbitrariamente, tempo de verde mínimo de 27 segundos e tempo de amarelo de 3 segundos. O tempo de *warm-up* de 960 segundos foi descontado, foi definido o contador de tempo com o tempo de *warm-up*. A rota utilizada foi do Tudo ou Nada pois o *script* não aceita a variabilidade de rotas do método Incremental.

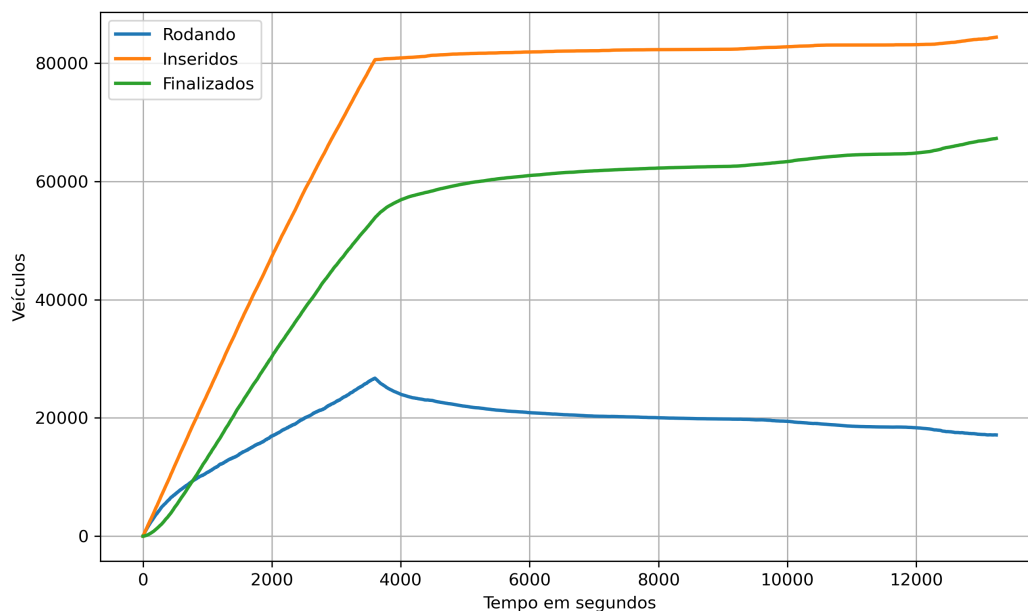
O resultado obtido rodando a simulação com o método Incremental com 5 iterações sem tempo de teletransporte e semáforos ativados é apresentado na Figura 33. A simulação finaliza com 83.753 veículos inseridos e 66.542 veículos com viagens finalizadas.

Figura 33: Gráfico da simulação com método de alocação Incremental com semáforos



A rede tranca devido a falta de coordenação semafórica entre os semáforos. Portando o *script tlsCoordinator.py* foi utilizado com o intuito de criar ondas verdes. A Figura 34 apresenta os resultados obtidos. Operando a simulação dessa forma 84.388 veículos foram inseridos e 67.267 finalizaram suas viagens. A aplicação do *tlsCoordinator.py* gerou melhora na simulação.

Figura 34: Gráfico simulação Incremental com semáforos e coordenação de fases



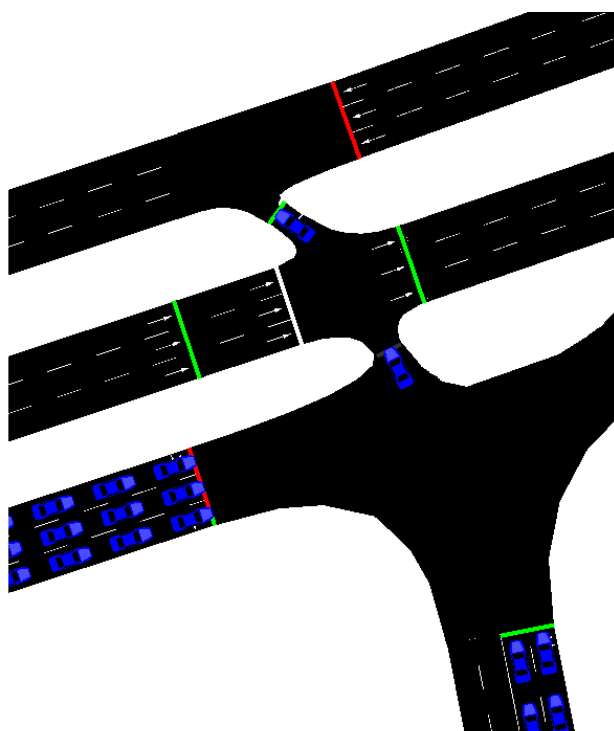
A mesma simulação foi rodada novamente com inserção de tempos de teletransporte. Os resultados foram tempos superiores a 30.000 segundos (8,3 horas) para todas as viagens serem finalizadas, o que torna a simulação não representativa.

Com estes resultados foi concluído que não é possível realizar uma programação semaforica adequada para a rede de forma sistemática com essas ferramentas para essa área de estudo.

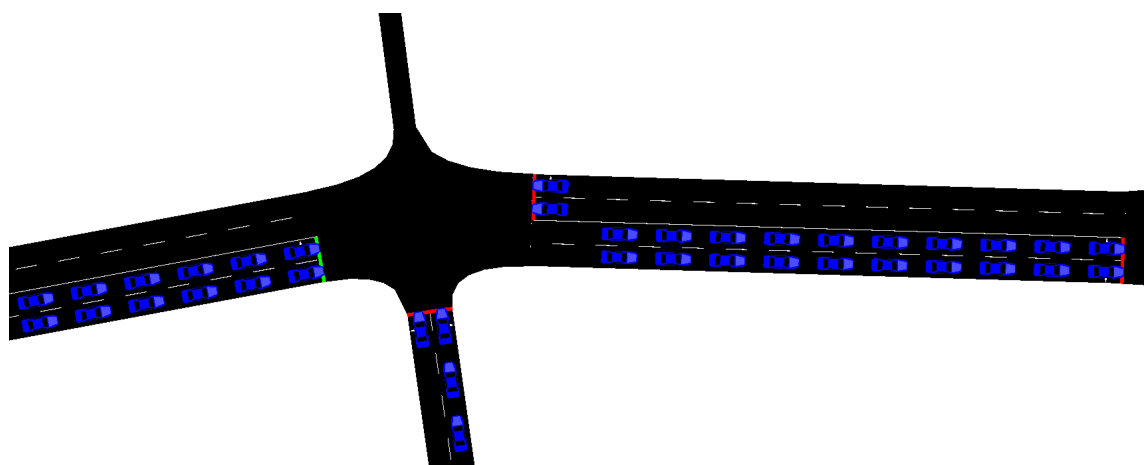
5.7.1 Modificações semaforicas

A falta de sincronismo dos semáforos foi a causa do trancamento da rede de tráfego. Dois problemas foram encontrados devido a falta de sincronismo: semáforos desagrupados, que trabalham de forma independente, com ciclos diferentes ou com *offsets*, que permitem movimentos em momentos não propícios, Figura 35 (a) e sinais que não foram coordenados de modo a obter onda verde, Figura 35 (b).

Figura 35: Semáforos sem sincronismo



(a) Erro de agrupamento



(b) Erro de coordenação

Os semáforos identificados nessas situações foram agrupados. O procedimento foi realizado no *netedit*, os semáforos com esses problemas foram apagados e recriados de forma agrupada. A Figura 36 apresenta um dos casos antes de realizar o procedimento de agrupamento. A Figura 37 apresenta o resultado após agrupar.

Figura 36: Semáforos antes do agrupamento

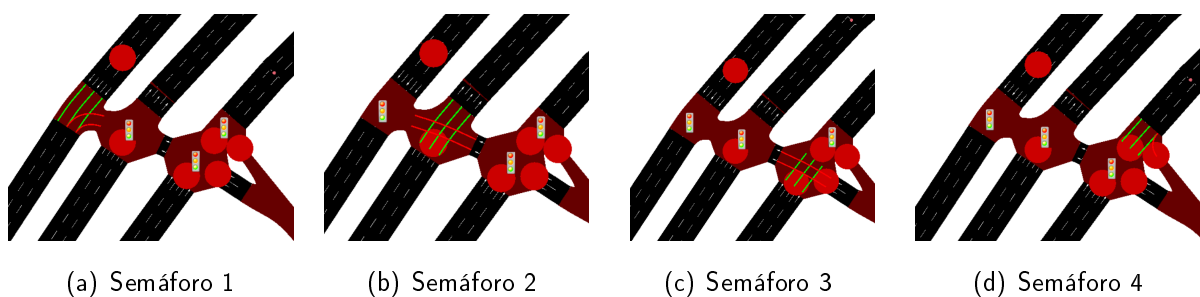
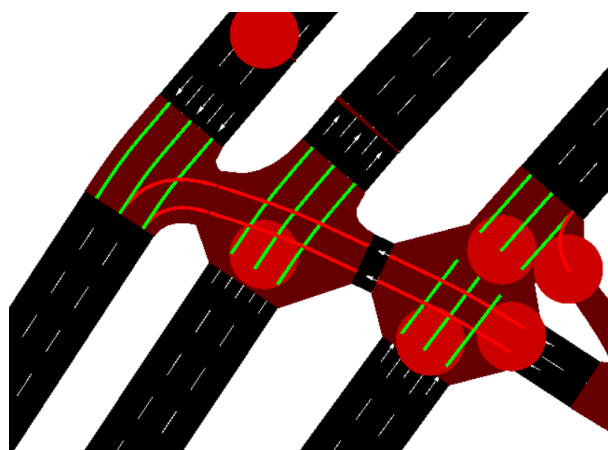
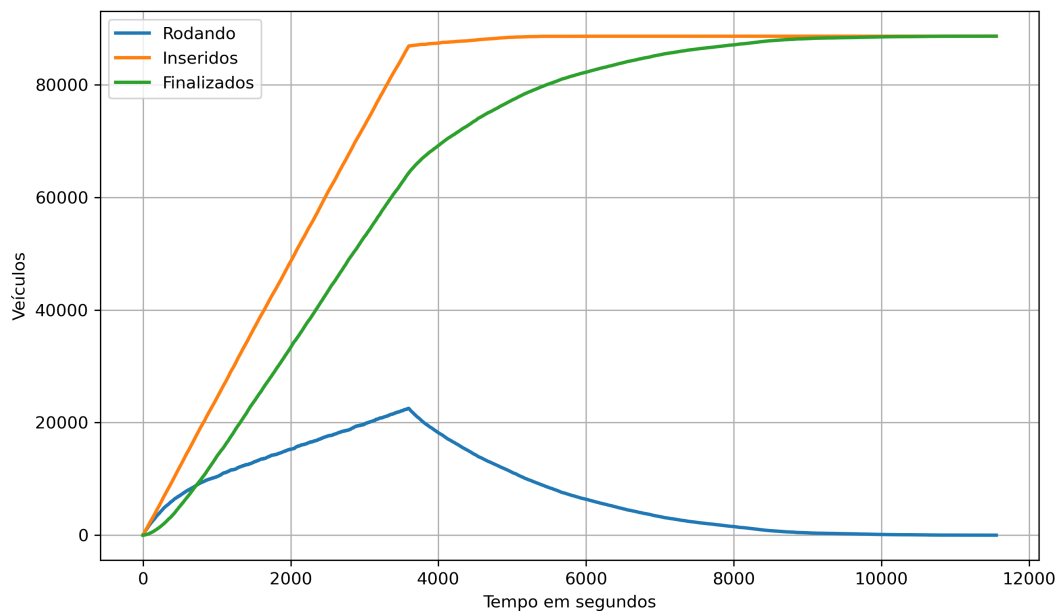


Figura 37: Semáforos agrupados



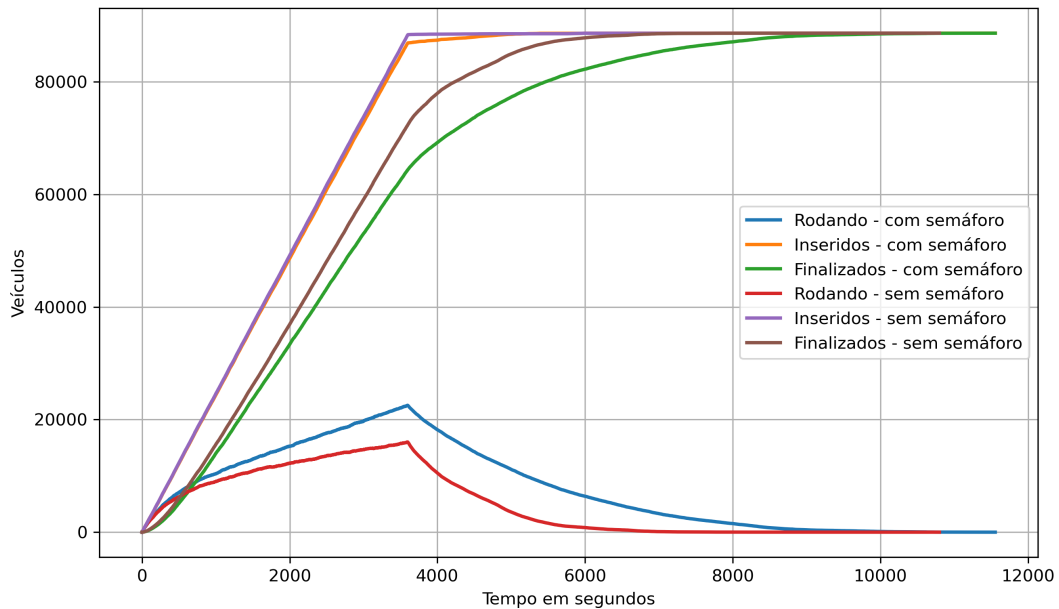
Foram realizados 21 agrupamentos semafóricos com objetivo de solucionar os dois problemas identificados. Novamente, o *tlsCycleAdaptation.py* e o *tlsCoordinator.py* foram utilizados, novos tempos semafóricos e *offsets* foram criados. A simulação rodou com o modelo Incremental de 5 iterações e tempo de teletransporte de 300 segundos. Ocorreram 78 teletransportes, o que representa 0,08% sobre as viagens realizadas. A Figura 38 apresenta os dados da simulação. O tempo necessário para ocorrer a a simulação foi de 11.555 segundos. O tempo de simulação e os valores obtidos com teletransporte, veículos inseridos e finalizados foram satisfatórios, não há trancamento na rede.

Figura 38: Gráfico da simulação com alocação Incremental com semáforos após a correção semafórica



A Figura 39 apresenta a comparação entre os valores obtidos com semáforos ativados e desativados. É possível notar que os semáforos retardam as viagens de modo geral, a taxa de veículos com viagens finalizadas é inferior durante o tempo de simulação e a de veículos rodando é superior.

Figura 39: Gráfico da simulação com método Incremental com e sem semáforos

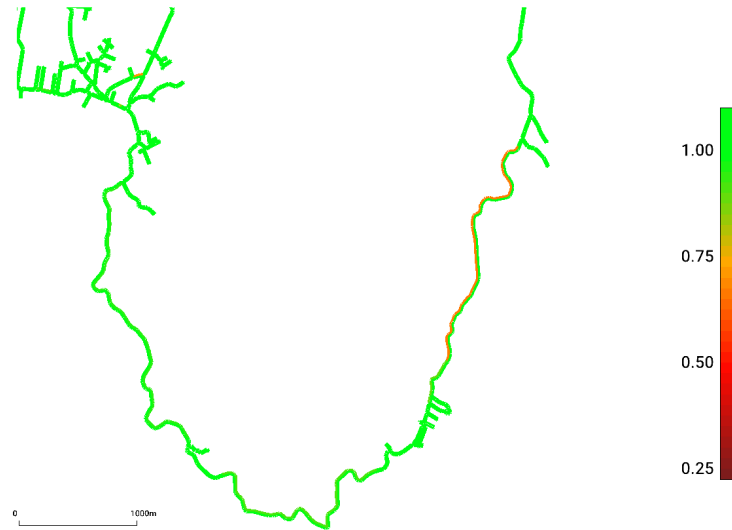


5.8 Avaliação dos resultados

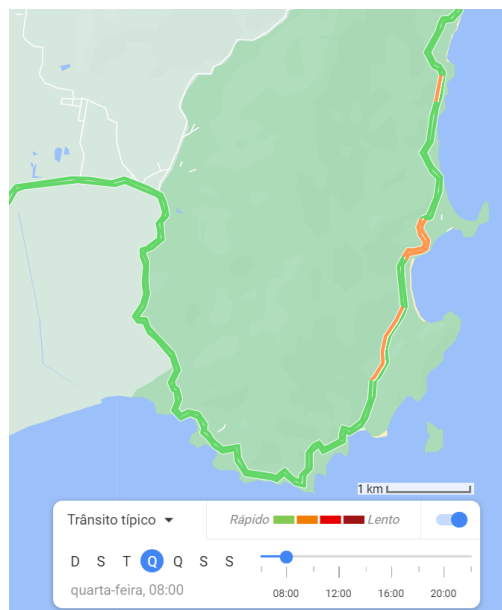
A simulação que se apresentou mais representativa foi a do método Incremental com 5 iterações com semáforos ativados. Essa simulação foi comparada ao trânsito típico da área de estudo fornecida pelo *Google Maps*, para uma quarta-feira às 8 horas da manhã. Os dados de tráfego do *Google Maps* utilizam dados históricos das vias para o período definido na busca.

O *Google Maps* classifica com escala de cores o trânsito de lento (vermelho) a rápido (verde). No SUMO, as imagens geradas foram de acordo com a velocidade relativa com relação a velocidade máxima permitida na via para as 8 horas, a escala de cor não é coincidente com a adotada pelo *Google Maps*. Os locais comparados são os que apresentam trânsito mais intenso na simulação. As Figuras 40, 41, 42, 43, 44, 45, 46, 47, 48, 49 apresentam comparações com relação a velocidade da simulação e a velocidade do trânsito típico do *Google Maps*.

Figura 40: Comparação da velocidade da simulação com a do trânsito típico do *Google Maps*, Governador Celso Ramos (latitude: -27.388, longitude: -48.590)



(a) Simulação SUMO

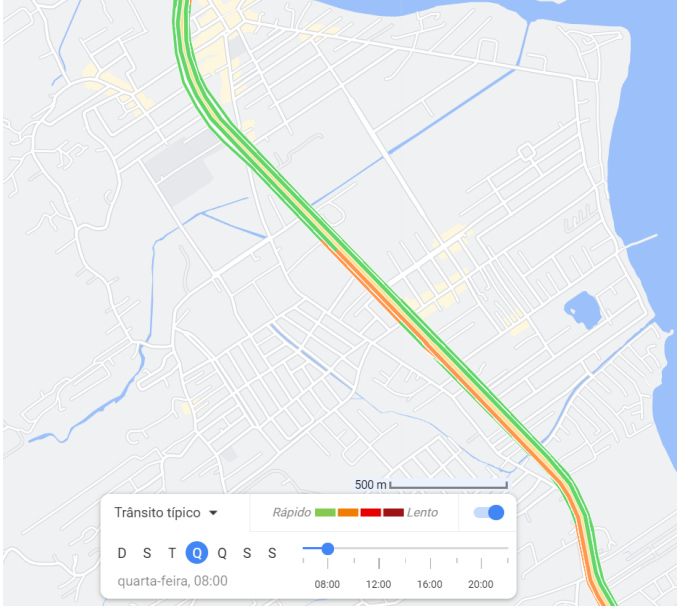


(b) Trânsito típico *Google Maps*

Figura 41: Comparação da velocidade da simulação com a do trânsito típico do *Google Maps*, BR-101, Biguaçu (latitude:-27.506 longitude: -48.648)



(a) Simulação SUMO



(b) Trânsito típico *Google Maps*

Figura 42: Comparação da velocidade da simulação com a do trânsito típico do *Google Maps*, BR-101, São José (latitude: -27.548, longitude: -48.619)

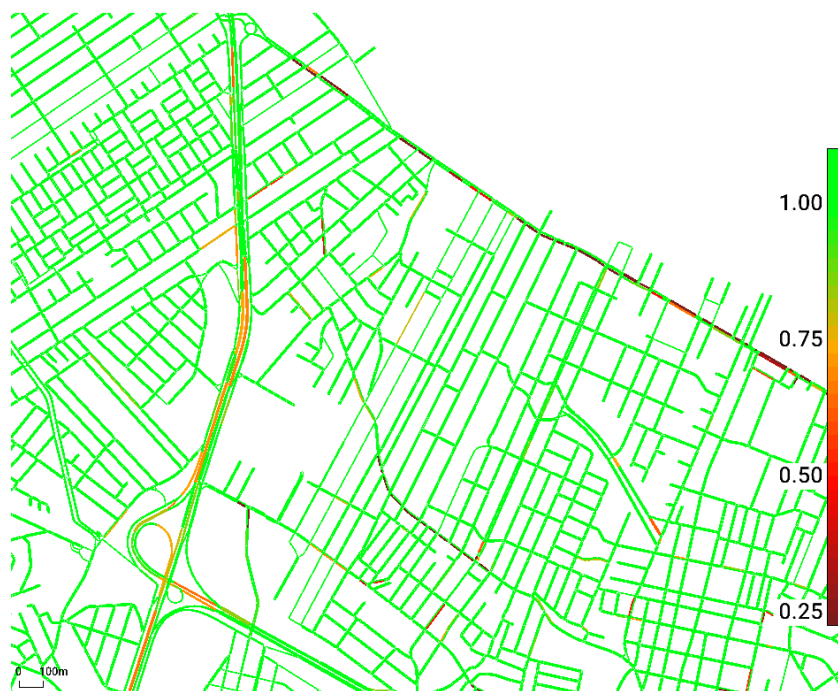


(a) Simulação SUMO

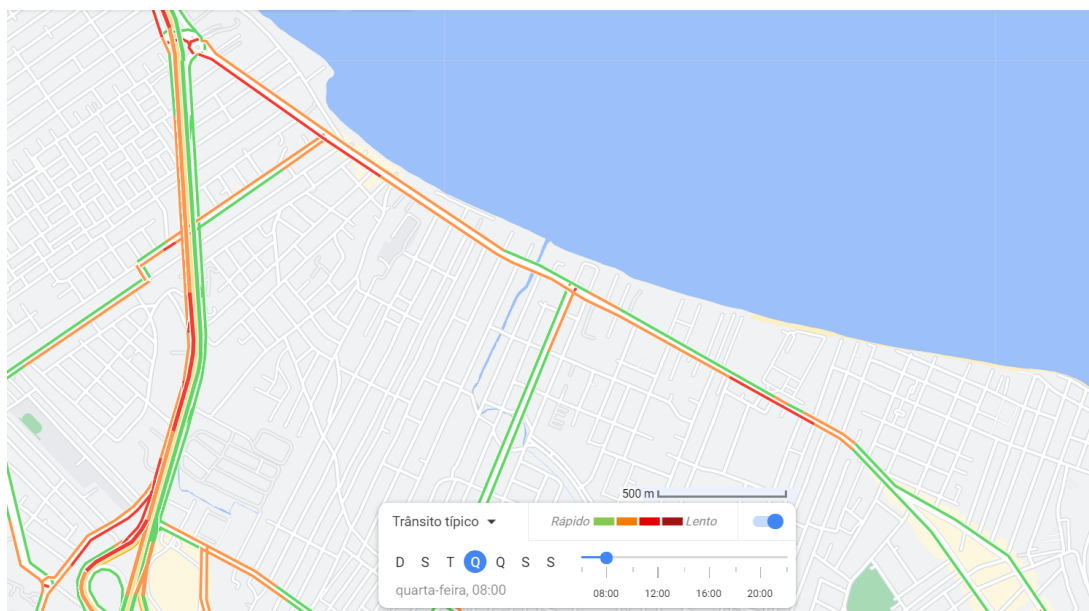


(b) Trânsito típico *Google Maps*

Figura 43: Comparação da velocidade da simulação com a do trânsito típico do *Google Maps*, São José (latitude: -27.570, longitude:-48.607)



(a) Simulação SUMO

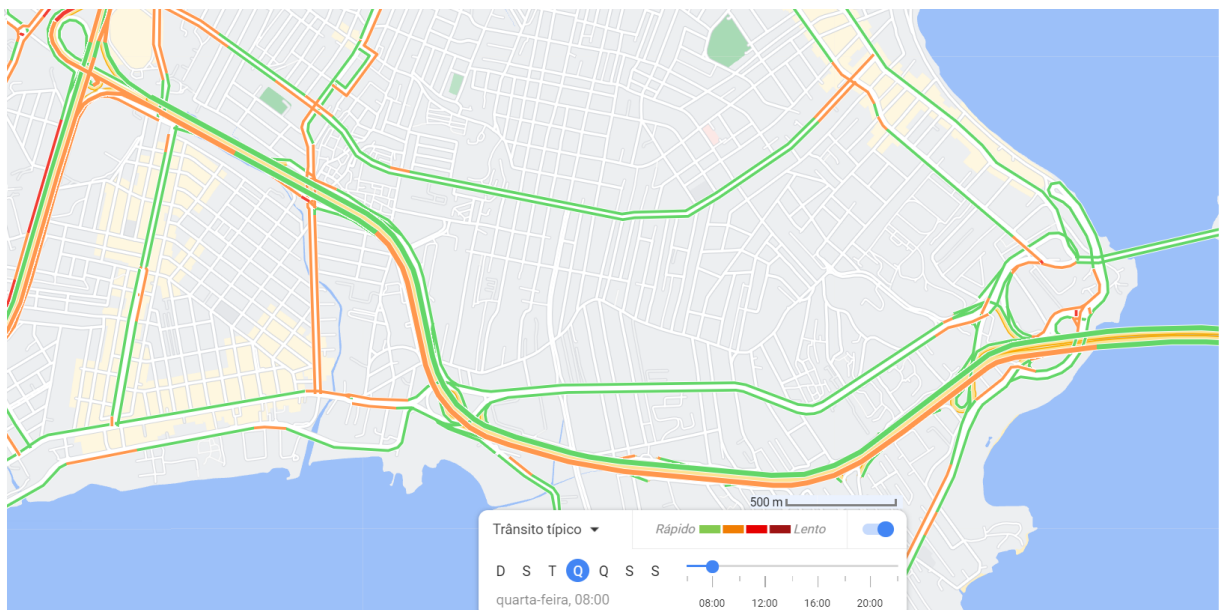


(b) Trânsito típico *Google Maps*

Figura 44: Comparação da velocidade da simulação com a do trânsito típico do *Google Maps*, Florianópolis (latitude: -27.603, longitude: -48.591)



(a) Simulação SUMO

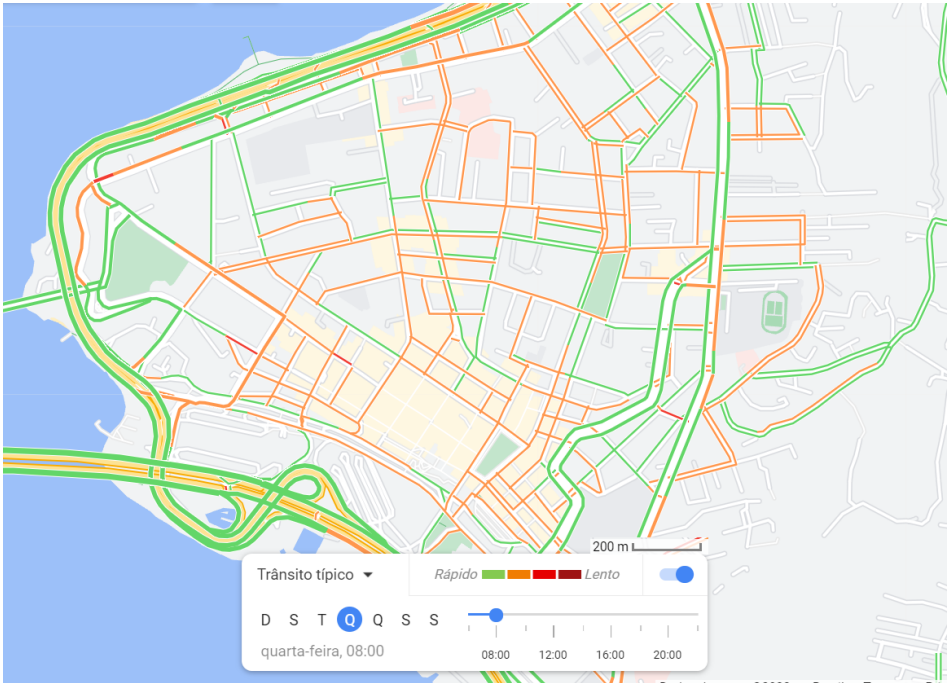


(b) Trânsito típico *Google Maps*

Figura 45: Comparação da velocidade da simulação com a do trânsito típico do *Google Maps*, Florianópolis (latitude: -27.593, longitude: -48.551)



(a) Simulação SUMO

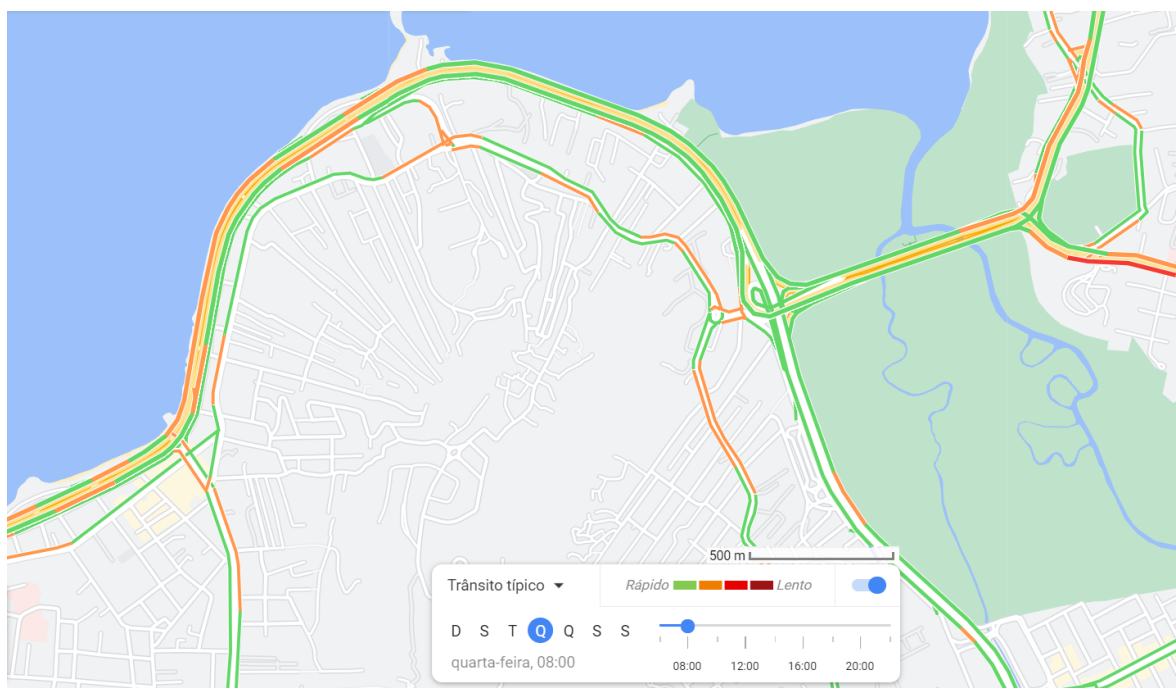


(b) Trânsito típico *Google Maps*

Figura 46: Comparação da velocidade da simulação com a do trânsito típico do *Google Maps*, Florianópolis (latitude: -27.579, longitude: -48.525)

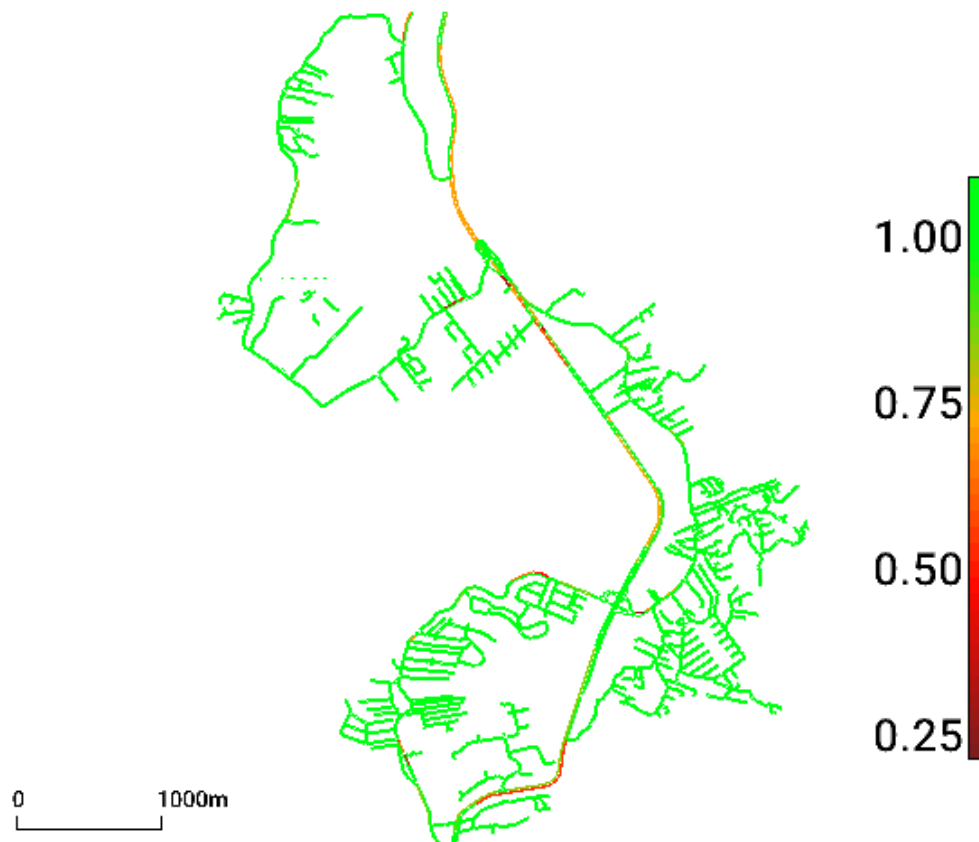


(a) Simulação SUMO

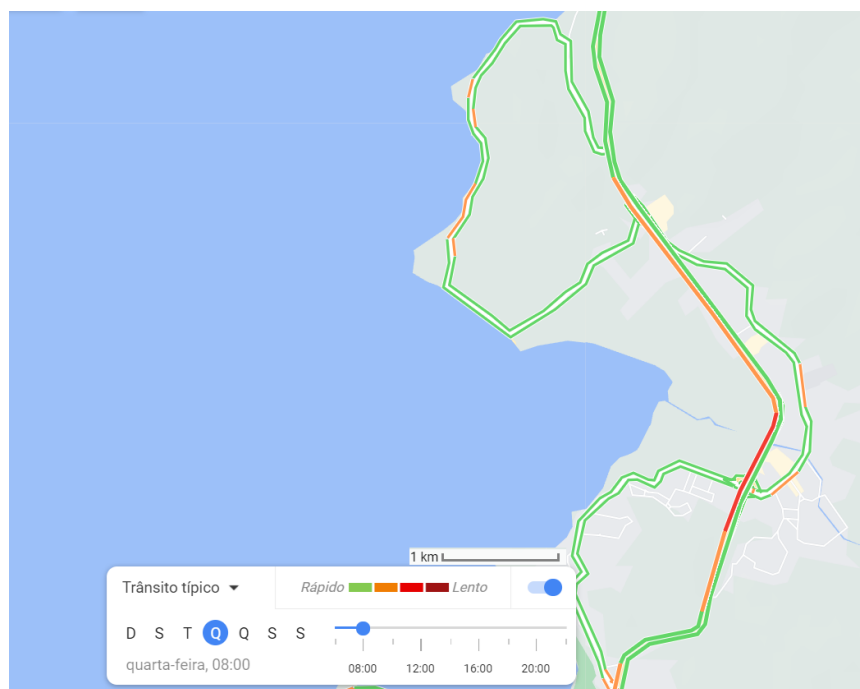


(b) Trânsito típico *Google Maps*

Figura 47: Comparação da velocidade da simulação com a do trânsito típico do *Google Maps*, Florianópolis (latitude: -27.543, longitude: -48.503)



(a) Simulação SUMO

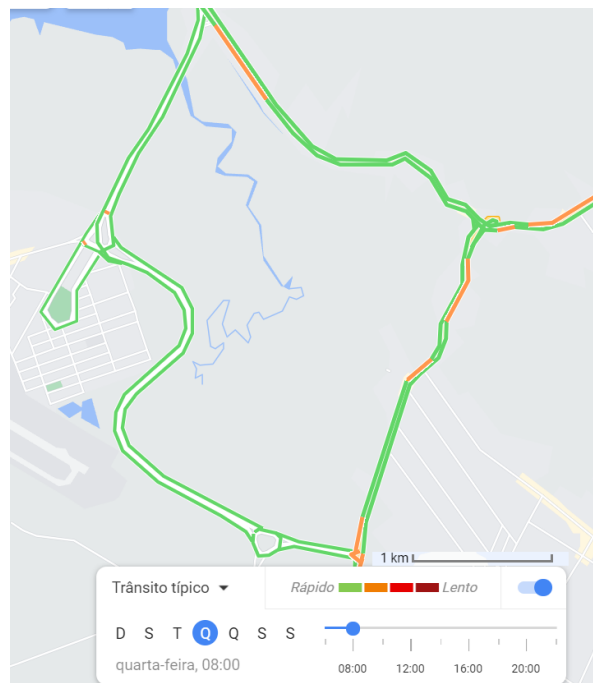


(b) Trânsito típico *Google Maps*

Figura 48: Comparação da velocidade da simulação com a do trânsito típico do *Google Maps*, Florianópolis (latitude: -27.650, longitude: -48.518) ,

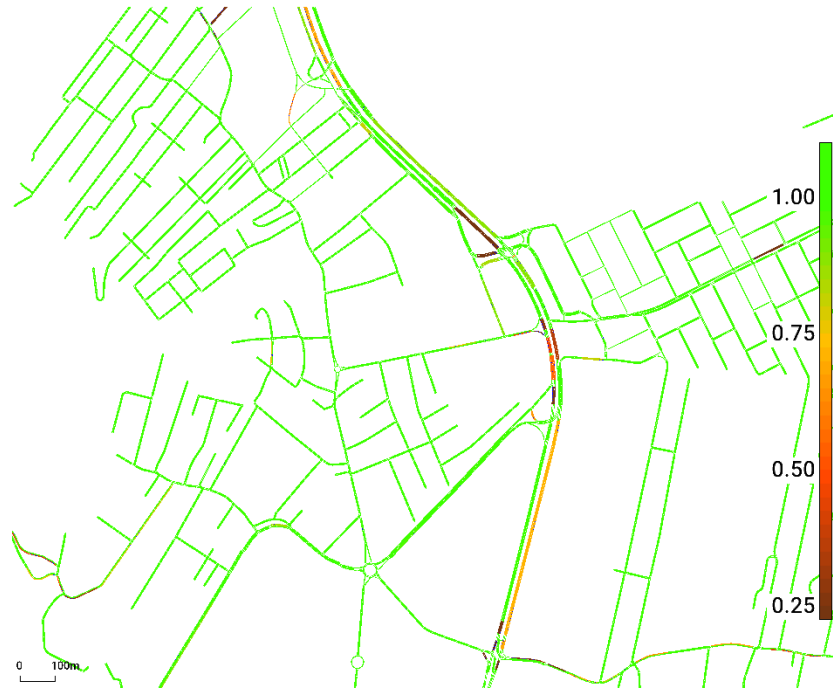


(a) Simulação SUMO

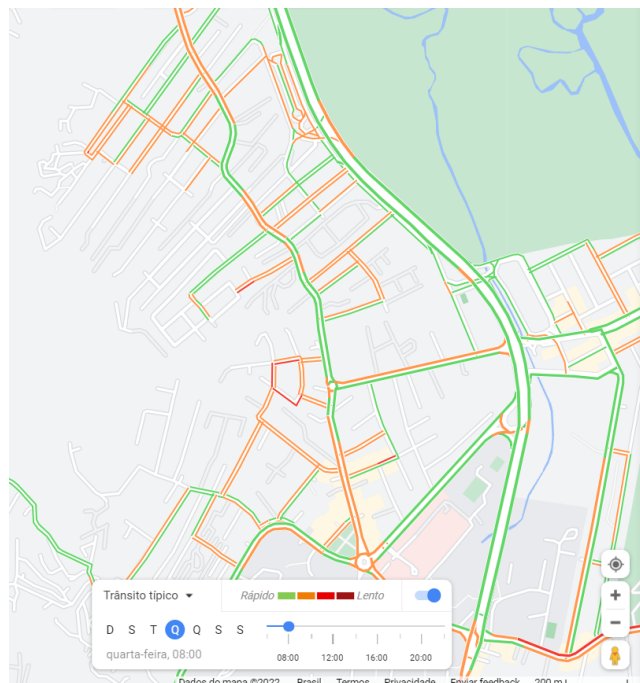


(b) Trânsito típico *Google Maps*

Figura 49: Comparação da velocidade da simulação com a do trânsito típico do *Google Maps*, bairro Trindade, Florianópolis (latitude: -27.595, longitude: -48.517) ,



(a) Simulação SUMO



(b) Trânsito típico *Google Maps*

A simulação apresenta pontos de semelhança e de divergência com a velocidade do trânsito típico do *Google Maps*. O tráfego mais lento possui pontos coincidentes e próximos com os obtidos na simulação. Essa forma de validação visual é de difícil confirmação. Além de ser necessário correlacionar o cenário temporal do trânsito típico com o tempo da simulação.

6 Conclusões

Com este Trabalho de Conclusão de Curso foi possível ter a experiência de rodar uma simulação com dados projetados de demanda reais do PLAMUS e dados abertos de infraestrutura do OSM. Promoveu a oportunidade de sentir a sensibilidade da simulação a diferentes métodos de alocação e parâmetros, além de estudar e conhecer o mecanismo do *software* SUMO.

Os dados abertos de infraestrutura da rede são suficientes para rodar a simulação em caráter de experimentação da ferramenta. O grau de realismo dos dados geométricos e dos movimentos permitidos não pode ser concluído, deve ser realizada verificação ou por meio de pesquisa de campo ou por meio de ferramentas *online* de forma manual. Foram notadas discrepâncias grandes em relação a velocidade permitida nas vias comparando com suas classificações do OSM. A correção foi realizada de forma sistemática para calibrar melhor o modelo. Os dados semafóricos das fases inseridas pelo SUMO não apresentam realismo e tentativa de adaptação das fases para a simulação somente com os *scripts* do SUMO não obteve sucesso. Os semáforos precisaram receber edições manuais para tornarem os resultados mais representativos.

O SUMO apresenta diversos modelos de alocação, o que dá a possibilidade de adaptação de diferentes cenários. Os métodos de alocação Tudo ou Nada, Incremental, variação do Incremental, Equilíbrio do Usuário e Equilíbrio Estocástico do Usuário foram aplicados na simulação. O método Tudo ou Nada mostrou-se não adequado para grandes áreas de estudo. A demanda fica concentrada nas vias principais e ocorre o trancamento, mesmo com valores baixos de tempo de teletransporte não obteve sucesso. O método Equilíbrio do Usuário e Equilíbrio Estocástico do Usuário também não são aptos para simulação de grande porte pois necessitam de tempo de processamento elevado para gerar o arquivo de rotas se atingirem o estado de equilíbrio. O método que se mostrou mais adequado para a área de estudo foi o método Incremental pois necessitou de baixo número de teletransporte, apresentou menor índice de veículos rodando ao mesmo tempo na simulação e conseguiu finalizar todas as viagens em tempo representativo.

A simulação rodou com tempo considerado possível de ocorrer. O total de 88.640 veículos finalizaram suas viagens para hora de pico da manhã com 11.555 segundos. O tempo de *warm-up* adotado foi com base no número de veículos inseridos, com tempo de 960 segundos.

A validação dos resultados obtidos na simulação foi de modo visual, comparando a imagens do *Google Maps* para o trânsito típico de uma quarta-feira às 8 horas. Essa forma de validação se mostrou difícil de verificar em uma grande área. A simulação apresentou pontos coincidentes e pontos discrepantes de trânsito intenso. Essa forma de validação é apenas a caráter de estudo, sendo necessário validação numérica para confirmar o grau de realismo da simulação. Outro fator é que os dados de demanda foram obtidos em pesquisa domiciliar em 2014 e projetados para 2020. A matriz OD resultante está sujeita a imperfeições do processo de estimativa.

O SUMO possui documentação, fóruns ativos em comunidades *online*, além de conferências anuais. Esses materiais fornecem recursos para entender melhor como operar o SUMO. No entanto, o *software* carece de informações importantes em sua documentação, principalmente tratando dos métodos de alocação que não possuem detalhadamente os procedimentos internos do SUMO. O *software* trabalha com diversos arquivos de definição e configuração com formato .xml, é necessária uma familiaridade maior com o ambiente de linha de comando e o conhecimento de uma linguagem de programação auxilia no desenvolvimento do trabalho. A interface *netedit* de edição de rede consegue manipular grandes áreas com bom desempenho e ferramentas intuitivas. A interface *sumo-gui* apresenta opções de visualização da simulação que facilitam a análise. As ferramentas em *python* que acompanham o programa SUMO fornecem mais possibilidades e opções de implementar a simulação. O SUMO se mostrou uma ferramenta que abrange diversos aspectos necessários à simulação.

Comparação da simulação com os modelos LuST e TuST

A síntese dos dados de infraestrutura dos modelos que serviram como referência e inspiração para grande parte deste trabalho é apresentada na Tabela 16. É possível notar que a área de estudo abrangida neste trabalho é consideravelmente maior. Os modelos de referência conseguiram realizar maior número de edições manuais na malha e conferência mais detalhada da infraestrutura.

Tabela 16: Comparação dos dados de infraestrutura dos modelos Lust, TuST e da Região Metropolitana da Grande Florianópolis

| Infraestrutura | Simulação LuST | Simulação TuST | Simulação Região Metropolitana da Grande Florianópolis |
|---|----------------|----------------|--|
| Total de nós | 2.372 | 32.936 | 43.778 |
| Total de <i>links</i> | 5.969 | 66.296 | 93.096 |
| Total de comprimento de <i>links</i> (km) | 931,12 | 6.570,28 | 12.554,48 |
| Total de comprimento de faixas (km) | 1.571,40 | 7.723,40 | 13.240,14 |
| Área de estudo (km^2) | 156 | 602,61 | 1288,49 |

Quanto aos métodos de alocação, tanto o modelo TuST para hora de pico quanto o modelo deste trabalho apresentaram resultados melhores com o método Incremental, porém com diferente número de iterações. Já o modelo LuST obteve sucesso com a variação do método Incremental.

O modelo LuST compara as velocidades da simulação com as obtidas no *Google Maps* e consegue concluir que a simulação é representativa. Neste trabalho, essa conclusão não é realizada, a

comparação com o *Google Maps* não foi considerada uma opção de qualidade. Já o modelo TuST concluí com base no tempo que todos os veículos levam para finalizar suas viagens, com a referência de tempo padrão observado na área de estudo, que a simulação pode ser representativa para um cenário pessimista. Os dois modelos não apresentam forma de validação com dados e métodos numéricos que são mais consistentes.

6.1 Recomendações para trabalhos futuros

Recomendações para trabalhos futuros:

1. realizar simulação com transporte público de uma área de estudo com o SUMO;
2. explorar simulações com ciclovias e pedestres no SUMO;
3. utilizar a ferramenta *ACTIVITYGEN* do SUMO para gerar dados de demanda e executar simulação;
4. experimentar a validação com dados e métodos numéricos consistentes.

Referências

- Barceló, J. et al. *Fundamentals of traffic simulation*. [S.l.]: Springer, 2010. v. 145.
- Brasil. Lei nº 9.503, de 23 de setembro de 1997. *CÓDIGO, DE TRÂNSITO BRASILEIRO*, 2003.
- BRASIL, D. Manual de projeto geométrico de travessias urbanas. *Rio de Janeiro: DNIT–Instituto de Pesquisas Rodoviárias*, 2010.
- Bruton, M. J. *Introdução ao planejamento dos transportes*. [S.l.]: Editora Intercincia, 1979.
- Campos, V. B. G. Planejamento de transportes: conceitos e modelos. *Rio de Janeiro: Interciência*, p. 2, 2013.
- Codeca, L.; Frank, R.; Engel, T. Luxembourg sumo traffic (lust) scenario: 24 hours of mobility for vehicular networking research. In: IEEE. *2015 IEEE Vehicular Networking Conference (VNC)*. [S.l.], 2015. p. 1–8.
- CONTRAN. Resolução 483.2014 - manual brasileiro de sinalização de trânsito, volume v, sinalização semafórica. *Brasília, DF*, 2014.
- DENATRAN. Estatísticas - frota de veículos - denatran. 2021. Disponível em: <<https://www.gov.br/infraestrutura/pt-br/assuntos/transito/conteudo-denatran/estatisticas-frota-de-veiculos-denatran>>.
- Denatran, M. Manual de semáforos. *Departamento Nacional de Trânsito, Brasília, Brazil*, 1984.
- DNIT, I. 723. *Manual de estudos de tráfego*. [S.l.]: Departamento Nacional de Infraestrutura e Transportes, Rio de Janeiro, Brazil, 2006.
- Dowling, R. et al. *Traffic analysis toolbox, volume III: Guidelines for applying traffic microsimulation modeling software*. [S.l.], 2004.
- Eissfeldt, N. G. *Vehicle-based modelling of traffic. Theory and application to environmental impact modelling*. Tese (Doutorado) — Universität zu Köln, 2004.
- Elefteriadou, L. et al. *An introduction to traffic flow theory*. [S.l.]: Springer, 2014.
- FHWA. *Traffic Analysis Toolbox Volume II: Decision Support Methodology for Selecting Traffic Analysis*. [S.l.], 2004.
- FHWA. *Traffic Analysis Toolbox Volume III: Guidelines for Applying Traffic Microsimulation Modeling Software*. [S.l.], 2004.
- Garber, N. J.; Hoel, L. A. *Traffic and highway engineering.: Cengage Learning*. [S.l.]: Inc, 2009.
- Hensher, D. A. *Handbook of transport geography and spatial systems*. [S.l.]: Elsevier, 2004.
- Hoel, L. A.; Garber, N. J.; Sadek, A. W. *Engenharia de infraestrutura de transporte: uma integração multimodal*. [S.l.]: Cengage Learning, 2012.
- Hutchinson, B. G. *Principles of urban transport systems planning*. 1974.
- Mannering, F. L.; Washburn, S. S. *Principles of highway engineering and traffic analysis*. [S.l.]: John Wiley & Sons, 2020.
- Ortúzar, J. de D.; Willumsen, L. G. *Modelling transport*. [S.l.]: John wiley & sons, 2011.
- Pignataro, L. J. et al. *Traffic engineering: theory and practice*. 1973.

- PNAD, I. Pesquisa nacional por amostra de domicílios. *Rio de Janeiro: IBGE*, 2015.
- Rapelli, M.; Casetti, C.; Gagliardi, G. Tust: from raw data to vehicular traffic simulation in turin. In: *IEEE. 2019 IEEE/ACM 23rd International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*. [S.l.], 2019. p. 1–8.
- Rodrigue, J.-P.; Comtois, C.; Slack, B. *The geography of transport systems*. [S.l.]: Routledge, 2016.
- Santos, M. *A urbanização brasileira*. [S.l.]: Edusp, 2013. v. 6.
- Saw, K.; Katti, B.; Joshi, G. Literature review of traffic assignment: static and dynamic. *International Journal of Transportation Engineering*, Tarrahan Parseh Transportation Research Institute, v. 2, n. 4, p. 339–347, 2015.
- Senna, L. A. dos S. *Economia e Planejamento dos Transportes*. [S.l.]: Elsevier Brasil, 2014.
- Spiegelman, C.; Park, E. S.; Rilett, L. R. *Transportation statistics and microsimulation*. [S.l.]: Chapman and Hall/CRC, 2011.
- Treiber, M.; Kesting, A. Traffic flow dynamics. *Traffic Flow Dynamics: Data, Models and Simulation*, Springer-Verlag Berlin Heidelberg, Springer, p. 983–1000, 2013.
- Wolshon, B.; Pande, A. et al. *Traffic engineering handbook*. [S.l.]: John Wiley & Sons, 2016.