



Universidade Federal de Santa Catarina

Centro Tecnológico

Programa de Pós-Graduação em Engenharia de Transportes e Gestão Territorial

Fernanda Beckhauser Mallon

**ELABORAÇÃO DE UM MODELO MULTIMODAL POR MEIO DE SIMULAÇÃO DE
TRÁFEGO BASEADA EM AGENTES: ESTUDO DE CASO NA GRANDE
FLORIANÓPOLIS/SC**

Florianópolis, 23 de junho de 2024

Fernanda Beckhauser Mallon

**ELABORAÇÃO DE UM MODELO MULTIMODAL POR MEIO DE SIMULAÇÃO DE
TRÁFEGO BASEADA EM AGENTES: ESTUDO DE CASO NA GRANDE
FLORIANÓPOLIS/SC**

Dissertação submetida ao Programa de Pós-Graduação em Engenharia de Transportes e Gestão Territorial, da Universidade Federal de Santa Catarina para a obtenção do título de Mestre em Engenharia de Transportes e Gestão Territorial.

Professor orientador: Alexandre Hering Coelho, Dr.

Florianópolis, 23 de junho de 2024

Mallon, Fernanda Beckhauser

Elaboração de um modelo multimodal por meio de simulação de tráfego baseada em agentes : estudo de caso na Grande Florianópolis/SC / Fernanda Beckhauser Mallon ; orientador, Alexandre Hering Coelho, 2024.

201 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Transportes e Gestão Territorial, Florianópolis, 2024.

Inclui referências.

1. Engenharia de Transportes e Gestão Territorial. 2. Planejamento de transportes urbanos. 3. Transporte público coletivo. 4. Simulação baseada em agentes. 5. MATSim. I. Coelho, Alexandre Hering. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia de Transportes e Gestão Territorial. III. Título.

Fernanda Beckhauser Mallon

**ELABORAÇÃO DE UM MODELO MULTIMODAL POR MEIO DE SIMULAÇÃO DE
TRÁFEGO BASEADA EM AGENTES: ESTUDO DE CASO NA GRANDE
FLORIANÓPOLIS/SC**

O presente trabalho em nível de Mestrado foi avaliado e aprovado, em 26 de março de 2024, pela banca examinadora composta pelos seguintes membros:

Prof. Ricardo Villarroel Dávalos, Dr.

UFSC

Prof. Werner Kraus Junior, Dr.

UFSC

Profa. Fabiana Serra de Arruda, Dra.

UnB

Certificamos que esta é a versão original e final do trabalho de conclusão que foi julgado adequado para obtenção do título de Mestra em Engenharia de Transportes e Gestão Territorial.

Coordenação do Programa de Pós-Graduação em Engenharia de Transportes e Gestão Territorial

Prof. Alexandre Hering Coelho, Dr.

Orientador

Florianópolis, 2024.

Agradecimentos

Ao professor e orientador Alexandre, pelo apoio, aprendizados e orientação irretocável ao longo de toda a minha trajetória acadêmica.

Aos desenvolvedores em código aberto de todo o mundo, pelo compartilhamento de ferramentas para o desenvolvimento de conhecimento de forma gratuita.

Ao LabTrans, em especial ao Rodolfo, Victor e Marcus, por todo o trabalho desenvolvido em conjunto e por todas as discussões técnicas ao longo dos anos que guiaram a minha trajetória profissional.

À minha irmã Luiza, e à minha amiga Kaísa, pelo apoio incondicional em todos os momentos da minha caminhada e por renovarem o significado da palavra amor a cada dia.

Aos meus amigos, em especial à Luísa e ao Sérgio, por proporcionarem as fugas da rotina, a alegria em compartilhar tempo e o engrandecimento em conhecer novos lugares juntos.

À minha família, por todo o suporte, carinho e compreensão ao longo dos últimos anos.

À minha mãe, Annelize, cuja memória de amor carrego em meu peito e uso como inspiração para tentar tornar o mundo um pouco melhor.

RESUMO

Com o crescimento da utilização de transportes individuais motorizados e os efeitos negativos ocasionados por sua utilização em massa nos centros urbanos, é evidenciada a necessidade de incentivo à migração para modos ativos e coletivos. A construção de modelos de transporte que simulem os efeitos do comportamento dos usuários pode ser utilizada com esse objetivo. Os modelos de simulação de tráfego que permitem explicitar o comportamento individual do usuário, como os modelos baseados em agente, destacam-se como ferramentas de compreensão dos efeitos de intervenções ao sistema de transporte no comportamento desses agentes. A disponibilidade de softwares em código aberto garante a acessibilidade da ferramenta para estudos com relativamente poucos recursos financeiros, apesar do alto grau de capacitação necessário ao técnico. Essa dissertação visa avaliar as potencialidades e as dificuldades da implantação de uma simulação multimodal baseada em agentes utilizando o *software* MATSim, com a elaboração de modelos de simulação das viagens realizadas por automóveis e no transporte coletivo relatadas na Pesquisa de Origem e Destino realizada em 2014. Os modelos têm seus parâmetros relacionados à infraestrutura, à demanda e ao comportamento dos usuários, os quais são calibrados de forma manual e por meio do calibrador automatizado *Cadyts*. No modelo calibrado para os automóveis e motocicletas, foram observados r^2 para o volume de tráfego diário de 0,9416, para o volume de tráfego nas horas-pico da manhã, de 0,7301, para o volume de tráfego nas horas-pico da tarde, de 0,7193, e para o tempo de viagem, de 0,8562. Posteriormente, foi elaborada a simulação multimodal, com a inclusão da alocação do transporte coletivo no modelo. Diante da indisponibilidade de dados sobre o transporte coletivo, foi modelada a oferta sintética de 63,6% das linhas de ônibus ativas. Os resultados do modelo apresentam alto índice de agentes presos e altos tempos de viagem. O último modelo incluiu a possibilidade de escolha do agente entre os modos automóvel, motocicleta e transporte coletivo, que foram modelados com o mesmo peso atribuído à utilidade do tempo de viagem. Esses fatores, somado à disponibilidade de motocicletas em relação à oferta limitada dos ônibus, e sua unidade de veículo padrão de 0,33 em relação ao carro, resultaram em uma migração modal com aumento de 26% da porcentagem de viagens totais realizadas por motocicletas, em detrimento do automóvel e do transporte coletivo.

Palavras-chave: planejamento de transportes urbanos; transporte público coletivo; simulação baseada em agentes; MATSim.

ABSTRACT

With the growing use of individual motorized transport and the negative effects caused by its mass use in urban centers, there is a clear need to encourage migration to active and collective modes of transport. The construction of transport models that simulate user behavior effects can be used with this objective. Traffic simulation models that allow for explicit individual user behavior, such as agent-based models, stand out as tools for understanding the effects of interventions on the transportation system on these agents' behavior. The availability of open-source software ensures the accessibility of the tool for studies with relatively few financial resources, despite the high level of technical training required. This dissertation aims to evaluate the potentialities and difficulties of implementing an agent-based multimodal simulation using the MATSim software, with the development of simulation models of trips made by cars and public transport reported in the Origin-Destination Survey conducted in 2014. The models have parameters related to infrastructure, demand, and user behavior, which are calibrated manually and through the automated calibrator Cadyts. In the model calibrated for cars and motorcycles, R^2 values were observed for daily traffic volume of 0.9416, for morning peak hour traffic volume of 0.7301, for afternoon peak hour traffic volume of 0.7193, and for travel time of 0.8562. Subsequently, a multimodal simulation was developed, including the allocation of public transport in the model. Due to the unavailability of public transport data, a synthetic supply of 63.6% of the active bus lines was modeled. The model results show a high rate of trapped agents and long travel times. The final model included the possibility of agent choice among car, motorcycle, and public transport modes, which were modeled with the same weight assigned to travel time utility. These factors, combined with the availability of motorcycles relative to the limited bus supply and their standard vehicle unit of 0.33 compared to cars, resulted in a modal shift with a 26% increase in the percentage of total trips made by motorcycles, to the detriment of cars and public transport.

Keywords: urban transport planning; public collective transport; agent-based simulation; MATSim.

Lista de Figuras

1	Relação entre o realismo comportamental e a complexidade computacional nos modelos de demanda. Fonte: Shiftan e Ben-Akiva (2010).	12
2	Etapas da modelagem e previsão de transportes. Fonte: adaptado de Ortúzar e Willumsen (2011).	14
3	Um grafo representando as origens (A e B), os destinos (C e D), os nós (1-5) e os <i>links</i> (a-h). Fonte: Hensher e Button (2008).	21
4	Sequência metodológica para calibração e validação do modelo. Fonte: Barceló (2010)	28
5	O algoritmo co-evolucionário no MATSim. Fonte: adaptado de Horni, Nagel e Axhausen (2016).	35
6	Módulos existentes no MATSim e sua integração com o processo de simulação. Fonte: adaptado de Horni, Nagel e Axhausen (2016).	37
7	Curva típica de desenvolvimento da pontuação. Fonte: Horni et al. (2009).	48
8	Layout dos principais visualizadores externos de resultados do MATSim	56
9	Rede de transporte público da região metropolitana de Berlim. Fonte: Oliveros (2014).	58
10	Calibração com <i>reroute</i> e <i>Cadyts</i> com peso crescente. Fonte: Oliveros (2014).	59
11	Modelo de rede viária de Joinville. Fonte: Bicudo (2015)	60
12	Evolução da pontuação dos planos no cenário de Joinville. Fonte: Bicudo (2015).	61
13	Gráfico de dispersão dos volumes de tráfego diários simulados e reais nos postos de contagem. Fonte: Kickhöfera et al. (2016).	63
14	Resultados da simulação em toda a área simulada, à esquerda, e na área central, à direita. Fonte: Kickhöfera et al. (2016).	64
15	Modelo da rede viária (em cinza) e a rede do sistema de transporte público (em azul). Fonte: Ziemke e Nagel (2017).	65
16	Distribuição horária dos erros médios relativos e absolutos após a calibração dos locais das atividades (esquerda) e após a calibração da divisão modal (direita). Fonte: Ziemke e Nagel (2017).	66
17	Evolução da pontuação da simulação do tráfego no DF. Fonte: Miranda (2017).	67
18	Densidade dos tempos de partida observados e simulados para diferentes motivos de viagem. Fonte: Zannat et al. (2024).	70

19	Concepção básica do método de elaboração do modelo de alocação do transporte individual motorizado na rede viária	71
20	Concepção básica do método de elaboração do modelo de alocação do transporte individual motorizado e público coletivo na rede viária	72
21	Área de estudo. Fonte: IBGE (2010) e PLAMUS (2014).	85
22	Densidade populacional das zonas de tráfego da área de estudo. Fonte: PLAMUS (2014).	86
23	Renda média mensal domiciliar das zonas de tráfego da área de estudo. Fonte: PLAMUS (2014).	87
24	Principais linhas de desejo no período da manhã identificadas pelo PLAMUS. Fonte: PLAMUS (2014).	88
25	Principais linhas de desejo no período da tarde identificadas pelo PLAMUS. Fonte: PLAMUS (2014).	89
26	Comparação visual entre a rede viária PLAMUS e OSM	93
27	Postos de contagem considerados pelo PLAMUS	96
27	Postos de contagem considerados pelo PLAMUS	97
28	Trechos da pesquisa de velocidade do PLAMUS	98
29	Rede viária PLAMUS antes e depois da simplificação	101
30	Quantidade de viagens originadas ou destinadas por coordenada no Experimento 4 .	104
31	Quantidade de viagens originadas ou destinadas por coordenada no Experimento 5 .	104
32	Quantidade de viagens originadas ou destinadas por coordenada no Experimento 32	105
33	Comparação entre volumes de tráfego diários no experimento 32	113
34	Comparação entre volumes de tráfego horários nos períodos de pico da manhã e da tarde	114
35	Carregamento da rede viária na hora-pico da manhã	116
36	Carregamento da rede viária na hora-pico da tarde	117
37	Frequências percentuais acumuladas dos tempos de viagem simulados e relatados na O/D	120
37	Frequências percentuais acumuladas dos tempos de viagem simulados e relatados na O/D	121
38	Evolução das pontuações em simulações com diferentes pesos do Cadyts	122

39	Evolução da pontuação	123
40	Histograma das viagens	125
41	Histograma das viagens simuladas a cada cinco minutos	126
42	Histograma das viagens simuladas, agregadas por hora	127
43	Histograma das viagens relatadas. Fonte: PLAMUS (2014).	127
44	Resultado da ordenação dos nós e <i>links</i> de um itinerários	129
45	Linhas de ônibus existentes, ativas e modeladas	130
46	Evolução da pontuação média dos agentes no experimento com transporte coletivo .	131
47	Histograma de todas as viagens realizadas no experimento com transporte coletivo .	132
48	Histograma das viagens de carro realizadas no experimento com transporte coletivo .	133
49	Histograma das viagens de ônibus realizadas no experimento com transporte coletivo	133
50	Histograma das viagens a pé realizadas no experimento com transporte coletivo . . .	134
51	Evolução da pontuação média dos agentes no experimento com transporte coletivo e migração modal	136
52	Divisão modal das viagens no experimento com transporte coletivo e migração modal	137
53	Histograma de todas as viagens realizadas no experimento com transporte coletivo com migração modal	139
54	Histograma das viagens de carro realizadas no experimento com transporte coletivo com migração modal	139
55	Histograma das viagens de ônibus realizadas no experimento com transporte coletivo com migração modal	140
56	Histograma das viagens a pé realizadas no experimento com transporte coletivo com migração modal	140

Lista de Tabelas

1	Trabalhos do PPGTG correlacionados a esse estudo	4
2	Divisão modal na área de estudo. Fonte: PLAMUS (2014).	90
3	Duração média das viagens por modo de transporte. Fonte: PLAMUS (2014).	90
4	Descrição dos dados disponíveis para elaboração do modelo de simulação na área de estudo	91
5	Estrutura do arquivo georreferenciado das rotas do transporte coletivo	99
6	Comparação dos dados entre a base do PLAMUS e do experimento	102
7	Perfis de modelo de demanda utilizados nos experimentos	107
8	Perfis de pontuação utilizados nos experimentos	107
9	Configuração dos experimentos	108
10	Medidas de performance dos experimentos analisados	109
11	Configurações do experimento final	111
12	Resultados dos experimentos com as diferentes redes viárias	111
13	Resultados dos experimentos com diferentes coordenadas O/D	112
14	R^2 da duração das viagens	119
15	Porcentagem de participação dos modos e duração média das viagens	137

Sumário

1	Introdução	1
1.1	Objetivos	3
1.1.1	Objetivo geral	3
1.1.2	Objetivos específicos	3
1.2	Limitações do trabalho	3
1.3	Aderência ao PPGTG	4
2	Revisão bibliográfica	8
2.1	Planejamento de transportes urbanos	8
2.1.1	Conceitos básicos	8
2.1.2	Histórico	10
2.2	Elaboração de modelos de simulação de tráfego	13
2.2.1	Concepção do modelo	15
2.2.2	Coleta de dados	18
2.2.3	Modelagem da oferta	20
2.2.4	Geração, distribuição e divisão modal de viagens	22
2.2.5	Alocação de tráfego	24
2.2.6	Calibração e validação do modelo	27
2.3	Modelos baseados em agentes	30
2.4	Simulador de transportes multi-agentes (MATSim)	32
2.4.1	O algoritmo coevolucionário	34
2.4.2	Elaboração dos arquivos de entrada	37
2.4.3	Execução da simulação (mobsim)	45
2.4.4	A pontuação dos planos diários	46
2.4.5	Replanejamento das atividades	49
2.4.6	Calibração e validação do modelo	50
2.4.7	Visualização dos resultados	54
2.5	Estudos de caso relacionados	57

3	Método	71
3.1	Levantamento e tratamento de dados	72
3.1.1	Diários de viagem	73
3.1.2	Rede viária	74
3.1.3	Contagens de tráfego	75
3.1.4	Oferta do transporte coletivo	75
3.2	Elaboração dos arquivos de entrada	76
3.3	Execução da simulação	79
3.4	Calibração do modelo	80
3.5	Validação do modelo de alocação do transporte individual	82
4	Área de estudo e dados disponíveis	85
4.1	Rede viária	92
4.2	Diário de viagens da população	93
4.3	Edificações	95
4.4	Contagens de tráfego	95
4.5	Pesquisa de velocidade	97
4.6	Oferta do transporte coletivo	98
5	Desenvolvimento e avaliação do modelo	100
5.1	Modelagem da rede viária	100
5.2	Modelagem da demanda	102
5.3	Verificação de erros no modelo	106
5.4	Calibração e validação do modelo	106
5.5	Elaboração da simulação multimodal com transporte coletivo	128
5.6	Inserção do módulo <i>ChangeMode</i> na simulação multimodal	135
6	Conclusões	142
7	Recomendações para trabalhos futuros	146
	Referências	147
A	Script de modelagem da rede viária	152

B	Script de modelagem da demanda	158
C	Parser de conversão dos dados de contagem de tráfego	165
D	Script para calibração das proporções de capacidade viária	169
E	Script de elaboração do <i>transitSchedule.xml</i>	171

1 Introdução

As condições da mobilidade urbana nas cidades brasileiras têm sido alvo de descontentamento crescente de boa parte da população nos últimos anos, principalmente em grandes centros urbanos. Segundo Carvalho e Pereira (2011), é possível identificar a deterioração das condições de mobilidade nesses locais, em virtude do aumento dos acidentes de trânsito com vítimas, dos congestionamentos urbanos e da emissão de poluentes veiculares. Segundo Carvalho (2016), isso se deve ao processo de urbanização acelerado e desordenado, e da crescente utilização do transporte motorizado individual pela população que vêm ocorrendo desde meados do século XX.

Em 2012, foi instituída a Política Nacional de Mobilidade Urbana, que tem como uma de suas diretrizes a “prioridade dos modos de transportes não motorizados sobre os motorizados e dos serviços de transporte público coletivo sobre o transporte individual motorizado” (BRASIL, 2012).

Apesar disso, Silva, Costa e Macêdo (2016) e Carvalho (2016) afirmam que a maioria das cidades no mundo, particularmente em países da periferia do capitalismo, continua a investir em políticas de incentivo à produção, comercialização e uso de veículos privados em detrimento de medidas de estímulo ao transporte público e ao transporte não motorizado.

A utilização exclusiva dos modos ativos na rotina diária da população ainda enfrenta muitos desafios, principalmente em cidades de ocupação espalhada, com centralidade bem definida. Isso porque, além da falta de investimentos em infraestrutura adequada e segura para esses modos, eles são limitados a pequenas e médias distâncias, e não são uma possibilidade para todos, como é o caso da utilização de bicicletas para pessoas portadoras de deficiências e idosos. Nesse sentido, o transporte coletivo (TC) realiza viagens de pequenas a grandes distâncias e, com o aumento de investimentos nesse setor, tem o potencial de ser acessível universalmente, aumentar a eficiência de mobilidade urbana e reduzir congestionamentos e tempos de viagem.

Para determinação dos investimentos a serem realizados pelos órgãos públicos para a efetiva migração modal para os modos coletivos, é necessário avaliar as consequências de implantação das medidas pretendidas. As simulações de tráfego são usualmente empregadas para esse fim, já que permitem a análise quantitativa dessas consequências no sistema, previamente a sua implantação.

Os modelos tradicionais de simulação de tráfego em sistemas de larga escala de mobilidade de um município ou de uma região metropolitana são macroscópicos e adotam uma série de simplificações sobre o comportamento dos usuários e dos sistemas de transporte. Esse processo tem como

fim a otimização do processamento computacional e o cálculo de indicadores resultantes agregados em nível de trecho de via sobre a performance dos sistemas de transporte. Esses modelos são perfeitamente capazes de orientar investimentos na ampliação de capacidade viária, e na definição de eixos de transporte de alta capacidade. No entanto, suas formas mais simples não consideram todas as vias da área de estudo e têm dificuldade de inclusão de modos não motorizados no modelo de alocação, e da consideração da interação entre os modos de transporte. Quanto mais complexo é o modelo para mitigar tais limitações, maior é seu custo financeiro e maior é a capacidade técnica necessária para manuseio.

Esse estudo levanta a pergunta se um modelo detalhado, em larga escala, multimodal, que explore as escolhas de tempo, rota e modos de transporte em conjunto por meio de um sistema de licença aberta é o ideal para a avaliação do impacto de medidas de alteração de infraestrutura, operação e políticas públicas na migração modal e na mobilidade urbana como um todo.

Recentemente, novas ferramentas e fontes de dados abertas e gratuitas têm sido desenvolvidas. Isso possibilita a economia de recursos para obtenção de dados e do *software* e, assim, facilita a elaboração de modelos dos sistemas de transporte.

Alguns *softwares* de simulação de tráfego de licença aberta mais usuais, como o SUMO¹, já permitem a utilização de demanda baseada em atividades, e a realização de viagens em etapas com diferentes modos de transporte. No entanto, não permitem a integração da etapa de escolha modal com a alocação de tráfego. Nesse sentido, o MATSim, um *software* de simulação de tráfego baseado em agentes, se destaca por permitir a fácil implantação da integração entre as diferentes escolhas inerentes ao processo de realizar uma viagem por meio de um modelo comportamental, facilitando as análises sobre o impacto da implantação de medidas pretendidas na migração modal em cenários futuros.

O MATSim é um *software* livre que utiliza o método baseado em agentes e é amplamente utilizado pela comunidade acadêmica. Sua aplicação, em conjunto com a utilização exclusiva de dados disponíveis de forma aberta permite a simulação com baixo custo de cenários complexos que exploram o processo decisório inerente aos deslocamentos urbanos de forma a orientar o investimento de recursos públicos.

São poucos estudos brasileiros que abordam a elaboração de modelos multimodais no MATSim, e os estudos atualmente disponíveis em que são elaborados modelos no MATSim no contexto

¹Disponível em: <<https://sumo.dlr.de/docs/index.html>>

brasileiro não abordam em profundidade os desafios encontrados na disponibilização dos dados necessários para a construção do modelo ou na calibração destes modelos.

Diante deste cenário, o presente trabalho visa contribuir para o entendimento do funcionamento e potencialidades da aplicação do MATSim em uma região metropolitana brasileira, a fim de avaliar as possibilidades e desafios na avaliação dos impactos de medidas sobre a migração modal para modos coletivos no contexto técnico, e contribuir com o desenvolvimento do estado da arte da simulação de tráfego no Brasil.

1.1 Objetivos

1.1.1 Objetivo geral

O objetivo deste trabalho é explorar o processo e os desafios de elaboração de um modelo de simulação de tráfego multimodal baseado em agentes da Região Metropolitana da Grande Florianópolis com base em *softwares* e dados gratuitos.

1.1.2 Objetivos específicos

Os objetivos específicos são:

- a) compreender as especificidades teóricas e práticas na elaboração de modelos baseados em agentes para simulação de tráfego multimodal;
- b) avaliar as potencialidades e limitações da utilização exclusiva de dados abertos ao público em geral;
- c) identificar os procedimentos a serem aplicados para a avaliação da migração modal em cenários alternativos;
- d) avaliar os resultados da migração modal em um cenário base não calibrado;
- e) desenvolver e apresentar *scripts* que realizem a conversão automatizada dos dados de entrada;
- f) identificar o nível de realismo adequado do modelo em relação ao nível de esforço e ganho nos resultados.

1.2 Limitações do trabalho

Esse trabalho possui as seguintes limitações:

- para avaliar especificamente a migração modal do transporte individual para o transporte coletivo, o modelo é composto somente por estes dois modos;
- não são aplicados procedimentos de projeção de demanda no caso de não haver homogeneidade temporal com dados da infraestrutura;
- no caso de ser constatada a falta de algum dado essencial para a construção de parte do modelo, é procurada alguma forma alternativa de suprir essa falta, mesmo que de forma sintética e com algum grau de arbitrariedade.

1.3 Aderência ao PPGTG

Esta dissertação trata da elaboração de uma simulação de tráfego multimodal baseada em agentes na Grande Florianópolis considerando a alocação na rede de modos de transporte individuais motorizados (veículos de passeio e motocicletas) e público coletivo (ônibus). Como tal, está inserida na área de concentração Sistemas de Transportes e na linha de pesquisa Planejamento de Sistemas de Transportes. No PPGTG, esta linha trata da adequação das necessidades de transporte de uma região ao seu desenvolvimento de acordo com suas características estruturais. Neste trabalho, focouse no detalhamento do processo de elaboração de um modelo de simulação baseado em agentes, incluindo a etapa de levantamento de dados e seus desafios, a etapa de modelagem da oferta e demanda dos modos individuais motorizados e do transporte coletivo, e a calibração e validação do modelo.

No histórico do PPGTG, foram encontradas onze dissertações que guardam afinidade com o tema deste trabalho. Na Tabela 1 a seguir, estão destacados os trabalhos considerados de contexto mais próximo ao desta dissertação.

Tabela 1: Trabalhos do PPGTG correlacionados a esse estudo

Ano	Autor	ABNT
Continua na próxima página		

Tabela 1 – Trabalhos do PPGTG correlacionados a esse estudo (continuação)

Ano	Autor	ABNT
2023	MARTENDAL	MARTENDAL, Haidi Rauber. Sistemas de transportes inteligentes e sua relação com a mobilidade urbana sustentável: proposta de framework para cidades brasileiras. Orientador: Lobo, Eduardo. 2023. 135 p. il., gráfs., tabs. Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Transportes e Gestão Territorial, Florianópolis, 2023.
2022	MAIA	MAIA, Tanise de Góes. Transporte público integrado ao planejamento urbano nos planos diretores: estudo multicaso do núcleo da Região Imediata de Itajaí - SC. Orientador: Lobo, Eduardo. 2022. 176 p. il., gráfs. Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Transportes e Gestão Territorial, Florianópolis, 2022.
2022	MARCON	MARCON, Aline Fileti. Using a multimodal database to assess the effect of the North-South line on the use of bicycle as a feeder mode to urban public transit in Amsterdam. Orientador: Seabra, Fernando. 2022. 86 p. il., gráfs. Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Transportes e Gestão Territorial, Florianópolis, 2022.
2022	SILVA	SILVA, Benício Silvio da. A utilização de subsídios e receitas extratarifárias ao transporte coletivo no Brasil e sua relação com a mobilidade urbana. Orientador: Lobo, Eduardo. 2022. 120 p. il., gráfs. Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Transportes e Gestão Territorial, Florianópolis, 2022.
2022	MACHADO	MACHADO, Silvia Dequech. Mobilidade urbana sustentável: o caso do sistema de transporte público de ônibus da Região Metropolitana de Florianópolis. Orientador: Souza, João Carlos. 2022. 102 p. il. Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Transportes e Gestão Territorial, Florianópolis, 2022.
Continua na próxima página		

Tabela 1 – Trabalhos do PPGTG correlacionados a esse estudo (continuação)

Ano	Autor	ABNT
2021	PEREIRA	PEREIRA, Hugo Alexander Martins. Método de análise de sistemas de transporte público urbano em cidades de médio porte: estudo de caso em São José dos Pinhais/PR. Orientador: Goldner, Lenise Grando. 2021. 106 p. il., gráfs. Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Transportes e Gestão Territorial, Florianópolis, 2021.
2021	DELLAROZA	DELLAROZA, Luiz Felipe Gomes. Modelo contínuo para desenho de rede de transporte público por ônibus: aplicação à cidade de Londrina/PR. Orientador: Coelho, Alexandre Hering. 2021. 153 p. il., gráfs. Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Transportes e Gestão Territorial, Florianópolis, 2021.
2019	BRAVO	BRAVO, Marcele Dorneles. Contribuição metodológica para estimativa da demanda transferida do automóvel para o transporte coletivo em um polo gerador de viagens. Orientador: Goldner, Lenise Grando. 2019. 188 p. il., gráfs., tabs. Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Transportes e Gestão Territorial, Florianópolis, 2019.
2018	BALVEDI	BALVEDI, Guilherme Arpini. Microssimulação de tráfego aplicada na avaliação do desempenho da segurança viária em rotatórias: um estudo de caso na rótula da UFSC/Trindade. Orientador: Goldner, Lenise Grando. 2018. 136 p. il., gráfs., tabs. Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Transportes e Gestão Territorial, Florianópolis, 2018.
2017	BORTOLAZZO	BORTOLAZZO, Shadia Silveira Assaf. Contribuição metodológica para planejamento de transporte urbano: uma proposta sob a ótica da sustentabilidade ambiental. Orientador: Valente, Amir Mattar. 2017. 217, [3] p. il., gráfs., mapas, tabs. Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Transportes e Gestão Territorial, Florianópolis, 2017.
Continua na próxima página		

Tabela 1 – Trabalhos do PPGTG correlacionados a esse estudo (continuação)

Ano	Autor	ABNT
2017	FERT	FERT, Luiza. Análise comparativa do uso de diferentes softwares para modelagem de tráfego em interseções urbanas. Orientador: Coelho, Alexandre Hering. 2017. 140 p. il., gráfs., tabs. Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Transportes e Gestão Territorial, Florianópolis, 2017.

Pode-se perceber que há trabalhos que têm contexto de aplicação no mesmo setor da presente dissertação e outros que usam os instrumentos metodológicos semelhantes aos aplicados nesta tese de dissertação. No primeiro caso, destacam-se os trabalhos de Maia (2022), que tem como foco a potencialidade do transporte público coletivo em gerar soluções e condições para transformações do espaço urbano por meio da análise da integração do transporte público ao planejamento urbano a partir da análise dos Planos Diretores e de Marcon (2022), em que foram realizadas análises baseadas na multimodalidade do transporte coletivo e bicicleta para acesso ao transporte coletivo frente às mudanças na infraestrutura de transportes. Já os trabalhos de Balvedi (2018) e Fert (2017) têm como abordagem a utilização de microssimulador, e avaliação de *softwares* para modelagem de tráfego em interseções urbanas. Os trabalhos utilizam os *softwares* VISSIM, HCS 2010, o Sidra Intersection e o Aimsun.

Ao observar o histórico de trabalhos do PPGTG, é possível notar que a presente dissertação traz como contribuição específica a elaboração de modelos de simulação de passageiros em grande escala utilizando modelos baseados em agentes. Nesse sentido, esta dissertação pode vir a ser subsídio para novas pesquisas na área de planejamento de transportes urbanos com interesse na migração modal para o transporte coletivo, na aplicação de modelos baseados em agentes no planejamento de transportes, e na elaboração de modelos de simulação multimodais, em geral.

2 Revisão bibliográfica

Este capítulo apresenta os conceitos necessários para o entendimento da elaboração de modelos de simulação de tráfego multimodais baseados em agentes em cenário urbano considerando o transporte individual motorizado e o transporte público coletivo por ônibus.

2.1 Planejamento de transportes urbanos

O conteúdo dessa seção contempla a apresentação de conceitos básicos relacionados ao planejamento de transportes urbanos, seguido de um breve histórico dos esforços realizados no âmbito de elaboração de modelos de planejamento de transportes urbanos.

2.1.1 Conceitos básicos

Segundo Campos (2013), o planejamento de transportes visa adequar a oferta de transporte com a demanda presente e futura, abrangendo desde intervenções simples em sistemas existentes até a implantação de novos sistemas. Uma estimativa precisa da demanda e da sua relação com a oferta existente é fundamental para identificar as alternativas mais adequadas para atender às necessidades da população local, e é realizada por meio de elaboração de modelos do sistema de transporte de interesse.

Nesse contexto, os modelos fornecem as ferramentas para descrever e prever os movimentos em um ambiente dado ou possível futuro, segundo Axhausen (2008). Eles estabelecem relações entre as quantidades, localizações, características e comportamentos de pessoas, empresas, infraestruturas, serviços e ambientes para calcular fluxos de movimento por tempo, rota, localização e modo.

O autor ainda pontua que as necessidades de dados variam consideravelmente de acordo com as formas específicas que esses modelos assumem. Elas podem variar desde modelos agregados grosseiros de fluxos de área até microsimulações detalhadas baseadas em agentes de tomadores de decisão individuais. De acordo com Axhausen (2008), a disponibilidade de dados e abordagens de modelagem se condicionam mutuamente. Para tanto, é preciso definir princípios básicos que orientem a elaboração desses modelos.

Ao descrever o processo de elaboração de modelos de planejamento de transporte urbano, Bruton (1979) delinea uma série de hipóteses e princípios, fornecendo as bases para as escolhas metodológicas. Esses princípios fundamentais orientam não apenas a seleção das abordagens de

modelagem, mas também a interpretação dos dados disponíveis e sua aplicação na construção do modelo. São eles:

1. padrões de viagens são tangíveis, estáveis e previsíveis;
2. demandas por movimentos estão diretamente relacionadas com a distribuição e intensidade de usos do solo, que são possíveis de ser exatamente determinados em alguma data futura.

Com a experiência na área, o autor verificou a necessidade de, em adição a estas hipóteses fundamentais, assumir que:

3. relacionamentos decisivos existem entre todos os modos de transportes e que o papel futuro de um particular modo não pode ser determinado sem se considerar todos os outros modos;
4. o sistema de transportes influencia no desenvolvimento de uma área tão bem quanto serve a esta área;
5. áreas de urbanizações contínuas requerem uma ampla consideração regional da situação de transporte;
6. o estudo de transportes é uma parte integrante do processo geral do planejamento e não pode ser considerado adequadamente de forma isolada;
7. o processo de planejamento dos transportes é contínuo e requer constante atualização, monitoramento e aperfeiçoamento.

Sobre a relação entre oferta e demanda, Ortúzar e Willumsen (2011) afirmam que, em oposição à visão do campo econômico dessa relação, o foco na área de transporte muitas vezes não tem sido nos itens monetários, mas sim nos tempos de viagem. Isso porque muitas das questões de demanda que interessam aos analistas de transporte são relacionadas ao impacto no desempenho do sistema de transporte, em vez dos custos monetários. O desempenho do sistema de transporte é traduzido então no custo generalizado de uma viagem, que combina custos financeiros e custos relacionados ao tempo e distância de viagem.

Segundo Bates (2008), a modelagem da "demanda" implica, portanto, em um procedimento para prever quais decisões de viagem as pessoas desejariam tomar, dado o custo generalizado de todas as alternativas. As decisões incluem a escolha do horário da viagem, rota, modo, destino e frequência ou supressão da viagem.

A compreensão acerca desses temas foi desenvolvida ao longo da evolução dos estudos de tráfego, cujo histórico é apresentado a seguir.

2.1.2 Histórico

Desde o século XIX, a necessidade de modelos de transporte tem sido reconhecida por planejadores urbanos, de transporte e pesquisadores, segundo Sivakumar (2007). Inicialmente, foram utilizadas abordagens agregadas, como modelos baseados em entropia e gravidade, para estimar movimentos e fluxos espaciais. No meio do século XX, foi difundido o modelo quatro etapas, ainda amplamente utilizado, para estimar a demanda de viagens. Paralelamente, os planejadores urbanos passaram a reconhecer as interações entre a rede de transporte e o restante do sistema urbano, incluindo influências da configuração do uso do solo e das necessidades de viagem, reguladas por políticas governamentais.

Segundo Melo (1981), desde os anos 1950, foi reconhecida a relação intrínseca entre o uso do solo e os padrões de movimentação, conforme destacado por Mitchell e Rapkin em 1954. Essa percepção orientou a formulação dos primeiros planos diretores de transportes urbanos, como o *Detroit Metropolitan Area Traffic Study*, iniciado em 1953, e o *Chicago Area Transportation Study*, iniciado em 1954, representando avanços significativos nesse campo.

Novais (1981) afirma que, com o avanço da tecnologia, especialmente a partir do advento do computador, os modelos de tráfego e transporte se desenvolveram rapidamente no final dos anos 1950 e ao longo da década seguinte. No entanto, o autor descreve limitações como a falta de retroalimentação nos modelos, que impedem a alteração das condições de ocupação do solo com base em melhorias nos sistemas de transporte. Apesar dessas limitações, diversos estudos e aplicações dos modelos sem retroalimentação foram realizados em todo o mundo, incluindo cidades brasileiras como São Paulo.

Segundo Sivakumar (2007), o primeiro modelo integrado de uso do solo e transporte (US-T) a ganhar destaque foi o modelo de Metropolis de Lowry em 1964. Desde então, vários modelos US-T integrados foram desenvolvidos em todo o mundo. A autora afirma ainda que, enquanto os componentes de uso do solo desses modelos US-T evoluíram rapidamente de representações agregadas simples para modelos econômicos e econométricos complexos dos processos de mercado, o modelo quatro etapas continua a representar o componente fundamental de modelagem de transporte.

A autora descreve que o modelo de demanda de viagem evoluiu muito desde os simples modelos quatro etapas da década de 1960. Avanços na modelagem econométrica levaram à adoção de modelos de escolha desagregados dentro da sua estrutura. Avanços conceituais, como o reco-

nhecimento das ligações entre viagens, resultaram em modelos quatro etapas baseados em *tours* e em atividades.

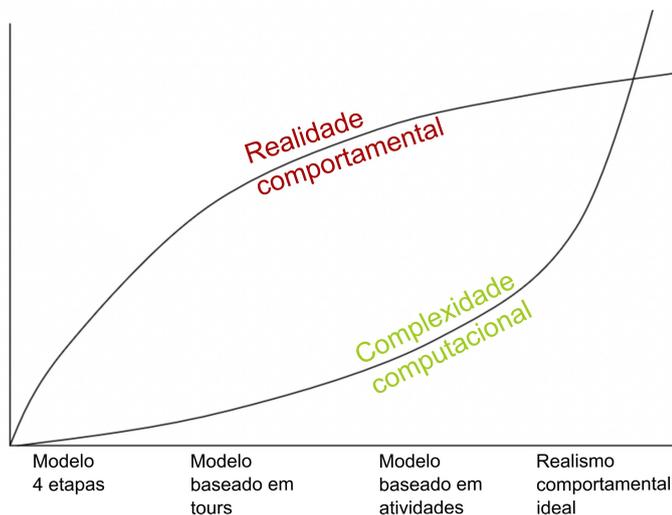
A autora afirma que, com o desenvolvimento da abordagem baseada em atividades para a modelagem da demanda de viagens, ocorreu uma mudança de paradigma. Esta abordagem, um resultado direto do crescimento na pesquisa de comportamento de viagem, reconhece que as necessidades de viagem de um indivíduo são impulsionadas pelo desejo (ou necessidade) de participar de atividades em diferentes locais geográficos.

Os modelos baseados em atividades mudam o foco de uma análise descritiva de fluxos para uma compreensão complexa da tomada de decisão por várias entidades, segundo Sivakumar (2007). Embora exista uma vasta literatura sobre modelagem baseada em atividades e o estado da arte dos modelos baseados em atividades seja bastante avançado, existem poucos modelos de demanda baseados em atividades totalmente operacionais.

Segundo Shiftan e Ben-Akiva (2010), diferentes métodos de pesquisa têm sido usados para estudar o comportamento de atividades, incluindo análise de duração, modelos de variáveis dependentes limitadas, modelos de equações estruturais e modelos de processos computacionais. Diferentes abordagens têm sido usadas para modelos baseados em atividades, mas geralmente seguem uma das duas principais abordagens: análise de escolha discreta (AED) e processo baseado em regras. A abordagem de modelagem de escolha discreta, é considerada pelos autores como a mais prática e, portanto, apontada como a mais comumente usada em modelos regionais reais.

Os autores descrevem que, com base na abordagem de AED, foram propostas estruturas práticas e abrangentes de modelagem de demanda baseada em atividades que captura as decisões de mobilidade, atividade e viagem de indivíduos e famílias. No entanto, essas aplicações atingiram um nível significativo de complexidade, o que pode dificultar seu uso prático. Nesse sentido, os autores ilustram a relação entre o realismo comportamental e a complexidade computacional nos modelos de demanda, conforme consta na Figura 1.

Figura 1: Relação entre o realismo comportamental e a complexidade computacional nos modelos de demanda. Fonte: Shiftan e Ben-Akiva (2010).



Segundo Shiftan e Ben-Akiva (2010), a Figura 1 mostra conceitualmente como a transição de modelos baseados em viagens (quatro etapas) para modelos mais avançados aumenta o realismo comportamental e a complexidade computacional. O custo da complexidade do modelo aumenta exponencialmente, em contraste aos benefícios do realismo comportamental, que aumentam em uma taxa decrescente. A análise das curvas permite identificar um nível ótimo de realismo comportamental que maximiza os benefícios do modelo.

Dada a complexidade e a escala dos atuais problemas no planejamento de transportes, Buisson et al. (2015) indicam que modelos de simulação de tráfego precisos e dinâmicos são cada vez mais utilizados para esses fins, em vez de métodos analíticos.

Buisson et al. (2015) apontam que muitos modelos comerciais de simulação de tráfego estavam disponíveis, e ainda mais modelos foram desenvolvidos por institutos de pesquisa e grupos de pesquisa em todo o mundo. No entanto, os resultados da simulação devem ser interpretados com grande cuidado. Como cada vez mais parâmetros devem ser definidos nos modelos de simulação de tráfego, situações podem ser modeladas de diferentes maneiras e os modelos exibem uma complexidade crescente, as habilidades de um usuário podem afetar a qualidade dos resultados da simulação. Portanto, é importante desenvolver métodos e procedimentos para ajudar desenvolvedores e usuários a aplicar corretamente, efetivamente e com resultados reproduzíveis os modelos de simulação de tráfego.

2.2 Elaboração de modelos de simulação de tráfego

Os modelos de simulação são utilizados para descrição e compreensão do comportamento e da evolução de um sistema, segundo Barceló (2010). Ou seja, um modelo é uma representação simplificada de uma parte do mundo real (o sistema de interesse) que foca em determinados elementos que são considerados importantes a partir de um ponto de vista específico, conforme afirmam Ortúzar e Willumsen (2011).

Com a evolução das tecnologias relacionadas aos computadores, aos *softwares* de engenharia e aos sistemas de transporte inteligente, a simulação de tráfego passou a ser uma abordagem comumente utilizada para análises de tráfego, conforme afirma Barceló (2010).

Essa ferramenta é considerada por Sloboden et al. (2004) como eficiente em avaliar a evolução dinâmica dos problemas de congestionamento de tráfego em sistemas de transporte, além de identificar a interferência dos congestionamentos de um local nos demais locais da rede viária.

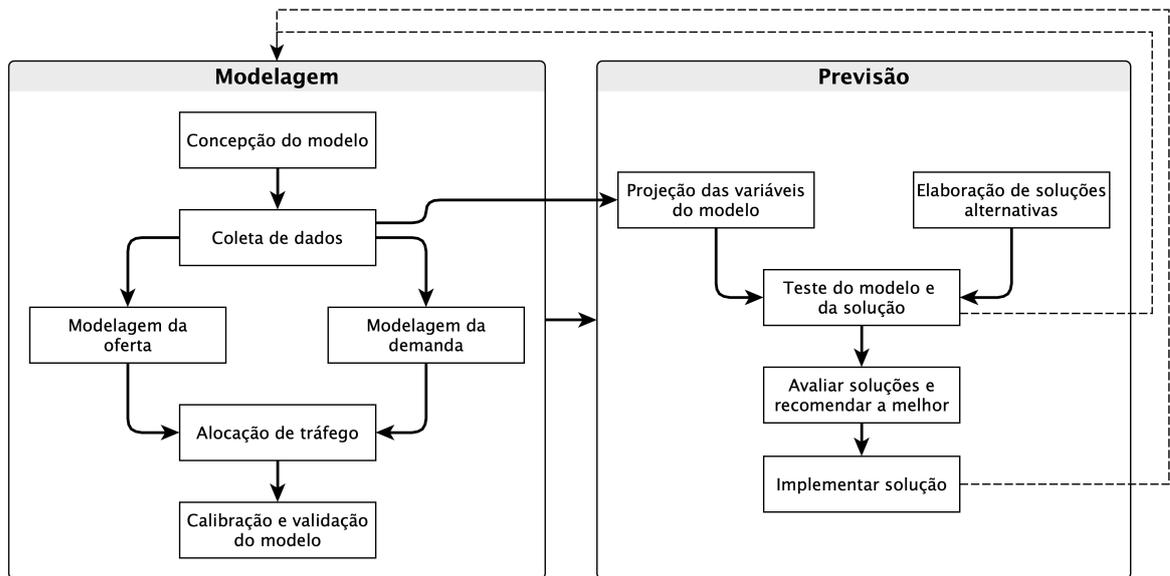
Contudo, Sloboden et al. (2004) mencionam algumas dificuldades relacionadas às ferramentas de simulação. São citadas a necessidade de grande volume de informações como dados de entrada, de extensa verificação de erros nos dados, e de manipulação de grande quantidade de parâmetros de calibração. Os processos de calibração e validação do modelo são necessários para sua aplicação, e podem ser bastante complexos e morosos.

De fato, Ortúzar e Willumsen (2011) afirmam que uma das primeiras tarefas que um modelador enfrenta é decidir quais variáveis serão previstas pelo modelo e quais possivelmente serão necessárias como entradas para ele. Algumas não serão incluídas, seja porque o modelador não tem controle sobre elas, porque os dados relativos a elas não estão disponíveis ou simplesmente porque a teoria por trás do modelo as ignora. Isso implica, para os autores, em um certo grau de erro e incerteza, que é ampliado por outros erros inerentes à modelagem devido às simplificações inevitáveis da realidade que o modelo exige para ser prático.

Os autores apontam ainda que há uma diferença sutil entre modelagem e previsão. A modelagem tem como foco a construção e a aplicação de ferramentas apropriadas que são sensíveis às escolhas de interesse e respondem logicamente às mudanças nos principais instrumentos políticos. A previsão é uma tentativa de visualizar e quantificar condições futuras. Normalmente, envolve a estimativa da demanda de viagens futuras e os fluxos e custos multimodais resultantes ao longo do tempo.

A Figura 2 apresenta uma versão adaptada das etapas de modelagem e previsão de transportes descritas por Ortúzar e Willumsen (2011), e as etapas de modelagem são descritas a seguir.

Figura 2: Etapas da modelagem e previsão de transportes. Fonte: adaptado de Ortúzar e Willumsen (2011).



- **Concepção do modelo.** O modelo tem como objetivo representar um problema, o que pode ser definido como um desajuste entre expectativas e realidade percebida. Dependendo do grau de complexidade do problema, o modelo deve seguir formulações que o representem adequadamente, considerando os tipos de modelos existentes e os dados disponíveis para sua construção.
- **Coleta de dados** sobre o estado atual do sistema de interesse para apoiar o desenvolvimento do modelo no cenário atual e projeções de futuro. Conforme já mencionado por Axhausen (2008) e, de acordo com Ortúzar e Willumsen (2011), a coleta de dados e o desenvolvimento do modelo estão intimamente correlacionados.
- **Construção de um modelo analítico do sistema de interesse, incluindo:**
 - modelagem de demanda que, no modelo quatro etapas inclui as etapas de geração, distribuição e divisão modal;
 - modelagem da oferta;
 - modelagem do desempenho do sistema (oferta) em relação à demanda por meio da alocação de tráfego;
 - calibração e validação do modelo.

As etapas envolvidas nesse processo são detalhadas nas seções a seguir.

2.2.1 Concepção do modelo

A aceitabilidade da modelagem, ou de uma abordagem de modelagem específica, dentro de um estilo de tomada de decisão é muito importante. Nesse sentido, Ortúzar e Willumsen (2011) listam várias características dos problemas de transporte e modelos que devem ser levadas em consideração ao especificar uma abordagem, descritas a seguir.

1. Precisão e exatidão necessárias. A exatidão é o grau em que uma medição ou resultado do modelo corresponde a valores verdadeiros ou aceitos. A exatidão é uma questão relacionada à qualidade dos dados e do modelo. A precisão refere-se ao nível ou unidades de medição usados para coletar dados e fornecer resultados do modelo.
2. O contexto da tomada de decisão. Isso envolve a adoção de uma perspectiva particular e a escolha de um escopo ou cobertura do sistema de interesse. A escolha de perspectiva define os requisitos dos modelos a serem usados, as variáveis a serem incluídas no modelo ou consideradas dadas ou exógenas.
3. Nível de detalhe necessário. O nível de resolução de um sistema de modelo pode ser descrito ao longo de quatro dimensões principais: geografia, unidade de análise, respostas comportamentais e tratamento do tempo.
4. A disponibilidade de dados adequados, sua estabilidade e as dificuldades envolvidas na previsão de seus valores futuros. Em muitos casos, os dados disponíveis serão o fator chave na decisão da abordagem de modelagem.
5. O estado da arte em modelagem para um tipo particular de intervenção no sistema de transporte. Isso, por sua vez, pode ser subdividido em: riqueza comportamental, praticidade matemática e computacional e disponibilidade de bons algoritmos de solução.
6. Recursos disponíveis para o estudo. Estes incluem dinheiro, dados, hardware e software de computador, habilidades técnicas, e assim por diante.
7. Requisitos de processamento de dados. Com o atual desenvolvimento dos recursos computacionais, o gargalo real no processamento de dados é a capacidade humana de coletar, codificar, inserir os dados, executar os programas e interpretar os resultados.
8. Níveis de treinamento e habilidades dos analistas.
9. Perspectiva e escopo de modelagem.

Os dois últimos têm mais a ver com as questões microeconômicas que afetam os fornecedores em sua escolha de combinações de insumos para minimizar custos.

Outras questões surgem na concepção do modelo, como a especificação do modelo. Ortúzar e Willumsen (2011) descrevem que, em um sentido amplo e mais interessante, essa questão considera os temas expostos a seguir.

- **Estrutura do Modelo.** É possível replicar o sistema a ser modelado com uma estrutura simples que assume, por exemplo, que todas as alternativas são independentes? Ou é necessário construir modelos mais complexos que procedem, por exemplo, ao cálculo de probabilidades de escolha condicionadas a seleções anteriores?
- **Forma Funcional.** É possível usar formas lineares ou o problema requer a postulação de funções não lineares mais complexas? As últimas podem representar o sistema de interesse com mais precisão, mas certamente serão mais exigentes em termos de recursos e técnicas para a calibração e uso do modelo.
- **Especificação da Variável.** Este é o significado mais comum atribuído à questão da especificação; quais variáveis usar e como (em que forma) elas devem entrar em um determinado modelo.

Dependendo das respostas para essas questões, devem ser escolhidas diferentes ferramentas de análise de tráfego. Law e Kelton (1999) as classificam em:

- **estática ou dinâmica:** modelos de simulação estática representam um sistema em um instante específico, desconsiderando, assim, o papel do tempo na simulação. Por outro lado, os modelos de simulação dinâmicos refletem um sistema à medida que o mesmo sofre alterações ao longo do tempo de simulação;
- **determinística ou estocástica:** modelos de simulação determinísticos são isentos de qualquer variável probabilística, em contrapartida, os modelos de simulação estocásticos consideram uma ou mais variáveis aleatórias;
- **discreta ou contínua:** modelos de simulação discreta são caracterizados pela contabilização do tempo em forma de incrementos, de modo que o sistema sofra alterações apenas em um número contável de instantes no tempo. Em alternativa, nos modelos de simulação contínua, tal contabilização ocorre de forma contínua, o que viabiliza a visualização dos valores das variáveis do estado a qualquer momento da simulação.

Além dessas especificações, Sloboden et al. (2004) indicam que a ferramenta de simulação indicada deve ser compatível com os objetivos do modelo em relação à capacidade de analisar o escopo geográfico, de modelar os diferentes elementos da rede viária, de analisar os modos de

transporte de interesse, de analisar estratégias de controle de tráfego, de estimar as respostas dos viajantes (incluindo desvio de rota, escolha do horário de partida, mudança de modo, escolha de destino e demanda induzida/renunciada), de produzir e fornecer medidas de desempenho, e de efetividade da ferramenta/custo para a tarefa, principalmente sob uma perspectiva de gestão ou operacional.

Outra questão importante nesse contexto é a da agregação. De fato, a questão é tópico de um subcapítulo de Ortúzar e Willumsen (2011), abordando as diferenças entre os modelos. Algumas das questões centrais abordadas são:

“Quantos estratos populacionais ou tipos de pessoas precisamos para obter uma boa representação e compreensão de um problema? Com que detalhe precisamos medir certas variáveis para replicar um fenômeno dado? O espaço é crucial no transporte; em que nível de detalhe precisamos codificar a origem e o destino dos viajantes para modelar seu comportamento de viagem?” (ORTÚZAR; WILLUMSEN, 2011)

Segundo Ortúzar e Willumsen (2011), os modelos agregados foram usados quase sem exceção em estudos de transporte até o final dos anos 1970. Eles se tornaram familiares, exigiam relativamente poucas habilidades por parte do analista e tinham a propriedade de oferecer uma “receita” para o processo completo de modelagem. Nesse sentido, Wise, Crooks e Batty (2017) afirmam que os primeiros modelos quatro etapas eram altamente agregados no sentido de que simulam fluxos totais entre os ditos origens e destinos, em diferentes modos de redes de trânsito e seus segmentos, geralmente no horário de pico, dividindo a cidade em pequenas zonas. Conforme foram desenvolvidos, passaram a ser desagregados em termos de tipos de viagem e categorias de viajantes.

Wise, Crooks e Batty (2017) apontam que o desenvolvimento da modelagem de escolha discreta na década de 1970 deu origem a modelos de microssimulação de tráfego. Essencialmente, esses modelos modelam as decisões das famílias ou dos tomadores de decisão de viagem em famílias com relação aos seus padrões de atividade diária, representando a cidade em detalhes completos com todos os agentes, ou uma porcentagem amostral. Nestes modelos, os agentes novamente não se comunicam, influenciando uns aos outros indiretamente através da congestão que geram um para o outro com seu custo resultante para o orçamento de atividades da família. Segundo Wise, Crooks e Batty (2017), o melhor exemplo desses tipos de modelos é o MATSIM, que foi desenvolvido

a partir da iniciativa dos EUA no modelo TRANSIMS no início dos anos 1990 (HORNI; NAGEL; AXHAUSEN, 2016).

Os modelos desagregados, que se tornaram cada vez mais populares durante os anos 1980, oferecem vantagens substanciais sobre os métodos tradicionais, ao mesmo tempo que permanecem práticos em muitos estudos de aplicação, como afirmam Ortúzar e Willumsen (2011). No entanto, um problema importante na prática é que eles demandam do analista um nível bastante elevado de habilidades estatísticas e econométricas para seu uso (em particular para a interpretação dos resultados), certamente muito maior do que no caso dos modelos agregados.

Os tipos de modelos existentes de acordo com seu nível de agregação são abordados ao longo das Seções 2.2.4 e 2.2.5.

2.2.2 Coleta de dados

Segundo McNally (2000), na área da pesquisa em transporte, nada é mais crucial, porém ao mesmo tempo mais restritivo para a validação de teorias e modelos do que a coleta de dados. Em diversas aplicações, são as limitações de tempo e custo que restringem a capacidade de coletar os dados necessários na pesquisa.

De acordo com Ortúzar e Willumsen (2011), a questão dos dados está intimamente ligada a questões como o tipo de variáveis a serem representadas no modelo e isso, é claro, está novamente intimamente ligado a questões sobre teoria. Os modelos preveem uma série de variáveis dependentes (ou endógenas) dadas outras variáveis independentes (ou explicativas). Para testar um modelo, normalmente precisaríamos de dados sobre cada variável, principalmente aquelas que o analista pode alterar para testar o valor de políticas ou medidas alternativas.

De forma geral, Axhausen (2008) lista diversos tipos de dados necessários para o planejamento de transporte, que incluem inventários detalhados de objetos e suas características, como a localização e as propriedades das vias, das rotas e dos demais atributos de serviços de ônibus. Além disso, são necessários dados provenientes de censos populacionais, obtidos por meio de pesquisas primárias ou outras fontes, e dados sobre comportamento atual, coletados por meio de observações ou pesquisas de viajantes.

Em relação aos modelos baseados em atividades, sua diversidade teórica e metodológica impede uma resposta definitiva sobre quais dados são necessários, dependendo da abordagem e

aplicação específicas. Arentze et al. (2000) descrevem os dados considerados universais nessas modelagens em quatro dimensões.

A primeira dimensão considera as atividades, sobre a qual Arentze et al. (2000) indicam que há evidências empíricas sugerindo que os dados sobre atividades devem ser coletados em um nível relativamente detalhado. E que isso também é verdadeiro para a segunda dimensão: a dimensão espacial ou localização das atividades. Pressupondo que os fluxos de tráfego reais são importantes, são necessários dados geocodificados sobre o uso do solo e a localização de várias instalações onde atividades específicas podem ser realizadas.

A terceira dimensão diz respeito ao tempo. Segundo Arentze et al. (2000), os modelos baseados em atividades incorporam o momento das decisões de viagem. Conseqüentemente, dados sobre horários de início e término, e portanto duração, de atividades e viagens relacionadas devem ser coletados. A última dimensão, a do modo de transporte, implica que dados sobre qual modo de transporte é usado para realizar as várias atividades devem ser coletados. Para tanto, devem ser coletados dados sobre variáveis que refletem a atratividade dos modos. Podem ser utilizadas medidas de custos, frequência, conforto, tempo de viagem e outros descritores comumente usados de atratividade.

Para além da disponibilidade dos dados, uma das preocupações na elaboração de um modelo é em relação à qualidade dos dados. Arentze et al. (2000) afirmam que um aspecto fundamental para a qualidade dos dados em modelos de transporte é a confiabilidade, que se refere à consistência na medição dos dados coletados. A confiabilidade é crucial, especialmente ao classificar atividades ou intervalos de tempo, onde diferentes abordagens podem levar a inconsistências. A qualidade dos dados pode ser aprimorada com instruções detalhadas e consistentes de codificação e classificação.

Além da confiabilidade, o autor descreve que a validade dos dados é essencial, garantindo que as variáveis selecionadas realmente meçam o conceito pretendido. Isso é crucial para garantir que os dados capturem os padrões de atividade representativos ao longo de todas as dimensões identificadas pelo modelo.

Outro requisito importante para Arentze et al. (2000) é o nível de detalhe dos dados, que deve estar alinhado com o objetivo do modelo. Por exemplo, modelos que exigem previsões detalhadas ao longo do dia precisam de dados igualmente detalhados sobre atividades e localização.

Finalmente, para garantir o uso eficiente e eficaz dos dados, o autor afirma que eles devem ser facilmente acessíveis, dinâmicos, flexíveis e integrados. Isso implica que os dados devem ser

disponibilizados em formatos modernos e atualizados continuamente para refletir as mudanças na realidade. A flexibilidade e integração dos dados são essenciais para a construção de modelos precisos e relevantes para a demanda de transporte.

No contexto brasileiro, o cadastro adequado de dados necessários para uma simulação de tráfego é raro, e é comum que analistas ou engenheiros precisem dedicar tempo para resolver problemas relacionados à qualidade dos dados incompletos ou desatualizados. Esses problemas geralmente só são identificados por meio de uma análise minuciosa dos dados, e as soluções envolvem, em primeira tentativa, correções sistemáticas. No entanto, é possível que isso não seja suficiente para alcançar resultados satisfatórios de validação.

2.2.3 Modelagem da oferta

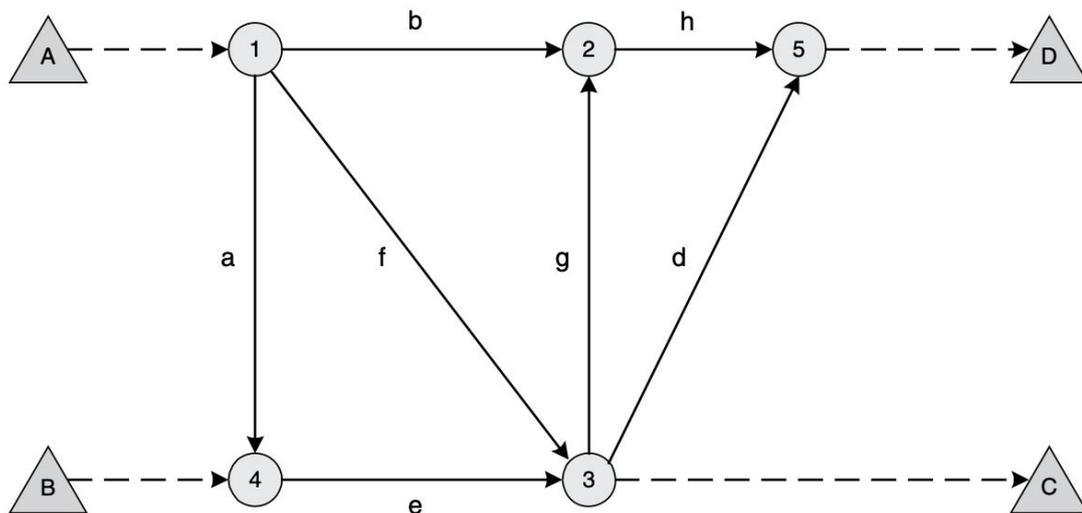
Na modelagem da oferta, assim como em outros elementos-chave da modelagem de transporte, Ortúzar e Willumsen (2011) afirmam que as escolhas na elaboração do modelo refletem um compromisso entre dois objetivos conflitantes: precisão e custo.

Segundo Hensher e Button (2008), as origens e os destinos podem corresponder a uma edificação específica (casas, lojas ou escritórios) ou a zonas tráfego, de acordo com o nível de agregação. Ortúzar e Willumsen (2011) afirmam que o sistema de zoneamento é usado para agregar os domicílios e instalações individuais em blocos gerenciáveis para fins de modelagem. Nesse sentido, os autores descrevem que as zonas são representadas nos modelos de computador como se todos os seus atributos e propriedades estivessem concentrados em um único ponto chamado centroide da zona. Os centróides são conectados à rede por meio de conectores que representam os custos médios (tempo, distância) de adesão ao sistema de transporte para viagens com origem ou destino naquela zona. Quase tão importante quanto o custo associado a cada conector de centróide é o nó na rede ao qual ele se conecta. Estes devem estar próximos aos pontos naturais de acesso/saída para a própria zona.

Segundo Ortúzar e Willumsen (2011), a rede de transporte é considerada um componente-chave do lado da oferta do esforço de modelagem, ou seja, o que o sistema de transporte oferece para satisfazer as necessidades de movimento dos viajantes na área de estudo. Hensher e Button (2008) complementam, indicando que a rede de transporte é uma construção analítica que facilita a identificação das rotas seguidas pelos viajantes e seus correspondentes “custos”.

Hensher e Button (2008) indicam que as redes de viagem podem ser representadas formalmente como um conjunto de *links* L e nós N . Um *link* conecta dois nós e um nó conecta dois ou mais *links*. A direção da viagem geralmente é especificada em um *link* e, nesse caso, são chamados de *links* direcionados. Um *link* não direcionado pode ser percorrido em ambas as direções. Ortúzar e Willumsen (2011) indicam que a prática comum é modelar a rede como grafo direcionado, ou seja, um sistema de nós e *links* que os unem, onde a maioria dos nós representam cruzamentos e os *links* representam trechos homogêneos de estrada entre cruzamentos. Essa representação é ilustrada na Figura 3.

Figura 3: Um grafo representando as origens (A e B), os destinos (C e D), os nós (1-5) e os *links* (a-h). Fonte: Hensher e Button (2008).



Além da conectividade, Hensher e Button (2008) indicam que um *link* pode ser representado com várias outras características úteis em redes de transporte, como seu comprimento (geralmente em metros, quilômetros ou outra unidade adequada), seu custo (itens como tempo de viagem e distância, mas geralmente uma combinação ponderada de tempo, distância e alguma outra propriedade relevante da infraestrutura), e sua capacidade (o fluxo máximo que pode passar por esse link por unidade de tempo).

O nível de desagregação pode ser aumentado ainda mais quando são usados modelos detalhados de simulação de tráfego. O nível de detalhe fornecido sobre os atributos dos *links* depende da resolução geral da rede e do tipo de modelo usado. (ORTÚZAR; WILLUMSEN, 2011)

Segundo Ortúzar e Willumsen (2011), as redes de transporte público são mais complexas do que as redes viárias. Elas exigem a identificação da rota seguida por cada serviço como uma

sequência única de *links*. Também é necessário identificar os locais onde as paradas são possíveis e também aqueles onde a transferência para outros serviços é permitida. A frequência do serviço e, em alguns casos, o horário real e a tarifa, também devem ser especificados e incluídos na descrição da rede. O acesso às paradas pode ser a pé ou por outro modo e isso pode ser representado por conectores de centróides nos modelos mais simples e por um ou mais redes auxiliares de modos de acesso em modelos mais realistas. Além do efeito do congestionamento viário nas simulações multimodais, às vezes é necessário considerar a questão do congestionamento de passageiros: lotação de ônibus e trens levando ao desconforto e até mesmo à impossibilidade de pegar um serviço porque estava lotado e impossível de embarcar.

2.2.4 Geração, distribuição e divisão modal de viagens

As etapas de geração, distribuição e divisão modal de viagens correspondem às etapas de modelagem da demanda anteriores à alocação dessas viagens na rede viária. Essa seção objetiva apresentar brevemente alguns métodos possíveis para cada uma dessas etapas, diferenciando-os em relação ao seu nível de agregação.

Em relação à geração de viagens, Ortúzar e Willumsen (2011) e Campos (2013) descrevem os métodos agregados de regressão linear, fator de crescimento, classificação cruzada, taxa de viagens e também o método logit que é particularmente apropriado para lidar com dados desagregados, apesar de ser possível aplicá-lo com dados agregados.

Ortúzar e Willumsen (2011) descreve como modelos agregados os seguintes modelos de distribuição de viagens: fator de crescimento, gravitacionais e maximização da entropia. Tradicionalmente, esses modelos são definidos em termos da capacidade de produção e atração de viagens das zonas de tráfego. No entanto, é possível aumentar o nível de desagregação desses modelos considerando diferentes perfis de usuários e diferentes motivos de viagem. Contudo, esse nível de desagregação pode não ser suficiente para capturar os efeitos complexos do comportamento de viagens. Para modelagem desagregada a nível do indivíduo, é necessário o uso de modelos probabilísticos, como um modelo logit multinomial.

Os autores declaram que, no que diz respeito à divisão modal, os modelos de escolha de modo podem ser agregados se baseados em informações zonais (e interzonais). Também é possível ter modelos desagregados se forem baseados em dados domiciliares e/ou individuais.

Os modelos de divisão modal *trip-end*, conforme descrevem Ortúzar e Willumsen (2011), são aplicados imediatamente após a geração de viagens, e visam preservar e utilizar as diferentes características dos indivíduos para estimar a divisão modal. No entanto, esses modelos omitem as características espaciais da viagem e dos modos, reproduzindo uma abordagem geral de planejamento que não considera a influência de políticas no comportamento da demanda.

Na Europa, os autores afirmam que o método mais comum se tornou a modelagem de divisão modal após a distribuição de viagens, chamados de divisão modal *trip interchange heuristics*. Isso facilitava a inclusão das características da viagem e dos modos alternativos disponíveis para realizá-las. No entanto, tornava mais difícil incluir as características do viajante, pois estas poderiam já ter sido agregadas na matriz (ou matrizes) de viagens.

Ainda segundo Ortúzar e Willumsen (2011), os modelos de demanda direta oferecem uma abordagem alternativa à metodologia sequencial convencional, pois permitem a estimativa simultânea de geração de viagens, distribuição e escolha de modo. Esses modelos podem ser usar uma única equação para todas as etapas, ou uma abordagem quase direta, que separa a divisão modal da demanda total de viagens.

Em relação ao nível de agregação dos modelos, Ortúzar e Willumsen (2011) indica que modelos de transporte de demanda agregada são baseados em relações observadas para grupos de viajantes ou em relações médias no nível da zona. Por outro lado, modelos de demanda desagregada são baseados em escolhas observadas feitas por viajantes individuais ou domicílios. A adoção dessa estrutura permite o desenvolvimento de modelos mais realistas, superando as deficiências dos modelos agregados convencionais.

Nesse sentido, segundo os autores, os modelos de escolha discreta postulam que a probabilidade de os indivíduos escolherem uma opção dada é uma função de suas características socioeconômicas e da atratividade relativa da opção. A teoria da utilidade aleatória é o paradigma mais comum para gerar esses modelos, postulando que os indivíduos agem de forma racional e possuem informações perfeitas, mas enfrentam uma incerteza aleatória ao fazer escolhas. O Modelo Logit Multinomial é o mais simples e popular modelo prático de escolha discreta, enquanto o Modelo Logit Aninhado agrupa opções correlacionadas em hierarquias. O Modelo Probit Multinomial e o Modelo Logit Misto oferecem abordagens alternativas para lidar com a incerteza aleatória na escolha discreta.

2.2.5 Alocação de tráfego

A premissa básica na alocação de tráfego é a suposição de um viajante que escolhe a rota com os menores custos individuais percebidos (e antecipados), de acordo com Ortúzar e Willumsen (2011). Vários fatores são considerados influentes na escolha da rota ao dirigir entre dois pontos, e a produção de uma expressão de custo generalizada que incorpore todos esses elementos é uma tarefa difícil. Além disso, não é prático e pode não ser possível tentar modelar todos eles, e portanto aproximações são inevitáveis.

A aproximação mais comum, segundo os autores, é considerar apenas dois fatores na escolha da rota: tempo e custo monetário, sendo que o custo monetário muitas vezes é considerado proporcional à distância de viagem. A maioria dos programas de alocação de tráfego permite que o usuário atribua pesos ao tempo de viagem e à distância para representar as percepções dos motoristas sobre esses dois fatores. A soma ponderada desses dois valores então se torna um custo generalizado usado para estimar a escolha da rota. Há evidências que sugerem que, pelo menos para o tráfego urbano de carros, o tempo é o fator dominante na escolha da rota.

De acordo com Ortúzar e Willumsen (2011), o fato de diferentes motoristas frequentemente escolherem rotas diferentes ao viajar entre os mesmos dois pontos pode ser atribuído a três motivos:

1. diferenças nas percepções individuais do que constitui a 'melhor rota': alguns podem desejar minimizar o tempo, outros o consumo de combustível e muitos uma combinação de ambos, introduzindo uma variedade de escolhas de rota;
2. o nível de conhecimento de rotas alternativas varia e isso introduz uma aparente irracionalidade (do ponto de vista do observador) nas escolhas;
3. efeitos de congestionamento afetando rotas mais curtas primeiro e tornando seus custos generalizados comparáveis a rotas inicialmente menos atraentes.

Normalmente lidamos com o primeiro problema por meio de múltiplas classes de usuários, o segundo por meio de 'efeitos estocásticos' e o terceiro por meio de alocação e equilíbrio congestionados.

Certos tipos de modelos são mais adequados para representar uma ou mais das influências acima. Os detalhes e características de cada método são discutidos abaixo.

As funções básicas da alocação são identificar um conjunto de rotas que podem ser consideradas atraentes para os motoristas (estágio de construção de árvores), atribuir proporções adequadas

da matriz de viagens a essas rotas ou árvores, resultando em fluxos nos *links* da rede e, por fim, procurar por convergência. Muitas técnicas seguem um padrão iterativo de sucessivas aproximações a uma solução ideal. A convergência para esta solução deve ser monitorada para decidir quando interromper o processo iterativo.

Ortúzar e Willumsen (2011) descreve que as técnicas de atribuição congestionada tentam se aproximar do verdadeiro equilíbrio de Wardrop, e que os métodos heurísticos frequentemente falharam em alcançar esse tipo de equilíbrio.

Ortúzar e Willumsen (2011) destaca que encontrar uma solução de Equilíbrio de Usuário Dinâmico (DUE) para um conjunto de volumes e tempos de viagem de *link* e rota variáveis no tempo que satisfaçam o equilíbrio de Wardrop para uma determinada rede e padrão de demanda variável no tempo não é trivial. A escolha da melhor rota de cada viajante depende dos níveis de congestionamento ao longo da viagem, e estes, por sua vez, dependem das escolhas de rota e do progresso pela rede de outros viajantes que partem em diferentes momentos. Essa interdependência significa que as soluções são encontradas por meio de um processo iterativo, começando de um conjunto inicial de escolhas de rota e gradualmente melhorando-as. Um objetivo prático de muitos modelos de alocação dinâmica de tráfego (DTA) atuais é encontrar algo próximo ao equilíbrio dentro de um período de tempo razoável.

As alocações dinâmicas de tráfego atuam com resolução mais desagregada de dados de oferta, e também de demanda. Esses modelos são caracterizados por Heydecker e Addison (2005) de acordo com os requisitos básicos para serem considerados verdadeiramente dinâmicos. São eles:

- positividade: são considerados apenas fluxos não negativos em links, caminhos, matrizes de viagens e custos;
- conservação: o modelo deve satisfazer requisitos de conservação de fluxo;
- FIFO: no tráfego real, o comportamento FIFO (Primeiro a Entrar, Primeiro a Sair, do termo inglês *First In, First Out*) geralmente prevalece e isso deve ser mantido no modelo se atrasos adequados forem estimados;
- tempo mínimo de viagem: os fluxos não se propagam instantaneamente;
- tempo de limpeza finito: não devem restar filas no final do período de modelagem, ou seja, atrasos infinitos não ocorrem;
- capacidade: os fluxos reais não podem exceder a capacidade mesmo por um curto período de tempo;

- causalidade: os atrasos agora são afetados pelo que outros veículos fazem ou fizeram no passado, não no futuro.

Segundo Ortúzar e Willumsen (2011), várias abordagens diferentes foram tentadas ao longo dos anos para fornecer maneiras práticas de resolver esses problemas de forma não analítica. À medida que o poder computacional aumentou, novos e melhores softwares levaram a soluções mais interessantes. Os autores destacam as abordagens de meso e microsimulação. Os modelos de mesossimulação vieram primeiro, e são considerados os modelos intermediários entre os modelos micro e macroscópicos. Sua abordagem consiste na simulação de escolha de rota por meio de um grupo de veículos liberados de forma conjunta e sequencial durante um período de tempo, já que tratá-los um por um não era possível na época.

A descrição dos modelos de alocação por meio de simulação de acordo com seus níveis de agregação são apresentadas a seguir, de acordo com Sloboden et al. (2004), Barceló (2010) e Treiber e Kesting (2013).

- Modelos macroscópicos: modelos agregados que consideram os fluxos de tráfego como análogos ao fluxo de líquidos ou gases e, por isso, eventualmente são chamados de modelos hidrodinâmicos. São baseados nas relações determinísticas do fluxo, velocidade e densidade do tráfego, ou seja, relações agregadas de velocidade/volume e demanda/capacidade. A simulação em um modelo macroscópico ocorre seção por seção, em vez de rastrear veículos individuais.
- Modelos microscópicos: modelos totalmente desagregados que descrevem o processo de um fluido das dinâmicas das partículas individuais que a compõe. Esses modelos podem ser simulações físicas, com modelos detalhados de *car-following*, simulações de autômato celular, em que as vias são discretizadas em células, ou simulações baseadas em filas (*queue-based*), onde a dinâmica do tráfego é modelada com filas de espera. Eles rastreiam veículos através da rede de transporte em intervalos de tempo breves e são influenciados por características da via. A calibração e validação desses modelos são realizadas ajustando fatores de sensibilidade do motorista. Os modelos microscópicos têm altos requisitos de tempo e armazenamento computacional, e aplicações usuais possuem limitações bastante restritas relacionados ao tamanho da rede (limitada à interseções, por exemplo) e ao número de simulações que podem ser concluídas.
- Modelos mesoscópicos: modelos intermediários baseados na simplificação das dinâmicas veiculares e são comumente utilizados quando não há informações suficientes para modelagem microscópica, necessitando de inferências baseadas na modelagem macroscópica ou vice-versa. Eles operam com

base no fluxo de tráfego de veículos individuais, mas não consideram relações dinâmicas de velocidade/volume. Embora forneçam menos detalhes que os modelos de microsimulação, os modelos mesoscópicos são superiores às técnicas analíticas de análise de planejamento.

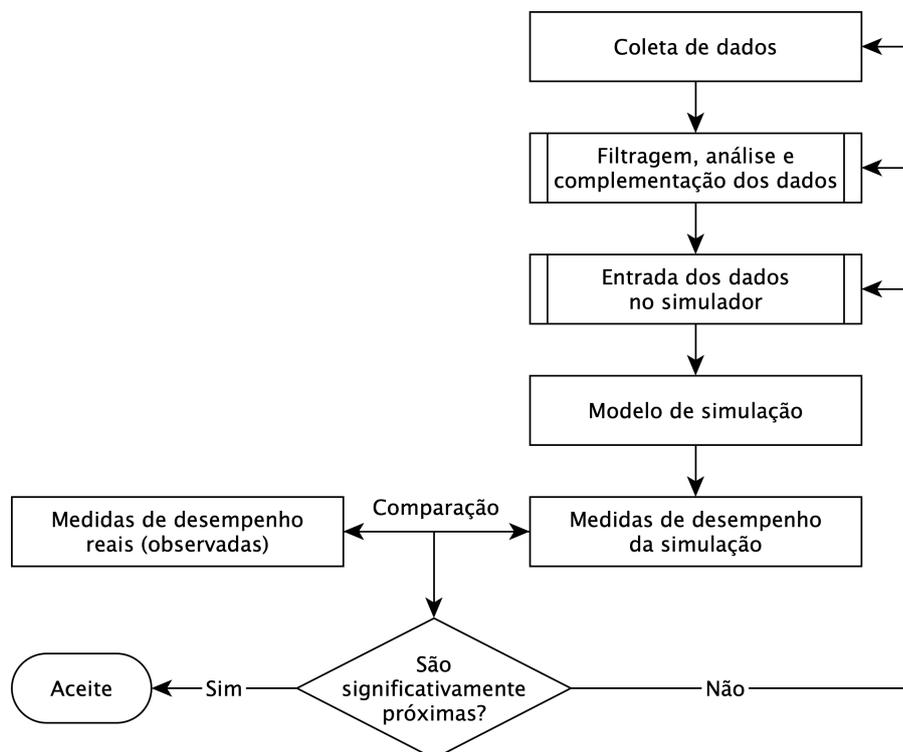
Segundo Treiber e Kesting (2013), os modelos macroscópicos são úteis quando efeitos microscópicos não precisam ser considerados (mudança de faixas, vários tipos de veículos), quando só se está interessado nas quantidades macroscópicas, quando o tempo de processamento computacional é crítico ou se os dados de entrada disponíveis são inconsistentes ou de diferentes fontes, e é necessária a fusão dos dados.

Os modelos microscópicos são utilizados, segundo Treiber e Kesting (2013), quando a modelagem de veículos individuais afeta o tráfego (como veículos autônomos ou sob demanda), em situações em que a heterogeneidade do tráfego é importante (como a simulação dos efeitos de limites de velocidade ou restrição à veículos de carga), na descrição do comportamento humano de direção, incluindo tempo de reação, falta de atenção e antecipação ou na visualização da interação entre diferentes participantes do tráfego (carros, caminhões, ônibus, ciclistas, pedestres).

2.2.6 Calibração e validação do modelo

A simulação pressupõe que a evolução do modelo do sistema imita adequadamente a evolução do sistema real ao longo do tempo. Assim, Barceló (2010) indica que informações que dizem respeito às variáveis observacionais de interesse devem ser coletadas. A partir dessa coleta, são realizadas análises da proximidade do modelo ao sistema real usando técnicas de análise estatística, no processo denominado validação, e aplicados ajustes ao modelo para que atinjam o nível de proximidade desejado, no processo denominado calibração. A Figura 4 ilustra a sequência metodológica para a calibração e a validação de modelos de simulação apresentada por Barceló (2010).

Figura 4: Sequência metodológica para calibração e validação do modelo. Fonte: Barceló (2010)



Segundo o autor, a metodologia pressupõe que, após o tratamento dos dados coletados do sistema, a eliminação das medições não confiáveis e a imputação dos dados faltantes, um conjunto de dados é obtido para a construção de um modelo de dados de entrada adequado para o modelo de simulação.

A validação é definida por Barceló (2010) como a confrontação dos dados simulados com os dados observados do sistema, usando métodos de análise estatística para verificar se as amostras de dados são suficientemente similares. Se sim, o modelo de simulação é validado, se não, é invalidado e os processos de calibração e validação devem ser refeitos. Isso requer um procedimento iterativo no qual, conforme o caso, mais ou novos dados devem ser coletados, o tratamento de dados deve ser revisto, ou o modelo de dados de entrada deve ser modificado ou aprimorado.

Esses processos fazem parte da etapa de calibração de um modelo. Segundo Barceló (2010) e Ortúzar e Willumsen (2011), essa etapa envolve a seleção dos parâmetros e seus valores, assumidos como não nulos, que resultarão em um modelo válido, avaliado por meio de uma ou mais medidas de ajuste do modelo que dependem dos dados observados.

Para Barceló (2010), a calibração e a validação de modelos de simulação ainda são um grande desafio no uso de simulação para fins práticos, especialmente no caso de modelos de simulação

de tráfego microscópico que combinam o alto nível de incerteza do sistema modelado com um grande número de parâmetros, alguns dos quais consideram aspectos comportamentais do sistema veículo-motorista.

Barceló (2010) afirma que todas as diretrizes metodológicas coincidem em recomendar a decomposição do principal problema de calibração em subproblemas para resolvê-lo de forma mais eficiente, levando em consideração a natureza diferente dos parâmetros a serem calibrados. Essa prática geralmente é de grande ajuda para determinar os intervalos prováveis de variabilidade dos parâmetros e constitui uma entrada útil para procedimentos simultâneos. De acordo com essas recomendações, as diretrizes de Dowling, Skabardonis e Alexiadis (2019) estruturam o processo em quatro etapas, listadas a seguir.

1. Verificação de erros: a rede de transporte codificada e os dados de demanda são revisados em busca de erros. Este passo é necessário para eliminar erros de codificação antes de prosseguir com a calibração.
2. Calibração de capacidade: uma calibração inicial é realizada para identificar os valores dos parâmetros de ajuste de capacidade que fazem com que o modelo reproduza melhor as capacidades de tráfego observadas no campo. Uma calibração global é realizada primeiro, seguida por ajustes finos específicos do *link*. O Highway Capacity Manual (HCM) pode ser usado como fonte alternativa para definição dos valores de capacidade viária se as medições de campo forem inviáveis.
3. Calibração de escolha de rota: se o modelo de microsimulação incluir ruas paralelas, a escolha da rota será importante. Nesse caso, um segundo processo de calibração é realizado, mas desta vez com os parâmetros de escolha de rota. Uma calibração global é realizada primeiro, seguida por ajustes finos específicos do *link*.
4. Validação de desempenho: finalmente, as estimativas gerais do modelo de desempenho do sistema (tempos de viagem e filas) são comparadas às medições de campo dos tempos de viagem e filas. Ajustes finos são feitos para permitir que o modelo corresponda melhor às medições de campo.

O autor descreve que estes passos não precisam ser seguidos de forma sequencial: uma vez que o passo 1 tenha sido alcançado satisfatoriamente, os passos 2-4 podem ser repetidos iterativamente no caso de não haver correspondência aceitável dos critérios de comparação entre as medidas selecionadas de desempenho e os dados de campo. O uso de uma ferramenta de simulação sem calibração e validação levanta questões sobre a capacidade preditiva da ferramenta.

Segundo Buisson et al. (2015), a literatura inclui várias abordagens para a validação estatística dos modelos de simulação. Essas abordagens envolvem medidas de bondade de ajuste,

intervalos de confiança e testes estatísticos das distribuições e processos subjacentes. Porém, elas podem não ser aplicáveis quando os processos de tráfego reais e simulados são não estacionários e autocorrelacionados. A escolha e a aplicação dos métodos adequados para a validação de modelos de simulação de tráfego dependem da natureza dos dados de saída. Os métodos e as saídas considerados são:

- medidas de desempenho de ponto único (por exemplo, atraso médio, volume de tráfego total na rede);
- medidas de desempenho multivariadas (por exemplo, medições de fluxo ou velocidade dependentes do tempo em diferentes locais, tempos de viagem em diferentes seções).

Em relação às medidas de bondade de ajuste dos dados utilizadas, Hollander e Liu (2008) apresentam uma revisão crítica acerca de suas possibilidades, conforme citam Buisson et al. (2015). Eles concluem que as medidas que dependem das diferenças ao quadrado penalizam grandes erros, o que é mais adequado, pois penalizar erros pequenos seria errado já que pequenas variações em torno da média são parte da natureza dos fenômenos de tráfego. Entre as medidas mais usadas, destacam-se: o erro quadrático médio ou RMSE (Root Mean Squared Error), que quantifica o erro geral; a percentagem do erro quadrático médio ou RMSP (Root Mean Squared Percent Error), que fornece informações sobre a magnitude dos erros em relação à medida média; o erro médio; e o erro normalizado médio.

Diversas aplicações de simulação de tráfego no contexto de Planos de Mobilidade Urbana no Brasil utilizam o coeficiente de determinação (r^2) como medida de bondade de ajuste, incluindo o Rio de Janeiro², Florianópolis³ e Salvador⁴.

2.3 Modelos baseados em agentes

Este capítulo visa introduzir os principais conceitos sobre os modelos baseados em agentes, demonstrando que sua abordagem permite uma representação detalhada e realista do comportamento dos diversos elementos que compõem o tráfego urbano, considerando fatores como preferências individuais, interações sociais e adaptação ao ambiente, e reconhecendo os desafios de implementação desses modelos.

²Disponível em: <<https://www.rio.rj.gov.br/web/pmud/documentos>>

³Disponível em: <<https://observatoriadamobilidadeurbana.ufsc.br/publicacao/projetos-de-pesquisa/>>

⁴Disponível em: <<http://www.planmob.salvador.ba.gov.br/index.php/15-planmob?ml=1>>

Os modelos baseados em agentes (ABM, sigla inglesa para *Agent-Based Model*) são utilizados em diversas áreas do conhecimento. Zhang e Levinson (2004) indicam que, de forma geral, os ABM são compostos de três elementos: os agentes, com suas características objetivos e comportamentos, o ambiente, que representa o espaço onde os agentes circulam, e as regras comportamentais, que definem como os agentes interagem com o ambiente e entre si. As características do ambiente podem ser alteradas como resposta às atividades dos agentes.

Na aplicação do ABM na simulação de tráfego, o modelo atua como uma microssimulação, onde os agentes são as pessoas que realizam as viagens, como mencionam Wise, Crooks e Batty (2017) e Klügl e Bazzan (2012). Esses agentes interagem estocasticamente entre si e com o seu ambiente com base em seus atributos pessoais e regras impostas pelo sistema.

Para Bazzan e Klügl (2013), o modelo baseado em agentes se destaca em relação às abordagens convencionais na simulação de tráfego por várias razões. Primeiramente, ele permite lidar de forma elegante com a diversidade dos agentes, já que tem como unidade básica de demanda o diário de atividades de cada indivíduo, que pode ser caracterizado de acordo com sua renda, posse de automóvel, entre outros. Em segundo lugar, por permitir a integração da escolha de rota, de modo de transporte e de tempo de saída, o ABM possibilita a modelagem de tomadas de decisão mais complexas e realistas, que consideram múltiplos fatores e comportamentos adaptativos.

As autoras descrevem que o alto grau de liberdade na fase de elaboração do modelo pode gerar sérios desafios, já que o nível de detalhamento necessário nem sempre é claro. Ao reduzir os detalhes do modelo, diminui a necessidade de justificativa, bem como a quantidade de parâmetros a serem calibrados, facilitando assim sua implementação. No entanto, a simplificação excessiva pode comprometer a relevância dos resultados obtidos.

Com essas características, os ABM são considerados por Klügl e Bazzan (2012) como particularmente adequados para sistemas como os seguintes:

1. sistemas cuja dinâmica é gerada por meio de interações locais e flexíveis, com tamanho de população, estruturas e interações variadas;
2. sistemas que requerem a representação da heterogeneidade, não apenas em relação aos estados comportamentais, mas também as suas regras;
3. sistemas onde as decisões ocorrem em diferentes níveis de agregação, como a tomada de decisões individuais dos indivíduos e a tomada de decisões políticas por autoridades regulatórias, e tais decisões afetam os indivíduos e os níveis mais agregados do sistema;

4. sistemas que incluem processos de aprendizado ou de evolução em nível individual ou populacional;
5. sistemas em que as premissas necessárias para uma modelagem baseada em equilíbrio são muito fortes, tais como a homogeneidade do espaço, a tomada de decisões de forma homogênea, informação perfeita, racionalidade, entre outros;
6. sistemas em que o foco não é no equilíbrio da função utilidade, mas sim no fenômeno da viagem em si e nos comportamentos que geram suas características.

Os modelos baseados em agentes são frequentemente utilizados em conjunto com a geração de demanda por meio da abordagem baseada em atividades. Segundo McNally (2000), a abordagem baseada em atividades para geração de demanda se contrapõe à abordagem baseada em viagens, pois age de acordo com a premissa de que as viagens são derivadas do desejo de realização de atividades, em oposição ao entendimento convencional centrado na viagem. O autor ressalta que, nesta abordagem, a viagem é considerada apenas um dos diversos atributos de uma atividade. Atributos como modo de transporte e tempo de deslocamento são tratados como características da atividade, não apenas da viagem em si.

O cerne da abordagem de atividades reside, segundo McNally (2000), na ideia de que as decisões de viagem são moldadas por uma série de atividades que compõem a agenda do indivíduo. Dessa forma, a análise das decisões de viagem só é completa quando considerada dentro do contexto dessa agenda diária de atividades.

Ao adotar uma visão holística, Bhat e Koppelman (1999) afirmam que essa abordagem reconhece as complexas interações entre atividades e viagens. Essa perspectiva destaca que o desejo de participar de atividades é mais fundamental do que as próprias viagens que podem resultar dessas atividades. Assim, ao analisar padrões de comportamento de atividade ao longo do dia, essa abordagem pode contribuir para o gerenciamento de congestionamento, explorando como as pessoas ajustam suas atividades em resposta a mudanças em suas rotinas diárias.

2.4 Simulador de transportes multi-agentes (MATSim)

O MATSim é um programa de código aberto em linguagem Java, definido em seu manual, escrito por Horni, Nagel e Axhausen (2016), como um simulador de tráfego baseado em agentes, idealizado para

utilização em cenários de grande escala⁵, utilizando a abordagem de alocação de tráfego baseada em filas (*queue-based*). O programa é o resultado da colaboração entre Kai Nagel e Kay W. Axhausen, e conforme consta em seu manual (HORNI; NAGEL; AXHAUSEN, 2016), tem como principais características:

- modelagem microscópica do tráfego: o MATSim realiza uma simulação microscópica integral dos fluxos de tráfego resultantes e do congestionamento que eles produzem;
- modelagem comportamental microscópica da demanda/modelagem baseada em agentes: o MATSim utiliza uma descrição microscópica da demanda, por meio do cronograma diário e das decisões dos viajantes sintéticos, no que pode ser chamado de “baseado em agentes”;
- física computacional: o MATSim realiza simulações microscópicas rápidas com 10^7 ou mais elementos modelados (incluindo *links* e agentes);
- sistemas adaptativos complexos/algoritmos coevolutivos: o MATSim otimiza as utilidades experimentadas de todo o cronograma por meio da busca coevolutiva pelo equilíbrio ou estado estacionário resultante.

O manual do MATSim (HORNI; NAGEL; AXHAUSEN, 2016) oferece uma gama de instruções e definições para elaboração de simulações de tráfego no *software*. A presente seção tem como objetivo apresentar um breve resumo desses itens, complementados pela experiência de outros autores no uso do MATSim.

Apesar de ser considerado um modelo baseado em agentes, pelo alto nível de desagregação, utilização do modelo de demanda baseado em atividades e modelos comportamentais, Wise, Crooks e Batty (2017) usa o MATSim como principal exemplo de simuladores parcialmente baseados em agentes, já que os agentes não interagem diretamente, influenciando nas decisões uns dos outros indiretamente por meio do congestionamento que causam um ao outro e o respectivo custo generalizado resultante.

Além disso, o MATSim pode ser considerado como um simulador parcialmente baseado em atividades. Isso porque ele modela os usos de forma desagregada, e é capaz de alterar o horário de realização das viagens conforme esse uso do solo, mas não de alterar os locais de origem e destino das viagens com base nesse mesmo critério.

⁵O conceito de “grande escala” nesse contexto representa as áreas de estudo compostas por bairros, municípios, regiões metropolitanas ou ainda maiores, em oposição à aplicação usual dos modelos de microssimulação que se atêm a alguns trechos de via e interseções.

A Seção 2.4.1 apresenta o algoritmo coevolucionário, no qual o MATSim se baseia. Segundo Horni, Nagel e Axhausen (2016), o funcionamento básico do algoritmo considera que cada agente otimiza iterativamente seu cronograma diário de atividades, competindo por “vagas” de espaço e tempo com todos os outros agentes na infraestrutura de transporte. Durante as iterações, a demanda inicial é otimizada individualmente por cada agente, que possui uma memória com quantidade fixa e configurável de planos diários, compostos por uma sequência de atividades diárias e sua pontuação associada.

Elaboração de modelos no MATSim A demanda inicial e a infraestrutura de transportes devem ser modeladas de acordo com o formato exigido pelo MATSim, conforme exhibe a Seção 2.4.2.

Antes da simulação do carregamento da demanda na rede com o MATSim, na etapa denominada *mobsim* e detalhada na Seção 2.4.3, cada agente seleciona um plano diário de sua memória de acordo com a pontuação desses planos, elucidada na Seção 2.4.4, e uma parte dos agentes é autorizada a clonar o plano selecionado e modificá-lo, no processo denominado replanejamento, apresentado na Seção 2.4.5.

Após a execução de todas as iterações, é realizado o processo de calibração e validação do modelo, cujas especificidades no MATSim são apresentadas na Seção 2.4.6. Com o fim da simulação, é possível visualizar os resultados obtidos. A Seção 2.4.7 descreve os arquivos de saída do MATSim, e as ferramentas disponíveis para representação visual dos resultados.

2.4.1 O algoritmo coevolucionário

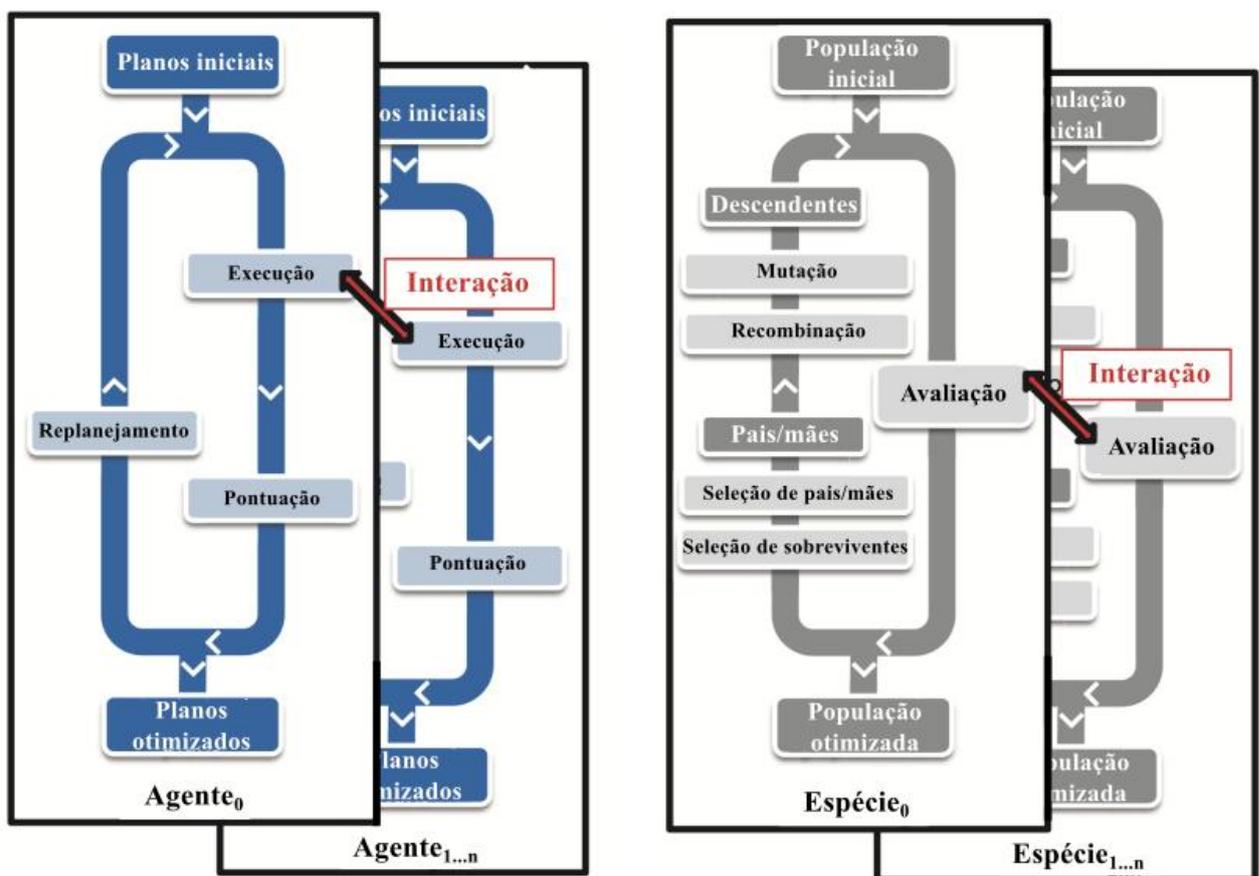
O MATSim utiliza um algoritmo coevolucionário para otimização dos planos dos agentes em busca do equilíbrio das pontuações. Segundo Popovici et al. (2012), os algoritmos coevolucionários (CoEAs) têm o mesmo objetivo dos algoritmos evolucionários (EAs) tradicionais, ou seja, tentam aproveitar as noções darwinianas de hereditariedade e sobrevivência dos mais aptos para fins de simulação ou resolução de problemas.

Os autores descrevem que um modelo é elaborado para representar aspectos de possíveis soluções para um problema dos indivíduos. O comportamento dos indivíduos é alterado durante a simulação por meio de ferramentas de variação semelhantes às genéticas, como mutação e cruzamento, e a busca é direcionada pela seleção dos melhores indivíduos, determinados por meio de uma avaliação de adequação.

De acordo com Popovici et al. (2012), os EAs começam com uma função de aptidão que atribui um valor real a cada genótipo possível. Dada tal função, a relação de aptidão entre quaisquer dois genótipos é clara: comparamos o resultado de duas funções para ver qual é mais apto. Em contraste, os CoEAs não usam uma métrica direta da aptidão. Em vez disso, dois genótipos são comparados com base em seus resultados das interações entre indivíduos. Ou seja, ainda são comparados os resultados das duas funções, mas essas são compostas por valores atribuídos a cada genótipo que são resultados da interação entre os indivíduos.

A Figura 5 ilustra a relação entre o modelo co-evolucionário tradicional e o algoritmo co-evolucionário utilizado no MATSim, ambos os quais co-evoluem diferentes espécies sujeitas à competição.

Figura 5: O algoritmo co-evolucionário no MATSim. Fonte: adaptado de Horni, Nagel e Axhausen (2016).



(a) Algoritmo co-evolucionário do MATSim

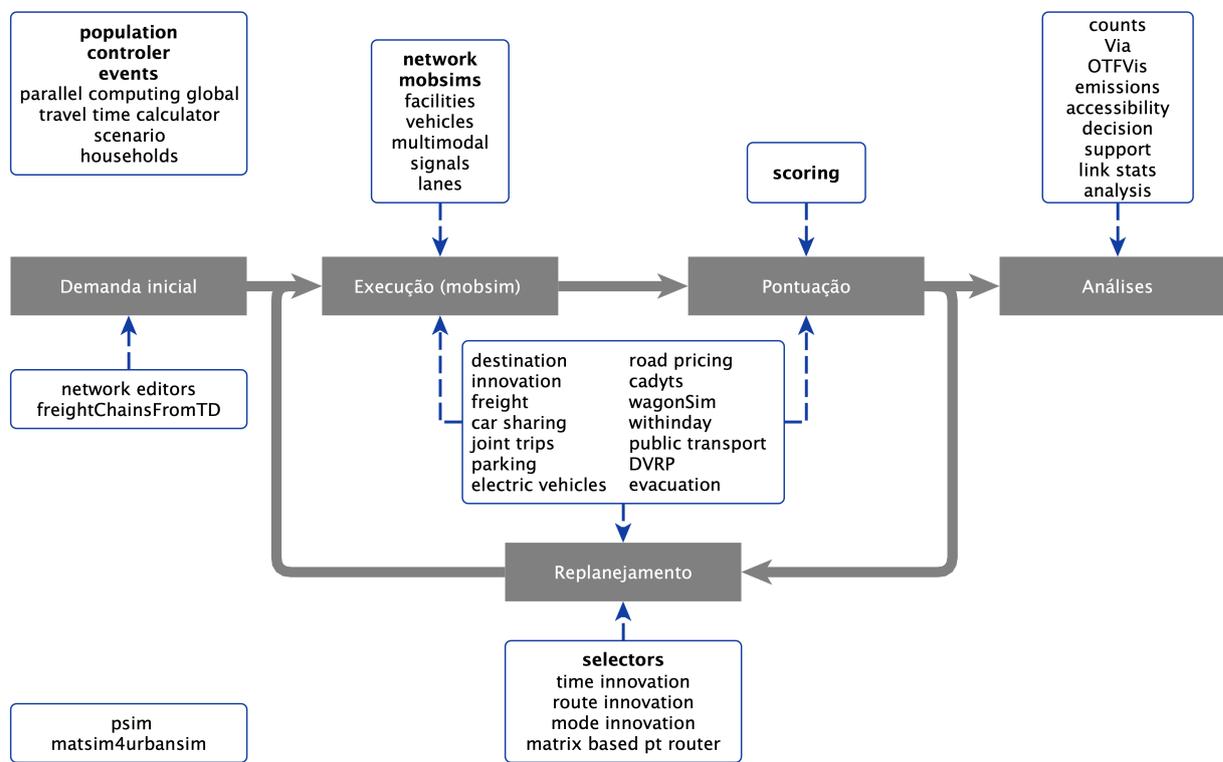
(b) Algoritmo co-evolucionário tradicional

Cada agente aprende pois mantém múltiplos planos, que são pontuados ao serem executados, selecionados de acordo com sua pontuação e, às vezes, modificados. De forma mais detalhada, Horni, Nagel e Axhausen (2016) descrevem os elementos do processo iterativo da seguinte forma:

- *execução* ou *mobsim*: a simulação de mobilidade seleciona um plano por agente e o executa em uma realidade modelada, em um processo que pode ser denominado carregamento da rede;
- *pontuação* ou *scoring*: a performance do plano na realidade simulada é utilizada para mensuração da pontuação do plano executado;
- *replanejamento* ou *replanning*: são definidos os planos a constarem na memória de cada agente na iteração seguinte, por meio desses três tipos de ação:
 - para uma porcentagem pré-estabelecida de agentes, um plano é duplicado, modificado e selecionado na próxima iteração (estratégias de inovação);
 - todos os outros agentes escolhem entre os seus planos (estratégia de seleção);
 - se um agente tem mais planos do que a quantidade máxima permitida (um dos parâmetros de configuração), alguns planos são removidos de acordo com um seletor configurável (estratégia de remoção);

O MATSim possui ainda diversos módulos que podem ser adicionados aos elementos do processo de simulação. No MATSim, os módulos representam uma variedade de tipos de funcionalidades, como funções de configuração, componentes de replanejamento, contribuições ou até ferramentas externas. Metaforicamente, Horni, Nagel e Axhausen (2016) descrevem que um módulo pode ser visto como o maior divisor comum das diferentes funcionalidades fornecidas pelo MATSim. A Figura 6 ilustra os módulos existentes em relação aos elementos em que se integram, onde os itens em negrito representam os módulos obrigatórios para funcionamento da simulação.

Figura 6: Módulos existentes no MATSim e sua integração com o processo de simulação. Fonte: adaptado de Horni, Nagel e Axhausen (2016).



Alguns módulos são integrados com apenas um dos elementos do processo de simulação, enquanto outros se integram com mais de um elemento, e os módulos não conectados a nenhum elemento em específico atuam de forma global na simulação.

2.4.2 Elaboração dos arquivos de entrada

Essa seção apresenta a descrição dos seguintes arquivos de entrada utilizados para a elaboração de um modelo de simulação no MATSim, que podem ser resumidos em:

- arquivos de entrada básicos:
 - *plans.xml*: representa a demanda a ser inserida na simulação, e contém os dados de diários de viagens da população a ser modelada;
 - *network.xml*: representa a oferta de infraestrutura viária no modelo, e contém os dados das vias a serem consideradas na alocação de tráfego.
- arquivos complementares:

- *laneDefinitions.xml*, *signalControl.xml*, *signalGroups.xml* e *signalSystems.xml*: representam os sistemas de controle de tráfego, os quais aumentam o nível de detalhamento da modelagem das vias e interseções da rede viária;
- *transitVehicles.xml* e *transitSchedule.xml*: representam o sistema de transporte coletivo, e em simulações multimodais com a inclusão desse modo de transporte, são considerados arquivos necessários para a simulação.

A estrutura completa desses e dos demais dados de entrada possíveis de serem incluídos no MATSim são listados no site <<http://matsim.org/files/dtd/>>.

Demanda A modelagem da demanda é composta pelo diário de viagens da população a ser modelada. O diário de viagens diz respeito à caracterização de todas as viagens realizadas durante um dia, as quais podem ser obtidas por meio de entrevistas em campo, ou podem ser gerados por meio de modelos baseados em atividades, conforme abordado na Seção 2.2.4.

Os diários de viagem servem de subsídio para elaboração do arquivo *plans.xml*, o qual deve conter a seguinte estrutura básica:

```

1 <population >
2   <!--indicação da identificação numérica única da pessoa -->
3   <person id="1">
4     <!-- descrição de um plano diário de viagens -->
5     <plan>
6       <!--as atividades devem ser caracterizada pelo seu tipo (casa, trabalho, educação, etc),
7       pelas coordenadas de sua localização e horário de fim-->
8       <activity type="home" x="741516.56" y="6946051.59" end_time="07:02"/>
9       <!--indicação do modo de transporte utilizado para deslocamento até as coordenadas da
10      atividade seguinte -->
11      <leg mode="walk"/>
12      <activity type="education" x="742123.30" y="6946139.87" end_time="12:38" />
13      <leg mode="walk"/>
14      <!--todos os agentes realizam a primeira e a última atividade do dia em seus
15      domicílios -->
16      <activity type="home" x="741516.56" y="6946051.5"/>
17    </plan>
18  </person>
19  <person id="2">
20  ...

```

```
18 </person>  
19 </population >
```

No caso das entrevistas de campo, os diários de viagem são usualmente obtidos para apenas uma amostra da população, que considera critérios espaciais e socioeconômicos para garantir a representatividade da amostra. Após a coleta de dados, é necessária a expansão dessa amostra para determinação da quantidade total de viagens. Para isso, são utilizados fatores de expansão que consideram os mesmos critérios espaciais e socioeconômicos utilizados para determinação da amostra entrevistada.

Dados os esforços computacionais exigidos para simulação com esse nível de detalhes em grande escala, o MATSim fornece a possibilidade de utilização de apenas uma fração dessa população. A partir disso, são considerados valores ajustados da capacidade viária de acordo com a fração populacional utilizada. Os detalhes desse ajuste são abordados na Seção 2.4.6.

Para estratificação da população, e consideração de diferentes comportamentos para diversos grupos de pessoas, o MATSim possibilita a utilização da extensão *subpopulation* ao arquivo *plans.xml*. O *GitHub*⁶ do MATSim apresenta um *script* para a geração da população sintética com base em dados censitários da população, que pode ser encontrado pesquisando pelo termo “*RunZPopulationGenerator*”.

Oferta Para modelagem da malha viária, os dados da oferta devem preferencialmente vir de cadastros atualizados das entidades responsáveis pelo seu gerenciamento. Conforme mencionado na seção 2.2.2, no Brasil é rara a existência desses cadastros e, ainda mais, a sua disponibilização de forma aberta. Há fontes de dados alternativas que podem ser utilizadas para esse fim, como o *OpenStreetMap* (OSM). Em ambos os casos, é necessária a conversão desses dados para formato *xml* na estrutura MATSim.

A utilização do OSM possibilita a conversão automática dos dados por meio do *software* JOSM⁷, o qual possui uma extensão denominada MATSim para a realização dessa atividade. O JOSM realiza a simplificação da geometria, a imputação de capacidade viária de acordo com a classificação funcional das vias, e a reestruturação e conversão dos dados para o formato MATSim. Outras fontes de dados podem ter estruturas variadas e exigem customização no processo de conversão.

⁶Disponível em: <<https://github.com/matsim-org/matsim-code-examples/find/11.x>>

⁷Disponível em: <<https://josm.openstreetmap.de/>>

A modelagem básica da rede viária no formato do arquivo *network.xml* deve conter a seguinte estrutura básica:

```
1 <network>
2   <nodes>
3     <!-- indicação da identificação numérica única de cada nó da rede, e suas coordenadas -->
4     <node id="1" x="743374.8374" y="6943803.8274"/>
5     <node id="2" x="743281.0623" y="6943618.4360"/>
6     ...
7   </nodes>
8   <!-- indicação do período a que a capacidade viária se refere, em horas, do tamanho de cada
9     "célula" considerada o tamanho do veículo padrão, e a largura das faixas, ambos em
10    metros-->
11  <links capperiod="01:00:00" effectivecellsize="7.5" effectivecellwidth="3.75">
12    <!-- indicação da identificação numérica única de cada link, da identificação dos nós de
13      início e fim do link, que atribui localização e sentido aos links (todos os links são
14      unidirecionais no MATSim), comprimento do trecho, em metros, velocidade de fluxo livre, em
15      metros por segundo, capacidade viária, em veículos padrão por "capperiod", quantidade de
16      faixas e modos permitidos a trafegar no link -->
17    <link id="101" from="1" to="2" length="207.69" freespeed="6.945" capacity="496.2406"
18      perlanes="1.0" oneway="1" modes="car, motorcycle"/>
19    <link id="102" from="2" to="1" length="207.69" freespeed="6.945" capacity="496.2406"
20      perlanes="1.0" oneway="1" modes="car, motorcycle"/>
21    ...
22  </links>
23 </network>
```

Duas características fundamentais para o modelo de fluxo de tráfego do MATSim são obtidas a partir da capacidade viária do *link*: (i) a capacidade de fluxo dos segmentos de via; e (ii) a capacidade de armazenamento nesses segmentos.

Segundo Horni, Nagel e Axhausen (2016), o item (i) especifica a quantidade de veículos que podem sair do segmento por período (definido em *capperiod*), e o item (ii) define a quantidade máxima de veículos que cabem em determinado segmento de via. Ou seja, o segundo determina quando o segmento para de receber novos veículos, e o primeiro determina de quanto em quanto tempo é permitida a saída dos veículos do segmento.

Sistemas de controle de tráfego O modelo básico do MATSim não considera os controles de tráfego e as restrições de movimento nas interseções. Contudo, é possível utilizar os módulos *Signals* e *Lanes* para inserir a consideração desses equipamentos no modelo, caso seja necessário para a validação do modelo, e haja disponibilidade dessas informações.

A definição dos sistemas semafóricos é realizada por meio de três arquivos *signalControl.xml*, *signalGroups.xml* e *signalSystems.xml*, cujas estruturas básicas são respectivamente apresentadas a seguir, conforme adaptado do exemplo que consta no repositório *GitHub* do MATSim⁸.

```
1 < signalControl >
2   < signalSystem reflD ="2">
3     < signalSystemController >
4       < controllerIdentifier > DefaultPlanbasedSignalSystemController </ controllerIdentifier >
5     < signalPlan id="1">
6       <cycleTime sec="120"/>
7       < offset sec="0"/>
8       < signalGroupSettings reflD ="1">
9         <onset sec="0"/>
10        <dropping sec="55"/>
11      </ signalGroupSettings >
12      < signalGroupSettings reflD ="2">
13        <onset sec="0"/>
14        <dropping sec="55"/>
15      </ signalGroupSettings >
16      ...
17    </ signalPlan >
18  </ signalSystemController >
19 </ signalSystem >
20 </ signalControl >
```

```
1 < signalGroups >
2   < signalSystem reflD ="2">
3     < signalGroup id="1">
4       < signal reflD ="1"/>
5     </ signalGroup >
6     < signalGroup id="2">
7       < signal reflD ="2"/>
```

⁸Disponível em: <<https://github.com/matsim-org/matsim-libs/tree/master/contribs/signals/examples/tutorial/example90TrafficLights/useSignalInput/withLanes>>

```

8   </signalGroup >
9   </signalSystem >
10  <signalGroups >

```

```

1  <signalSystems >
2  <signalSystem id="2">
3    <signals >
4      <signal linkIdRef ="12" id="1">
5        <lane reflId ="12.l"/>
6      </signal >
7      <signal linkIdRef ="12" id="2">
8        <lane reflId ="12.r"/>
9      </signal >
10     <signal linkIdRef ="32" id="3"/>
11     <signal linkIdRef ="72" id="4"/>
12   </signals >
13 </signalSystem >
14 </signalSystems >

```

Para a modelagem dos movimentos proibidos e permitidos nas interseções, é utilizado o módulo *Lanes*, o qual requer a elaboração de um arquivo denominado *laneDefinitions.xml*, que deve ter a seguinte estrutura básica.

```

1  <laneDefinitions >
2  <lanesToLinkAssignment linkIdRef ="12">
3    <lane id="12.l">
4      <leadsTo>
5        <toLink reflId ="23"/>
6      </leadsTo>
7      <representedLanes number="1.0"/>
8      <capacity vehiclesPerHour ="1800.0"/>
9      <startsAt meterFromLinkEnd="150.0"/>
10     <alignment>1</alignment>
11     <attributes />
12   </lane>
13   <lane id="12.ol"
14     ...
15   </lane>

```

```
16 </ lanesToLinkAssignment >  
17 <\ laneDefinitions >
```

Dada a complexidade da modelagem dos sistemas de controle de tráfego, além da possível indisponibilidade desses dados, alguns estudos, como Ziemke e Nagel (2017) e Kickhöfer et al. (2016), realizaram ajustes nas velocidades das vias urbanas para estimativa dos impactos de semáforos, assim como aceleração e desaceleração em interseções. A seleção dos fatores de ajuste foi realizada de forma aparentemente arbitrária pelos autores, utilizando valores entre 0,5 e 1,0 (os detalhes sobre esses estudos são apresentados na Seção 2.5).

Transporte coletivo No MATSim, é possível incluir diversos modos na mesma rede viária, segundo Horni, Nagel e Axhausen (2016). Com isso, o modo da viagem é analisado e, se esse modo corresponder a um dos modos listados como “principais” (*main*), um veículo é gerado para aquela viagem, o qual entra na rede designada ao seu modo.

Segundo Horni, Nagel e Axhausen (2016), em modos de “passageiros”, como o transporte coletivo ou táxi, o agente embarca no veículo, o qual se move ao longo da rede, enquanto a única decisão do agente é de desembarcar ou não no próximo ponto. O ônibus, por sua vez, é um participante normal do sistema de tráfego, trafegando na rede viária e sujeitos a atrasos devido a congestionamentos. Para simulação do transporte coletivo, o MATSim possui um módulo específico denominado “*transit*” que pode ser ativado por meio do arquivo de configuração do MATSim.

Atualmente, Horni, Nagel e Axhausen (2016) destacam que o MATSim proporciona um modelo de transporte coletivo bastante detalhado, em que os ônibus se deslocam ao longo dos itinerários das linhas, embarcando e desembarcando passageiros nos pontos de parada, enquanto monitora a capacidade e velocidade máxima dos ônibus.

Esse modelo já permite a modelagem avançada de alguns aspectos do transporte coletivo, conforme citado por Horni, Nagel e Axhausen (2016), como a variação de velocidades ao longo de uma mesma rota em horários diferentes do dia, a utilização de diversos tipos de veículos em horários diferentes do dia, e a reutilização de veículos em uma ou mais linhas.

Para modelagem da oferta de transporte público coletivo, são necessárias informações a respeito de todas as linhas operadas pelo sistema, incluindo seus itinerários (rotas), horários ofertados, veículos empregados e pontos de embarque e desembarque. No Brasil, esses dados não são comumente disponibilizados em regime aberto, e devem ser obtidos junto à administração pública.

Após a obtenção desses dados, é necessária a elaboração de dois arquivos *xml*. O primeiro é denominado *transitVehicles.xml*, e visa caracterizar os veículos utilizados na operação. Essa caracterização deve ter a seguinte estrutura básica:

```
1 < vehicleDefinitions >
2   < vehicleType id="1">
3     < description >Comum</description>
4     < capacity >
5       <!--indicação da quantidade de pessoas sentadas e em pé que o tipo de veículo é capaz de
6         transportar simultaneamente -->
7       <seats persons="43"/>
8       <standingRoom persons="40"/>
9     </ capacity >
10    <!--indicação do comprimento do tipo de veículo, em metros-->
11    <length meter="15.0"/>
12  </ vehicleType >
13  <!--indicação do identificador único e do tipo a cada veículo existente -->
14  < vehicle id="Comum_1" type="1"/>
15  < vehicle id="Comum_2" type="1"/>
16 </ vehicleDefinitions >
```

O segundo arquivo necessário é denominado *TransitSchedule.xml*, e visa caracterizar a operação das linhas existentes. Essa caracterização deve ter a seguinte estrutura básica:

```
1 < transitSchedule >
2   < transitStops >
3     <!--indicação do identificador único do ponto de parada, suas coordenadas e o link em que o
4       ponto se encontra -->
5     < stopFacility id="1" x="733327.1168" y="6953636.7420" linkRefId="12"/>
6     < stopFacility id="2" x="733379.2636" y="6953593.1768" linkRefId="13"/>
7   </ transitStops >
8   < transitLine id="Linha_01">
9     < transitRoute id="1">
10    <transportMode>pt</transportMode>
11    < routeProfile >
12      <!--indicação da lista ordenada dos pontos de parada (de acordo com o itinerário da
13        linha), o tempo esperado de chegada e saída do veículo no ponto a partir da primeira
14        parada-->
15      <stop refId="1" departureOffset="0:00:00"/>
```

```

13     <stop refId ="2" departureOffset ="0:19:28" arrivalOffset ="0:19:58"
awaitDeparture =" false "/>
14 </ routeProfile >
15 <route>
16     <!--indicação da lista ordenada de links de acordo com o itinerário da linha -->
17     <link refId ="12"/>
18     <link refId ="33"/>
19     <link refId ="15"/>
20     <link refId ="13"/>
21 </route>
22 <departures >
23     <!--indicação das viagens realizadas ao longo do dia, caracterizadas pelo tempo de
partida do ponto inicial e pelo veículo utilizado para sua realização (em consonância com o
arquivo transitVehicles )-->
24     <departure id ="1" departureTime ="5:14:04" vehicleRefId ="Comum_1"/>
25     <departure id ="2" departureTime ="7:06:34" vehicleRefId ="Comum_2"/>
26 </departures >
27 </ transitRoute >
28 </ transitLine >
29 < transitLine id ="Linha_02">
30     ...
31 </ transitLine >
32 </ transitSchedule >

```

2.4.3 Execução da simulação (mobsim)

Segundo Horni, Nagel e Axhausen (2016), o MATSim inclui duas possibilidades de execução da simulação: QSim (*Queue Simulation*) e JDEQSim (*Java Discrete Event Queue Simulation*), além de possibilitar a integração com ferramentas externas de simulações de tráfego.

O QSim é a ferramenta padrão do MATSim, e é baseada em intervalos de tempo. O método de alocação de tráfego aplicado é o método baseado em filas (*queue-based*), utilizando o conceito FIFO (*first in, first out*). Essa abordagem é muito eficiente, mas, segundo Horni, Nagel e Axhausen (2016), não é tão realista quanto outros modelos mais sofisticados, que são capazes produzir os efeitos de ultrapassagem de veículos, por exemplo.

No JDEQSim, por razões computacionais, a abordagem de fila de espera é combinada com uma etapa de atualização baseada em eventos (CHARYPAR et al., 2008). Em outras palavras, não há um processo de atualização baseado em intervalos de tempo para nenhum agente no cenário. Em vez disso, os agentes são processados apenas se realmente precisarem de ação, o que, no experimento de Charypar et al. (2008), aumentou a eficiência da alocação de tráfego em mais de dez vezes.

O MATSim permite a simulação multimodal, com a indicação dos modos “principais” (*main modes*) que devem ser simulados na rede no arquivo de configurações. Todos os modos não descritos como principais, são processados por meio do “teletransporte”. De acordo com Horni, Nagel e Axhausen (2016), o teletransporte gera eventos de saída e de chegada em atividades, calculando o tempo de viagem de acordo com a distância em linha reta⁹ entre origem e destino, com a velocidade média do modo teletransportado.

2.4.4 A pontuação dos planos diários

A pontuação dos planos diários de cada agente é a etapa que visa avaliar as escolhas realizadas ao longo do dia, de forma que o agente aprenda com suas decisões em iterações prévias e tome novas decisões que levem ao equilíbrio estocástico do sistema modelado.

Devido ao fato de o MATSim ser baseado em planos de dias completos e da ausência de funções de utilidade para dia completo na literatura, Horni, Nagel e Axhausen (2016) descreve que o MATSim utiliza a chamada função de pontuação ou utilidade Charypar-Nagel.

A função de pontuação do plano de cada usuário no MATSim pode ser configurada por vários parâmetros já inclusos no MATSim, ou ainda por meio da inserção de sua própria função de pontuação customizada.

A primeira e ainda básica função de pontuação do MATSim foi formulada por Charypar e Nagel (2005), baseada vagamente no modelo de Vickrey para congestionamento de estradas, conforme descrito por Vickrey (1969) e Arnott et al. (1993). Ela é descrita na Equação 1, a qual considera que a última atividade do plano do agente é unida a sua primeira atividade de forma a gerar uma quantidade igual de viagens e atividades. Ainda, a função de pontuação específica das

⁹Enquanto o teletransporte é utilizado em outros simuladores de tráfego, como o SUMO, nem todos utilizam as distâncias em linha reta para teletransporte dos agentes. No SUMO, por exemplo, apesar de ser retirado da rede viária, o veículo se desloca ao longo de sua rota com a velocidade média do *link* onde está atualmente "passando". O veículo é reinserido na rede se houver espaço suficiente para ser colocado em uma faixa que permita continuar sua viagem (LOPEZ et al., 2018).

atividades é descrita na Equação 2, e das viagens, na Equação 3.

$$S_{plan} = \sum_{q=0}^{N-1} S_{act,q} + \sum_{q=0}^{N-1} S_{trav,mode(q)} \quad (1)$$

Onde:

S_{plan} é a utilidade do plano;

$S_{act,q}$ é a utilidade das atividades realizadas;

$S_{trav,mode(q)}$ é a (des)utilidade das viagens realizadas;

N é a quantidade de atividades;

q é o indicador numeral da viagem e atividade, de forma que a viagem q é aquela que segue a atividade q .

$$S_{act,q} = S_{dur,q} + S_{wait,q} + S_{late.ar,q} + S_{early.dp,q} + S_{short.dur,q} \quad (2)$$

Onde:

$S_{dur,q}$ é a utilidade atribuída à duração da atividade;

$S_{wait,q}$ é a utilidade atribuída à espera para início da atividade;

$S_{late.ar,q}$ é a (des)utilidade atribuída ao atraso na chegada à atividade;

$S_{early.dp,q}$ é a (des)utilidade atribuída a não permanecer na atividade por tempo o suficiente;

$S_{short.dur,q}$ é a (des)utilidade atribuída a uma atividade “curta demais”.

$$S_{trav,q} = C_{mode(q)} + \beta_{trav,mode(q)} \times t_{trav,q} + \beta_m \times \Delta m_q \\ + (\beta_{d,mode(q)} + \beta_m \times \lambda_{d,mode(q)}) \times d_{trav,q} + \beta_{transfer} \times x_{transfer,q} \quad (3)$$

Onde:

$C_{mode(q)}$ é a constante específica do modo;

$\beta_{trav,mode(q)}$ é utilidade direta do tempo gasto viajando por modo;

$t_{trav,q}$ é o tempo de viagem entre as atividades q e $q + 1$;

β_m é a utilidade do dinheiro (normalmente positiva);

Δm_q é a mudança no custo financeiro causado por tarifas ou pedágios (normalmente negativo ou zero);

$\beta_{d,mode(q)}$ é a utilidade da distância (normalmente negativa ou zero);

$\lambda_{d,mode(q)}$ é a taxa financeira da distância específica para cada modo (normalmente negativa ou zero);

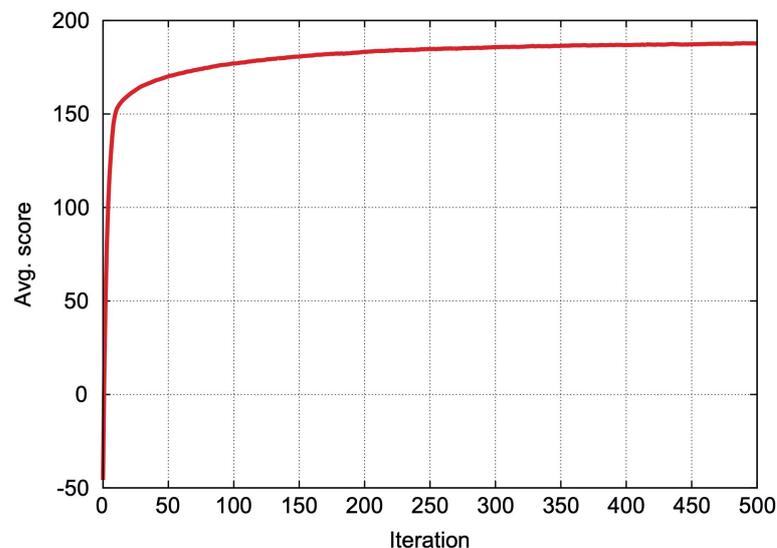
$d_{trav,q}$ é a distância viajada entre as atividades q e $q + 1$;

$\beta_{transfer}$ é a penalidade por transferência de veículo do transporte coletivo (normalmente negativa);

$x_{transfer,q}$ é uma variável binária (0 ou 1) que sinaliza se a transferência ocorreu entre o trecho anterior e atual da viagem.

Segundo Horni, Nagel e Axhausen (2016), o processo iterativo é repetido até que a pontuação média da população se estabilize. A curva típica de desenvolvimento da pontuação, apresentada na Figura 7 (HORNI et al., 2009), assume a forma de um progresso de otimização evolutiva (EL-BEN, 2015). Como as simulações são estocásticas, não se pode utilizar critérios de convergência apropriados para algoritmos determinísticos.

Figura 7: Curva típica de desenvolvimento da pontuação. Fonte: Horni et al. (2009).



A estabilização da pontuação depende dos valores dos parâmetros supracitados, assim como dos demais parâmetros de calibração do modelo. Estes dois, em conjunto, influenciam também os resultados relacionados aos volumes de tráfego, tempos de viagem e quantidade de agentes que não são capazes de finalizar suas viagens até o final do dia (agentes presos).

2.4.5 Replanejamento das atividades

As estratégias de replanejamento constituem os módulos básicos de inovação disponíveis no MAT-Sim, conforme descrevem Horni, Nagel e Axhausen (2016). Os módulos de replanejamento podem ser diferenciados entre os que afetam o conjunto de planos que cada agente possui e os que apenas selecionam entre esses planos.

A cada módulo é atribuído um peso que determina a probabilidade pela qual a decisão representada pelo módulo é tomada. Os módulos de inovação são indicados por Horni, Nagel e Axhausen (2016), e listados a seguir.

- O módulo de inovação de tempo altera aleatoriamente os horários de término das atividades dentro de uma faixa configurável, conforme descrito por Balmer et al. (2005) e Raney (2005).
- O módulo de inovação de rota é especificado pelo algoritmo de roteamento no arquivo de configuração do controlador.
- O módulo de inovação de modo possui três configurações: *ChangeLegMode*, *ChangeSingleLegMode* e *SubtourModeChoice*. O *ChangeLegMode* escolhe aleatoriamente um dos planos de uma pessoa e muda o modo de transporte. Apenas um modo de transporte por plano é suportado. Ao usar modos diferentes para sub-períodos em um único dia, o módulo *SubtourModeChoice* é necessário. O *ChangeSingleLegMode* escolhe aleatoriamente um dos planos de uma pessoa e muda o modo de transporte de uma única etapa da viagem (escolhida aleatoriamente). Ao contrário da *ChangeLegMode* permite múltiplos modos em um único plano. Além disso, este módulo pode (opcionalmente) respeitar a disponibilidade de carros.

Os módulos de seleção dos planos são descritos por Horni, Nagel e Axhausen (2016) como:

- No módulo de remoção de plano, por padrão, o plano com a pontuação mais baixa é removido se a memória do agente estiver cheia, definida pela quantidade máxima de planos na memória dos agentes.
- A seleção de plano tem diversas possibilidades de modos, cujos nomes e descrições são listados a seguir.
 - *KeepLastSelected* mantém o plano selecionado na iteração anterior.
 - *BestScore* seleciona o plano com a pontuação mais alta da iteração anterior.
 - *SelectExpBeta* realiza a seleção MNL (Modelo Logit Multinomial) entre os planos. Pode ser configurado pelo parâmetro *BrainExpBeta* do grupo de configuração de pontuação, sendo o

parâmetro de escala em modelos de escolha discreta, os autores recomendam o valor padrão de 1.0.

- *ChangeExpBeta* muda para um plano diferente, com probabilidade dependente da diferença de pontuação entre os dois planos e de amostragem de um MNL.
- *SelectRandom* realiza seleção aleatória entre os planos.
- *SelectPathSizeLogit* seleciona um plano existente de acordo com o logit. Pode ser configurado pelo parâmetro *PathSizeLogitBeta* do grupo de configuração de pontuação. Contudo, este seletor nunca foi investigado sistematicamente.

2.4.6 Calibração e validação do modelo

Os parâmetros apresentados na Seção 2.4.4 e na Seção 2.4.5 devem ser calibrados de acordo com a realidade do sistema estudado. Os processos de calibração do modelo visam definir os valores desses parâmetros de forma que seus resultados sejam o mais próximos possíveis da realidade, o que é avaliado no processo de validação.

No modelo inicial, podem ser utilizados os valores padrão do MATSim, indicados por Horni, Nagel e Axhausen (2016) como:

- $\beta_m = 1$ utils/unidade monetária;
- $\beta_{dur} = 6$ unidades de utilidade/hora;
- $\beta_{trav,mode(q)} = -6$ unidades de utilidade/hora;
- $\beta_{wait} = 0$ unidades de utilidade/hora;
- $\beta_{short.dur} = 0$ unidades de utilidade/hora;
- $\beta_{late.ar} = -18$ unidades de utilidade/hora;
- $\beta_{early.dp} = 0$ unidades de utilidade/hora.

Esses valores foram definidos vagamente com base no modelo de congestionamento de Vickrey (Arnott, 1990). Para a escolha modal, Horni, Nagel e Axhausen (2016) indicam uma possível abordagem para a calibração dos parâmetros, enumerada a seguir.

1. Defina $\beta_m \equiv$ Utilidade Marginal do Dinheiro para o que quer que seja o peso do parâmetro monetário no seu modelo logit de escolha de modo. Caso a análise não inclua um modelo logit de escolha de modo, o valor padrão para esse parâmetro é + 1,0, e sempre positivo, já que ter mais dinheiro normalmente aumenta a utilidade.

2. Defina $\beta_{dur} \equiv$ realização para o que quer que seja o peso do parâmetro de tempo de viagem de carro no seu modelo logit de escolha de modo, enquanto altera o sinal desse parâmetro de seu típico - para um +. Caso a análise não inclua um modelo logit de escolha de modo, defina como +6,0, e sempre positivo, já que realizar uma atividade por mais tempo normalmente aumenta a utilidade.
3. Defina $\beta_{tt,carro} \equiv$ Utilidade Marginal de Viajar como 0,0. Mesmo definido como zero, viajar de carro será implicitamente punido pelo custo de oportunidade do tempo: se você estiver viajando de carro, não poderá realizar uma atividade; portanto, você está (marginalmente e aproximadamente) perdendo β_{dur} .
4. Defina todas as outras utilidades marginais do tempo de viagem por modo em relação ao valor do carro, conforme descrito em seu modelo logit. Caso a análise não inclua um modelo logit de escolha de modo, defina todos os valores de $\beta_{tt,modo} \equiv$ Utilidade Marginal de Viajar como 0,0.
5. Defina as taxas de custo de distância *monetaryDistanceRate* para valores plausíveis, se esse estiverem disponíveis. Este valores devem ser sempre negativos, já que a distância consome dinheiro.
6. Para a calibração da divisão modal, os valores das constantes específicas de cada alternativa C_{modo} devem ser alterados.
7. Caso a divisão modal se assemelhe à real, mas a distribuição de distância não, as velocidades dos modos devem ser analisadas. Horni, Nagel e Axhausen (2016) identificaram que essa medida é mais eficiente do que a calibração do $\beta_{tt,modo}$.

Além dos parâmetros da pontuação, outro parâmetro a ser ajustado é a quantidade de iterações realizadas pela simulação. Esse deve ter seu valor atrelado à estabilização dos valores médios de pontuação, considerando que, quanto maior o número de iterações, maior é a capacidade computacional necessária. O estudo de Kickhöfera et al. (2016) realiza 200 iterações, o estudo de Farinha (2013) realiza 400 iterações, o de Bicudo (2015) e o de Ziemke e Nagel (2017) realizam 500 iterações. Informações sobre as respectivas área de estudo são apresentadas na Seção 2.5.

Conforme abordado na Seção 2.4.2, quando é utilizada apenas uma porção da população para realização das simulações, é necessário ajustar a capacidade viária de acordo. Contudo, a relação entre população e capacidade viária não é, necessariamente, linear. Outros estudos que reduziram a amostra da população apresentam valores de capacidade viária próximos aos valores utilizados para a amostra, mas não iguais. Ziemke e Nagel (2017) utilizam 10% de amostra da população e 12% de capacidade viária, enquanto Kickhöfera et al. (2016) utilizam 0,65% de amostra

da população e 0,64% da capacidade viária. Nenhum dos autores descrevem uma relação explícita entre os parâmetros, e sua definição deve fazer parte do processo de calibração do modelo.

Além da capacidade viária, outro parâmetro a ser ajustado é a capacidade de armazenamento dos *links*, que diz respeito a quantidade de veículos que podem ocupar simultaneamente o mesmo *link*, e é calculado pela Equação 4. Ziemke e Nagel (2017) utilizam 24% da capacidade de armazenamento, para 12% de capacidade viária, enquanto Kickhöfer et al. (2016) utilizam 1,9% de capacidade de armazenamento, para 0,64% de capacidade viária.

$$Cap_a = \frac{F_t \times C_l}{C_{e_c}} \quad (4)$$

Onde:

Cap_a é a capacidade de armazenamento do *link*;

F_t é a quantidade de faixas de tráfego;

C_l é o comprimento do *link*; e

C_{e_c} é o comprimento efetivo de uma célula (igual ao *headway*¹⁰).

Os autores não descrevem a relação entre os parâmetros, contudo Nicolai (2012) utiliza o fator da capacidade viária igual à amostra de população utilizada, e descreve a relação entre os fatores da capacidade viária e da capacidade de armazenamento por meio da Equação 5.

$$F_{ca} = \frac{F_{cv}}{F_h}, \text{ onde } F_h = \sqrt[4]{F_{cv}} \quad \therefore \quad F_{ca} = F_{cv}^{0,75} \quad (5)$$

Onde:

F_{ca} é o fator de redução da capacidade de armazenamentos dos *links*;

F_{cv} é o fator de redução da capacidade viária dos *links*; e

F_h é o fator heurístico, definido por uma função de ajuste baseada na heurística.

No MATSim, a calibração do modelo pode ser realizada, ainda, de forma automatizada por meio do módulo *Cadyts* (*Calibration of dynamic traffic simulations*). De forma geral, o efeito da calibração do *Cadyts* atua como um componente adicional da pontuação, que avalia o quão bem os planos (e respectivas viagens) executados durante a simulação correspondem às observações de tráfego reais, conforme descrevem Ziemke e Nagel (2017).

¹⁰*Headway* é a distância entre a frente de um veículo e a frente do veículo imediatamente atrás desse. O valor padrão do MATSim para esse parâmetro é de 7,5 metros.

Segundo Horni, Nagel e Axhausen (2016), o Cadyts adiciona um componente de correção $\Delta S_a(k)$ para cada *link* a e cada hora de simulação k . Esse componente é definido pela Equação 6.

$$\Delta S_a(k) = \frac{y_a(k) - q_a(k)}{\sigma_a^2(k)} \quad (6)$$

onde $y_a(k)$ é o volume de tráfego real no *link* a na hora k , $q_a(k)$ é o volume de tráfego simulado correspondente, e $\sigma_a^2(k)$ é uma estimativa da variância dos volumes de tráfego reais.

A pontuação do agente soma, portanto, o componente $\Delta S_a(k)$ a sua pontuação original, de acordo com os *links* e horários em que se deslocou ao longo de seu plano diário.

Para realização da calibração e validação da simulação, é necessário utilizar dados de movimentação reais e independentes dos diários de viagem. Conforme apresentado na Seção 2.2.6, são usualmente utilizados dados de contagem de tráfego e pesquisas de velocidade para calibração e validação da simulação dos modos privados, e dados de embarque e desembarque para calibração e validação da simulação do transporte público coletivo.

Tanto para a calibração quanto para a validação, são utilizados dados disponíveis no cenário espacial e temporal utilizado. Contudo, conforme descrevem Sloboden et al. (2012), a validação implica na comparação do modelo calibrado com dados reais não utilizados na construção do modelo. Essa definição se aplica para todos os modelos de simulação de tráfego.

Para validação das viagens feitas por automóvel, o MATSim oferece módulo de comparação de volumes de tráfego nos *links* em que há contagem de tráfego, o que requer a elaboração de arquivo denominado *counts.xml*. Esse arquivo deve conter, no mínimo, o *link* relacionado ao posto de contagem, horário da contagem, e volume de tráfego horário.

Para validação das viagens feitas por transporte coletivo, é oferecido o módulo de comparação da quantidade de embarques e desembarques nos pontos E/D, o qual é considerado individualmente para cada linha de ônibus. Para isso, é necessária a elaboração de três arquivos: *counts_alighting.xml*, contendo a quantidade de desembarques nos pontos; *counts_boarding.xml*, contendo a quantidade de embarques nos pontos; e *counts_occupancy.xml* contendo a ocupação no ônibus. Os valores considerados em todos os arquivos devem ser agregados por hora, ao longo de um dia.

2.4.7 Visualização dos resultados

Para compreensão e análise dos resultados obtidos com a simulação, é necessário o levantamento de informações sobre os arquivos de saída e de suas formas de visualização.

Segundo Horni, Nagel e Axhausen (2016), o MATSim fornece arquivos de saída que permitem a análise dos resultados e o monitoramento do andamento da simulação. Alguns arquivos resumem toda a execução da simulação, e outros dizem respeito a apenas uma iteração. Os arquivos seguintes resumem toda a execução da simulação:

- arquivo de registros (*Log File*): durante a execução do MATSim, um arquivo de registros é simultaneamente elaborado, contendo informações que podem ser necessárias depois para a análise dos resultados ou de uma falha na execução;
- arquivo de registro de erros e advertências (*Warnings and Errors Log File*): esse arquivo apresenta todos os erros e advertências apresentados no *Log File* isoladamente;
- estatísticas da pontuação (*Score Statistics*): as estatísticas da pontuação são disponibilizadas em forma de imagem e de texto. Ambos arquivos apresentam a média da melhor e da pior pontuação, da pontuação executada e sua média geral para os planos de todos os agentes para cada iteração;
- estatísticas das distâncias das viagens (*Leg Travel Distance Statistics*): esses arquivos são comparáveis com as estatísticas da pontuação, mas apresentam dados da distância das viagens;
- cronômetro (*Stopwatch*): o arquivo de saída contém a quantidade de tempo utilizada para realizar ações, como o replanejamento e a execução do mobsim, para cada iteração.

Os arquivos de saída que representam os resultados de apenas uma iteração são descritos por Horni, Nagel e Axhausen (2016) como:

- eventos (*Events*): todas as ações ocorridas na simulação são registradas como um evento MATSim, seja o início de uma atividade ou a mudança de link na rede, e o arquivo com seus registros é utilizado como base nos programas visualizadores (abordados adiante);
- planos (*Plans*): é gerado um arquivo contendo os planos diários de atividades e viagens dos agentes;
- histograma das viagens (*Leg Histogram*): é criado um histograma geral e para cada modo de transporte que ilustra a quantidade de agentes iniciando, terminando uma viagem ou em rota por unidade temporal;

- duração das viagens (*Trip Durations*): é criado um arquivo de texto listando a quantidade de viagens e suas durações, para cada tipo de atividade de início e fim de viagem (casa-trabalho, ou trabalho-compras, etc);
- estatísticas dos *links* (*Link Stats*): é criado um arquivo de texto contendo o volume e o tempo de deslocamento por hora em cada *link* da rede, o qual é especialmente importante para comparação com dados de contagem de tráfego reais.

O MATSim produz muitos arquivos de saída com dados desagregados, sendo necessário agregá-los para elaborar conclusões ou previsões sobre o cenário. Para realização dessa atividade de forma automatizada, até o início de 2023, o MATSim possuía duas opções de *softwares* visualizadores: Via e OTFVis.

De acordo com Horni, Nagel e Axhausen (2016), o Via¹¹ é um programa comercial desenvolvido pela Senozon AG para visualizar e analisar os resultados das simulações do MATSim. Ele oferece a possibilidade de visualizar a rede viária, os estabelecimentos, os veículos e as atividades. São disponibilizados *plug-ins* para visualizar resultados do transporte coletivo, da comparação de volume real e simulado, de análises agregadas, para realizar gravação de vídeos da simulação, e usar imagens do Google Maps ou OSM como fundo.

Os autores descrevem ainda que o *plug-in* de transporte coletivo permite a visualização de diferentes veículos de diferentes formas, assim como a ocupação absoluta ou relativa desses veículos. Para os pontos de parada, a quantidade de passageiros aguardando por um veículos pode ser plotada ao longo do dia, e a ocupação ao longo da rota de uma linha pode ser visualizada.

O OTFVis¹² (sigla para *On The Fly Visualizer*) é uma versão simplificada para visualização de simulações do MATSim em código aberto. Ele é utilizado, sobretudo, para resolver os problemas com a execução do MATSim, e também é usado para gravação de vídeos da simulação. Contudo, não há versão estável disponível desse *software* atualmente.

O SimWrapper¹³ foi inserido no guia do usuário do MATSim em Julho de 2023, e representa uma alternativa gratuita e de fácil utilização. Segundo Horni, Nagel e Axhausen (2016), é uma ferramenta de visualização de dados baseada na web para pesquisadores que constroem simulações de transporte desagregadas de softwares como MATSim e ActivitySim.

¹¹Disponível em: <<https://www.simunto.com/via/>>

¹²Disponível em: <<https://github.com/matsim-org/matsim-libs/tree/master/contribs/otfvis>>

¹³Disponível em: <<https://vsp.berlin/simwrapper/>>

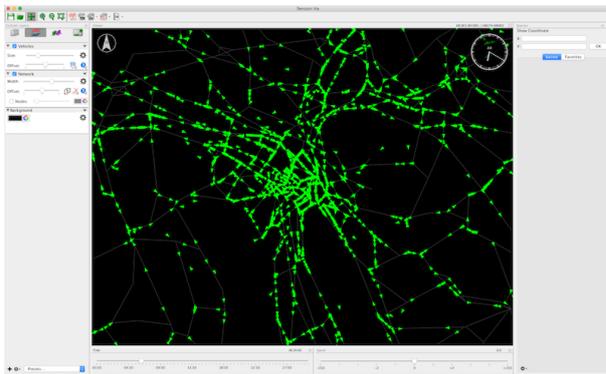
Se os seus resultados de simulação estão na sua máquina local, você pode acessar os arquivos imediatamente pelo Google Chrome e Microsoft Edge, selecionando “Adicionar pasta local” na página principal do SimWrapper e autorizando o acesso a uma pasta local. É possível a utilização de outros navegadores com a criação de *scripts* para acesso à rede local.

Não é necessário nenhum conhecimento prévio de qualquer linguagem de programação, já que o SimWrapper é essencialmente apenas um site, que lê os arquivos dos usuários diretamente e constrói visualizações de acordo com os arquivos disponíveis e a configuração nas pastas de saída do modelo.

Uma das principais características do SimWrapper é a sua capacidade de organizar múltiplos painéis de visualização em uma única página, conhecida como *dashboard*. Todos os tipos de visualização descritos acima podem ser organizados como painéis em um *dashboard* do SimWrapper.

A Figura 8 apresenta o layout dos três visualizadores.

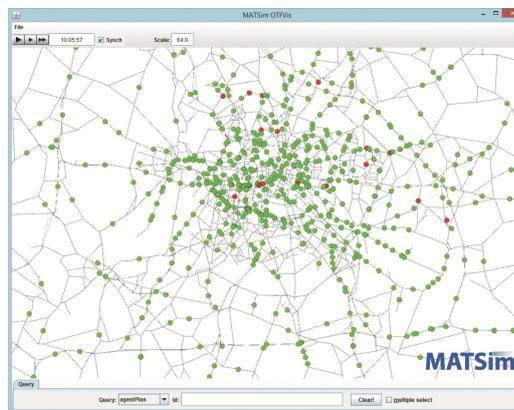
Figura 8: Layout dos principais visualizadores externos de resultados do MATSim



(a) Via Simunto



(b) SimWrapper



(c) OTFVis

Outra possibilidade no futuro é o pacote Python “*matsim-tools*” que permite a análise de dados por meio da integração com os pacotes Pandas e Geopandas. Atualmente, o *site* desse projeto afirma que ele está em fase inicial de desenvolvimento e, portanto, a API será alterada, os procedimentos podem falhar e certamente ocorrerão falhas. Os autores não recomendam sua utilização atualmente.

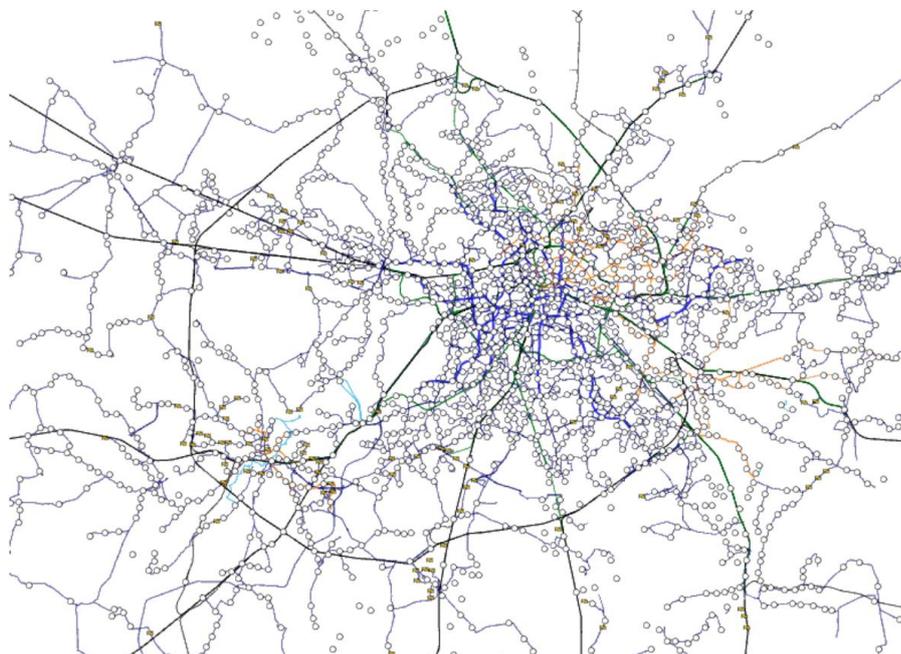
2.5 Estudos de caso relacionados

Essa seção apresenta as principais características e resultados de estudos de caso relacionados ao tema desse trabalho. Os estudos abordam a elaboração de modelos de simulação no MATSim em áreas urbanas.

Berlim (Alemanha) – Oliveros (2014) Considerando o cenário de Berlim, Oliveros (2014) realizou diversos testes de calibração para alocação das viagens nas linhas de transporte coletivo. Dois subcenários são utilizados, sendo o primeiro uma pequena área do distrito de Neukölln, o qual possui uma linha de ônibus que viaja ao longo de 17 pontos de parada, nos quais são disponíveis os volumes horários de ocupação de passageiros. A demanda inclui 36.119 usuários do transporte coletivo que realizam suas atividades em locais próximos aos pontos de ônibus.

O segundo subcenário considera toda a rede de transporte coletivo de Berlim, com 328 linhas e demanda de 231.369 pessoas. Para esse cenário, os volumes de ocupação de passageiros de 2.723 pontos de parada são disponíveis. A Figura 9 ilustra a rede de transporte coletivo de Berlim.

Figura 9: Rede de transporte público da região metropolitana de Berlim. Fonte: Oliveros (2014).



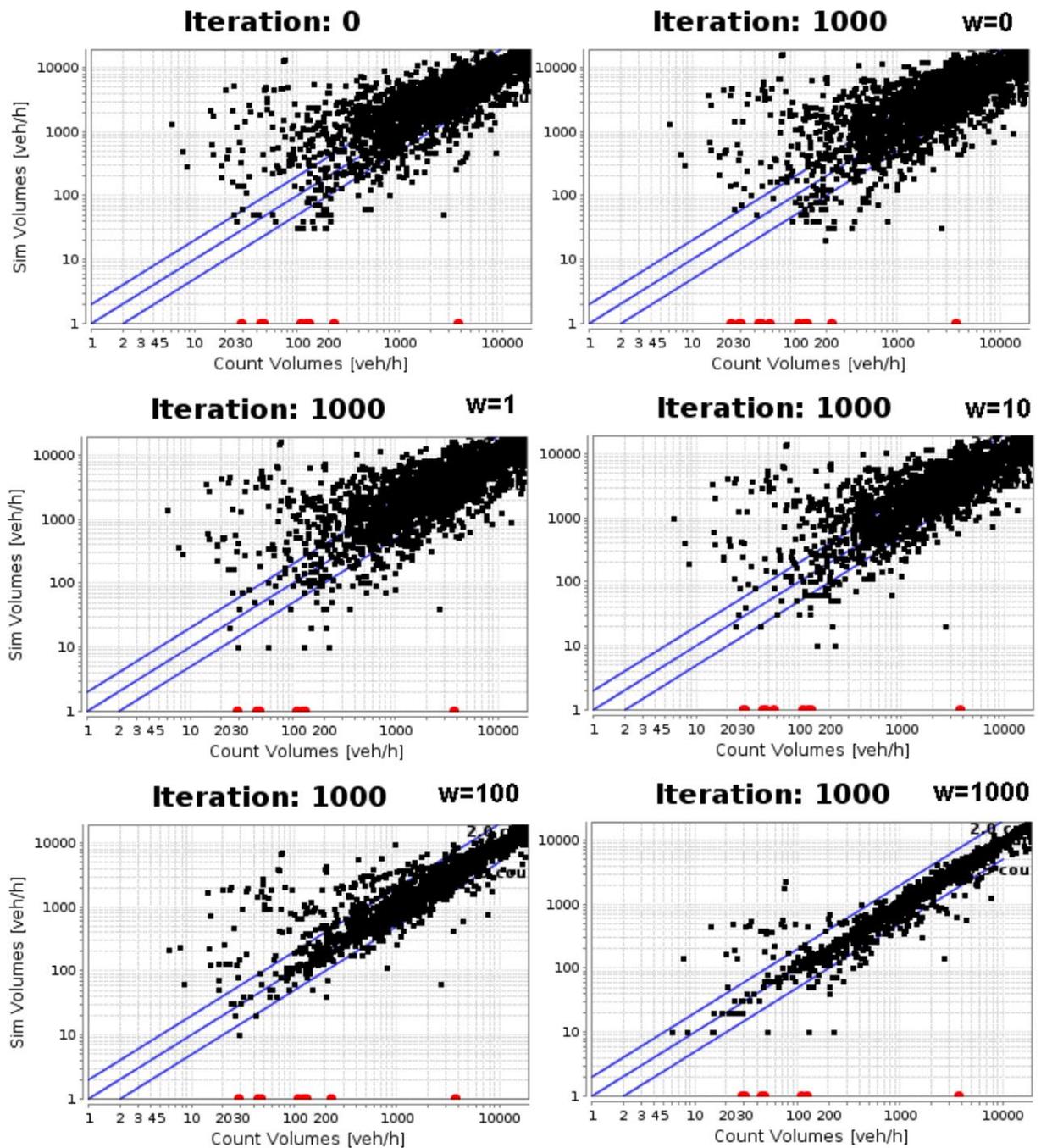
Em ambos cenários, a demanda de viagens é gerada a partir de informações de uma pesquisa domiciliar que é estruturada para descrever um dia típico completo das atividades de uma pessoa em diferentes locais da cidade. O modelo considera apenas as viagens realizadas de transporte coletivo.

A calibração foi efetuada primeiramente de forma manual e, em seguida, foi utilizado o calibrador automático *Cadyts*. A performance dos métodos de calibração testados são avaliados por meio do MRE (desvio relativo, traduzido do termo inglês *Mean Relative Error*).

A calibração manual consistiu na alteração do peso de três parâmetros de pontuação: tempo de caminhada, distância viajada de transporte coletivo, e utilidade na troca de linhas. A simulação com melhor resultado utilizou os pesos de -10 para o tempo de caminhada (comparado ao -6 do padrão MATSim), 0 para a distância viajada de transporte coletivo (igual ao padrão MATSim), e 240 para a utilidade na troca de linhas (comparado a 60 do padrão MATSim). A porcentagem de MRE resultante da comparação entre volumes de embarque e desembarque reais e simulados flutua entre 30% a 50%.

Para a calibração automática com o *Cadyts*, Oliveros (2014) apresenta os resultados exibidos na Figura 10 e aponta que pesos baixos para o parâmetro definido têm baixo impacto sobre os resultados. Por outro lado, com peso de 100 e 1000, as alterações podem ser vistas já na primeira iteração. Nas simulações em que o peso *Cadyts* foi alto, foi possível perceber uma queda brusca no erro após a parada das inovações nos planos e início da calibração total.

Figura 10: Calibração com *reroute* e *Cadyts* com peso crescente. Fonte: Oliveros (2014).



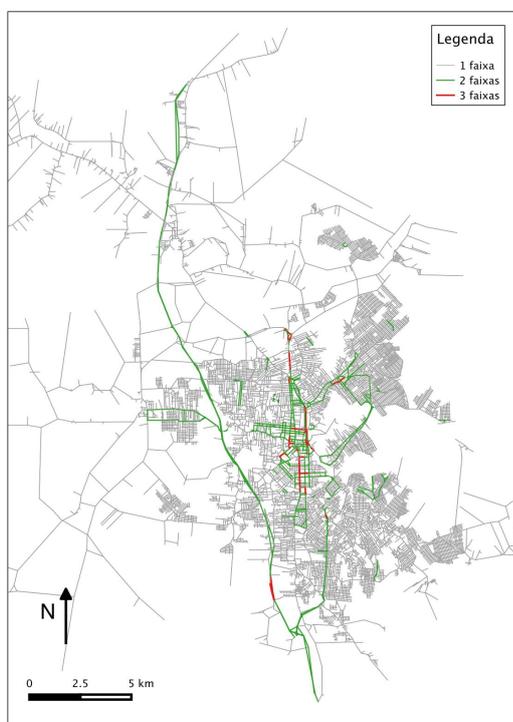
A ferramenta se mostrou adequada, e teve como MRE resultados entre 10% e 20%, sendo que as linhas que não foram calibradas não apresentaram maior erro do que apresentavam antes da calibração nas demais linhas.

Joinville/SC – Bicudo (2015) Segundo o autor Bicudo (2015), a aplicação do MATSim em Joinville/SC simula as viagens da população do município em que são utilizados carros para realização de atividades de moradia, trabalho, lazer, compras e educação de acordo com a localização desses estabelecimentos.

Os dados obtidos e utilizados para a confecção do modelo de Joinville são listados abaixo, e a rede viária modelada é apresentada na Figura 11.

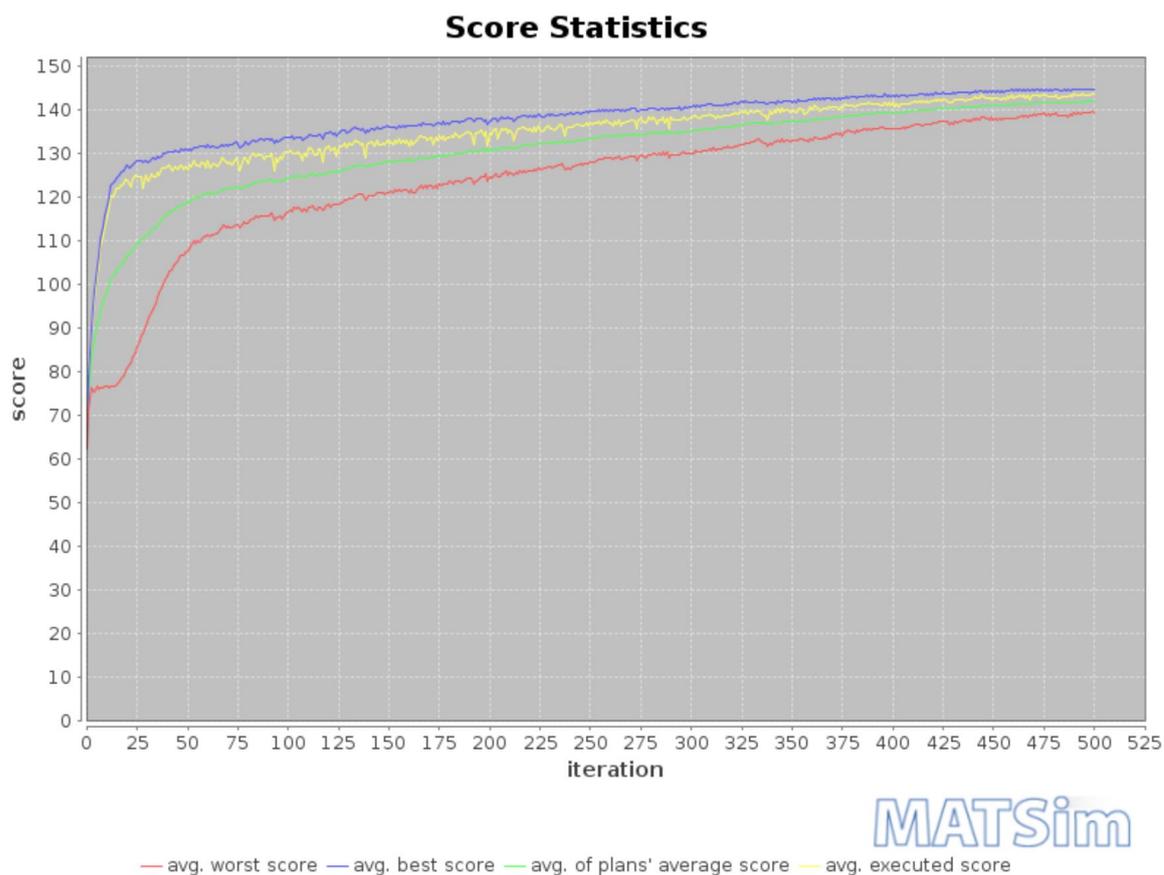
- rede viária municipal, advinda do Fundação Instituto de Pesquisa e Planejamento para o Desenvolvimento Sustentável de Joinville (2015);
- pesquisa origem-destino do município de Joinville do ano de 2010, advinda do Fundação Instituto de Pesquisa e Planejamento para o Desenvolvimento Sustentável de Joinville;
- censo demográfico 2010, advinda do IBGE (2010);
- banco de dados do Sistema de Gestão Cadastral (SGC) da Prefeitura Municipal de Joinville; e
- contagens de tráfego da Prefeitura Municipal de Joinville.

Figura 11: Modelo de rede viária de Joinville. Fonte: Bicudo (2015)



Foram utilizados os parâmetros de simulação padrão do MATSim, e realizadas 500 iterações, sendo que o modelo apresentou estado de equilíbrio na 350ª iteração. A Figura 12 ilustra a evolução da pontuação dos planos dos agentes ao longo das iterações.

Figura 12: Evolução da pontuação dos planos no cenário de Joinville. Fonte: Bicudo (2015).



O modelo foi calibrado e validado por meio de contagens de tráfego em 20 postos de contagem localizados nos semáforos. O autor não calculou os coeficientes de correlação dos volumes de tráfego simulados e reais, mas apresentou os resultados comparativos ao longo do dia para os dois postos com melhor e pior performance, e os resultados comparativos de todos os postos na hora pico da manhã e da tarde. o melhor resultado foi no período entre 8h e 10h, com erros relativos próximos de 30%. O pico da tarde, entre 16h e 17h também apresentou erros relativos próximos a 55%. De forma geral, os volumes simulados são inferiores aos volumes reais, mas os horários de pico apresentaram uma boa comparação.

Santiago (Chile) – Kickhöfera et al. (2016) O trabalho de Kickhöfera et al. (2016) simula o cenário analisado pela pesquisa de origem e destino de 2012 em Santiago, com população estimada de 6,65 milhões de pessoas. Na simulação é utilizada uma amostra de apenas 0,65% da população, para a qual é realizada a diminuição da capacidade viária para 0,64%.

Para modelagem da rede de tráfego, foram utilizados os dados do *OpenStreetMap*, cujas velocidades foram ajustadas devido à diminuição da velocidade de tráfego causada pelo tráfego de pedestres e ciclistas nas vias, assim como a existência de semáforos, que não foram incluídos explicitamente no modelo. Os critérios de ajuste e seus fatores de redução da velocidade de fluxo livre, escolhidos aparentemente de maneira arbitrária, são:

- $v_{livre,nova} = 0,50 \times v_{livre,OSM}$ para vias de até 40 km/h e todas as vias de até 60 km/h com apenas uma faixa;
- $v_{livre,nova} = 0,75 \times v_{livre,OSM}$ para vias de até 60 km/h com duas faixas;
- $v_{livre,nova} = 1,00 \times v_{livre,OSM}$ para todas as outras vias.

Os parâmetros de comportamento utilizados para atribuição das pontuações dos planos tiveram como referência o estudo de Munizaga et al. (2008), enquanto os parâmetros ASC (Constante de Alternativa Específica, traduzido livremente de *Alternative Specific Constants*) são determinados no processo de calibração do modelo. A determinação desses parâmetros é ajustada de forma iterativa por meio da Equação 7.

$$C_{mode,n+1} = C_{mode,n} - \log \frac{p_{mode,n}}{p_{mode,it.0}} \quad (7)$$

onde:

n é a iteração da simulação;

$C_{mode,n+1}$ é a ASC calculada para a próxima iteração;

$C_{mode,n}$ é a ASC na iteração atual;

$p_{mode,n}$ é a divisão modal ao final das iterações; e

$p_{mode,it.0}$ é a divisão modal na iteração zero da simulação.

A simulação contemplou a alocação das viagens realizadas por carros, por transporte coletivo e a pé e com a possibilidade de migração modal ativada (*ChangeMode*), e foi calibrada com base na divisão modal resultante da pesquisa OD. Os volumes de tráfego nos *links* foram validados apenas de forma visual com base nos resultados de contagens de tráfego em 36 postos, das 6:00

às 23:30. O gráfico de dispersão do volume de tráfego diário simulado e obtido nas contagens é ilustrado na Figura 13, e a Figura 14 ilustra as atividades dos agentes em toda a área simulada à meia-noite, à esquerda, e o movimento de carros e veículos de transporte público às 8 horas da manhã, à direita. Triângulos vermelhos indicam carros em congestionamento, enquanto triângulos verdes mostram carros em fluxo livre. .

Figura 13: Gráfico de dispersão dos volumes de tráfego diários simulados e reais nos postos de contagem. Fonte: Kickhöfer et al. (2016).

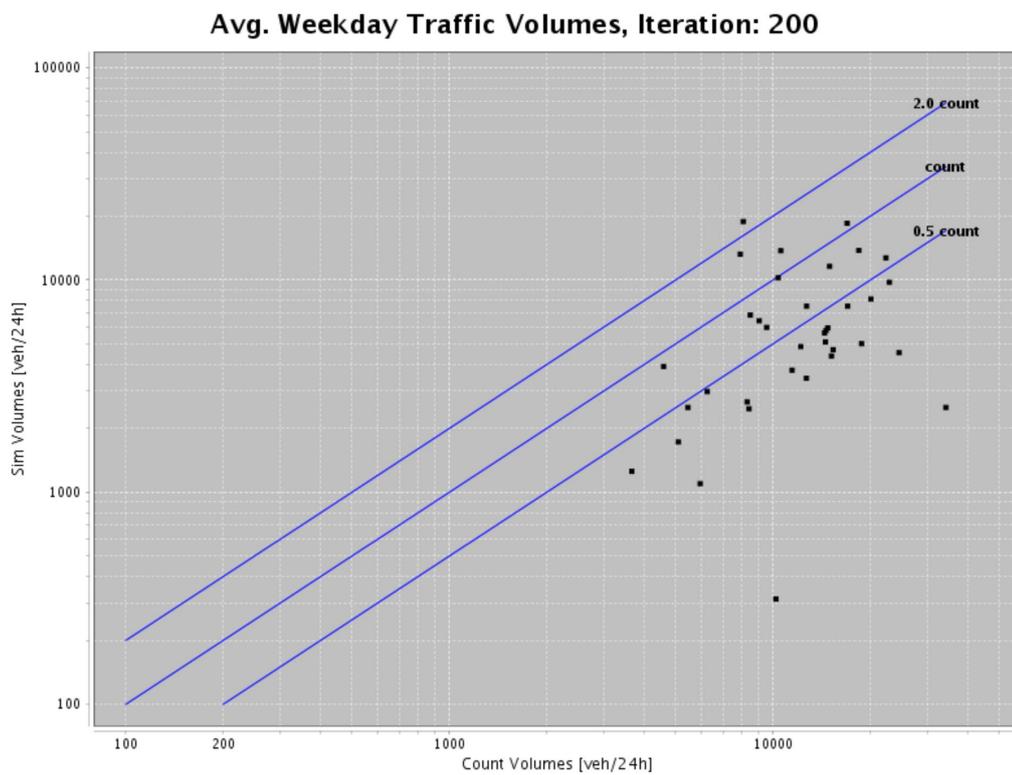
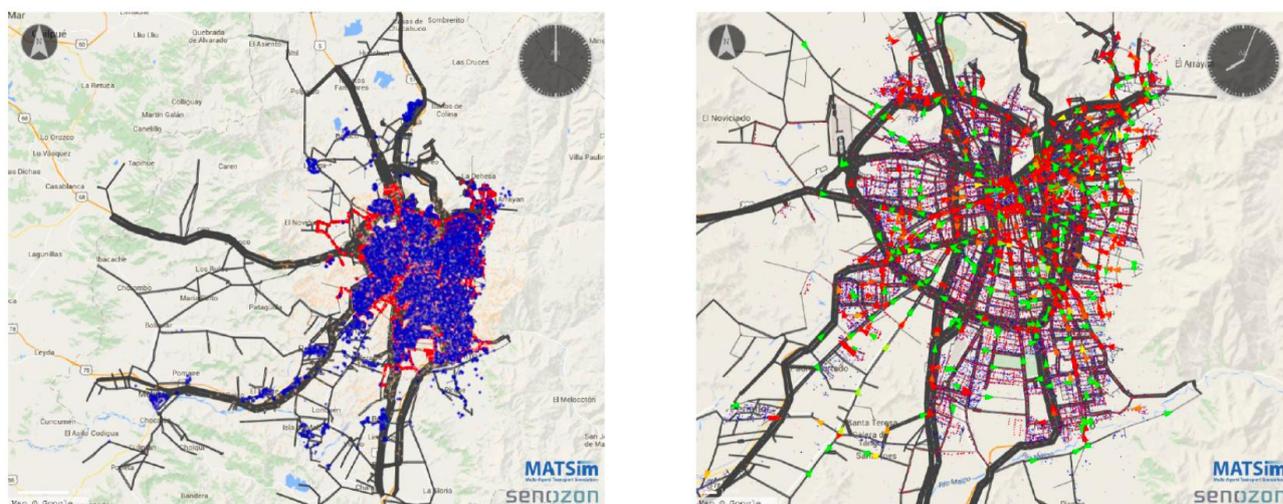


Figura 14: Resultados da simulação em toda a área simulada, à esquerda, e na área central, à direita. Fonte: Kickhöfera et al. (2016).



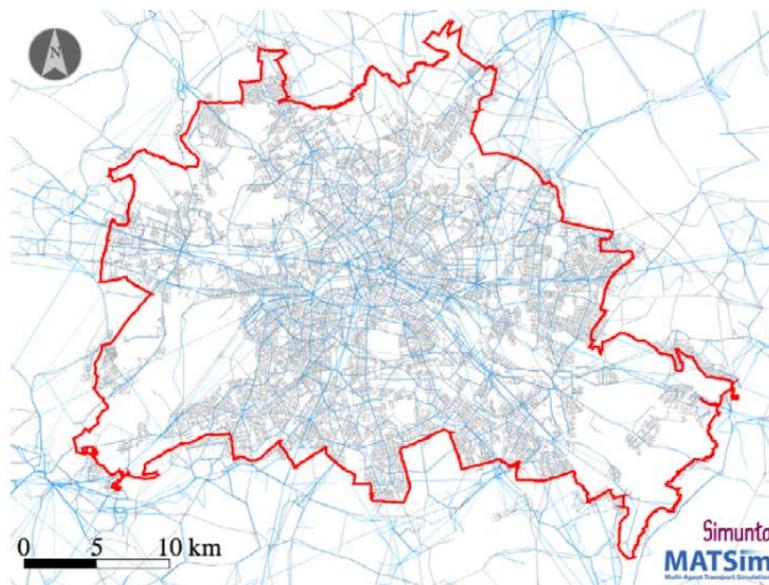
Devido à pequena amostra utilizada, os padrões de congestionamento resultantes não foram considerados suficientemente compatíveis com a realidade, já que os volumes de tráfego simulados são, de forma geral, menores que os volumes reais. Os autores recomendam a utilização de uma amostra populacional maior.

Berlim (Alemanha) – Ziemke e Nagel (2017) No trabalho de Ziemke e Nagel (2017), foi realizada simulação das viagens realizadas pela população da região metropolitana de Berlin e Brandenburgo na Alemanha em todos os modos de transporte disponíveis (veículos de passeio como motoristas ou passageiros, ciclistas, pedestres e transporte público). O modelo foi configurado para ser sensível às políticas em termos de escolha de rota, modo e horário de partida.

As seguintes fontes de dados são utilizadas: 'Zensus 2011', um censo nacional da Alemanha, 'Pendlerstatistik 2009' (estatísticas de deslocamento), *OpenStreetMap*, dados GTFS locais, contagens de tráfego local, contagens da BAST sobre tráfego de carga, shapefiles que descrevem as geometrias municipais em Brandenburgo e as geometrias LOR1 em Berlin, e dados de uso do solo CORINE. Para validação de várias propriedades de viagem, são utilizadas as pesquisas de viagem Berlin SrV 2008 e Berlin MiD 2008.

A rede viária modelada (em cinza) e a rede do sistema de transporte público (em azul) são ilustradas na Figura 15.

Figura 15: Modelo da rede viária (em cinza) e a rede do sistema de transporte público (em azul).
Fonte: Ziemke e Nagel (2017).



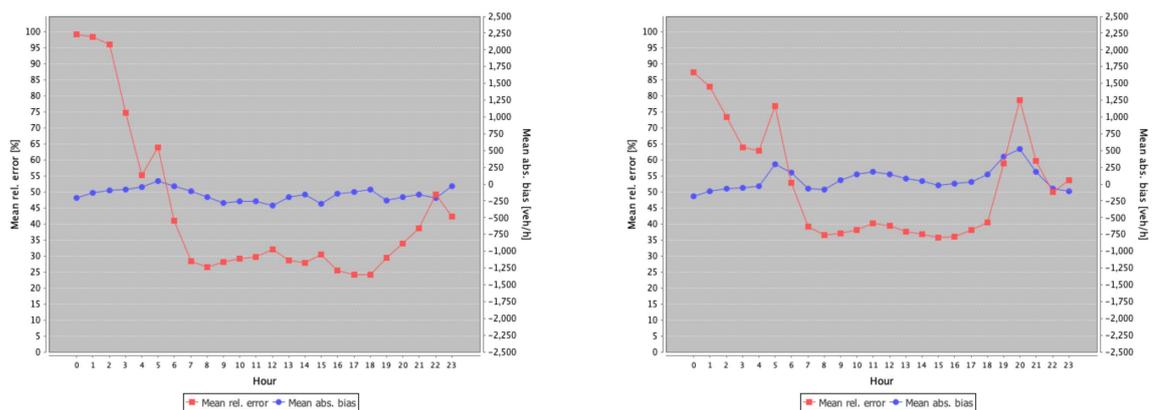
O modelo utilizou 10% da população, para 10% da capacidade viária de fluxo e de armazenamento, e peso das estratégias de inovação igual a 5%, sendo elas a alteração do modo de transporte, do tempo de partida e da rota utilizada. Os parâmetros específicos de cada modo foram definidos nessa etapa da calibração.

Para a modelagem da demanda, foi utilizado o CEMDAP (*Comprehensive Econometric Micro-simulator for Daily Activity-travel Patterns*) para criação dos padrões diários de viagens e atividades de cada agente. O *Cadyts* foi utilizado nessa etapa para obter planos adequados (principalmente no que toca a localização das atividades) a partir dos planos iniciais gerados pelo CEMDAP, e a subsequente calibração dos modos de transporte.

Para a primeira etapa de calibração do modelo (em relação ao CEMDAP), o *Cadyts* foi executado em conjunto com o MATSim, utilizando os dados de 346 postos de contagem de tráfego. Para a segunda etapa de calibração (em relação às viagens), o *Cadyts* não foi utilizado, e as pessoas poderiam escolher outros modos de transporte além do carro, os quais foram simulados por “teletransporte” no modelo (o funcionamento do teletransporte no MATSim é apresentado na seção 2.4.3). Para obtenção de velocidades de fluxo livre mais realistas, a velocidade das vias dentro do núcleo urbano são reduzidas pela metade para considerar semáforos, aceleração e desaceleração em interseções, de forma aparentemente arbitrária.

A validação do modelo foi realizada apenas de forma gráfica, e não são apresentados resultados numéricos de validação. A Figura 16 ilustra o erro médio relativo e absoluto após a calibração dos locais das atividades (esquerda) e após a calibração da divisão modal (direita).

Figura 16: Distribuição horária dos erros médios relativos e absolutos após a calibração dos locais das atividades (esquerda) e após a calibração da divisão modal (direita). Fonte: Ziemke e Nagel (2017).



Distrito Federal – Miranda (2017) No estudo de Miranda (2017), foi desenvolvida uma metodologia para tratar dados e simular um modelo multiagente utilizando o MATSim. As informações foram coletadas de bases de dados de código aberto e relatórios censitários, e foram processadas e analisadas usando softwares gratuitos. Isso permitiu a modelagem detalhada do transporte em situações onde há limitações financeiras no planejamento.

Foram utilizados os dados do OSM como base para modelagem da rede viária e dos pontos de atividades na região do Distrito Federal.

A modelagem da população sintética exigiu muitas informações censitárias e um código mais longo e complexo do que o usado para os dados geográficos. Para simplificar, as pessoas foram agrupadas de acordo com seu perfil de atividades: (i) só trabalham; (ii) só estudam; (iii) trabalham e estudam; e (iv) sem atividade. Foram utilizadas informações sobre a situação de ocupação e sobre a frota veicular da Pesquisa Distrital por Amostra de Domicílios do Distrito Federal (CODEPLAN, 2012) e os dados da pesquisa origem-destino realizada para a área de estudo no ano de 2009 (Secretaria de Transportes, 2011).

No computador utilizado no estudo, a simulação no MATSim 0.8.1 com 50 iterações e com escala de 10% da população do DF, levou 51 horas, 11 minutos e 22 segundos para ser realizada. A pontuação obtida nessa simulação é apresentada na Figura 17.

Figura 17: Evolução da pontuação da simulação do tráfego no DF. Fonte: Miranda (2017).



Os dados tratados foram usados para alimentar um modelo MATSim em um estudo de caso na região do Distrito Federal e Entorno, resultando em um arquivo com informações detalhadas sobre o comportamento de transporte de cada indivíduo ao longo do dia. A comparação entre contagens volumétricas em quatro postos de contagem resultou em erros médios entre 33% e 111%, e o modelo mostrou-se subdimensionado, possivelmente devido à exclusão de ônibus e outros veículos do sistema de transporte da região.

Luxemburgo – Bigi, Schwemmle e Viti (2023) O estudo de Bigi, Schwemmle e Viti (2023) tem como área de estudo Luxemburgo, e visa avaliar os resultados da medida de gratuidade universal no transporte público gratuito, implantada em 2020 pelo governo federal, já que o impacto dessa política no uso de carros é controverso.

O cenário desenvolvido simula os principais modos de transporte, incluindo carros, transporte público, bicicletas e caminhadas. Ele fornece horários detalhados para serviços de transporte público, como ônibus, bondes e trens. A modelagem da rede de transporte foi configurada usando duas fontes de dados abertos: informações de rede do OSM e horários de transporte público gerados a partir do GTFS. Os arquivos GTFS foram convertidos para o formato MATSim por meio

da ferramenta *pt2matsim*. A geração de demanda e cadeias de viagens, bem como as matrizes origem-destino, foram obtidas na pesquisa de viagens LuxMobil realizada em 2017. Após a geração da demanda, a calibração do cenário MATSim foi realizada usando os dados da pesquisa de viagens, bem como contagens de tráfego.

Para calibrar o cenário antes da política de TP gratuito, foram executadas diferentes simulações com 10% da população total (cerca de 64.000 agentes) para alcançar o equilíbrio da pontuação. Isso foi alcançado realizando múltiplas simulações, cada uma com 250 iterações, a partir dos parâmetros usados para o cenário de Berlim, conforme Ziemke e Nagel (2017). Os parâmetros de pontuação foram ajustados após cada simulação até que a convergência fosse alcançada com os dados de contagem de tráfego, bem como com a divisão modal, tempo médio de viagem e distância média por modo da pesquisa de viagens Luxmobil.

Os resultados da calibração do modelo não foram apresentados, e a apresentação de resultados do estudo teve como foco as alterações no comportamento dos usuários com a implantação da gratuidade universal no TP, que apresentou 10% dos usuários migrando para o transporte coletivo.

Dhaka (Bangladesh) – Zannat et al. (2024) O estudo de Zannat et al. (2024) previu a demanda de BRT (Bus Rapid Transit) na cidade de Dhaka, utilizando o MATSim. Um modelo de escolha de modal foi desenvolvido usando dados de preferência observada e de preferência declarada para avaliar os impactos da implantação de uma linha de BRT.

Foram utilizados dados de diários de viagens em toda a área da Rajuk realizado por pesquisa domiciliar de um Projeto de Metrô de Dhaka. Os dados de preferência declarada contém entrevistas com 1.016 indivíduos, realizadas por meio de aplicação de questionário à beira da estrada.

Uma vez que o escopo do trabalho foi limitado ao BRT, apenas os componentes do modelo conjunto relacionados à escolha entre os modos existentes (caminhada, bicicleta, riquixá, ônibus, transportador humano, motocicleta, auto-rickshaw) e BRT foram usados para a simulação dos cenários futuros.

Foram gerados planos de atividade usando microdados de uma amostra representativa da população. No entanto, o diário de viagem incluía apenas a localização detalhada do domicílio dos participantes, e as localizações das diferentes atividades estavam disponíveis apenas no nível de Zona de Tráfego (ZT). As localizações das várias atividades foram atribuídas aleatoriamente dentro do

limite da ZT usando o Sistema de Informação Geográfica (SIG) de forma a corresponder ao tempo de viagem declarado pelos usuários.

A rede para a área de estudo foi obtida a partir do *OpenStreetMap* (OSM). No MATSim, os modos disponíveis são carro, transporte público (TP), bicicleta e caminhada. Outros modos, como riquixás motorizados e motocicletas foram modelados utilizando especificações veiculares especiais dentro do arquivo *mode-vehicles*. O ajuste do lado da oferta foi feito no MATSim considerando a capacidade de fluxo e o fator de capacidade de armazenamento de 0,015.

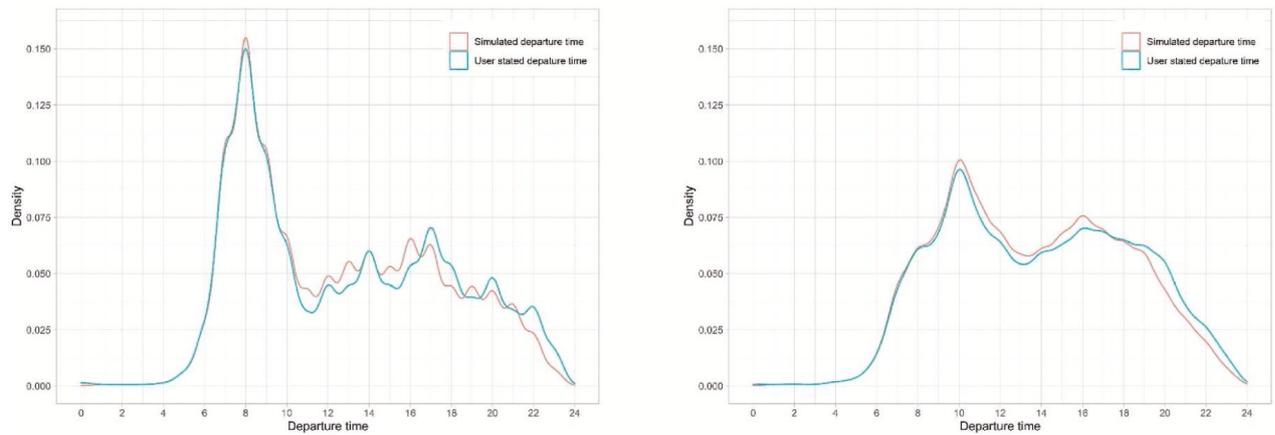
As configurações padrão de configuração da simulação MATSim foram usadas como ponto de partida neste estudo. No entanto, no módulo padrão, o modo de acesso/saída para o TP é limitado apenas à caminhada. No cenário de acesso/saída multimodal para este estudo, foram consideradas caminhada, riquixá e bicicleta. Parâmetros de raio de pesquisa adicionais foram especificados para riquixás e bicicletas: se nenhuma parada de TP fosse encontrada dentro do raio inicial (4 km), o raio de pesquisa era expandido até encontrar uma parada (até o raio máximo de pesquisa de 6 km).

Para estimar os parâmetros de pontuação para a atribuição de modo, foram desenvolvidos modelos de escolha de modo seguindo o *framework* de utilidade aleatória. Os modelos de escolha de modo desenvolvidos previram a escolha entre os modos existentes (por exemplo, carro, ônibus, riquixá, transportador humano, riquixá motorizado, motocicleta, caminhada, bicicleta) e BRT.

Devido à exclusão das ruas estreitas nos arquivos de rede disponíveis (que são aplicáveis para caminhada, ciclismo e riquixá), os modos não motorizados foram simulados usando 'distâncias em linha reta ajustadas' por meio do 'módulo de teletransporte' dentro do MATSim. Devido à ausência de informações confiáveis sobre as rotas de ônibus, eles também foram simulados usando o recurso de 'teletransporte', mas com distâncias de viagem derivadas da rede.

O estudo não apresenta medidas de desempenho para comparação entre observações reais e simuladas, mas os autores afirmam que os modelos desenvolvidos funcionaram razoavelmente bem em todas as dimensões consideradas neste estudo (tempo de viagem, custo de viagem e modo de acesso/saída). A Figura 18 ilustra esses resultados.

Figura 18: Densidade dos tempos de partida observados e simulados para diferentes motivos de viagem. Fonte: Zannat et al. (2024).



Em termos de aplicação prática, o modelo desenvolvido neste estudo pode ajudar a entender os padrões de atividade e o comportamento de viagem da população após o início de um novo serviço de BRT.

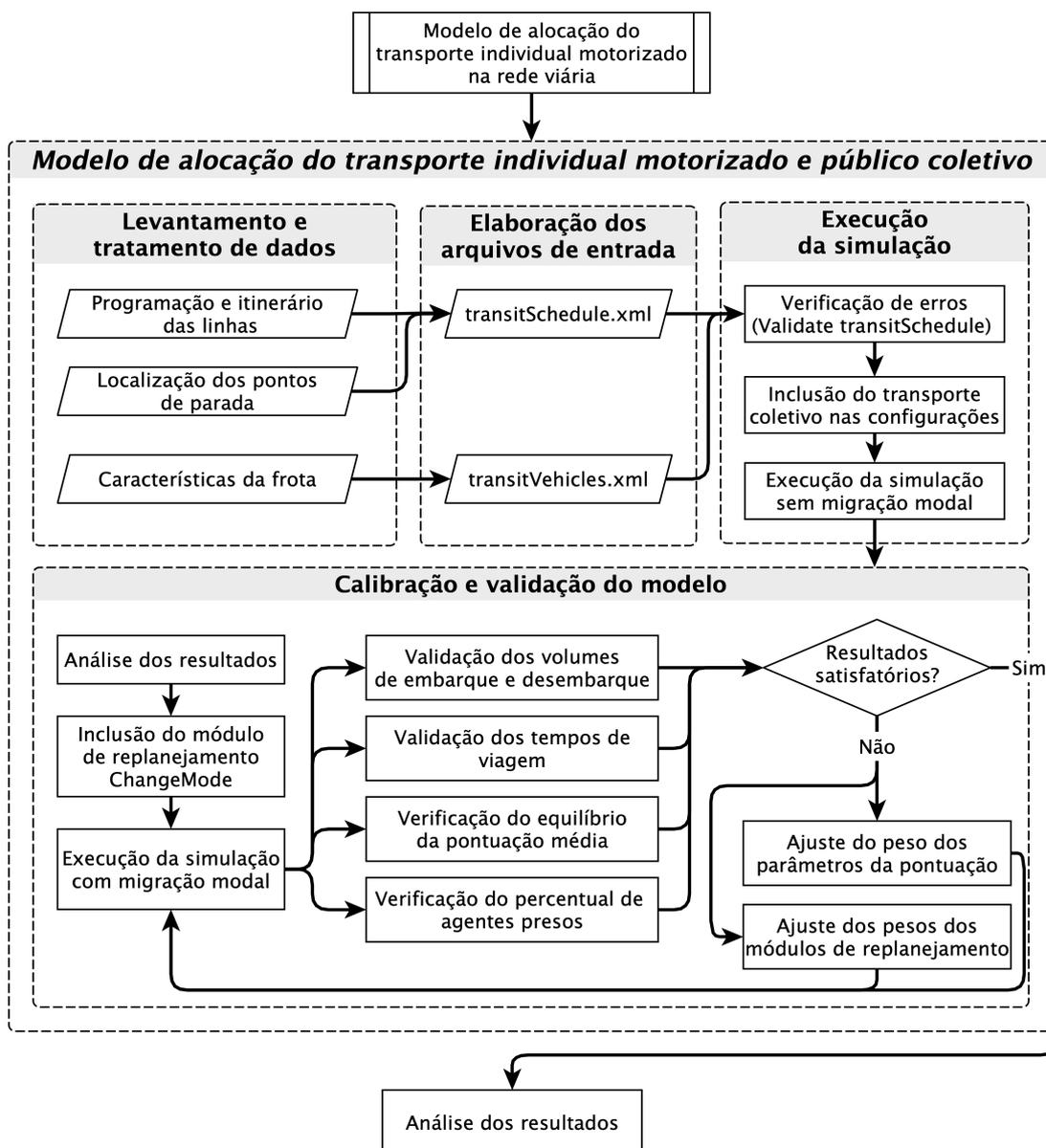
3 Método

A concepção do modelo multimodal, em termos gerais, é composta pelo método de elaboração do modelo de alocação do transporte individual motorizado, apresentado na Figura 19, e pelo método de elaboração do modelo de alocação multimodal do transporte individual motorizado e público coletivo, ilustrada na Figura 20. Apesar da ordem de aplicação do método seguir os fluxogramas, o detalhamento dos itens segue a ordem das atividades em comum realizadas em cada modelo, e abrange ambos modelos de alocação.

Figura 19: Concepção básica do método de elaboração do modelo de alocação do transporte individual motorizado na rede viária



Figura 20: Concepção básica do método de elaboração do modelo de alocação do transporte individual motorizado e público coletivo na rede viária



3.1 Levantamento e tratamento de dados

Esta seção apresenta o método de levantamento dos dados necessários para a elaboração de um modelo de simulação no MATSim, assim como do tratamento dos dados levantados.

3.1.1 Diários de viagem

Para a modelagem da demanda no MATSim, são utilizados os diários de viagem da população, abordados na Seção 2.4.2.

O levantamento primário desses dados é realizado por meio de entrevistas realizadas na pesquisa de origem e destino (O/D), a qual requer alto nível de recursos financeiros, técnicos e humanos. Nesse sentido, o levantamento de dados reais sobre os planos diários dependem da existência prévia de uma pesquisa O/D, e de sua disponibilização pelas respectivas entidades responsáveis. Portanto, na maioria dos casos, a sua realização para fins acadêmicos pode ser considerada inviável, e estudos recorrem à elaboração de populações sintéticas (vide Seção 2.5).

No caso da utilização de diários de viagens advindos de entrevistas, a primeira etapa após o levantamento da informação é a verificação de erros. Isso porque as entrevistas estão sujeitas a erros humanos durante o processo de aplicação do questionário ou na tabulação de seus resultados. São retiradas da base de dados as viagens sem a indicação de origem ou destino, sem a indicação do horário de saída ou chegada aos destinos, ou horários incompatíveis (negativos ou com duração improvável).

Dados seus altos custos de realização, a pesquisa O/D contempla apenas uma amostra da população. A definição da amostra é realizada considerando indicadores socioeconômicos relacionados à geração de viagens, como a renda domiciliar, a quantidade de moradores no domicílio, a posse de automóveis. Os fatores a serem utilizados dependem da disponibilidade desses dados de forma distribuída pelo território. Para garantir a representatividade estatística dos resultados, após a realização das entrevistas, o resultado obtido com essa amostra deve ser expandido para a obtenção dos diários de viagens para toda a população. A expansão é calculada com base em fatores de expansão que consideram os mesmos fatores socioeconômicos utilizados na etapa de definição da amostra.

Após a realização da multiplicação de cada registro de viagem pelo seu fator de expansão final (igual a multiplicação de todos os fatores supracitados), é realizada a redução da população para processamento eficiente da simulação sem perder suas proporções. Para isso, cada registro de viagem expandido deve ser multiplicado pela proporção da população que se deseja representar no modelo de simulação. No MATSim, as viagens são vinculadas a uma pessoa específica e, portanto, não é possível utilizar dados decimais de quantidade de viagens e são sempre consideradas quantidades inteiras de viagens. Como esse arredondamento ocorre para todos os registros de viagem, é esperado

certo grau de erro na quantidade de viagens modeladas, que deve ser monitorado ao longo do tratamento de dados.

A utilização dos fatores de expansão tem como resultado a cópia dos diários de viagens exatamente como são. Isso implica que o horário de saída e chegada de todas as viagens e os locais de origem e destino nos planos diários gerados a partir daquela entrevista são exatamente iguais. Para a adequação da demanda à abordagem baseada em atividades, é necessário aplicar outros procedimentos para a modelagem dos planos diários da população sintética, descritos em detalhes na Seção 3.2.

3.1.2 Rede viária

Para a modelagem da rede viária, é necessária a utilização de uma representação georreferenciada da malha, que contenha informações da localização dos nós (interseções entre os *links*), e da capacidade viária, da velocidade de fluxo livre, do sentido de tráfego e dos modos permitidos nos *links*.

Idealmente, as informações sobre a rede viária a ser utilizada advém de fontes oficiais e possuem data de referência próxima à data de coleta dos dados de demanda de viagens na área de estudo.

No caso destes dados não existirem ou não estarem publicamente acessíveis, outras fontes de dados alternativas podem ser exploradas. O OpenStreetMap (OSM), por exemplo, oferece uma base de dados no formato adequado. No entanto, considerando que o OSM é constituído de dados editados pelo público em geral, não estando submetidos a um controle de qualidade ou veracidade, é aconselhável realizar uma verificação das características da malha por meio da comparação amostral com outra base de dados, preferencialmente advinda de fonte oficial ou de observação em campo. Essa verificação deve comparar os valores indicados de sentido, capacidade viária e velocidade nas vias. Para verificação eficiente das informações, devem ser consideradas vias de todas as hierarquias viárias existentes, em diferentes locais da área de estudo. A seleção da quantidade de vias a serem analisadas por hierarquia viária e local deve considerar a extensão e diversidade da malha viária da área de estudo.

No MATSim, não há diferenciação entre vértice e nó. Portanto, os dados da rede viária devem passar por processo de simplificação de geometria, de forma a evitar que cada vértice da malha se torne um nó, ou interseção viária. Isso porque cada entrada em um novo *link* requer certo tempo de processamento, e quanto maior a quantidade de *links*, mais demorado é o processamento

da simulação. Para que os *links* sejam representados apenas por uma linha reta entre seus pontos inicial e final, apenas as coordenadas desses pontos devem ser indicadas na criação do arquivo *network.xml*.

Como a geometria da malha viária é alterada, é importante registrar o comprimento de cada segmento de via anteriormente ao processo de simplificação da geometria, para sua posterior inclusão no modelo de simulação. Além disso, para a visualização dos resultados da simulação na malha viária original, é importante criar códigos compatíveis entre os *links* modelados e os *links* originais. Assim, é possível unir as tabelas de atributo por meio do código em comum e visualizar os volumes de tráfego, velocidades e outros resultados na malha viária com a geometria original.

3.1.3 Contagens de tráfego

Assim como a Pesquisa O/D, a realização de contagens de tráfego requer alto nível de recursos. Principalmente porque é necessária a realização das contagens em vários pontos distribuídos ao longo da área de estudo, incluindo as *screen lines* e *cordon lines*, e em vários períodos do dia. Nesse sentido, a utilização desses dados em pesquisas acadêmicas requer a disponibilização dessas informações pelas entidades responsáveis por sua realização.

Nas *screen lines*, ou em outros pontos onde há interesse prévio (pela grande quantidade de volume de tráfego ou pelo padrão de distribuição temporal de viagens atípico em relação aos demais), é recomendada a realização de contagens de dia inteiro para definição dos horários de pico. Nos demais pontos, a contagem pode ser realizada apenas nos horários de pico do sistema.

Ainda, é importante que as informações utilizadas para construção e validação do modelo sejam independentes entre si. Assim, os dados de contagens de tráfego disponíveis devem ser separados em dois grupos para servir a esses propósitos. Os grupos devem conter proporção semelhante de postos com diferentes características, como a hierarquia viária do *link* onde se encontra, a duração da contagem, os volumes de tráfego registrados e a sua localização.

3.1.4 Oferta do transporte coletivo

Para modelagem da oferta de infraestrutura do TC, são necessárias informações a respeito dos veículos operantes da frota (capacidade em pé e sentada, comprimento do veículo e fator de equivalência com carro de passeio, ou unidade veículo padrão - UVP), dos pontos de parada (localização e existência de baia) e das linhas (itinerário, horários ofertados, veículos empregados).

O itinerário deve conter a lista dos pontos de parada em ordem de passagem, assim como indicação espacial dos *links* em que o ônibus trafega. Devem ser indicados os horários de saída de cada viagem e o respectivo veículo empregado para operação da linha, com suas respectivas características.

As informações sobre a frota, as linhas operadas e os pontos de parada do TC possuem organização e formato variados. Caso os dados sejam disponibilizados em formato GTFS¹⁴, a conversão é mais simples e o próprio MATSim possui um módulo que realiza esse procedimento. Quando há disponibilização dos dados necessários, mas em formato diferente do GTFS, é necessário elaborar *scripts* de conversão automatizada dos dados, utilizando dados georreferenciados dos pontos de ônibus e dos itinerários das linhas para sua compatibilização à rede viária modelada.

3.2 Elaboração dos arquivos de entrada

A partir do levantamento e tratamento dos dados, é necessária a adaptação dos dados para o formato requerido para a execução da simulação no MATSim. Essa seção apresenta os métodos de elaboração dos arquivos de diários de viagens, rede viária, contagens de tráfego e oferta do transporte coletivo.

Diários de viagens No caso da utilização de dados advindos de pesquisa O/D, a modelagem da demanda e consequente elaboração do arquivo *plans.xml* inclui as etapas listadas a seguir.

1. **Definição das coordenadas de início e final da viagem.** Nas pesquisas de origem e destino, não é usual ter conhecimento sobre o endereço exato de origem e destino das viagens, com exceção do domicílio (onde é realizada a entrevista). Portanto, nos modelos de alocação de tráfego tradicionais as viagens costumam ser alocadas para o centróide das ZT, de onde são distribuídas para a malha a partir de conectores (vide Seção 2.2.3). No MATSim, as origens e destinos das viagens ocorrem nas edificações e não existem as entidades de centróides e de conectores das ZT. Portanto, para a origem e destino dessas viagens, é necessária a indicação de coordenadas distribuídas nas edificações da ZT. Caso existam informações sobre parâmetros relacionados à atração ou produção de demanda, é possível realizar a distribuição das viagens de forma ponderada de acordo com esses valores.

¹⁴Segundo a GTFS.org, disponível em <<https://gtfs.org/schedule/reference/>>, GTFS, do termo inglês *General Transit Feed Specification* e traduzido como Especificação Geral de Feed de Trânsito, representa uma série de arquivos que descrevem a operação do transporte coletivo.

2. **Definição do horário de saída das viagens.** Conforme exposto na Seção 3.1.1, a expansão das entrevistas gera horários de saída e de chegada das viagens exatamente iguais para agentes gerados a partir da mesma entrevista, e pode congestionar a malha viária por acessos simultâneos em vias locais. Assim, é necessária a randomização do horário de saída em um intervalo de tempo pré-definido.

A realização dessas etapas, em conjunto com a reestruturação dos dados para elaboração do arquivo de entrada pode ser realizada por meio de *scripts*.

Independente dos dados utilizados como demanda, deve haver a definição dos modos de transporte e suas características. A indicação do modo de transporte no arquivo *plans.xml* deve ser compatível com a caracterização do modo no arquivo de configuração (*config.xml*), e no arquivo de definição dos modos de transporte (*mode-vehicles.xml*).

Esse último é necessário quando se deseja considerar diferentes tipos de veículos alocados na malha viária, como motocicletas e veículos de passeio. Nele, são indicadas a velocidade máxima do tipo de veículo, seu comprimento e largura, o tempo do usuário ao entrar ou sair do veículo e seu fator de equivalência em relação ao veículo de passeio.

Rede viária A modelagem da rede viária se resume à reestruturação dos dados georreferenciados da malha viária para um arquivo em formato *.xml* requerido pelo MATSim. Para a rede viária do OpenStreetMap (OSM), a conversão da base para a estrutura de dados em *.xml* é automatizada pela utilização do JOSM.

Caso estejam disponíveis dados georreferenciados oficiais da rede viária com as informações necessárias para composição do modelo, é possível utilizá-los com anterior reestruturação dos dados por meio de elaboração de *script*. Essa reestruturação deve abranger os itens mínimos indicados na Seção 2.4.2.

Contagens de tráfego A modelagem das contagens de tráfego se resume à reestruturação dos dados de localização e dos resultados das contagens de veículos motorizados por hora em arquivo em formato *.xml*. Em relação a sua localização, cada posto de contagem deve estar relacionado a um *link* da malha viária. O MATSim utiliza *links* sobrepostos, separados para cada sentido, dificultando a automatização da atribuição dos *links* aos postos de contagem. Para garantir a correta modelagem, recomenda-se a atribuição manual dos *links* indicados, ou baseada em dados auxiliares que possam estar disponíveis.

Oferta do transporte coletivo Conforme apresentado na Seção 3.1.4, os arquivos em formato GTFS podem ser convertidos por meio de módulo específico do MATSim, denominado “GTFS2MATSim”¹⁵.

Caso os dados sejam disponibilizados em outros formatos, é necessária a reestruturação dos dados por meio de *script* específico para elaboração do arquivo *transitSchedule.xml*. Esse *script* deve realizar os procedimentos listados a seguir.

1. Indicação das coordenadas dos pontos de parada existentes no sistema, e respectivo *link* da rede viária em que se encontra.
2. Para cada linha do sistema de transporte coletivo, deve ser indicada a lista de pontos de parada na ordem do itinerário.
3. Indicação da lista de *links* em que os ônibus trafegam ao longo do itinerário de cada linha por ordem de passagem. A compatibilização do itinerário com os *links* descritos no modelo da rede viária podem ser obtidos por meio de algoritmos de roteamento considerando os dados georreferenciados da rede viária e do itinerário completo.
4. Cada ponto de parada deve ter um horário previsto de chegada e de saída das viagens da linha, o intervalo estimado entre os pontos pode ser obtido por meio dos resultados do algoritmo de roteamento (caso o custo associado seja o tempo de deslocamento no *link* em velocidade de fluxo livre), ou a divisão do tempo de percurso da linha e a quantidade de pontos de parada existentes na linha. Esses tempos são ajustados ao longo da simulação de acordo com a velocidade de fluxo nos *link* no momento da viagem.
5. Listagem de todas as viagens programadas da linha em um dia, indicando o horário de saída do ponto inicial e o veículo empregado para operação da linha. Caso não seja disponibilizado o quadro de horários das linhas, é necessário estimar os horários de saída de acordo com a quantidade de viagens diárias e a frequência de viagens durante o período de pico e durante o período de vale. O tipo de veículo empregado deve ser conhecido, mas caso o objetivo da simulação não seja otimização do transporte coletivo, não é necessária a indicação de veículos reais da frota do TC e podem ser utilizados veículos sintéticos para realização das viagens programadas.

Os veículos indicados nas viagens de cada linha devem ser descritos em arquivo separado, denominado *transitVehicles.xml*. Devem ser descritos todos os tipos de veículos existentes, conside-

¹⁵Mais informações em <<https://github.com/matsim-org/GTFS2MATSim>>.

rando sua capacidade em pé e sentada e seu comprimento. Cada veículo indicado nas viagens deve ser listado, com seu código de identificação e seu tipo.

3.3 Execução da simulação

Após a modelagem da demanda e da oferta na área de estudo, é necessário realizar a verificação de erros nos arquivos de entrada do modelo por meio de uma simulação teste. Para isso, é realizada uma simulação com apenas uma iteração e com a configuração inicial, a fim de analisar os erros de forma rápida.

O MATSim é configurado por meio do arquivo *config.xml*. É nesse arquivo que são indicados os arquivos de entrada do modelo, assim como as características da simulação. Esse arquivo pode conter uma grande quantidade de módulos, que adicionam funcionalidades ao simulador. Todos os módulos e seus respectivos parâmetros são listados no arquivo *fullConfig.xml*, que pode ser obtido por meio de linha de código ou no GUI¹⁶.

O manual do MATSim (HORNI; NAGEL; AXHAUSEN, 2016) recomenda que, no início da modelagem, seja configurado o menor número possível de opções no arquivo de configuração. Portanto, para configuração inicial do modelo, é necessário:

1. incluir apenas os módulos obrigatórios (*controler*, *counts*, *global*, *network*, *PlanCalcScore*, *plans*, *PlansCalcRoute*, *strategy* e *QSim*);
2. definir os valores dos parâmetros de configuração conforme padrão do MATSim;
3. indicar os arquivos de entrada e sistema de coordenadas.

Para configuração do transporte coletivo, os módulos obrigatórios são: *transit* e *transitRouter*.

Após a execução da simulação, os erros são identificados no arquivo *log*, onde são apresentadas as informações relativas ao processamento da simulação, incluindo quaisquer erros ou advertências sobre o modelo. Apesar de não haver uma lista de todos os erros possíveis, a comunidade de usuários do MATSim utiliza a seção *Issues* nos fóruns do GitHub¹⁷ para troca de experiências e colaboração na solução dos erros encontrados.

Para verificação dos erros na modelagem do transporte coletivo, pode ser utilizada a ferramenta *ValidateTransitSchedule* no MATSim GUI.

¹⁶GUI significa *Graphical User Interface* (Interface Gráfica do Usuário, em português).

¹⁷Localizados em <<https://github.com/matsim-org>>

Após a execução sem erros de uma simulação teste com apenas os módulos obrigatórios, o arquivo deve ser complementado de acordo com os objetivos do estudo em questão e procede-se à execução da simulação com a quantidade de iterações inicial. A quantidade final de iterações é definida durante o processo de calibração.

3.4 Calibração do modelo

Essa seção apresenta o método de calibração dos parâmetros do modelo relativos à função de pontuação, às estratégias de replanejamento, ao Cadyts, à proporção de população e de capacidade viária e os ajustes nos arquivos de entrada.

Ajuste do peso dos parâmetros da função de pontuação A função de pontuação possui diversos parâmetros, definidos no módulo *planCalcScore*, e apresentados na Seção 2.4.4. Para início da simulação, podem ser utilizados os valores padrão do MATSim (vide Seção 2.4.6). Alternativamente, Horni, Nagel e Axhausen (2016) indicam que é possível utilizar modelos estatísticos pré-existentes para definição dos pesos dos parâmetros.

Ainda, a depender dos dados disponíveis para elaboração do modelo, é possível eliminar o peso de determinados parâmetros, de forma que não influenciem na tomada de decisões. Por exemplo, se não foram informados dados sobre o horário de abertura dos estabelecimentos, e os valores indicados na simulação não representam a realidade, os parâmetros de saída com antecedência e de chegada com atraso podem ser zerados, para que não influenciem na simulação.

Com os resultados de uma primeira simulação teste com os valores padrão para os pesos dos parâmetros, devem ser realizados ajustes nos pesos dos parâmetros relacionados aos problemas encontrados na análise dos resultados. Na divisão modal, por exemplo, caso um modo de transporte apresente mais viagens do que o originalmente modelado, é necessário ajustar o peso das constantes específicas de cada modo de transporte (β_m) para que representem corretamente a divisão modal real na área de estudo.

Ajuste dos pesos das estratégias de replanejamento As estratégias de replanejamento dizem respeito às mudanças permitidas aos agentes a partir dos planos indicados no arquivo de entrada. Essas mudanças podem estar relacionadas com a escolha do plano a ser executado (seleção), a

retirada de um plano de sua memória (retirada) ou a criação de planos novos (inovação), conforme apresentado na Seção 2.4.5.

Para determinação da estratégia, recomenda-se a utilização de pelo menos uma estratégia de seleção de planos, uma estratégia de retirada de planos, e as estratégias de inovação de interesse. Por exemplo, caso se deseje investigar a migração modal, deve ser utilizada a estratégia *ChangeMode*. É atribuído um peso a cada uma das estratégias utilizadas, que determinam a probabilidade das mudanças indicadas ocorrerem. Assim, os pesos variam entre zero e um e são normalizados, caso a soma não totalize um.

Dado que as estratégias de inovação realizam as alterações nos planos de forma aleatória, é recomendada a atribuição de pesos menores a essas estratégias do que às estratégias de seleção. Caso contrário, a evolução da pontuação média da população pode apresentar instabilidade.

Ajuste do Cadyts Em relação à configuração do Cadyts, seu manual (FLÖTTERÖD, 2010) apresenta os parâmetros de configuração, assim como uma breve descrição de como defini-los.

O Cadyts é implantado por meio de sua inclusão no arquivo de configuração, e no código-fonte do MATSim. Sua adição no código-fonte inclui a definição do peso do parâmetro incluído pelo Cadyts na função utilidade. Quanto maior for o peso deste parâmetro, maior será a influência no Cadyts na simulação. No entanto, pesos muito altos podem contribuir para a instabilidade da pontuação média dos agentes.

Ajuste da proporção de população e capacidade viária Conforme apresentado na Seção 2.4.3, o MATSim prevê a possibilidade de redução da população, com a adaptação de outras características do modelo para garantir a compatibilidade entre demanda e oferta.

O ajuste da proporção da população deve ser definido com base nas limitações computacionais imputadas ao experimento. Ou seja, dependendo do tamanho da população a ser modelada, da quantidade de iterações necessárias e do tempo necessário para execução da simulação nas ferramentas computacionais disponíveis, deve ser ajustada a proporção, buscando mantê-la o mais alta possível, para melhor representação da realidade.

A capacidade viária deve ser então ajustada a essa redução na população. A proporção das capacidades de armazenamento e de fluxo das vias pode ser considerada igual à proporção da população no modelo, a princípio. Contudo, espera-se que a relação entre esses valores não seja linear e, portanto, devem ser calibrados para obtenção dos resultados mais verossímeis.

Para essa calibração, devem ser realizadas simulações teste com diferentes proporções de capacidade viária de fluxo, e de armazenamento. Essas diferentes proporções podem ser definidas com base nos métodos apresentados na Seção 2.5. São selecionados os fatores de capacidade viária do modelo que reproduzirem com maior fidelidade o sistema analisado.

Ajustes nos arquivos de entrada Ao longo da calibração, pode ser necessário realizar alterações nos arquivos da rede viária e da demanda. Por exemplo, a inclusão de novos modos de transporte requer a alteração do arquivo de demanda (*plans.xml*), assim como do arquivo da rede viária (*network.xml*), a fim de garantir que o modo pode ser simulado na rede viária. Além disso, podem ser avaliadas diferentes formas de pulverização da demanda em suas respectivas ZT de origem e destino ou de seus horários de saída, o que requer ajustes no arquivo *plans.xml*.

3.5 Validação do modelo de alocação do transporte individual

No processo de validação são avaliados indicadores de comparação entre as viagens reais e simuladas. Conforme indicam Buisson et al. (2015), diversos autores recomendam a comparação dos dados reais e modelados dos volumes de tráfego em pontos estratégicos da área de estudo, e dos tempos de viagem. A comparação deve ser realizada por meio de testes estatísticos de bondade de ajuste dos dados, conforme apresentado na Seção 2.2.6.

Além disso, no contexto do MATSim, é necessária a verificação da estabilidade das pontuações e da existência de agentes presos (que não finalizam suas viagens) na execução da simulação.

Volumes de tráfego Para calibração e validação, é necessária a utilização de uma parte dos postos de contagem para a construção do modelo e outra parte para o processo de calibração, para chegar na validação. No caso do MATSim, todos os postos de contagem e seus volumes de tráfego horário são indicados no arquivo *counts.xml*.

No MATSim, o grupo de postos para a construção do modelo, baseado no processo de calibração, é exclusivamente utilizado pelo Cadyts. A indicação dos postos a serem considerados nesse procedimento deve ser realizada no parâmetro *calibratedLinks* do módulo de configuração do Cadyts no arquivo *config.xml*.

Para validação, o arquivo resultante da simulação *countscompare.txt* deve ser filtrado para conter apenas os postos para validação. A partir desses resultados, são calculadas as medidas estatísticas de bondade de ajuste.

Para que cada um dos grupos possua representantes de todos os postos de contagem, esses devem ser agrupados de acordo com sua categoria (postos de dia inteiro, postos de hora-pico, postos *cordon lines* e postos *screen lines*), e ordenados de acordo com seus volumes diários ou volumes na hora-pico. Os grupos devem ser compostos de forma que existam postos de todos os tipos e de todos os intervalos de volume de tráfego representativos dos postos.

Após a definição dos grupos, é realizado o cálculo das medidas estatísticas de bondade de ajuste dos volumes de tráfego em diferentes períodos de tempo. Podem ser realizadas comparação de volumes diários, volumes horários, incluindo os volumes na hora-pico.

Tempos de viagem Para verificação da correlação entre os tempos de viagem reais e simulados, podem ser utilizados os resultados de pesquisa em campo para aferir os tempos de viagem em diversos trechos ao longo da área de estudo. O método aplicado para a estruturação de dados necessária ao cálculo do coeficiente de determinação (r^2) é o seguinte:

1. identificação dos trechos da pesquisa;
2. levantamento do tempo de viagem nos trechos;
3. georreferenciamento dos trechos pesquisados;
4. identificação dos nós necessários para traçado da rota;
5. traçado da rota na malha simulada e identificação dos links que a compõem com auxílio do Via;
6. identificação do tempo de viagem simulado nos horários da pesquisa;
7. cálculo do r^2 do tempo de viagem para os trechos nos dois sentidos (quando houver).

Estabilização das pontuações Em cada iteração, a demanda anterior é otimizada individualmente por meio da função de pontuação. O processo iterativo deve ser repetido até a estabilização da pontuação média dos agentes.

Para determinação da quantidade de iterações necessárias, é aconselhada a realização de testes com diferentes quantidades de iterações. O resultado dos testes é avaliado a partir do gráfico de evolução das pontuações médias dos planos executados da população. Caso a pontuação dos

planos executados presente estabilidade ao final da simulação, a quantidade de iterações realizada é considerada adequada.

Além disso, outros parâmetros influenciam na rapidez com que a convergência é atingida. O parâmetro *learning rate*, ou taxa de aprendizado, varia de zero a um, e é dado por $P_n = (1 - L_r) \times P_a + L_r \times P_{mobsim}$, onde P_n é a nova pontuação, P_a é a pontuação da iteração anterior, P_{mobsim} é a pontuação calculada na execução da simulação e L_r é a taxa de aprendizado. Valores próximos de zero emulam a média das pontuações, mas atrasam a convergência inicial.

O parâmetro *fractionOfIterationsToDisableInnovation* também possui influência na convergência da pontuação, já que determina em qual porcentagem das iterações, as estratégias de inovação são desativadas, e os agentes escolhem apenas entre os planos que estão em sua memória nessa iteração. Nesse sentido, o parâmetro *maxAgentPlanMemorySize* também deve ser ajustado, de forma que permita a escolha do melhor plano, sem gasto desnecessário de memória computacional. Horni, Nagel e Axhausen (2016) recomendam valores de 0,8 para o *fractionOfIterationsToDisableInnovation* e cinco (5) para o *maxAgentPlanMemorySize*.

Quantidade de agentes presos A verificação da quantidade de agentes presos é necessária para identificar trechos em que o fluxo não escoar em tempo adequado ao longo da simulação, que podem apontar para problemas na modelagem da rede e/ou da distribuição de pontos de inserção da demanda

A identificação da quantidade de agentes presos é realizada por meio do arquivo *legHistogram.txt*, onde é indicada a quantidade de veículos saindo de suas origens, chegando em seus destinos, em rota e presos na rede a cada cinco minutos. Deve ser realizada a soma da quantidade de agentes presos ao longo da simulação, e a soma da quantidade de agentes saindo de suas origens, para então calcular a porcentagem de agentes presos em relação à quantidade de agentes que saíram de suas origens.

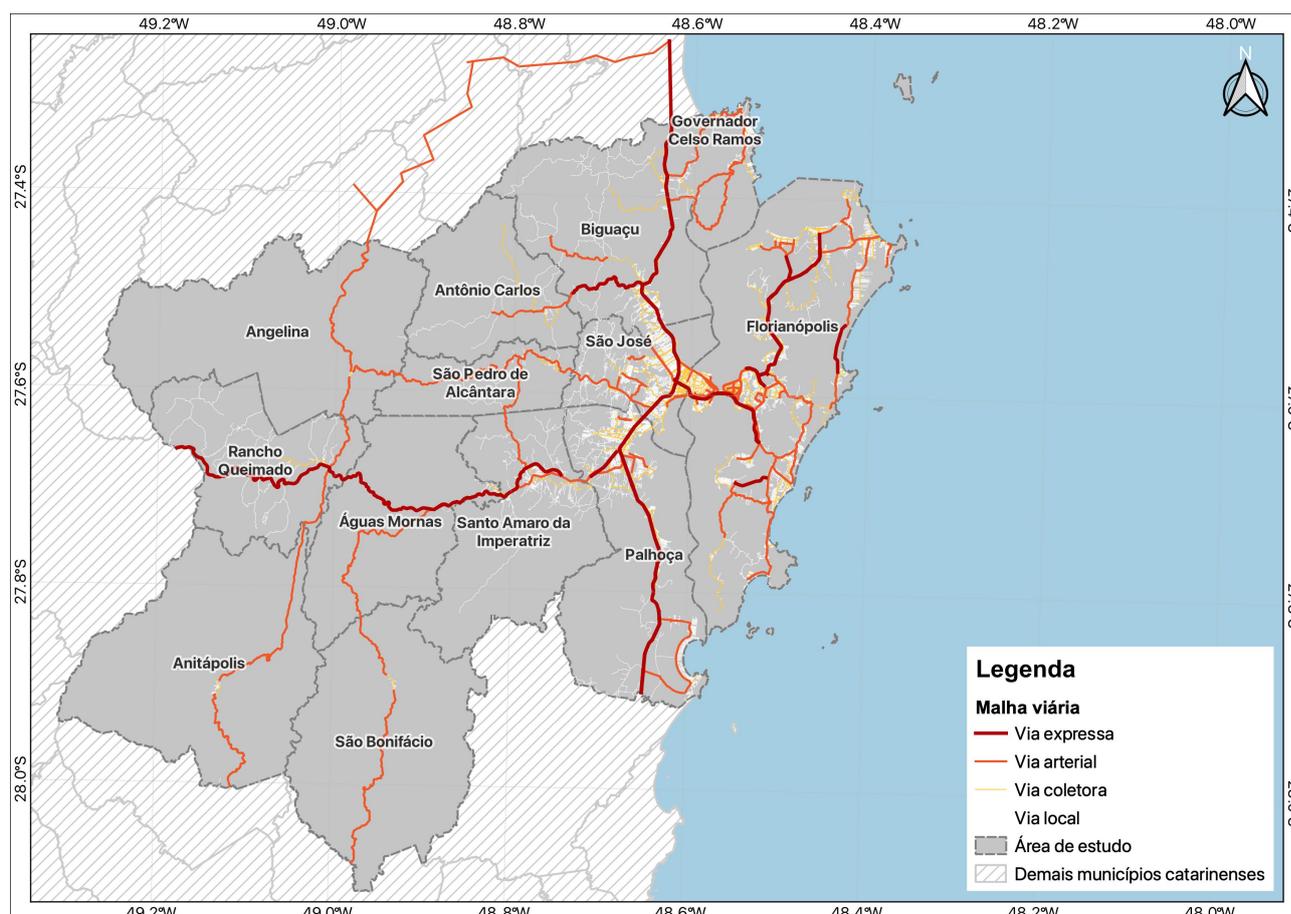
4 Área de estudo e dados disponíveis

Esse estudo tem como cenário espacial a região da Grande Florianópolis, composta pelos municípios de Águas Mornas, Angelina, Anitápolis, Antônio Carlos, Biguaçu, Florianópolis, Governador Celso Ramos, Palhoça, Rancho Queimado, Santo Amaro da Imperatriz, São Bonifácio, São José e São Pedro de Alcântara, e como cenário temporal o ano de 2014.

Esta área de estudo foi adotada em acordo com os dados disponíveis para a construção do modelo de simulação, os quais advêm, majoritariamente do repositório do Plano de Mobilidade Urbana Sustentável da Grande Florianópolis (PLAMUS)¹⁸.

A Figura 21 representa os municípios analisados e sua rede viária (PLAMUS, 2014) de acordo com sua classificação funcional.

Figura 21: Área de estudo. Fonte: IBGE (2010) e PLAMUS (2014).



¹⁸Disponível em: <https://observatoriodamobildadeurbana.ufsc.br/publicacao/projetos-de-pesquisa/>. Acesso em: 19 fev. 2024.

Segundo os estudos realizados no PLAMUS, a área de estudo pode ser entendida como duas partes distintas. A primeira é o município de Florianópolis, que concentra a maior parte dos serviços e locais de trabalho, e a segunda são os demais municípios da área, onde a maior parte da ocupação é de uso residencial. Essas assumem um papel periférico perante à capital, numa dinâmica “centro-periferia” que evidencia o desenvolvimento desequilibrado e a falta de coesão espacial entre os municípios. A Figura 22 e a Figura 23 ilustram a densidade populacional e a renda média mensal domiciliar nas zonas de tráfego na área de estudo, que refletem, em parte, a dinâmica citada.

Figura 22: Densidade populacional das zonas de tráfego da área de estudo. Fonte: PLAMUS (2014).

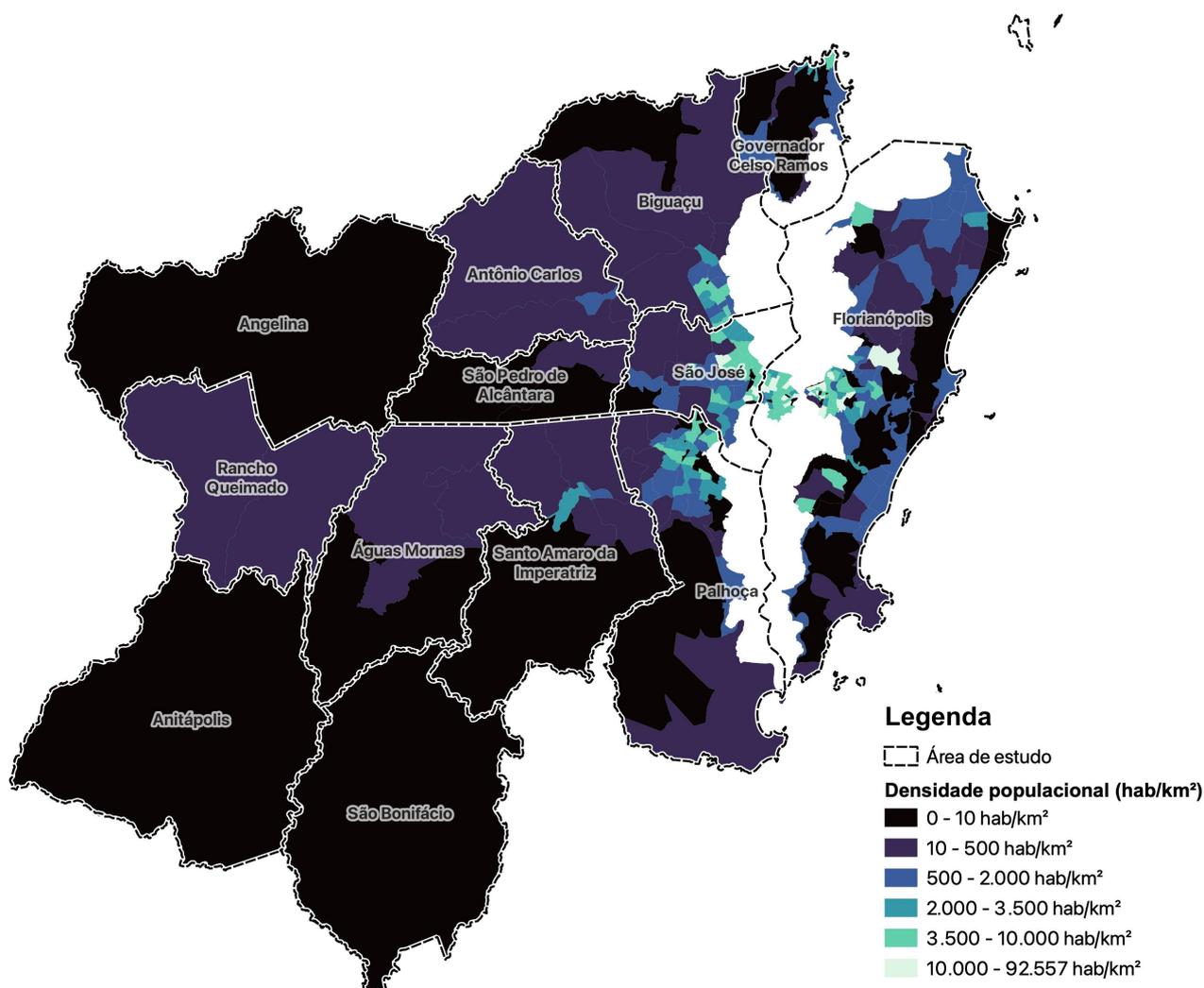
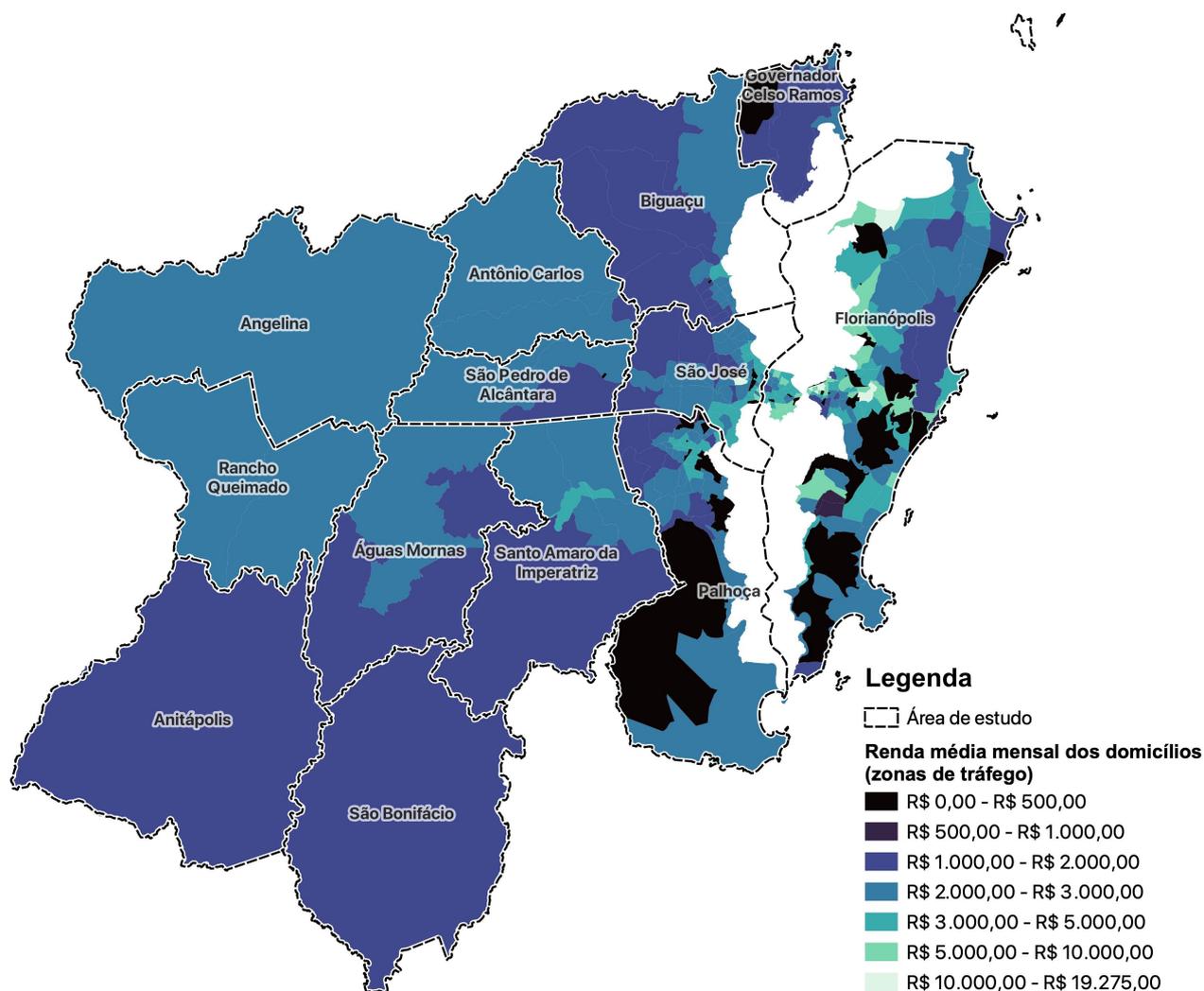


Figura 23: Renda média mensal domiciliar das zonas de tráfego da área de estudo. Fonte: PLAMUS (2014).



O diagnóstico da mobilidade da Grande Florianópolis pelo PLAMUS, em termos gerais, aponta a grande dependência econômica dos demais municípios em relação à capital, especialmente quanto à oferta de emprego, que ocasiona congestionamento de vias vinculado aos deslocamentos pendulares entre habitação e trabalho. A ocupação urbana se expande rumo a áreas ambientalmente frágeis, como encostas, áreas alagáveis e dunas, com um padrão periférico de crescimento, urbanização dispersa, baixas densidades e ausência de usos mistos, em todos os municípios da área de estudo. Em relação à malha urbana, o PLAMUS identificou um descompasso entre a forma e a função de vias, como rodovias que assumem a função de avenidas urbanas e ruas estreitas que desempenham uma função viária estrutural.

A Figura 24 e a Figura 25 apresentam os mapas elaborados pelo PLAMUS para ilustração das principais linhas de desejo do pico da manhã e da tarde, respectivamente, identificadas por meio de sua Pesquisa O/D. São apresentados apenas os pares de viagens com volume superior a 1.200 viagens, que representam 33% das viagens realizadas em cada um dos períodos.

Figura 24: Principais linhas de desejo no período da manhã identificadas pelo PLAMUS. Fonte: PLAMUS (2014).

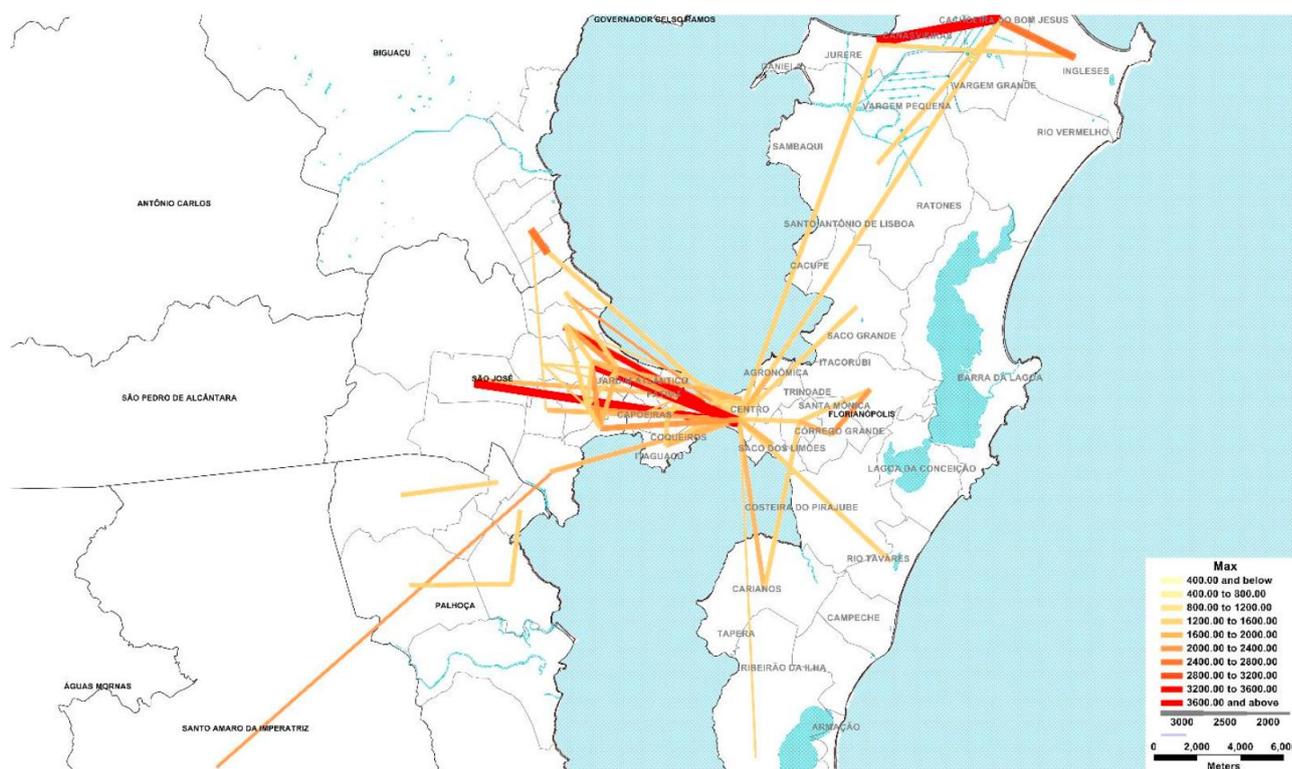


Figura 25: Principais linhas de desejo no período da tarde identificadas pelo PLAMUS. Fonte: PLAMUS (2014).



O diagnóstico do PLAMUS indica que no período da manhã, a maioria das linhas de desejo se concentra na região central de Florianópolis, sendo os pares O/D de maior volume localizados entre o Município de São José e o centro de Florianópolis. Também destacam as linhas de desejo entre Canasvieiras e Ingleses. No período da tarde, foi indicado que há maior distribuição das viagens, com diminuição na concentração de viagens em algumas linhas de desejo. Mesmo assim, as principais linhas de desejo são as mesmas observadas para o pico da manhã.

A área de estudo apresenta uma taxa de viagens por pessoa de 1,83, e uma taxa de viagens motorizadas por pessoa de 1,39. A Tabela 2 apresenta a divisão modal e quantidade de viagens realizadas na área de estudo, conforme dados do PLAMUS (2014).

Tabela 2: Divisão modal na área de estudo. Fonte: PLAMUS (2014).

Modo de transporte	Quantidade de viagens	Participação modal (%)
Individual	854.743	47,7
Coletivo	445.132	24,9
A Pé	370.439	20,7
Bicicleta	66.504	3,7
Outros	45.876	2,6
Táxi	8.018	0,4
Total	1.790.711	100,0

Tabela 3: Duração média das viagens por modo de transporte. Fonte: PLAMUS (2014).

Modo de transporte	Duração média (minutos)
A Pé	15,1
Outros	23,4
Individual	31,2
Coletivo	57,6
Táxi	27,4
Total	33,9

Considerando a área de estudo apresentada, a Tabela 4 apresenta suas informações disponíveis para elaboração do modelo de simulação, suas descrições e fontes disponíveis.

Tabela 4: Descrição dos dados disponíveis para elaboração do modelo de simulação na área de estudo

Dados	Descrição	Fontes disponíveis
Rede viária	<ul style="list-style-type: none"> • Links e nós da rede viária georreferenciada • Velocidade em fluxo livre • Capacidade viária • Quantidade de faixas de tráfego 	OSM e PLAMUS
Diário de viagens da população	<ul style="list-style-type: none"> • Origem e destino das viagens • Duração das viagens (tempo de espera, a pé e total) • Atividades realizadas • Perfil socioeconômico dos viajantes • Fator de expansão da amostra pesquisada 	PLAMUS
Edificações	<ul style="list-style-type: none"> • Localização dos lotes, com respectiva indicação de existência de edificação 	Prefeitura de Florianópolis
Contagens de tráfego	<ul style="list-style-type: none"> • Quantidade de veículos motorizados em postos de contagem estrategicamente localizados 	PLAMUS
Pesquisa de velocidade	<ul style="list-style-type: none"> • Duração e velocidade de viagens em trechos estrategicamente localizados na área de estudo 	PLAMUS
Rotas das linhas do transporte coletivo	<ul style="list-style-type: none"> • Itinerários georreferenciados das linhas • Pontos de parada da linha 	PLAMUS
Características operacionais das linhas do transporte coletivo	<ul style="list-style-type: none"> • Tarifa • Tipo de veículo utilizado na linha (comum, executivo ou articulado) 	PLAMUS
Dados dos pontos de embarque e desembarque do transporte coletivo	<ul style="list-style-type: none"> • Localização georreferenciada dos pontos E/D • Existência de baía para E/D 	Prefeitura de Florianópolis e OSM
Mapas de carregamento viário	<ul style="list-style-type: none"> • Mapas em PDF do carregamento viário simulado no TransCAD 	PLAMUS

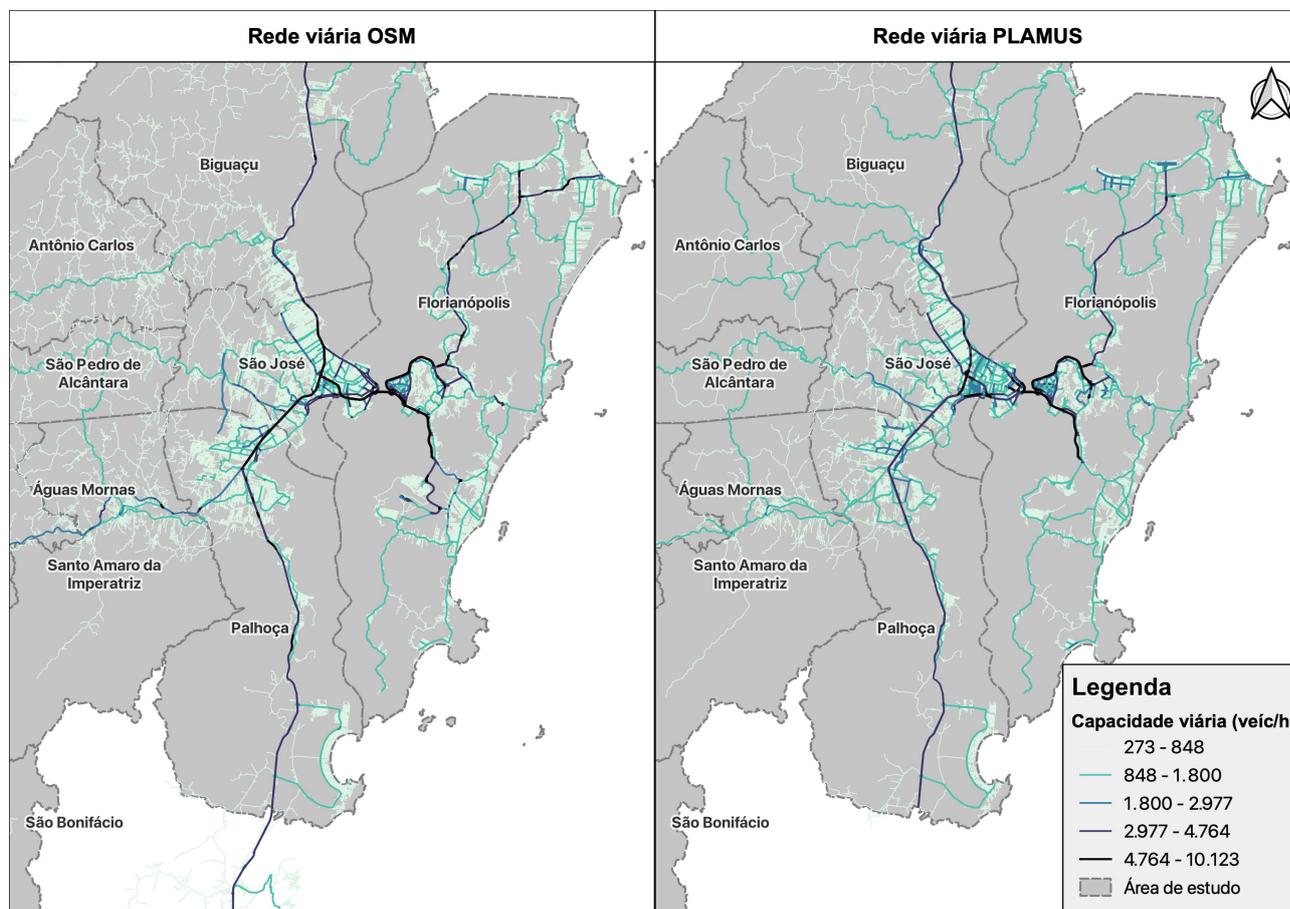
4.1 Rede viária

Para modelagem da infraestrutura viária, tanto o OSM quanto o PLAMUS disponibilizam arquivo de dados georreferenciados que apresenta informações sobre a geometria, sentido, velocidade e quantidade de faixas. O arquivo do PLAMUS dispõe ainda da capacidade viária por sentido, enquanto o plugin do MATSim do JOSM imputa capacidades de acordo com a classificação funcional das vias.

As fontes apresentam informações com diferentes datas de referência. O PLAMUS fornece dados de 2014, quando os dados de demanda foram coletados. Os dados do OSM são atualizados frequentemente, mas não é possível ter um controle sobre o ano de atualização de cada via já que os dados são atualizados por diversos colaboradores. Nesse sentido, os dados do OSM consideram a abertura da Ponte Hercílio Luz para tráfego de veículos (considerada desativada na rede do PLAMUS), a duplicação de alguns trechos de rodovia e todas as demais alterações viárias realizadas após 2014.

A rede do OSM possui detalhamento homogêneo ao longo da área de estudo, enquanto a rede do PLAMUS passa a considerar apenas as vias principais nas áreas mais externas à área de estudo. A Figura 26 apresenta uma comparação visual entre as redes viárias disponibilizadas por ambas fontes.

Figura 26: Comparação visual entre a rede viária PLAMUS e OSM



4.2 Diário de viagens da população

A pesquisa domiciliar foi realizada pela equipe do PLAMUS (2014) em todos os municípios da Grande Florianópolis, onde foram entrevistados todos os moradores maiores de sete anos. A pesquisa foi realizada no período de 10/04/2014 a 12/07/2014 em 5.464 domicílios da Grande Florianópolis. O questionário era composto de três blocos de informações:

1. dados do domicílio:
 - endereço;
 - número de famílias;
 - tipo de domicílio;
 - telefone de contato.
2. dados socioeconômicos:
 - nome do morador;

- idade;
- sexo;
- grau de instrução;
- renda;
- número de veículos, bicicletas e/ou motocicletas.

3. dados sobre deslocamentos (viagens):

- origem da viagem;
- destino da viagem;
- motivo da viagem;
- meio de transporte utilizado para a viagem;
- tempo a pé;
- tempo de espera;
- tempo da viagem;
- custo da viagem;
- horário de saída;
- horário de chegada.

Além disso, sobre os domicílios e moradores, são apresentados dados sobre as coordenadas das residências, e os fatores de expansão considerados para os moradores entrevistados. Os fatores de expansão considerados pelo PLAMUS são listados a seguir.

1. Fator de expansão de renda: calculado por meio do coeficiente entre a quantidade de domicílios de determinada faixa de renda existentes na zona de tráfego (ZT) pela quantidade de domicílios de determinada faixa de renda entrevistados na ZT.
2. Fator de expansão de tamanho de domicílio: calculado por meio do coeficiente entre a quantidade de domicílios de determinado número de moradores existentes na ZT pela quantidade de domicílios de determinado número de moradores entrevistados na ZT.
3. Fator de expansão de ajuste da população: calculado por meio do coeficiente entre a quantidade de moradores existentes na ZT pela quantidade de moradores entrevistados na ZT.
4. Fator de expansão de ajuste das viagens com as contagens de tráfego: com base nas contagens de tráfego em *Screen Lines*, é calculado o coeficiente entre a quantidade de viagens em determinado

modo de transporte e período do dia registradas na contagem de tráfego, e nas entrevistas. Como o MATSim utiliza os agentes como unidade básica, não é possível aplicar esse fator de expansão nas simulações no MATSim.

4.3 Edificações

Como colocado na Seção 2.4.2, o MATSim espacializa a demanda em nível de edificações, e não em nível de zonas de tráfego. Para suprir esta informação para a construção do modelo são necessários os dados das edificações existentes na área de estudo.

No município de Florianópolis, é disponibilizado o arquivo georreferenciado dos polígonos dos lotes cadastrados na Planta Genérica de Valores (PGV) de 2023 por meio do geoportal da prefeitura¹⁹. Nesse arquivo, quando os valores referentes às unidades prediais são iguais a zero, o lote não possui edificação cadastrada. E é, portanto, considerado um terreno desocupado.

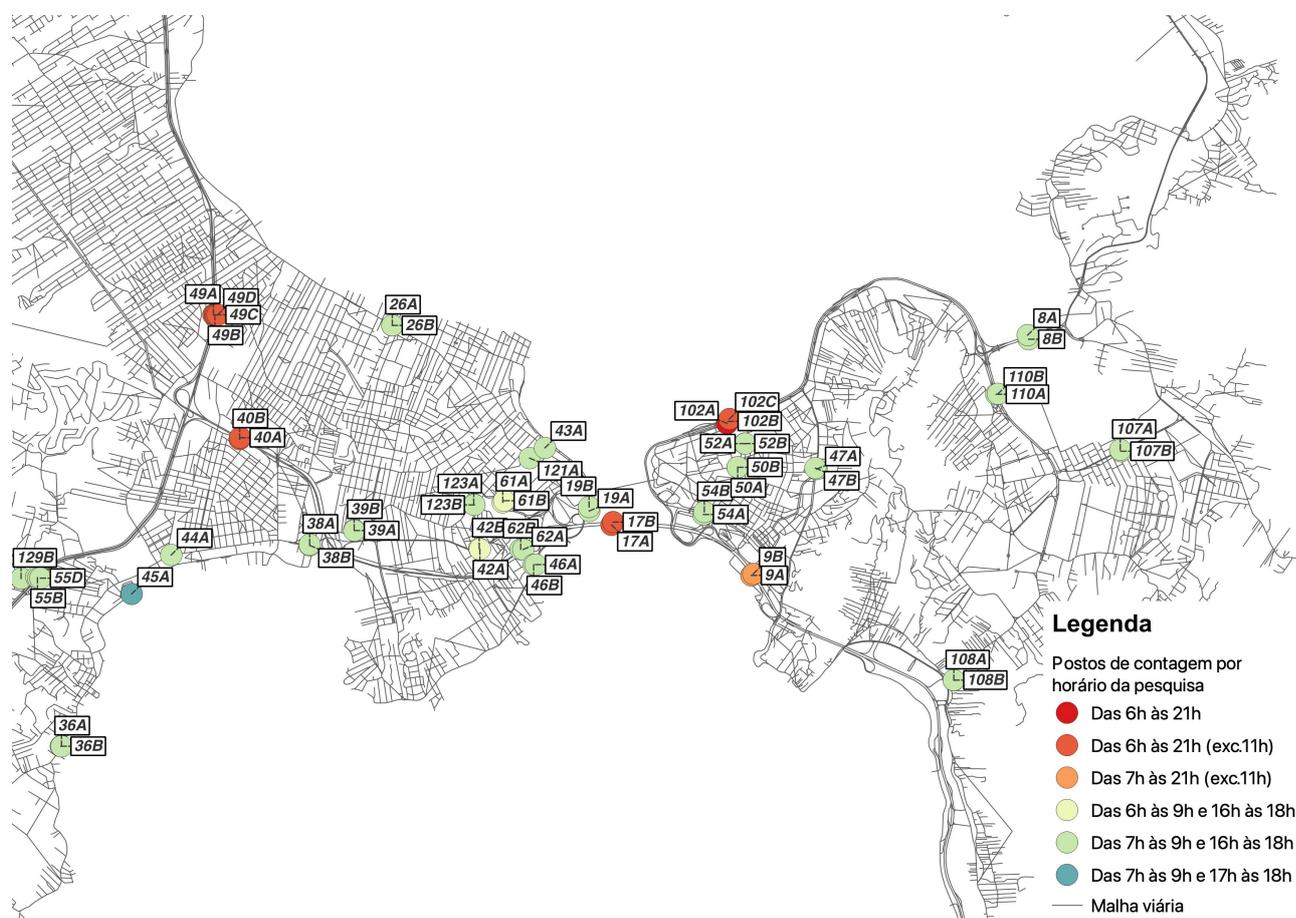
Os demais municípios não disponibilizam essa informação.

4.4 Contagens de tráfego

As contagens de tráfego do PLAMUS foram realizadas em 99 postos de contagem, considerando que cada posto realiza a contagem de apenas um sentido de tráfego em diferentes períodos do dia. A Figura 27 apresenta a localização dos postos de acordo com os horários do dia em que são realizadas as contagens de tráfego.

¹⁹Disponível em: <https://geoportal.pmf.sc.gov.br/downloads>. Acesso em: 19 fev. 2024.

Figura 27: Postos de contagem considerados pelo PLAMUS

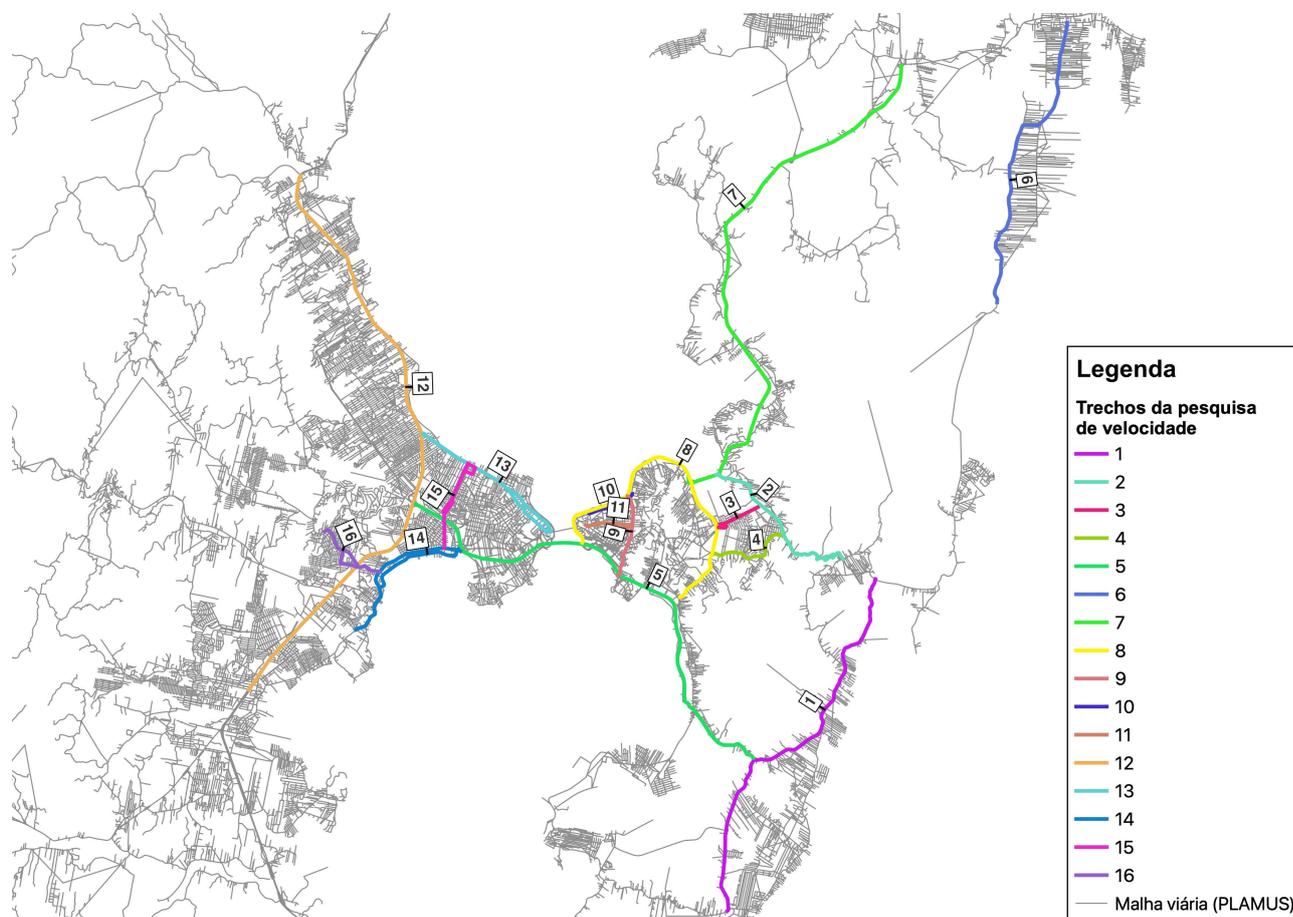


(b) Postos de contagem - área central

4.5 Pesquisa de velocidade

A pesquisa de velocidade do PLAMUS foi realizada nos períodos de pico da manhã e da tarde, em 16 trechos ao longo da área de estudo, dos quais 15 possuem duplo sentido. A Figura 28 apresenta os trechos pesquisados.

Figura 28: Trechos da pesquisa de velocidade do PLAMUS



4.6 Oferta do transporte coletivo

Os dados sobre a oferta do transporte coletivo são disponibilizados exclusivamente em formato *shapefile*, contendo informações sobre os itinerários existentes no período de realização do PLAMUS. A estrutura e a descrição dos dados georreferenciados das linhas são apresentadas na Tabela 5.

Tabela 5: Estrutura do arquivo georreferenciado das rotas do transporte coletivo

Variável	Descrição
ID	Identificador numérico da rota
NOME_ROTAS	Identificador alfanumérico da rota
FREQHPM	Frequência de viagens por hora durante a hora-pico da manhã
FREQHPT	Frequência de viagens por hora durante a hora-pico da tarde
INTHPM	Intervalo de viagens durante a hora-pico da manhã
INTHPT	Intervalo de viagens durante a hora-pico da tarde
TARIFA	Preço da passagem
CATEGORIA	Categoria da linha: municipal, intermunicipal ou rodoviária
OPERAÇÃO	Tipo de operação: expressa, direta, semidireta ou paradora
SERVIÇO	Tipo de serviço: convencional ou executivo
TIPOVEICUL	Tipo de veículo empregado na rota: comum, articulado ou executivo
LENGTH	Comprimento da linha, em quilômetros
TEMPO_COLE	Tempo de ciclo da rota, em minutos

Apesar de indicar a frequência de viagens durante as horas-pico, não são disponibilizadas a frequência durante o período fora das horas-pico ou o quadro de horários das linhas existentes no período do estudo.

O PLAMUS não disponibiliza, ainda, o *shapefile* dos pontos de ônibus existentes. No município de Florianópolis, a localização dos pontos de ônibus existentes em 2018 podem ser obtidos no site do geoportal da prefeitura²⁰. Nos demais municípios, os dados não são disponibilizados de forma oficial. Como alternativa, o OSM apresenta essa informação para todos os municípios da área de estudo, apesar de sua incompatibilidade temporal com o cenário modelado, e da dificuldade em validar seus dados.

Enquanto a localização dos pontos de ônibus é passível de conhecimento, a indicação dos pontos de ônibus em que são realizados os embarques e os desembarques de cada linha no ano de 2014 não são disponibilizadas de forma aberta.

²⁰Disponível em: <https://geoportal.pmf.sc.gov.br/downloads/camadas-em-sig-do-mapa>. Acesso em: 09 jan. 2024.

5 Desenvolvimento e avaliação do modelo

Esse capítulo apresenta os principais resultados obtidos durante a elaboração do modelo de simulação e nos processos de calibração e validação do modelo. A ordem das seções segue a sequência de elaboração do modelo, com início na modelagem da rede viária e da demanda, posterior execução, verificação de erros, calibração e validação do modelo em que apenas os veículos individuais motorizados são alocados na rede viária e, em seguida, a modelagem da oferta de transporte público coletivo e execução do modelo em que as viagens em veículos individuais motorizados e em transporte público coletivo são alocadas na rede viária. Por fim, são apresentados os resultados do modelo não calibrado com a integração da divisão modal à alocação, permitindo a escolha de modo aos agentes na seleção de seus planos diários.

5.1 Modelagem da rede viária

Para modelagem da rede viária, foram realizados testes com os dados do OSM, e com os dados oficiais disponibilizados pelo PLAMUS.

Foi realizada a verificação da coincidência das especificações das vias por meio da comparação entre os valores indicados de capacidade viária e velocidade nas vias pelo OSM e pelo PLAMUS, sendo esse último considerado já validado, por representar um plano oficial do município. Essa comparação considerou apenas as vias dos postos de contagem existentes, as quais são de diferentes hierarquias viárias, e se encontram em diferentes locais da área de estudo (vide Figura 27).

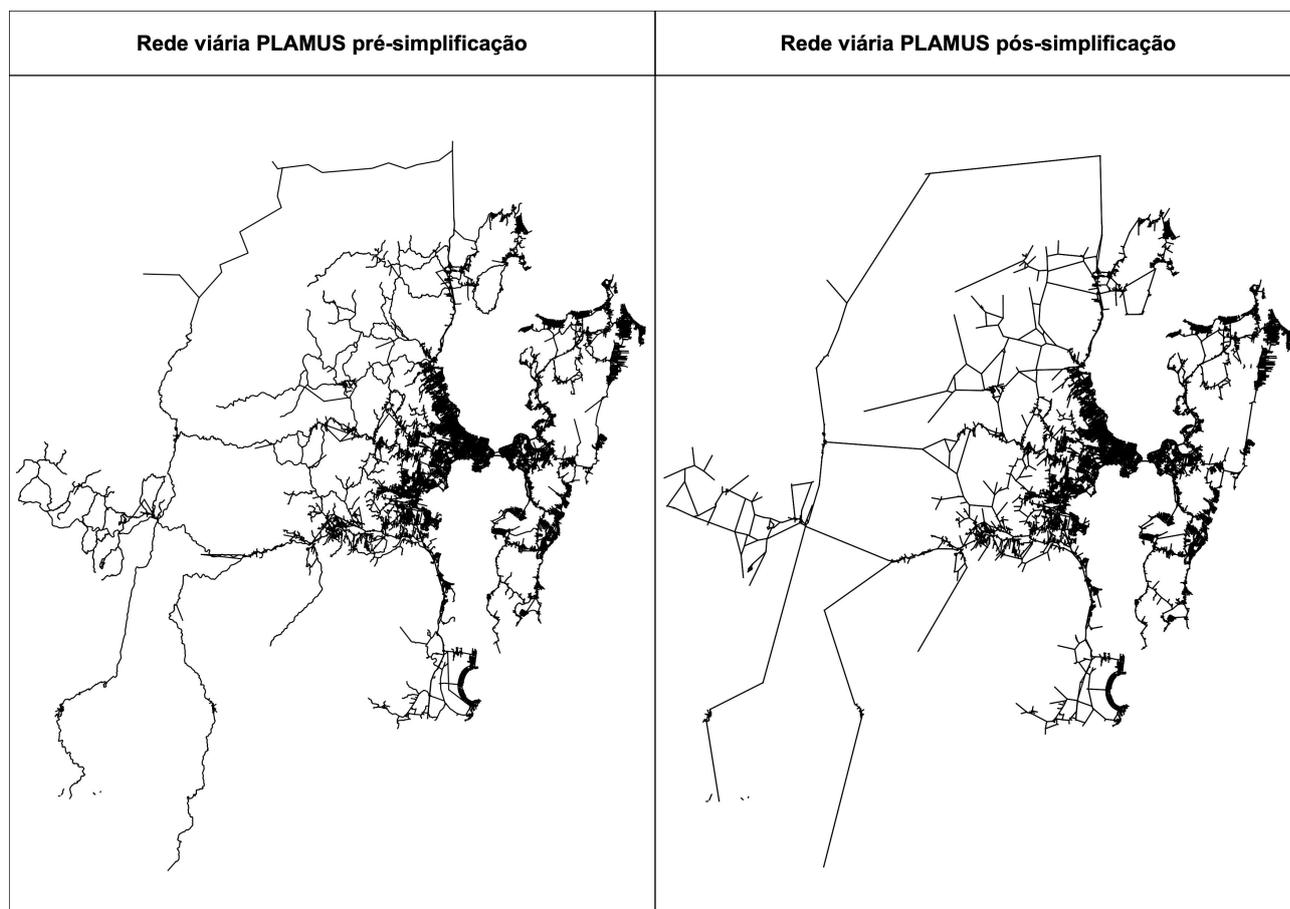
Na verificação, a rede do OSM apresenta capacidades viárias semelhantes à rede do PLAMUS, mas apresenta vias com maior velocidade, e inclui locais onde houveram alterações viárias desde a realização do PLAMUS. Ambas as redes são apresentadas na Figura 26 (vide Seção 4.1), onde são caracterizadas pela sua capacidade viária.

Os dados do OSM foram reestruturados para o formato requerido pelo MATSim por meio do programa JOSM, conforme procedimento apresentado na Seção 2.4.2.

Os dados do PLAMUS são condizentes com a demanda utilizada, já que foram modelados no mesmo estudo. No entanto, por não ser padronizada como os dados do OSM, a sua reestruturação para o formato requerido pelo MATSim requer procedimento específico que considera a estrutura de dados elaborada no contexto do PLAMUS.

A primeira etapa da reestruturação é a simplificação da rede viária, conforme apresentado na Seção 3.1.2, de forma a diminuir a quantidade de *links* existentes na malha. A Figura 29 representa a malha viária antes e depois do processo de simplificação.

Figura 29: Rede viária PLAMUS antes e depois da simplificação



Com a geometria simplificada, são coletadas as coordenadas dos *links* e dos nós, e as características dos *links* para estruturação do dados conforme requer o arquivo de entrada. O Apêndice A apresenta o *script* elaborado para a reestruturação e conversão dos dados da rede viária do PLAMUS no formato do arquivo *network.xml*.

Em relação à capacidade viária, em ambas as redes foi considerada a relação entre a capacidade de fluxo (C_f) e de armazenamento (C_a) definida como $C_a = C_f^{0.75}$ (ver Seção 2.4.6). A definição da capacidade de fluxo é realizada no processo de calibração, onde foram realizadas diferentes simulações com valores de C_f entre 10% e 30%, testadas exclusivamente para a rede viária OSM.

5.2 Modelagem da demanda

Para modelagem da demanda, foram utilizados os diários de viagem obtidos na Pesquisa O/D realizada pelo PLAMUS, descritos na Seção 4.

Para o tratamento inicial dos dados, foram considerados dados inválidos aqueles que não indicam local de origem ou de destino, ou aqueles que não indicam o horário de saída ou de chegada da viagem. Foram desconsiderados os registros que apresentam mais de uma etapa de viagem, os quais representam apenas 0,2% das viagens registradas. A dificuldade de inserir esse dado no modelo, em conjunto com a baixa quantidade de viagens com essa característica, tornou o custo-benefício de inserção inviável.

Para representação reduzida da população, foi considerada uma amostra de 10% representativa da população total da área de estudo. Ou seja, cada morador foi representado no modelo tantas vezes quanto o valor do fator de expansão dos moradores (definido pelo PLAMUS) dividido por dez. Para a representação múltipla de um mesmo morador registrado, os dados de um mesmo morador na entrevista foram utilizados em vários agentes do MATSim, diferenciados por um sufixo de três dígitos. Conforme abordado na Seção 2.4.2, esse processo imputa certo grau de erro ao modelo pelo arredondamento dos valores.

A Tabela 6 apresenta os dados base e expandidos do PLAMUS em comparação com os dados modelados para inserção no MATSim por meio do *parser* apresentado no Apêndice B.

Tabela 6: Comparação dos dados entre a base do PLAMUS e do experimento

Dado	PLAMUS (amostra)	PLAMUS (amostra expandida)	PLAMUS (10% da amostra expandida)	Experimento (10% da amostra expandida)
Etapas de viagem	20.839	1.780.273	178.027	163.780
Viagens (total)	20.795	1.777.094	177.709	163.780
Viagens (car)	10.038	818.278	81.828	75.183
Moradores	12.497	976.800	97.680	76.202

As coordenadas de origem e destino das viagens também foram alvos de processamento. Isso porque, como estão sendo utilizados dados de entrevistas domiciliares de origem e destino com base no modelo quatro etapas, a origem e destino das viagens é indicada por meio das coordenadas do

centroide da zona de tráfego (ZT) em que a atividade está localizada. No MATSim, o procedimento de inserção da demanda na malha é de utilizar o *link* mais próximo à coordenada indicada como *link* de início da viagem. Por esse motivo, a utilização direta dos centroides das ZTs pode levar à geração de agentes presos na simulação, por inserção de demanda exacerbada para um *link* de baixa capacidade.

Portanto, foram analisadas três situações, as quais são listadas a seguir, e ilustradas nas figuras indicadas por meio das respectivas quantidades de origens e destinos por coordenada.

1. **Zona de tráfego e domicílios:** para viagens com origem ou destino em suas próprias casas, são utilizadas as coordenadas dos domicílios informadas nas entrevistas, e para as demais origens e destinos são utilizadas as coordenadas dos centroides das ZT. Ver experimento 4, exposto na Figura 30.
2. **Apenas zonas de tráfego:** todas as viagens têm origem e destino nas coordenadas dos centroides das ZT. Ver experimento 5, exposto na Figura 31.
3. **Domicílios, lotes edificadas em Florianópolis, e zonas de tráfego nos demais municípios:** para viagens com origem ou destino em suas próprias casas, são utilizadas as coordenadas dos domicílios informadas nas entrevistas, e para as demais origens e destinos são utilizadas as coordenadas de alguma edificação construída na ZT, selecionada aleatoriamente. Os dados espacializados dos lotes existentes nos demais municípios da área de estudo não foram encontrados e, portanto, permanece a utilização dos centroides das ZTs nesses municípios. Ver experimento 32, exposto na Figura 32.

Figura 30: Quantidade de viagens originadas ou destinadas por coordenada no Experimento 4

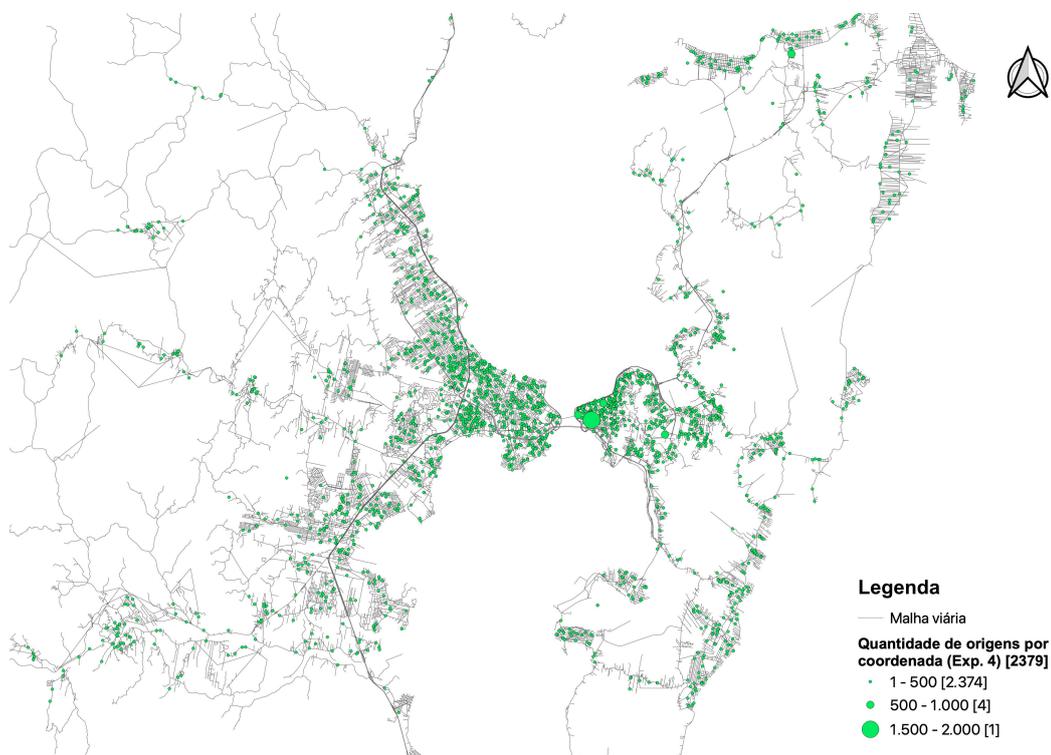


Figura 31: Quantidade de viagens originadas ou destinadas por coordenada no Experimento 5

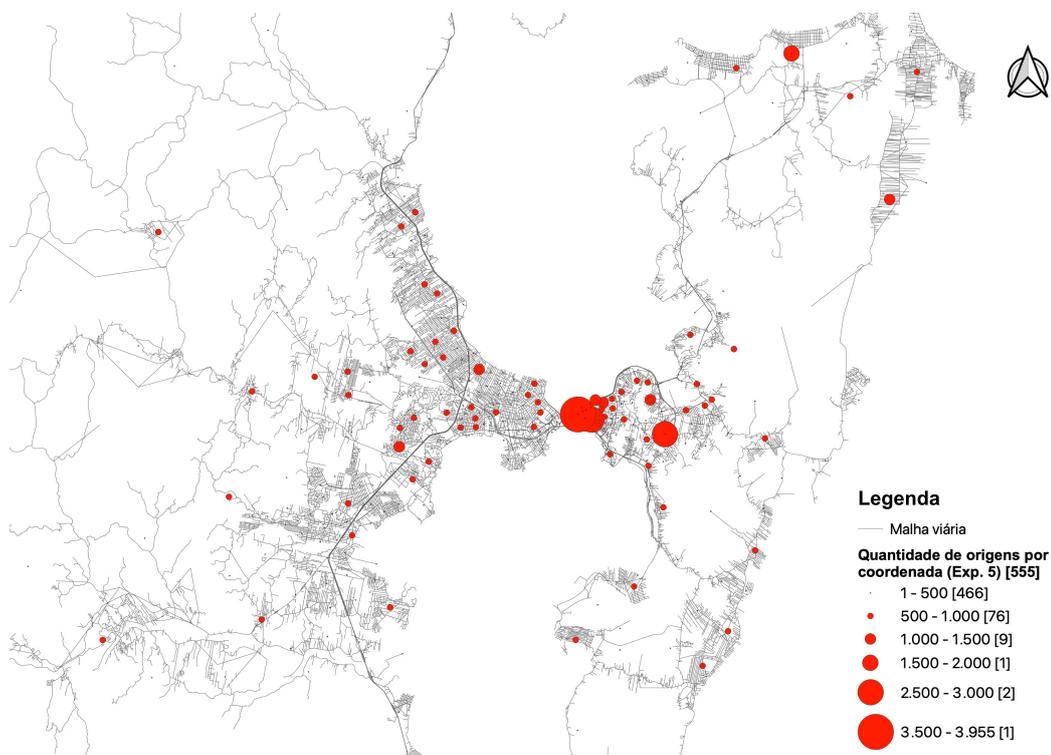
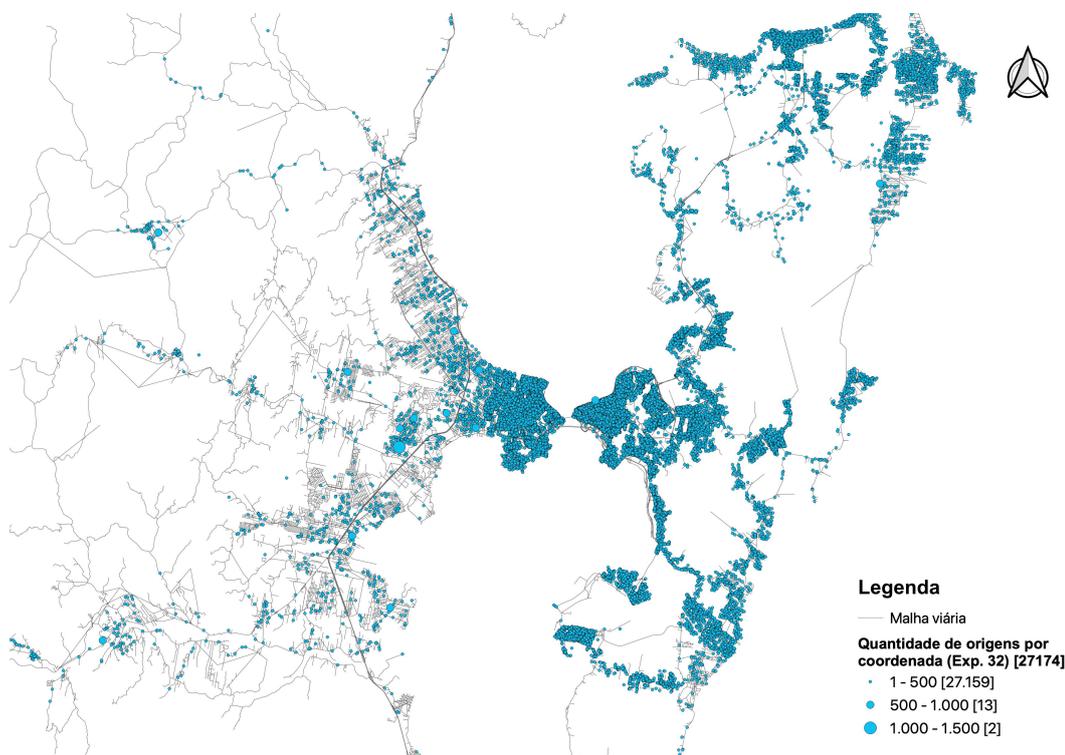


Figura 32: Quantidade de viagens originadas ou destinadas por coordenada no Experimento 32



Enquanto a situação 1 (experimento 4) considera 2.379 pontos de origem distintos, com 99,8% desses originando até 500 viagens por dia, a situação 2 (experimento 5) considera apenas 555 pontos de origem distintos, com 84% desses originando até 500 viagens, 13,7% originando entre 500 e 1.000 viagens, e 2,3% originam entre 1.000 e 3.955 viagens. Na situação 3 (experimento 32), são utilizadas as coordenadas do ponto central das edificações nas ZT de Florianópolis. Foram considerados 27.174 pontos de origem distintos nesse experimento, com 99,9% desses originando até 500 viagens por dia.

Além disso, foram realizadas simulações com diferentes modos alocados na malha. O modo motocicleta foi descrito em um arquivo denominado *mode-vehicles.xml*, em que o comprimento, a largura, a velocidade máxima e o tempo de acesso e de egresso são iguais para as motocicletas e automóveis, e o fator de equivalência utilizado foi de 0,33, conforme o valor referência de DENATRAN (2014).

Os dados de contagem de tráfego foram convertidos para o formato requerido pelo MATSim por meio da elaboração de um *script*, apresentado no Apêndice C. Com a modelagem desses dados, o MATSim gera um arquivo de saída de comparação dos volumes de tráfego obtidos na contagem e na simulação.

5.3 Verificação de erros no modelo

Na simulação teste, com apenas uma iteração e atribuídos os valores padrão indicados pelo MATSim (vide Seção 2.4.6), foram encontrados alguns erros. A seguir são descritos os erros encontrados e as respectivas soluções adotadas:

- planos sem atividade ao final da viagem: em algumas viagens, não existiam informações sobre os seus destinos, impedindo que essas viagens sejam realizadas, então os moradores que apresentaram esse erro foram retirados da simulação;
- agentes impossibilitados de sair de seus locais de origem ou chegar ao seu destino: a existência de vias unidirecionais em becos sem saída impede que a viagem seja realizada, então as vias que apresentaram essa condição foram consideradas como bidirecionais.

5.4 Calibração e validação do modelo

É no arquivo de configuração do modelo que é ajustada a maior parte dos parâmetros da calibração, com exceção do peso do Cadyts, que é ajustado no código fonte do MATSim, e dos ajustes nos arquivos de entrada.

Os parâmetros do modelo foram alterados ao longo da calibração por meio de processo de tentativa e erro: execução da simulação, análise dos resultados (validação) e aplicação das alterações (calibração). Os parâmetros alterados e seus respectivos pesos atribuídos em cada experimento analisado²¹ são apresentados na Tabela 9, considerando os perfis de demanda apresentados na Tabela 7 e os perfis de pontuação apresentados na Tabela 8.

Apesar de não serem vinculados entre si, os perfis de demanda apresentados agrupam os modos alocados na malha, os modos teletransportados e a forma de indicação das coordenadas O/D por conveniência de apresentação dos dados.

Um processo de calibração automatizado foi aplicado para a identificação da capacidade viária de melhor desempenho por meio da elaboração de um *script*, apresentado no Apêndice D.

²¹Apenas os experimentos com resultados que agregaram à construção do modelo são apresentados, com omissão dos casos em que foram realizados ajustes no modelo que não resultam em novas conclusões e reflexões para sua melhoria.

Tabela 7: Perfis de modelo de demanda utilizados nos experimentos

Perfil	Modos alocados na malha	Modos teletransportados	Coordenadas O/D
1	carro, moto e táxi como "car"	pedestre, ciclista, transporte coletivo e outros	centroide ZT e domicílios
2	carro e táxi como "car", moto como "motorcycle", passageiro como "ride"	nenhum	centroide ZT
3	carro e táxi como "car", moto como "motorcycle", passageiro como "ride"	pedestre, ciclista, transporte coletivo e outros	domicílios e edificações

Tabela 8: Perfis de pontuação utilizados nos experimentos

Perfil pontuação	Parâmetros gerais									Parâmetros do carro					Parâmetros dos modos						
	<i>BrainExpBeta</i>	<i>learningRate</i>	<i>earlyDeparture</i>	<i>lateArrival</i>	<i>MU Money</i>	<i>performing</i>	<i>utilityOfLineSwitch</i>	<i>waiting</i>	<i>waitingPt</i>	<i>constant</i>	<i>DMConstant</i>	<i>DUConstant</i>	<i>MU Distance util m</i>	<i>MU Traveling util hr</i>	<i>MDRate</i>	<i>constant</i>	<i>DMConstant</i>	<i>DUConstant</i>	<i>MU Distance util m</i>	<i>MU Traveling util hr</i>	<i>MDRate</i>
1	1	0,9	0	-18	1	6	-1	0	-6	0	0	0	0	-6	-3	0	1	1	0	-6	0
2	1	1	0	-10	0	6	0	0	0	0	0	0	0	-6	0	0	0	0	0	-6	0
3	1	1	0	0	1	6	-1	0	-6	0	0	0	0	-6	0	0	0	0	0	-6	0
4	2	1	0	-10	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MU é *marginal utility*, ou utilidade marginal
DM é *daily monetary*, se referindo à constante financeira diária
MD é *monetary distance*, se referindo à taxa monetária de distância
DU é *daily utility*, ou utilidade diária

Tabela 9: Configuração dos experimentos

ID	Rede viária	Porcentagem de capacidade viária	Quantidade de iterações	Stucktime	Modos alocados na malha	Modos teletransportados	Coordenadas O/D	Peso Cadyts	Peso Reroute	Peso BestScore	Peso SelectExp-Beta	Perfil da pontuação
1	OSM	14%	1000	10	1	1	1	-	0,1	0,1	0,8	1
2	OSM	14%	1000	10	1	1	1	1000	0,1	0,1	0,8	1
3	PLAMUS	14%	1000	10	1	1	1	-	0,1	0,1	0,8	1
4	PLAMUS	14%	1000	10	1	1	1	1000	0,1	0,1	0,8	1
5	OSM	14%	1000	10	2	2	2	-	0,1	0,1	0,8	2
6	OSM	14%	1000	30	1	1	1	-	0,1	0,1	0,8	3
7	OSM	14%	1000	30	1	1	1	10	0,1	0,1	0,8	3
8	OSM	14%	1000	30	1	1	1	100	0,1	0,1	0,8	3
9	OSM	14%	1000	30	1	1	1	1000	0,1	0,1	0,8	3
10	OSM	10%	50	30	1	1	1	1000	0,5	0,9	-	4
11	OSM	11%	50	30	1	1	1	1000	0,5	0,9	-	4
12	OSM	12%	50	30	1	1	1	1000	0,5	0,9	-	4
13	OSM	13%	50	30	1	1	1	1000	0,5	0,9	-	4
14	OSM	14%	50	30	1	1	1	1000	0,5	0,9	-	4
15	OSM	15%	50	30	1	1	1	1000	0,5	0,9	-	4
16	OSM	16%	50	30	1	1	1	1000	0,5	0,9	-	4
17	OSM	17%	50	30	1	1	1	1000	0,5	0,9	-	4
18	OSM	18%	50	30	1	1	1	1000	0,5	0,9	-	4
19	OSM	19%	50	30	1	1	1	1000	0,5	0,9	-	4
20	OSM	20%	50	30	1	1	1	1000	0,5	0,9	-	4
21	OSM	21%	50	30	1	1	1	1000	0,5	0,9	-	4
22	OSM	22%	50	30	1	1	1	1000	0,5	0,9	-	4
23	OSM	23%	50	30	1	1	1	1000	0,5	0,9	-	4
24	OSM	24%	50	30	1	1	1	1000	0,5	0,9	-	4
25	OSM	25%	50	30	1	1	1	1000	0,5	0,9	-	4
26	OSM	26%	50	30	1	1	1	1000	0,5	0,9	-	4
27	OSM	27%	50	30	1	1	1	1000	0,5	0,9	-	4
28	OSM	28%	50	30	1	1	1	1000	0,5	0,9	-	4

Continua na próxima página

Tabela 9: Configuração dos experimentos (continuação)

ID	Rede viária	Porcentagem de capacidade viária	Quantidade de iterações	Stucktime	Modos alocados na malha	Modos teletransportados	Coordenadas O/D	Peso Cadyts	Peso Reroute	Peso BestScore	Peso SelectExp-Beta	Perfil da pontuação
29	OSM	29%	50	30	1	1	1	1000	0,5	0,9	-	4
30	OSM	30%	50	30	1	1	1	1000	0,5	0,9	-	4
31	PLAMUS	14%	1000	30	3	3	3	-	0,1	0,1	0,8	3
32	PLAMUS	14%	1000	30	3	3	3	100	0,1	0,1	0,8	3

Na validação do cenário para veículos individuais motorizados, foram verificados coeficientes de determinação (r^2) entre os valores reais e simulados dos volumes diários e dos volumes horários, o tempo médio de viagem, a pontuação executada média (média das impedâncias, ou utilidade, das viagens realizadas na última iteração da simulação), a quantidade de agentes presos (quantidade de viagens que não conseguem ser finalizadas ao final da simulação) nos diferentes experimentos apresentados.

Os resultados são apresentados na Tabela 10.

Tabela 10: Medidas de performance dos experimentos analisados

ID	r^2 volume horário	r^2 volume diário	Tempo médio de viagem	Pontuação média	Porcentagem de agentes presos
1	0,4153	0,5486	01:38	-65.426	2,37
2	0,4502	0,6003	01:11	-66.764	0,86
3	0,4739	0,5687	01:42	-60.665	4,73
4	0,4909	0,6335	01:26	-61.653	2,62
5	0,4265	0,5529	01:02	-27.779	3,59
6	0,6139	0,8817	00:42	-14	0,83
7	0,6259	0,8902	00:42	-18	0,82
8	0,6253	0,8884	00:42	-43	0,82

Continua na próxima página

Tabela 10: Configuração dos experimentos (continuação)

ID	r^2 volume horário	r^2 volume diário	Tempo médio de viagem (hh:mm)	Pontuação média	Porcentagem de agentes presos (%)
9	0,6234	0,8873	00:42	-490	0,82
10	0,6718	0,8315	00:57	305	1,42
11	0,6665	0,8313	00:54	260	1,26
12	0,6278	0,8294	00:51	50	1,09
13	0,6406	0,8316	00:50	170	0,94
14	0,6392	0,8326	00:48	110	0,84
15	0,6010	0,8337	00:46	0	0,75
16	0,6071	0,8341	00:46	40	0,65
17	0,6088	0,8375	00:44	-20	0,57
18	0,5767	0,8338	00:42	-90	0,49
19	0,5725	0,8339	00:41	50	0,40
20	0,5604	0,8388	00:41	20	0,31
21	0,5405	0,8333	00:39	30	0,23
22	0,5414	0,8355	00:39	-20	0,14
23	0,5185	0,8284	00:38	-40	0,09
24	0,5162	0,8311	00:37	-20	0,06
25	0,5004	0,8221	00:36	-120	0,06
26	0,5004	0,8243	00:35	-60	0,06
27	0,4848	0,8213	00:34	-100	0,06
28	0,4724	0,8179	00:34	-90	0,06
29	0,4690	0,8120	00:33	-95	0,06
30	0,4604	0,8066	00:33	-50	0,06
31	0,6128	0,9209	00:30	73	0,04
32	0,6296	0,9416	00:30	87	0,04

Considerando os resultados apresentados, pode-se concluir que o experimento de melhor desempenho é o de ID 32. Esse experimento possui as características apresentadas na Tabela 11.

Os resultados obtidos nos experimentos realizados, com ênfase nos resultados do experimento 32, são discutidos em detalhes a seguir.

Tabela 11: Configurações do experimento final

Quantidade de iterações	1.000
Rede viária	PLAMUS
Proporção da capacidade viária	14%
<i>stucktime</i>	30 segundos
Peso do Cadyts	100
Modos alocados na malha	Carro e táxi como “car”, motocicletas como “motorcycle” (considerando fator de equivalência de 0,33 em relação ao carro), passageiros como “ride”
Coordenadas O/D	Alocadas em domicílios e edificações
Horário de saída	Ajustado com parcela de até dez minutos em distribuição normal
Peso dos parâmetros de pontuação	Perfil 3 (vide Tabela 8)

Volumes de tráfego É possível verificar que os coeficientes de correlação entre os volumes diários são consistentemente maiores do que os coeficientes de correlação entre os volumes horários.

Nos experimentos para seleção do fator de ajuste da capacidade viária, utilizando a rede viária OSM (experimentos de ID 10 a 30), o r^2 dos volumes horários apresenta relação inversa com a porcentagem de capacidade viária, apresentando pior desempenho (r^2 de 0,4604) com 30% de capacidade, e melhor desempenho com 10% de capacidade (r^2 de 0,6718), e a maior variação dentre os r^2 (0,2114). Em contramão, a correlação entre os volumes diários apresenta pouca variação (0,0322) entre os experimentos, com valor máximo de 0,8388 para capacidade de 20%, e valor mínimo de 0,8066 para capacidade de 30%.

Para decisão entre qual malha viária utilizar, foram realizadas simulações teste com ambas malhas disponíveis. Os principais resultados desses experimentos são apresentados na Tabela 12.

Tabela 12: Resultados dos experimentos com as diferentes redes viárias

Experi- mento	Dados da malha	Uso do Cadyts	Pontuação	Agentes presos (%)	r^2 volume horário	r^2 volume diário
1	OSM	Não	-65.426	2,37	0,4153	0,5486
2	OSM	Sim	-66.764	0,86	0,4502	0,6003
3	PLAMUS	Não	-60.665	4,73	0,4739	0,5687
4	PLAMUS	Sim	-61.653	2,62	0,4909	0,6335

Os resultados do modelo com a malha viária do PLAMUS apresentam maiores r^2 . Esses modelos, no entanto, também resultam em maior porcentagem de agentes presos durante a simulação.

Para investigação da causa dos agentes presos, foram avaliadas três situações possíveis em relação à escolha das coordenadas de origem e destino das viagens. Para avaliação das três situações possíveis em relação à escolha das coordenadas de origem e destino das viagens, e as duas fontes possíveis para a rede viária foi realizada a análise dos resultados de quatro experimentos, de ID 1, 3, 5 e 31. Isso porque podem ser pareados um a um para identificação da melhor rede viária e do melhor método de distribuição das viagens entre as coordenadas O/D. Suas principais configurações, seus coeficientes de correlação dos volumes de tráfego diários e horários e a porcentagem de agentes presos são apresentados na Tabela 13.

Tabela 13: Resultados dos experimentos com diferentes coordenadas O/D

Experi- mento	Dados da malha	Coordenadas O/D	r^2 volume horário	r^2 volume diário	Agentes presos (%)
1	OSM	Centroide ZT e domicílios	0,4153	0,5486	2,37
3	PLAMUS	Centroide ZT e domicílios	0,4739	0,5687	4,73
5	OSM	Centroide ZT	0,4265	0,5529	3,59
31	PLAMUS	Domicílios e edificações	0,6128	0,9209	0,04

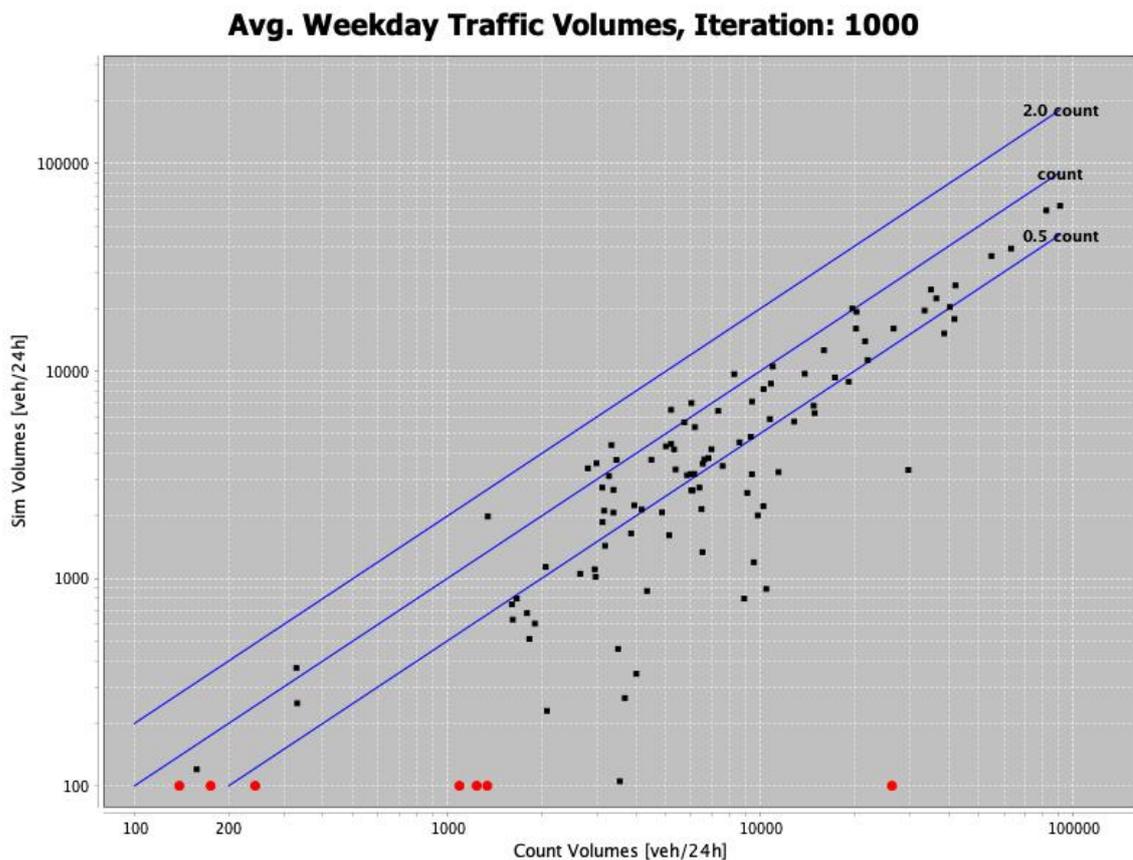
O experimento que considera apenas os centroides das ZT como origens e destinos das viagens (ID 5) apresenta r^2 de volume horário e diário cerca de 0,01 maior do que o experimento que utiliza a mesma malha viária (ID 1), mas considera os centroides e os domicílios como pontos O/D. Apesar disso, o experimento 5 apresenta 1,22% a mais de agentes presos do que o experimento 1.

O experimento 31 considera a demanda de forma distribuída ao longo das edificações e domicílios da ZT, e apresenta r^2 de volume horário 0,1389 (29,3%) maior do que o experimento que utiliza a mesma malha viária (ID 3), mas considera os centroides e os domicílios como pontos O/D. E o r^2 de volume diário é 0,3522 (61,9%) maior no experimento 31. Além disso, o experimento 31 é o único com ambos r^2 maiores que 0,6.

Apesar da porcentagem de agentes presos no experimento 3 ter sido a maior entre todos os experimentos analisados, com o uso da rede viária do PLAMUS, o experimento 31 demonstra que a rede viária não era a causa para o problema, já que usa a mesma rede viária e obteve apenas 0,04% de agentes presos. Assim, são selecionadas como de melhor desempenho a rede viária PLAMUS e a utilização dos domicílios e edificações como coordenadas de origem e destino das viagens.

Na avaliação dos volumes de tráfego diários no experimento 32, foi identificado que o r^2 nos postos de contagem considerados é igual a 0,9416. A comparação entre os volumes de tráfego diários pode ser observada na Figura 33.

Figura 33: Comparação entre volumes de tráfego diários no experimento 32

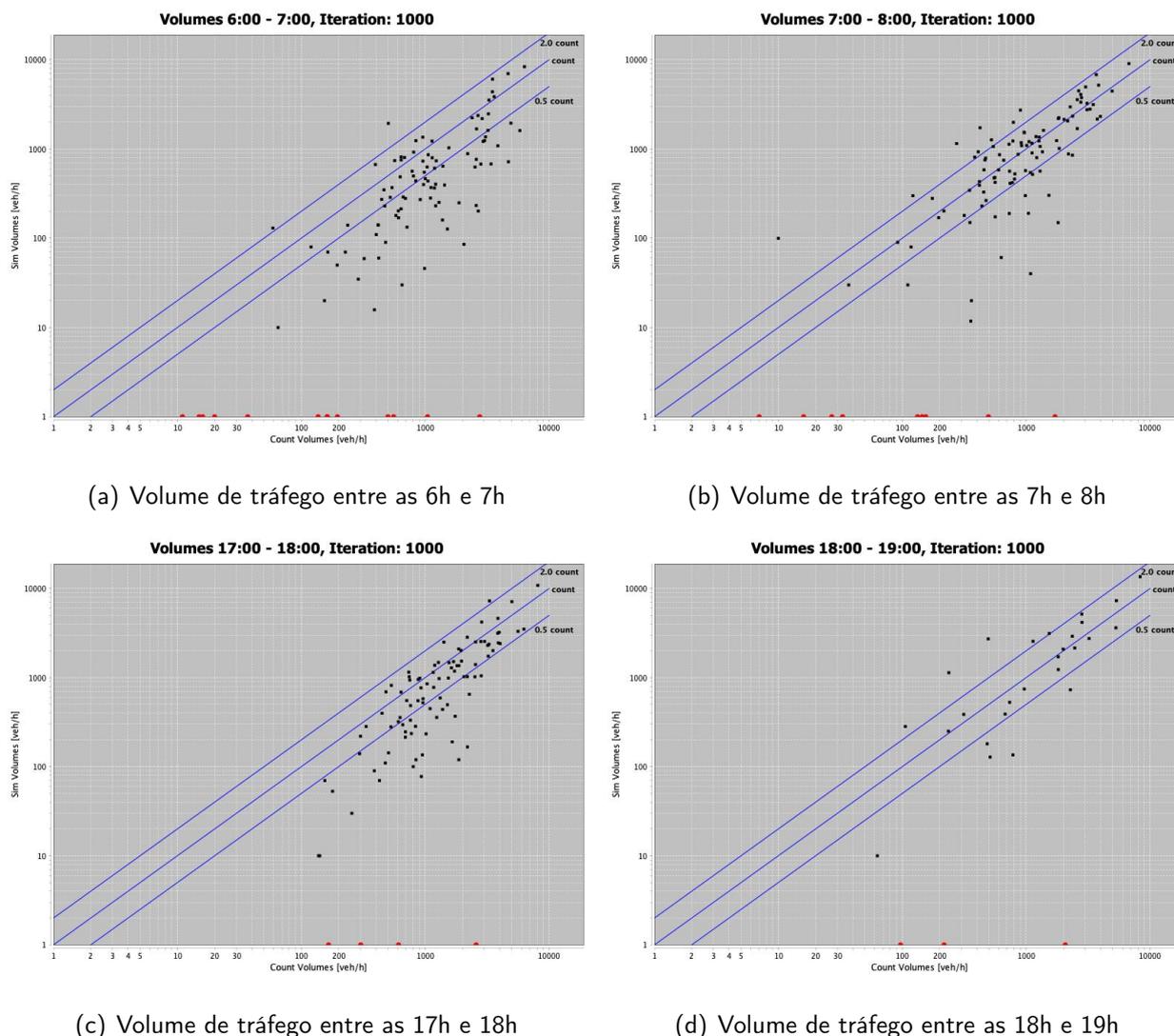


Na maior parte dos postos (59,6%) a simulação resulta em volumes diários maiores que metade do valor das contagens, e menores que o dobro desse valor. De forma geral, a simulação tende a resultar em volumes de tráfego menores do que nas contagens de tráfego, com algumas exceções.

Considerando a variação horária da demanda, também foi avaliado o coeficiente de relação dos volumes de tráfego horários nos postos de contagem. Considerando os postos e horários em que

ocorreram contagens de tráfego, apresentados na seção 4, o r^2 resultou em 0,6296. A Figura 34 apresenta a comparação entre os volumes de tráfego nos períodos de pico da manhã e da tarde.

Figura 34: Comparação entre volumes de tráfego horários nos períodos de pico da manhã e da tarde

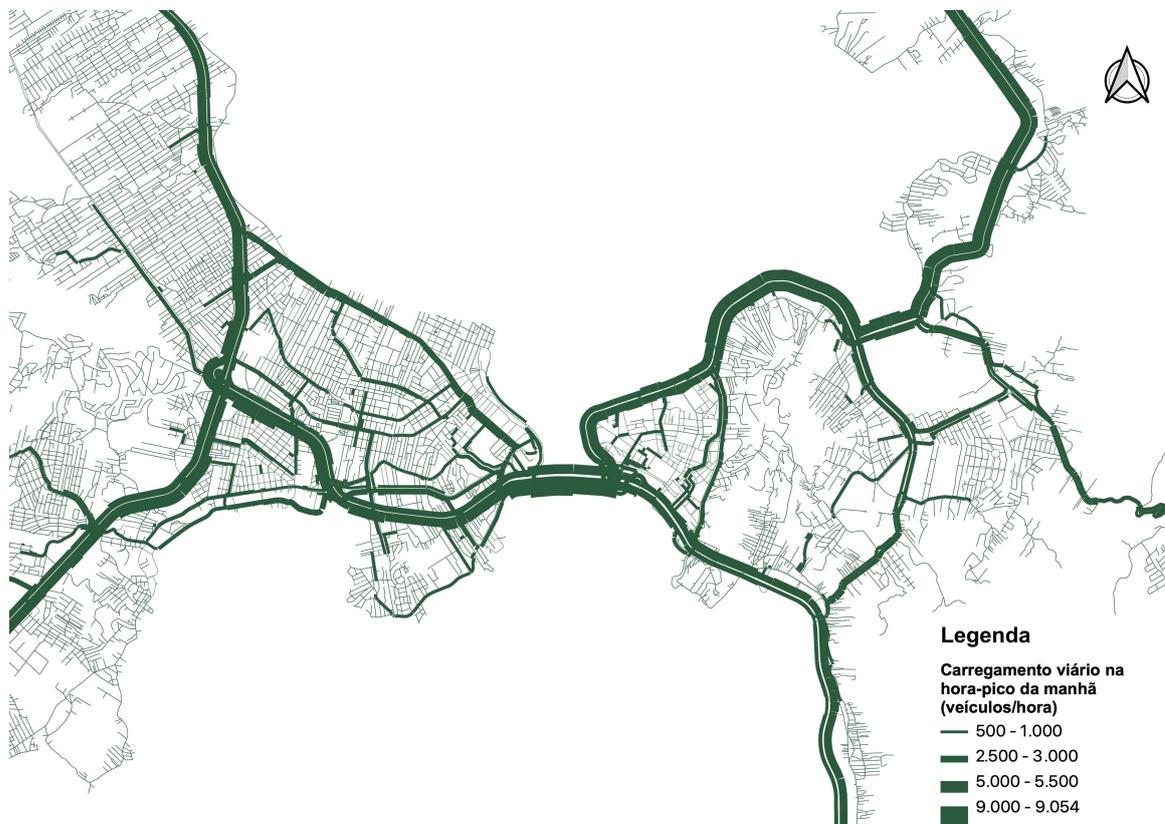


No horário das 6h e das 17h, a simulação resulta em volumes de tráfego inferiores aos constatados nas contagens de tráfego. Já no horário das 7h e das 18h, a maior parte dos volumes simulados fica entre metade e o dobro do volume de tráfego contado em pesquisa de campo. Esses horários são considerados a hora-pico da manhã e da tarde, respectivamente.

Foram calculados os r^2 dos postos contagem no período de pico da manhã e no período de pico da tarde, iguais a 0,7301 e 0,8068, respectivamente. Esses resultados são inferiores aos obtidos no PLAMUS, iguais a 0,9775 e 0,9800, respectivamente.

Nesses horários, são apresentados os carregamentos da rede viária realizados pelo PLAMUS, os quais são comparados com o carregamento simulado. A Figura 35 apresenta o carregamento na hora-pico da manhã, e a Figura 36 ilustra o carregamento na hora-pico da tarde.

Figura 35: Carregamento da rede viária na hora-pico da manhã

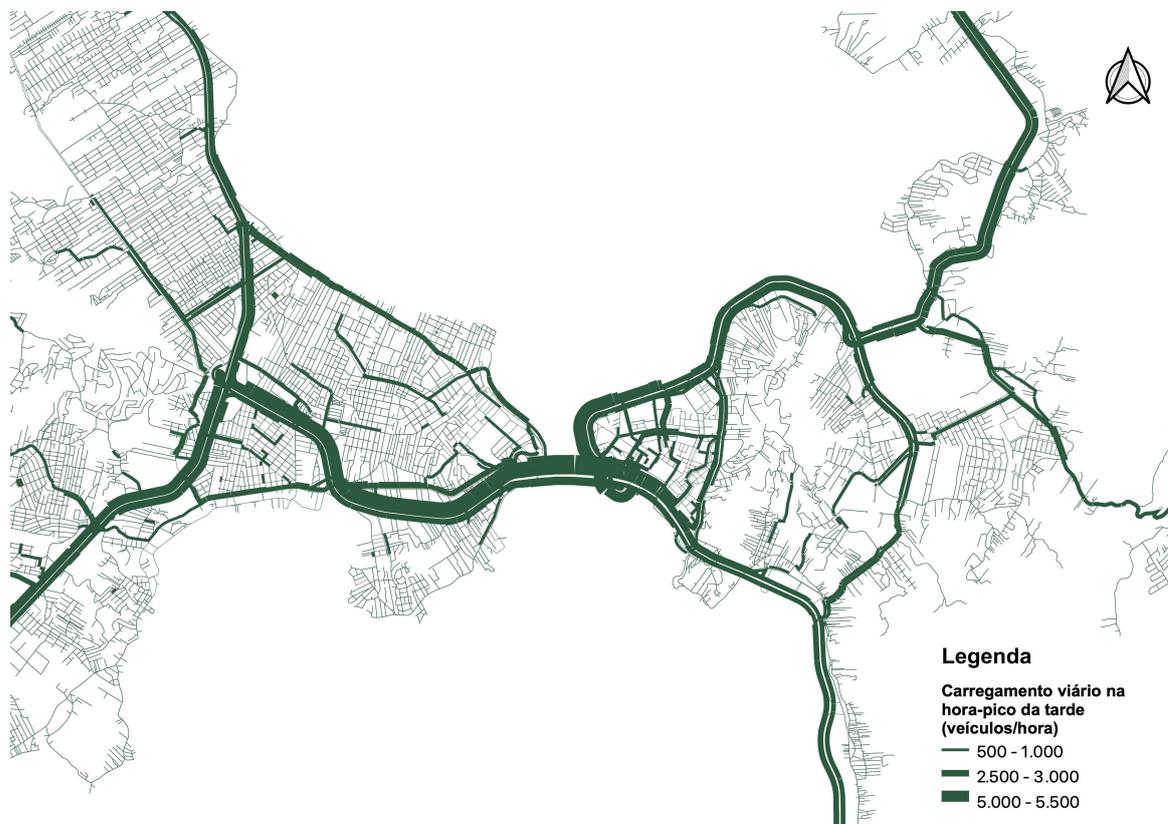


(a) MATSim

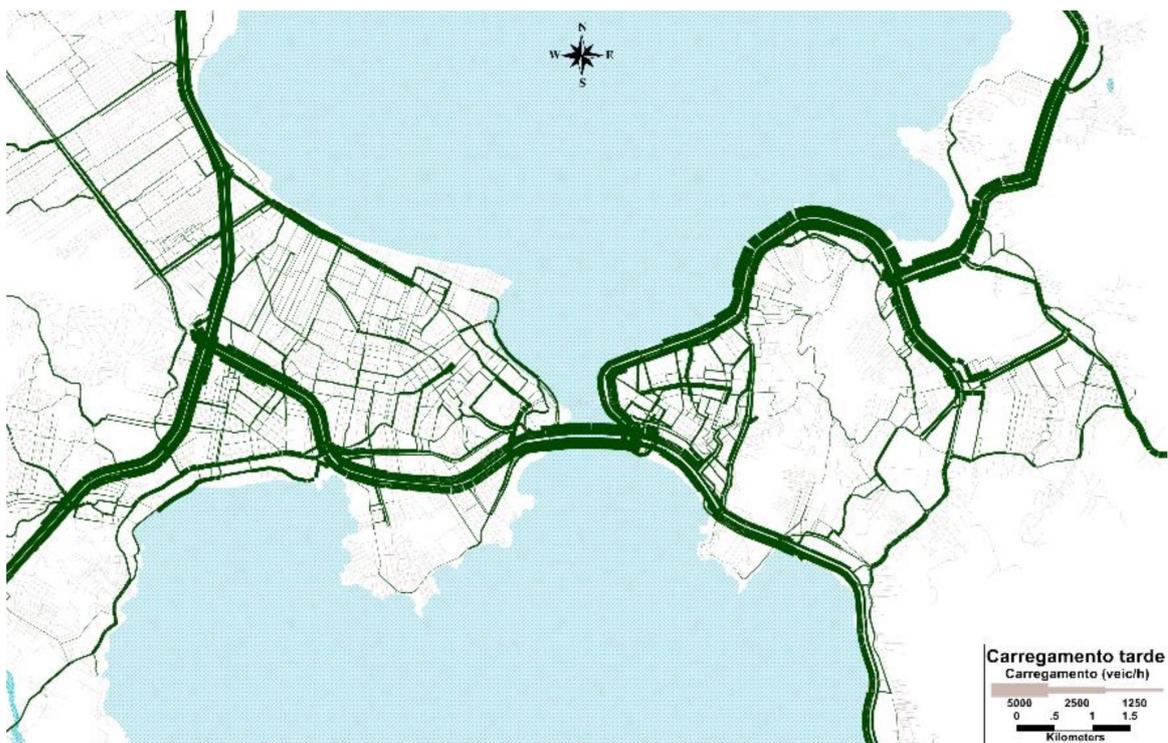


(b) PLAMUS

Figura 36: Carregamento da rede viária na hora-pico da tarde



(a) MATSim



(b) PLAMUS

Apesar da dificuldade em reproduzir fielmente a representação gráfica do carregamento do PLAMUS, realizado no TransCAD, é possível identificar que os padrões de comportamento simulados nesse trabalho são visualmente semelhantes aos resultados do PLAMUS, tanto na hora-pico da manhã quanto na hora-pico da tarde. No entanto, a simulação no MATSim representa com maior ênfase a dinâmica pendular e a dependência dos demais bairros ao Centro de Florianópolis, com maiores diferenças nos dois sentidos das vias que conectam o Centro ao restante da área de estudo.

Tempos de viagem Inicialmente, foram analisados os tempos de viagem médio das viagens em relação ao tempo de viagem médio em modos motorizados levantado por meio da Pesquisa O/D do PLAMUS, igual a 31,2 minutos.

Os experimentos com perfil de pontuação 1 (experimentos 1, 2, 3 e 4) apresentam os maiores tempos de viagem, todos acima de uma hora. Além disso, o experimento 5 também apresenta tempo médio acima de uma hora, e o ponto de convergência entre todos esses experimentos é a utilização do tempo de *stucktime* igual a 10 segundos, e a agregação da demanda em ZT, pelo menos em alguns casos.

Os experimentos de verificação do peso do Cadyts (ID 6, 7, 8 e 9) não apresentam variação de tempo médio de viagem entre si, indicando que o Cadyts não exerce influência nesse quesito.

Nos experimentos com diferentes capacidades viárias para a malha do OSM, o tempo médio é de 57 minutos para 10% da capacidade. Com o aumento da capacidade, o tempo médio de viagem diminui até o experimento 30, com 30% de capacidade, onde o tempo médio de viagem é de 33 minutos, o valor mais próximo do tempo médio relatado pelo PLAMUS, em relação aos experimentos que utilizam os centroides e domicílios como pontos de origem e destino das viagens.

Os experimentos que consideram os domicílios e edificações como pontos O/D apresentam tempo médio de 30 minutos: os menores tempos de viagem e mais próximos aos relatados pelo PLAMUS, em relação a todos os experimentos.

Ainda, foram calculados os coeficientes de correlação (r^2) entre as durações das viagens simuladas e das auferidas em pesquisa de velocidade, considerando 16 trechos, dos quais 15 possuem sentido de ida e volta, totalizando 31 trechos analisados na hora de pico da manhã (7h) e da tarde (18h).

O cálculo do coeficiente de correlação dos tempos de viagem foi realizado apenas para os experimentos que utilizam a rede viária PLAMUS (experimentos com ID 3, 4, 31 e 32), já que a análise foi realizada após a definição da rede viária com melhor desempenho. A Tabela 14 apresenta os resultados desse cálculo para os experimentos citados.

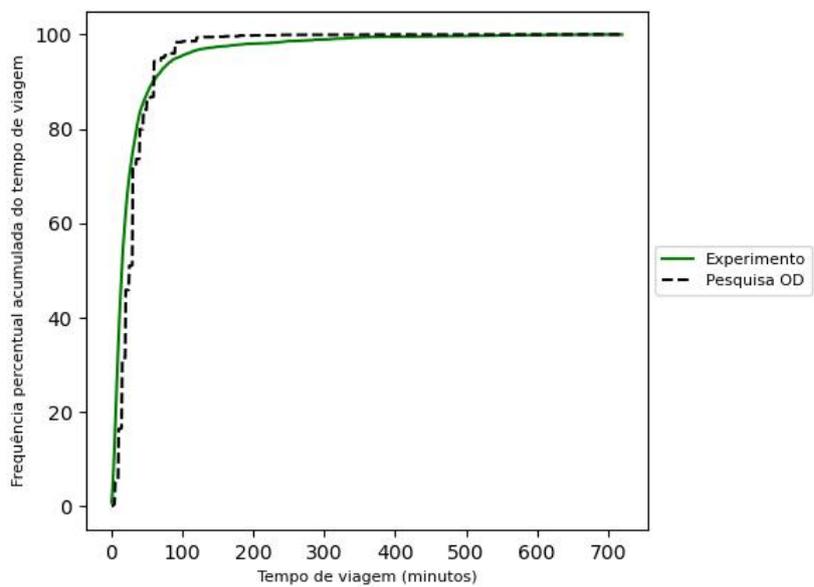
Tabela 14: R^2 da duração das viagens

ID	Peso do Cadyts	Coordenadas O/D	Tempo médio de viagem	r^2 tempo de viagem
3	0	Centróide ZT e domicílios	01:42	0,3445
4	1000	Centróide ZT e domicílios	01:26	0,4615
31	0	Domicílios e edificações	00:30	0,8542
32	100	Domicílios e edificações	00:30	0,8562

Confirmando as análises sobre os tempos médios de viagem, os experimentos que utilizam os domicílios e edificações como pontos O/D apresentam os melhores coeficientes de correlação de tempos de viagem. O uso do Cadyts não apresentou influência no r^2 , alterando-o em apenas 0,002.

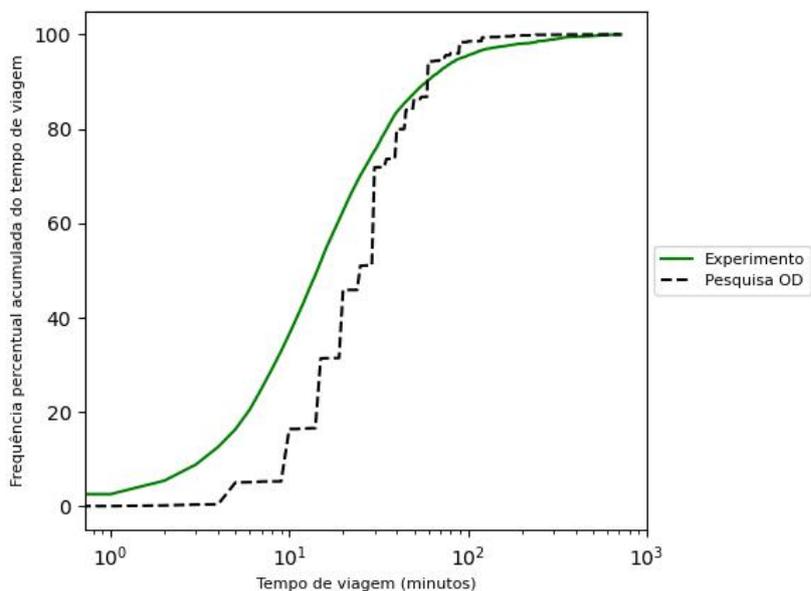
Como comparação entre os tempos de viagem simulados no experimento 32 e relatados na pesquisa O/D, é apresentado a Figura 37, que ilustra o percentual de frequência de tempo de viagem acumulada para os resultados da O/D e para a simulação em escala linear e em escala logarítmica.

Figura 37: Frequências percentuais acumuladas dos tempos de viagem simulados e relatados na O/D



(a) Escala linear

Figura 37: Frequências percentuais acumuladas dos tempos de viagem simulados e relatados na O/D



(b) Escala logarítmica

O comportamento dos tempos de viagem é semelhante nos dois casos, conforme ilustra a Figura 37(a). Com a utilização da escala log, é possível ter maior visualização das diferenças das viagens de até 100 minutos. Assim, a Figura 37(b) ilustra que o experimento possui maior quantidade de viagens de curta duração do que as relatadas na pesquisa O/D. Este também foi o experimento com duração média das viagens igual a 30 minutos, menor média entre os experimentos e semelhante à duração média relatada na Pesquisa O/D, igual a 33,9 minutos.

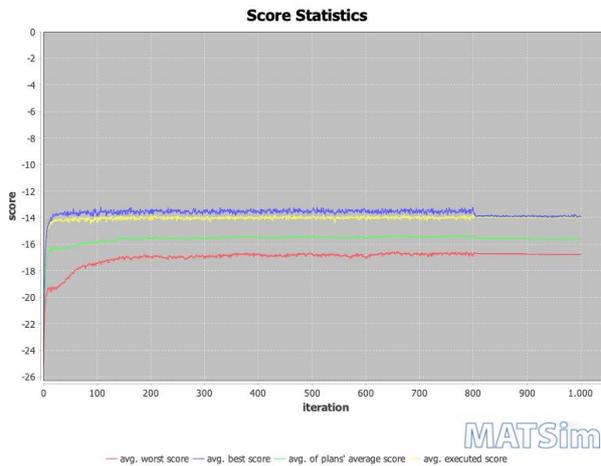
Na avaliação do coeficiente de correlação da duração de viagens em trechos selecionados (onde foram realizadas pesquisas de velocidade na ocasião do PLAMUS), apresentados na seção 4, foi identificado que o r^2 nesses trechos para a hora-pico da manhã e da tarde é igual a 0,8562.

Estabilização da pontuação Os parâmetros que influenciam a pontuação dos agentes são apresentados na seção 2.4.4, além da influência causada pelo Cadyts.

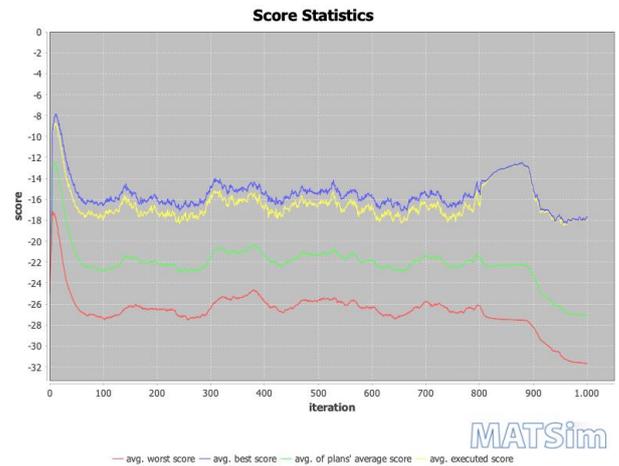
Foram realizados testes com diferentes pesos para a pontuação do Cadyts, com peso de 0, 10, 100 e 1.000 unidades de utilidade e com 1.000 iterações. A Figura 38 apresenta as evoluções

das pontuações (soma total da impedância) dos planos de viagem executados em quatro testes com diferentes pesos do Cadyts.

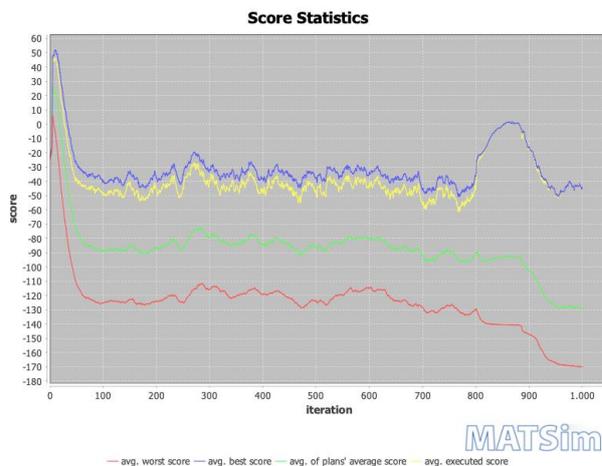
Figura 38: Evolução das pontuações em simulações com diferentes pesos do Cadyts



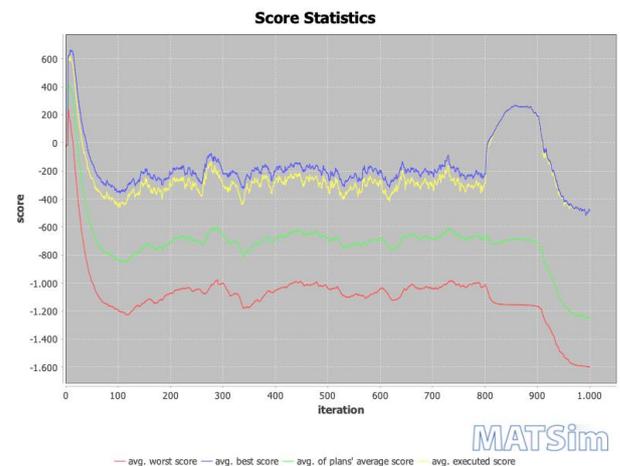
(a) Cadyts Peso 0 (experimento 6)



(b) Cadyts Peso 10 (experimento 7)



(c) Cadyts Peso 100 (experimento 8)



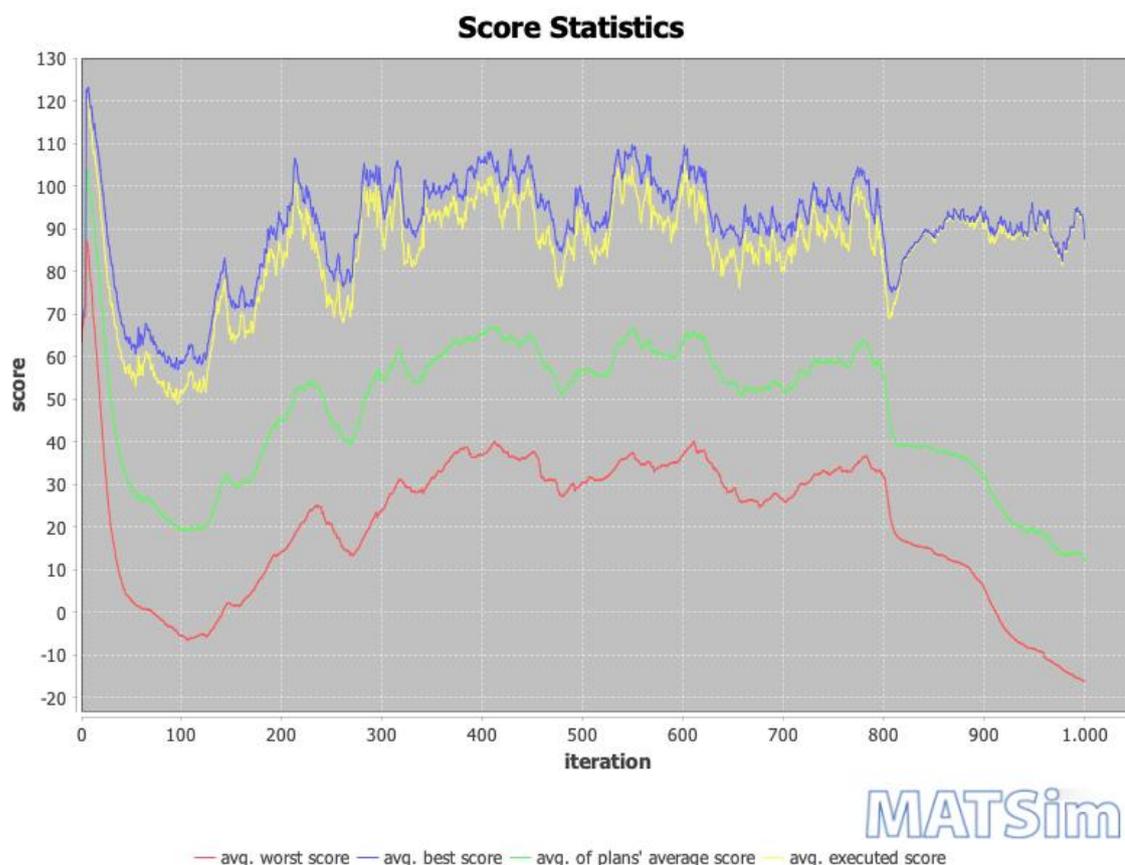
(d) Cadyts Peso 1000 (experimento 9)

A influência do Cadyts no equilíbrio da pontuação é maior quanto maior for o peso atribuído a ele e, de acordo com a Figura 38, essa influência gera instabilidade na evolução da pontuação média executada ao longo das iterações. Enquanto o experimento com peso 0 apresenta estabilidade a partir da 200ª iteração, o experimento com peso 10 apresenta equilíbrio após a 950ª iteração, o experimento com peso 100 entra em equilíbrio após a 975ª iteração, e o experimento com peso mil não chega ao equilíbrio.

Para utilização dos benefícios do Cadyts, mantendo certa estabilidade na evolução da pontuação, o peso 100 foi considerado como o mais adequado.

Os resultados da evolução da pontuação média dos planos do experimento 32 são apresentados na Figura 39.

Figura 39: Evolução da pontuação



Com o peso do Cadyts igual a 100, a evolução da pontuação não apresenta tanta instabilidade quanto nos experimentos em que seu peso é maior. Isso porque a cada nova iteração os agentes escolhem aleatoriamente novas rotas e, caso essa rota não contribua para a reprodução dos volumes de tráfego identificados nos postos de contagem, são atribuídas pontuações negativas multiplicadas por 100 ou 1000. Ou seja, pequenas mudanças nas rotas ocasionam grandes mudanças na pontuação, e essas mudanças nas rotas são realizadas a cada iteração por 10% dos agentes (vide peso do parâmetro *Reroute*). Nesse sentido, é esperada certa instabilidade pela utilização do Cadyts.

O início do efeito do Cadyts ocorre na iteração 5, após as iterações preparatórias (ver *preparatory iterations* em Flötteröd (2010)), onde é possível ver a queda na pontuação média. Após

a 800ª iteração, as estratégias de replanejamento são desativadas, e os agentes apenas selecionam entre os planos existentes em sua memória (igual a 5 nesse experimento). Nas últimas 200 iterações a pontuação executada média é ainda mais próxima à pontuação média dos melhores planos.

Além disso, foi utilizado *learningRate* igual a 1, o que pode causar instabilidade da evolução, mas acelera a convergência inicial entre a melhor pontuação e a pontuação executada, conforme Horni, Nagel e Axhausen (2016).

Além disso, o perfil da pontuação é o mais simples possível, e considera apenas a realização de atividades, com peso 6 positivo, e a duração da viagem, com peso 6 negativo. Nesse sentido, pode-se concluir que o modelo incentiva a realização das atividades previstas no plano diário em viagens o mais curtas possíveis, ao mesmo tempo que incentiva que os agentes reproduzam o comportamento indicado nas contagens de tráfego.

Quantidade de agentes presos Na análise da porcentagem de capacidade viária, entre os experimento de ID 10 a 30, é possível perceber que a quantidade de agentes presos diminui com o aumento da capacidade viária, chegando ao mínimo de 98 agentes presos a partir da simulação com 24% da capacidade viária. Na simulação com maior restrição de capacidade viária (10%), 1,42% das viagens não são finalizadas.

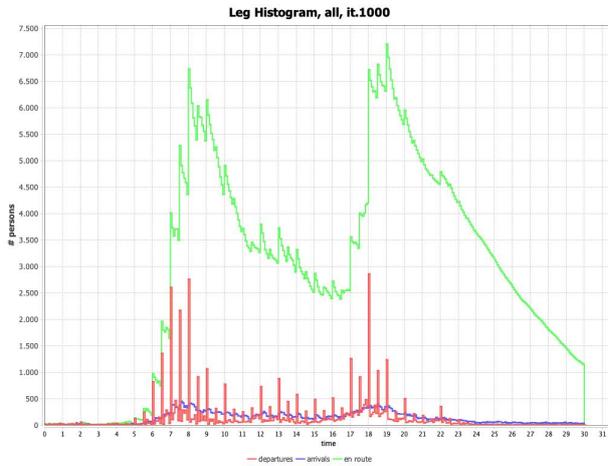
O peso do Cadyts não possui grande influência sobre a quantidade de agentes presos, conforme pode ser concluído por meio dos resultados dos experimentos 6, 7, 8 e 9, os quais diferem apenas em relação ao peso do Cadyts, e apresentam diferença de menos de 0,01% na porcentagem de agentes presos.

Foi verificado, então, a influência da malha viária sobre a quantidade de agentes presos, do parâmetro de *stucktime* e da distribuição da demanda em relação aos pontos de origem e destino das viagens. A Figura 40 apresenta o histograma de viagens de quatro experimentos que possuem variações em relação a esses parâmetros entre si.

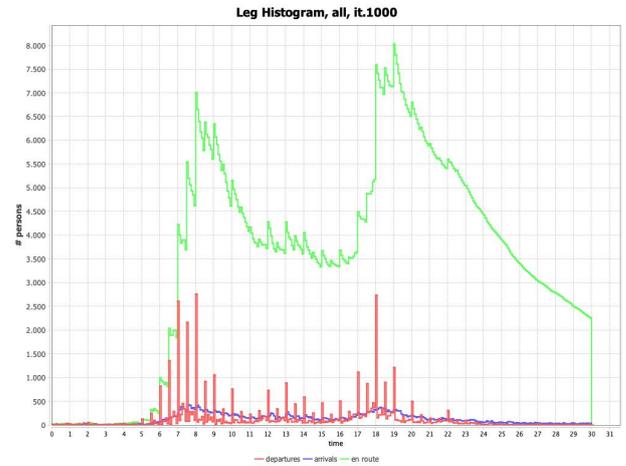
Os experimentos com *stucktime* de 10 segundos (ID 1 e 3) apresentaram a maior quantidade de agentes presos em relação aos experimentos com *stucktime* de 30 segundos. Da mesma forma, o experimento com a malha viária do PLAMUS (ID 3) apresentou maior quantidade de agentes presos em relação ao experimento de mesmas características com a malha viária do OSM (ID 1).

No entanto, a influência desses dois fatores não é suficiente para garantir uma proporção de agentes presos inferior a 0,1%. Apenas o experimento 31, que considera que as viagens têm origem

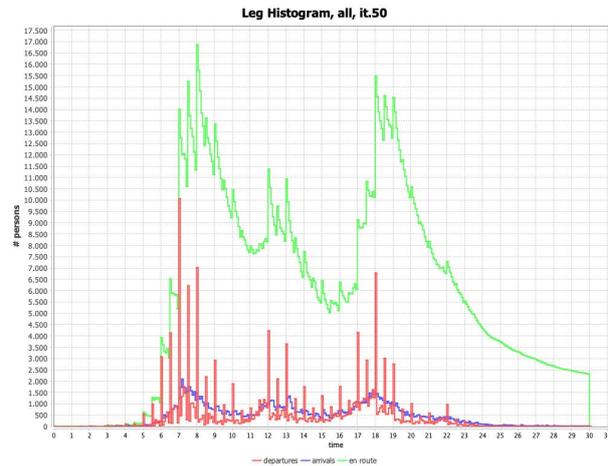
Figura 40: Histograma das viagens



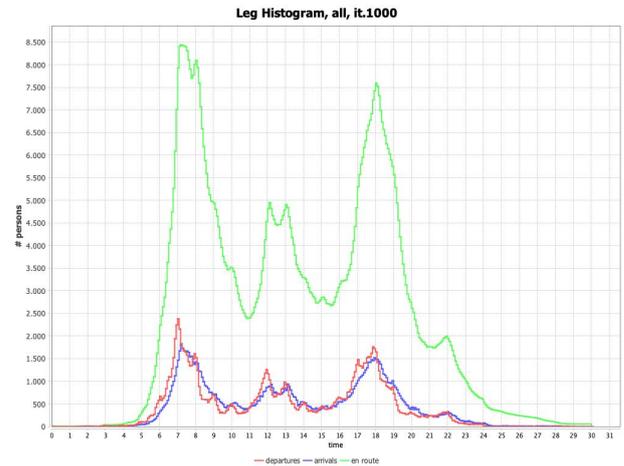
(a) Experimento 1



(b) Experimento 3



(c) Experimento 10



(d) Experimento 31

e destino em domicílios e edificações e atribui os horários de saída com a aplicação da variação de até dez minutos em distribuição normal passa por esse critério.

Além disso, é possível observar que os demais histogramas apresentam diversas variações locais ocasionadas pela atribuição dos horários de saída sem a devida aplicação da variação do horário de saída, e a demora em realizar as viagens pela quantidade exacerbada de demanda para a capacidade viária de determinados trechos.

Os resultados do histograma das viagens simuladas (carros, motocicletas e caronas) no experimento 32 são apresentados na Figura 41 a cada cinco minutos, e na Figura 42 com valores somados por hora, o qual pode ser comparado ao histograma das viagens relatadas pelo PLAMUS, ilustrado na Figura 43.

Figura 41: Histograma das viagens simuladas a cada cinco minutos

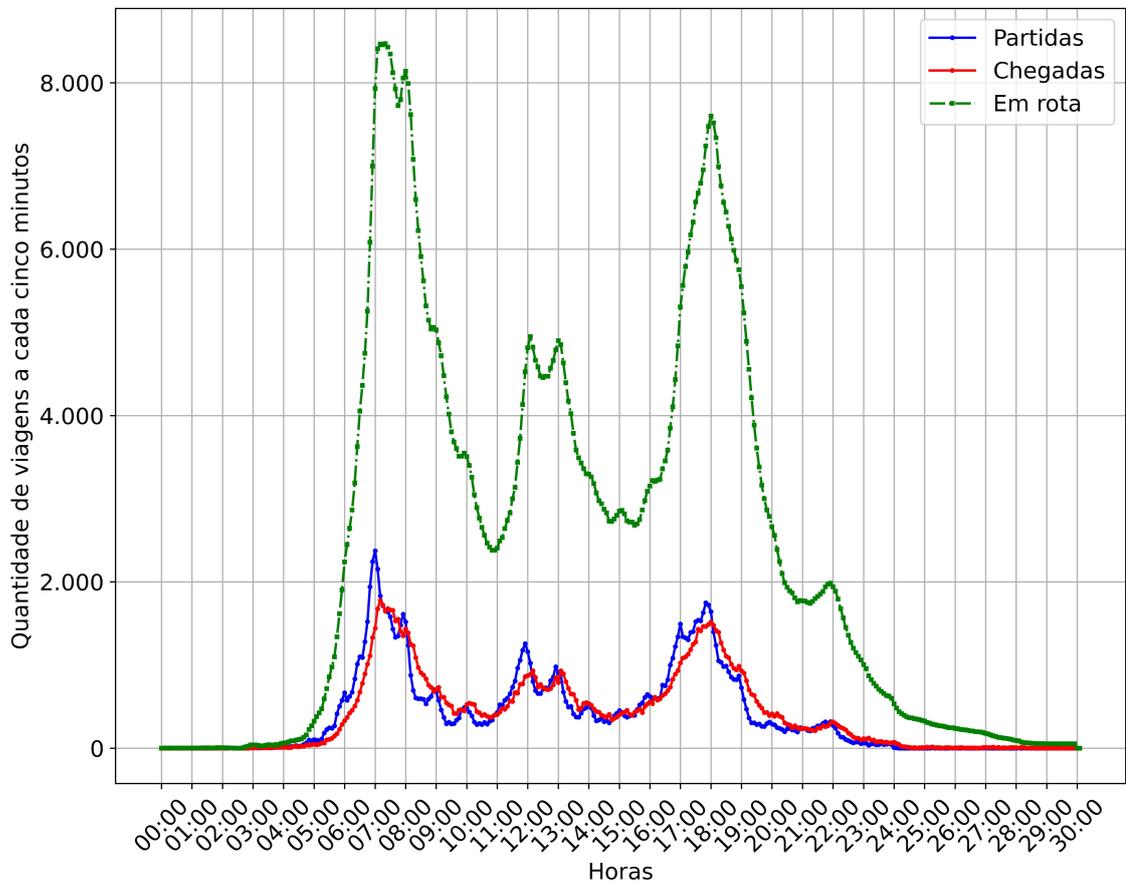


Figura 42: Histograma das viagens simuladas, agregadas por hora

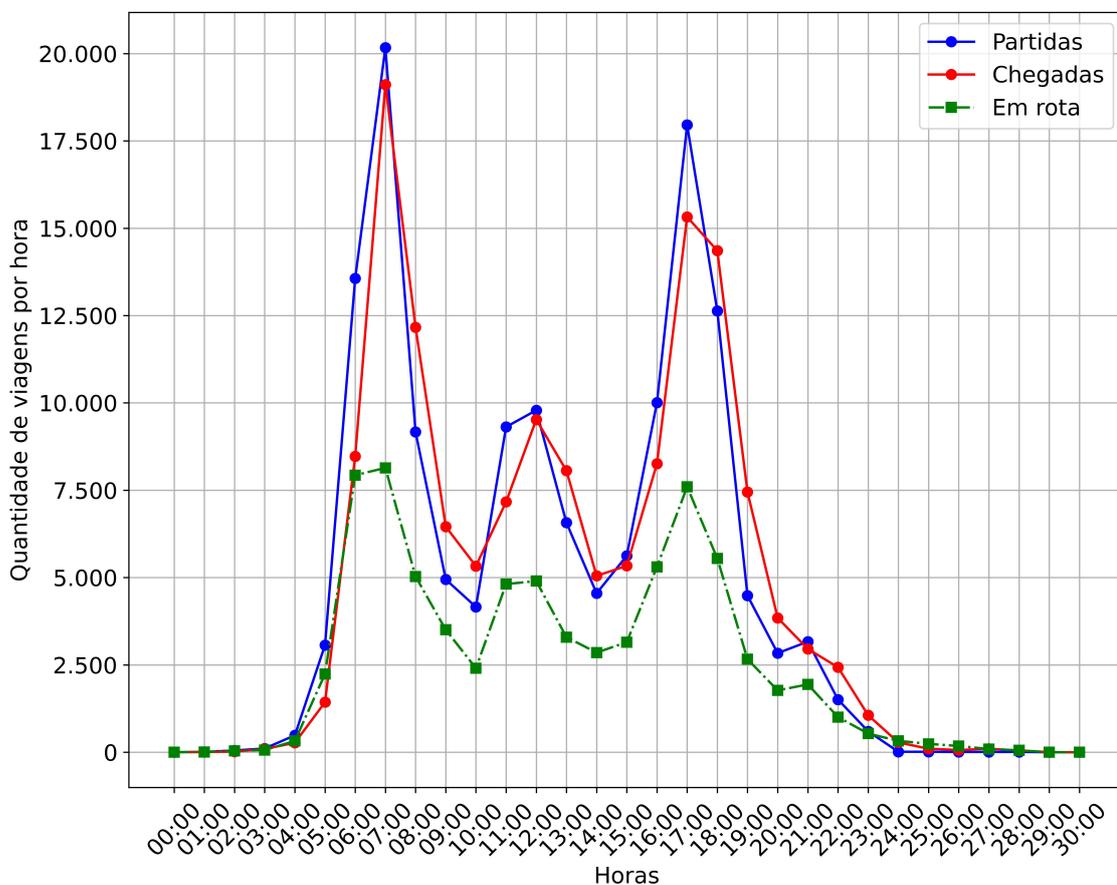
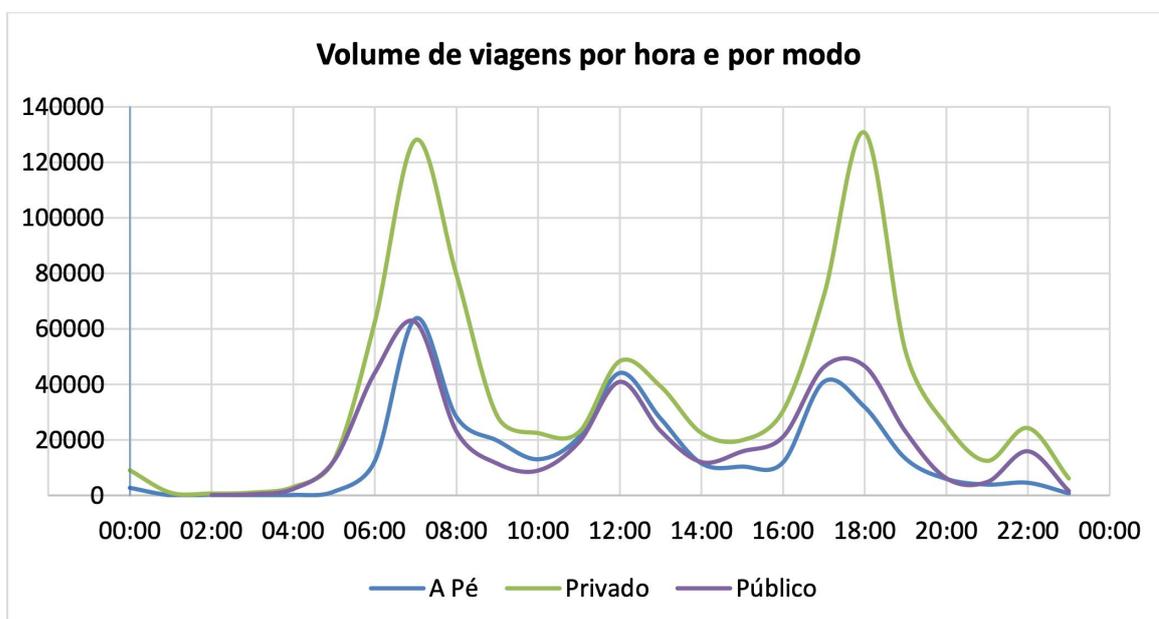


Figura 43: Histograma das viagens relatadas. Fonte: PLAMUS (2014).



O comportamento das viagens simuladas é semelhante ao das viagens relatadas, com horário pico da manhã entre as 7h e 8h, e da tarde entre as 18h e 19h. Ainda, os picos locais também são semelhantes, com destaque para o horário das 12h às 13h e das 22h. Com a dispersão dos pontos de origem e destino nas edificações, e a dispersão dos horários de saída das viagens, nenhum dos agentes simulados na rede ficaram presos na simulação.

5.5 Elaboração da simulação multimodal com transporte coletivo

Essa seção apresenta os resultados obtidos no processo de modelagem sintética das linhas de transporte coletivo, caracterizadas pelo seu itinerário, pontos de ônibus, horários de saída e veículos utilizados, assim como os resultados de uma simulação teste não-calibrada com os dados do transporte coletivo.

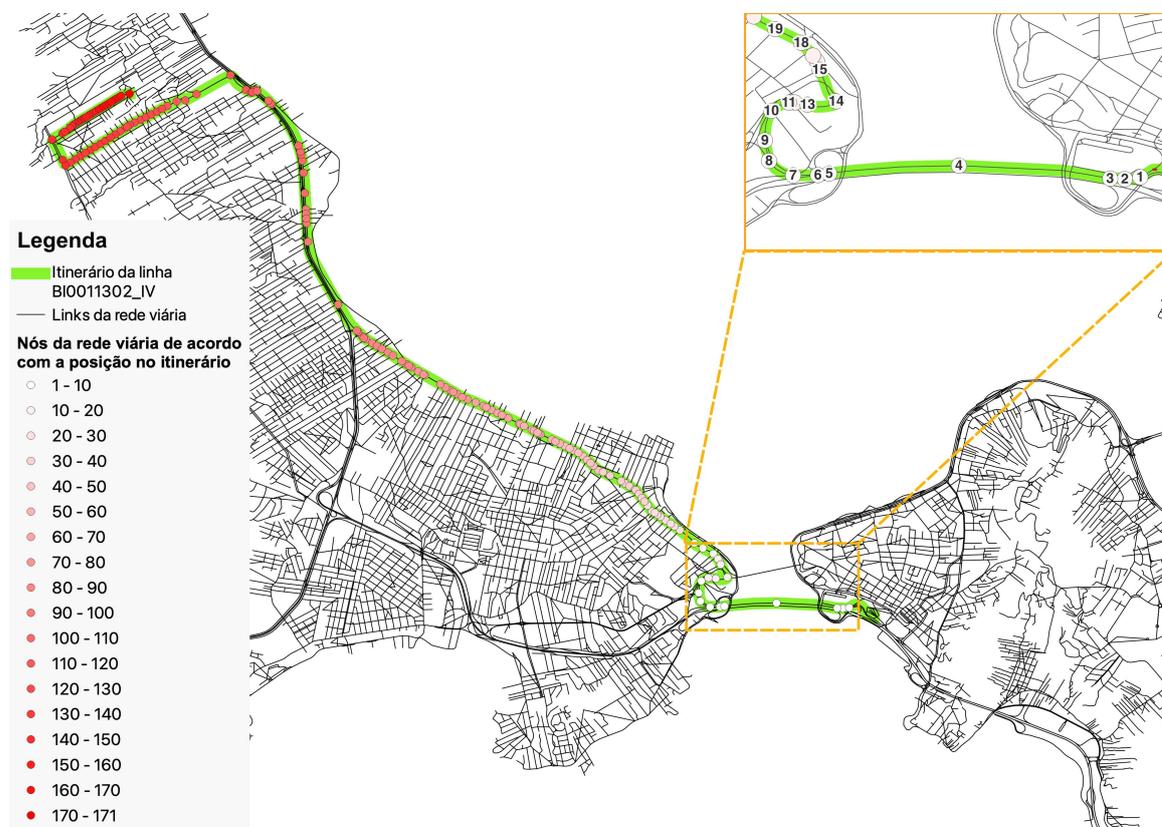
Não existiam todos os dados necessários completos então foram aplicados procedimentos para gerar os dados de forma sintética, seguindo premissas gerais, de forma a aproximar o máximo possível da situação real.

A criação das linhas no MATSim envolve a descrição dos *links* e pontos de ônibus em que essa linha passa, em ordem de chegada. Cada ponto de ônibus deve ser referenciado a um *link* da rede viária.

A indicação dos *links* em ordem é realizada por meio de geoprocessamento. Como o shapefile dos itinerários e da rede viária utilizados nesse estudo foram elaborados pela equipe do PLAMUS, ambos são espacialmente correspondentes. Portanto, foram selecionados os links que estão contidos em um *buffer* de cinco metros a partir do shapefile do itinerário. Para ordenamento dos *links*, foi realizado roteamento entre os nós dos *links* selecionados, partindo do nó inicial do itinerário, considerado como o ponto de início da linha. A Figura 44 ilustra o resultado dessa ordenação no itinerário de uma das linhas do transporte coletivo.

Como não foi possível obter a informação sobre os pontos de ônibus pertencentes a cada linha, foram realizados procedimentos de geoprocessamento para atribuição sintética dos pontos às linhas. Para isso, foram selecionados todos os pontos que estão atrelados a *links* que fazem parte da linha analisada. Esse processo acarreta em erros, dadas às características das linhas de ônibus. As linhas expressas, semi-diretas e diretas não param em todos os pontos intermediários, mesmo que estejam em seu itinerário. No entanto, o método utilizado não diferencia o tipo de operação das linhas, e inclui como parada todos os pontos em seu trajeto. Essa característica também causa

Figura 44: Resultado da ordenação dos nós e *links* de um itinerários



situações em que não são encontrados pontos de ônibus no itinerário da linha. Nesses casos, a linha é removida do modelo.

O método de elaboração dos arquivos de entrada do transporte coletivo foi aplicado por meio da elaboração de um *script*, apresentado no Apêndice E.

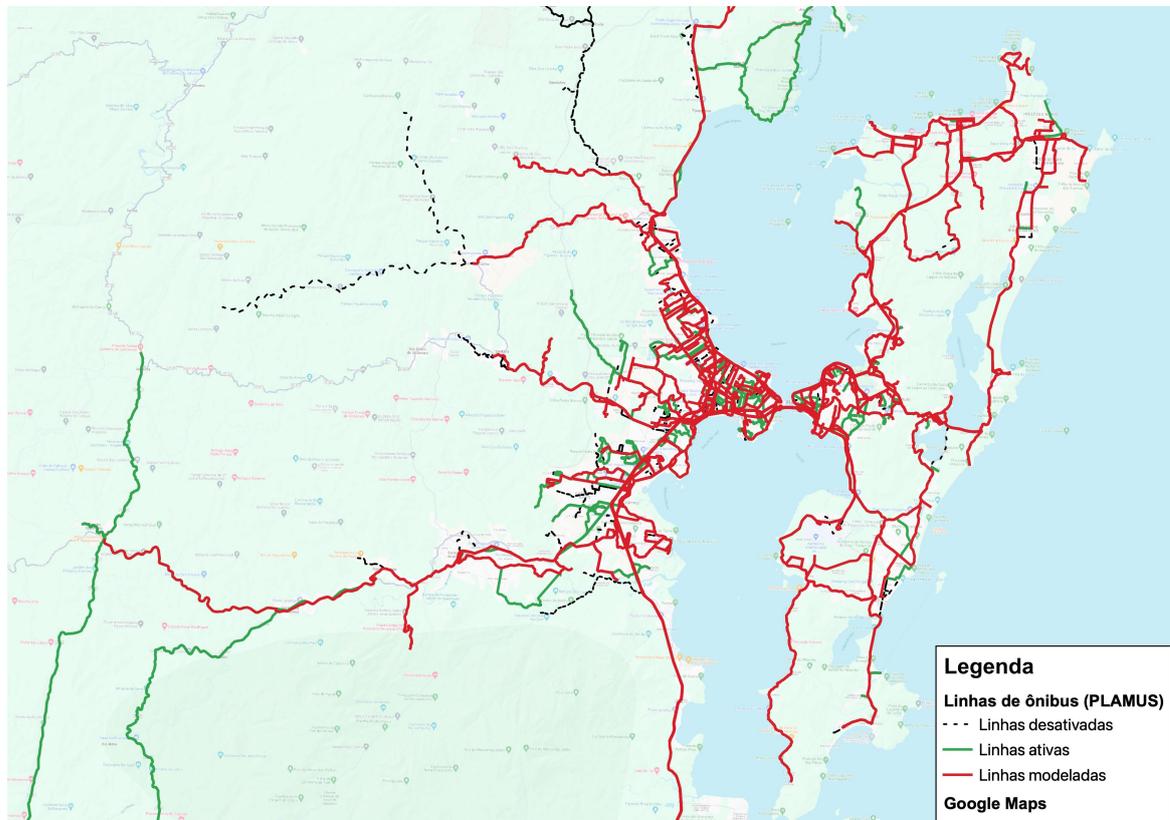
Dada a indisponibilidade de informações sobre as linhas ativas na base de dados do PLAMUS, foram consideradas ativas as linhas que possuem valores não-nulos sobre a frequência de viagens na hora-pico da manhã e/ou da tarde. Das 916 linhas constantes no *shapefile* disponibilizado pelo PLAMUS, 454 foram consideradas ativas. Entre as ativas, 289 linhas foram modeladas (63,6%).

A Figura 45 apresenta o total de linhas na base de dados, as linhas consideradas ativas e as linhas modeladas.

Para definição da quantidade de viagens diárias, e sua frequência, foram utilizados os seguintes critérios, selecionados arbitrariamente:

- a base para o cálculo da quantidade de viagens no dia é a frequência na hora-pico;
- a quantidade de viagens durante uma hora-pico representa 10% da quantidade de viagens diárias;

Figura 45: Linhas de ônibus existentes, ativas e modeladas



- cada linha opera durante dezenove horas, das quais quatro são consideradas horas-pico (das 7h00min às 9h00min, e das 17h00min às 19h00min);
- O horário exato de início varia de acordo com uma função normal que adiciona ou subtrai até quinze minutos das 5h30min da manhã.

Para definição da frequência fora do pico, é dividida a quantidade de viagens diárias fora da hora-pico, calculada conforme lista anterior, pelas suas quinze horas de operação.

Caso o horário de início da viagem ocorra dentro do período de pico, a próxima viagem ocorrerá na frequência definida deste período. Caso contrário, é adicionado o tempo entre viagens fora do pico ao horário de início da viagem anterior.

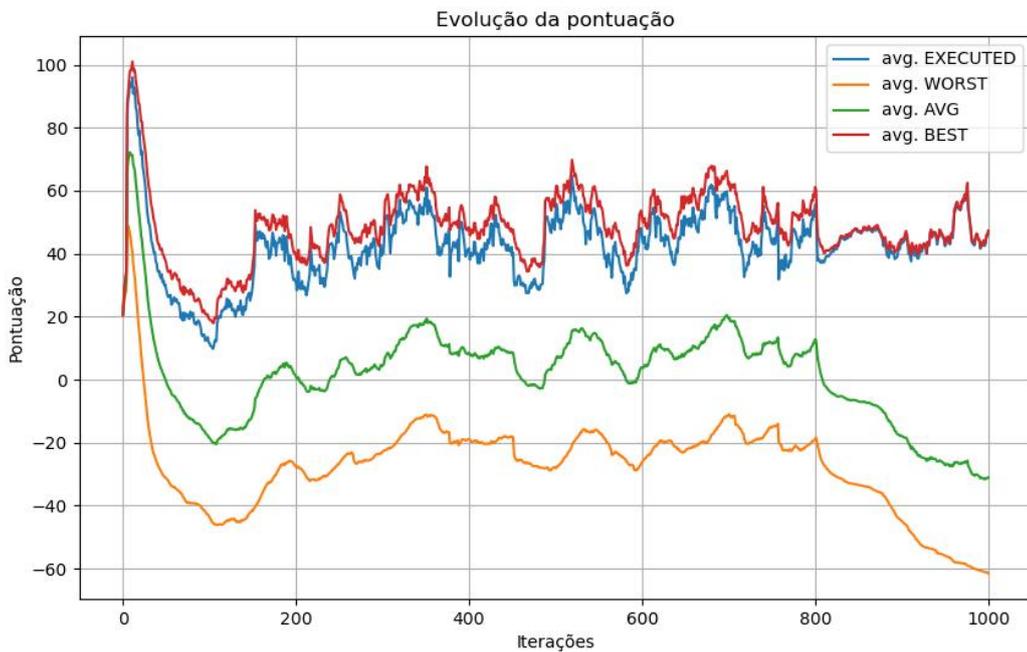
Para definição do tempo de viagem entre os pontos de ônibus, foi realizada uma interpolação linear entre a quantidade de pontos existentes e o tempo de viagem da linha indicado no *shapefile* do PLAMUS.

Finalizada a modelagem do transporte coletivo, foi realizada uma simulação teste para analisar os resultados obtidos com uma simulação multimodal. Como configurações do transporte

coletivo, foi utilizado o algoritmo de roteamento *SwissRailRaptor*, indicado por Horni, Nagel e Axhausen (2016), a busca por pontos de ônibus a partir do início ou fim da viagem é de até 10 km, aumentada em até 1km caso nenhum ponto seja encontrado, e a distância entre pontos para ser considerada uma integração entre linhas do transporte coletivo é de 1 km.

A Figura 46 apresenta a evolução da pontuação média dos agentes desse experimento.

Figura 46: Evolução da pontuação média dos agentes no experimento com transporte coletivo



A evolução da pontuação ocorre no mesmo padrão apresentado pelo experimento 32, sem o transporte coletivo, mas com cerca de 20 unidades de pontuação abaixo das pontuações no experimento 32. Como os parâmetros de pontuação do ônibus, e seus valores, são exatamente os mesmos do automóvel, a repetição do padrão era esperada. Como o único parâmetro não zerado da pontuação dos modos é a utilidade marginal do tempo de viagem ($\beta_{trav,mode(q)}$), e entre os parâmetros gerais de pontuação é considerado a espera pelo ônibus no ponto de parada ($waitingPt$) igual a -6, e a utilidade de integração entre linhas ($utilityOfLineSwitch$) igual a -1, é esperado que as viagens de ônibus tenham uma pontuação menor do que quando não eram simuladas na rede.

Isso porque as viagens tendem a demorar mais do que o tempo simulado fora da rede por meio de velocidade média do veículo e porque, dada a incapacidade de modelagem de toda a oferta do transporte coletivo, os usuários tendem a esperar mais nos pontos de parada por seus ônibus

(que podem já estar lotados) e tendem a realizar mais integração entre linhas do que a realidade pelo esparsa atendimento espacial das linhas modeladas.

A Figura 47 ilustra o histograma de todas as viagens, e as Figuras 48, 49 e 50 ilustram o histograma das viagens de carro, de ônibus e a pé, respectivamente.

Figura 47: Histograma de todas as viagens realizadas no experimento com transporte coletivo

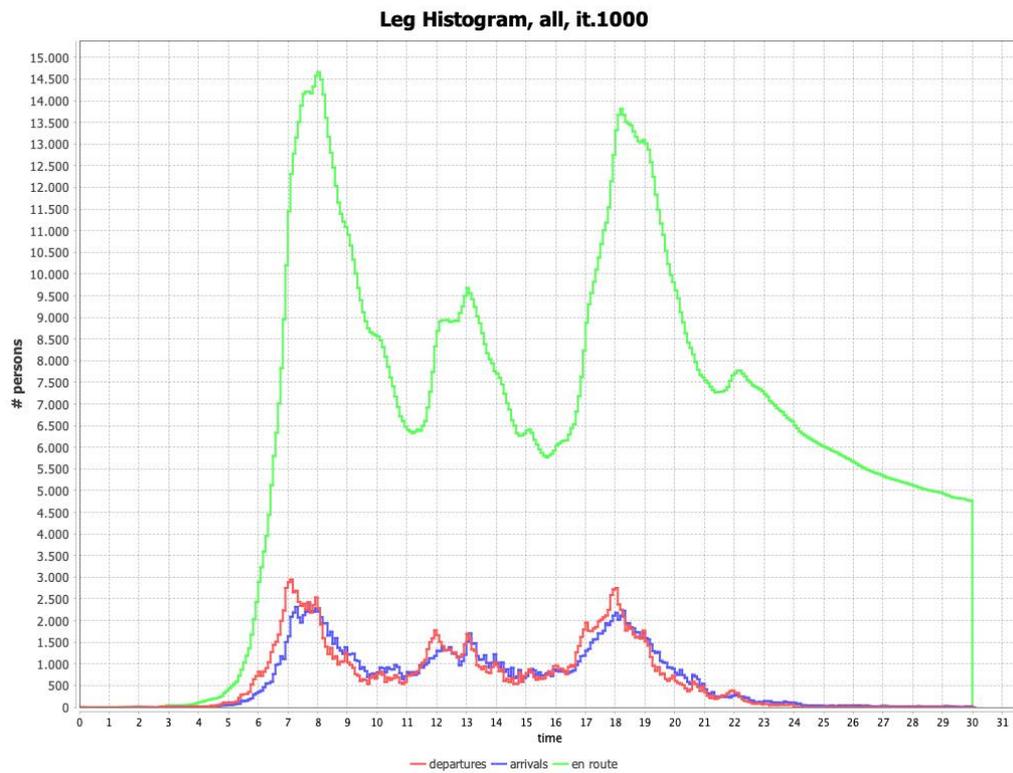


Figura 48: Histograma das viagens de carro realizadas no experimento com transporte coletivo

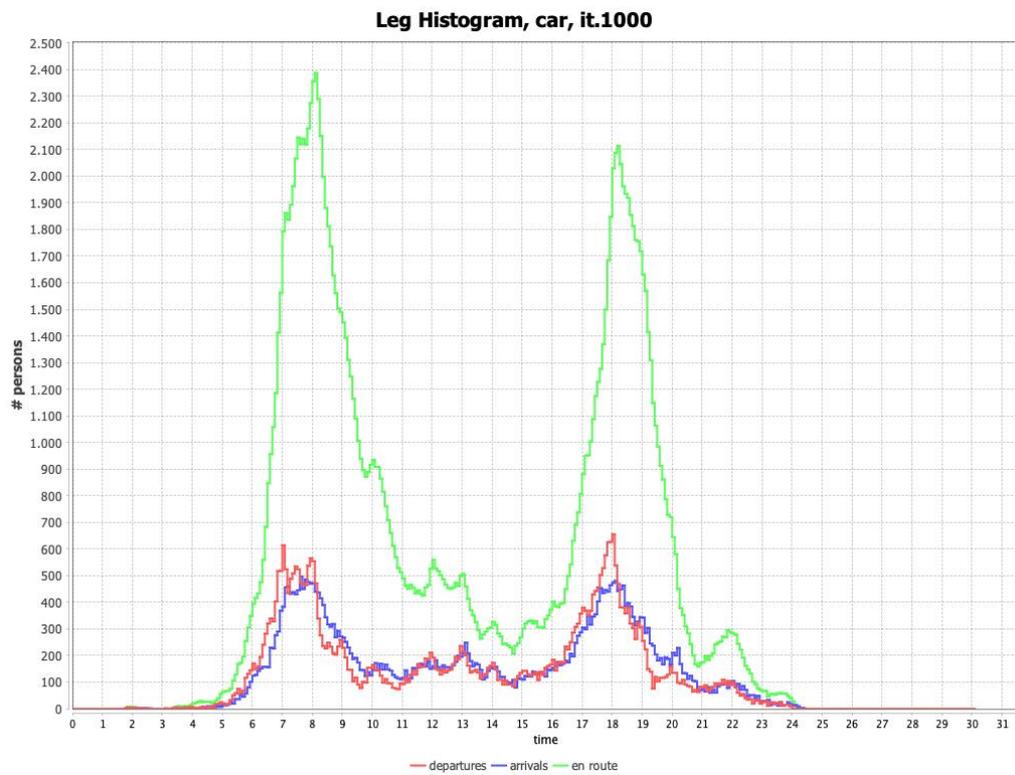


Figura 49: Histograma das viagens de ônibus realizadas no experimento com transporte coletivo

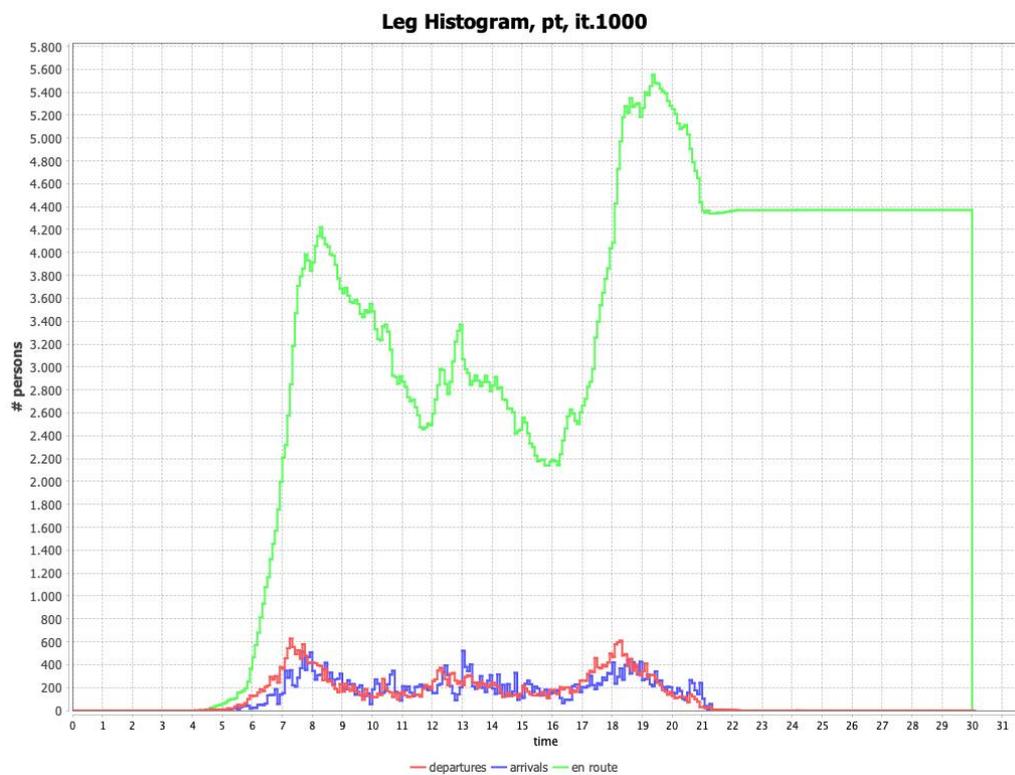
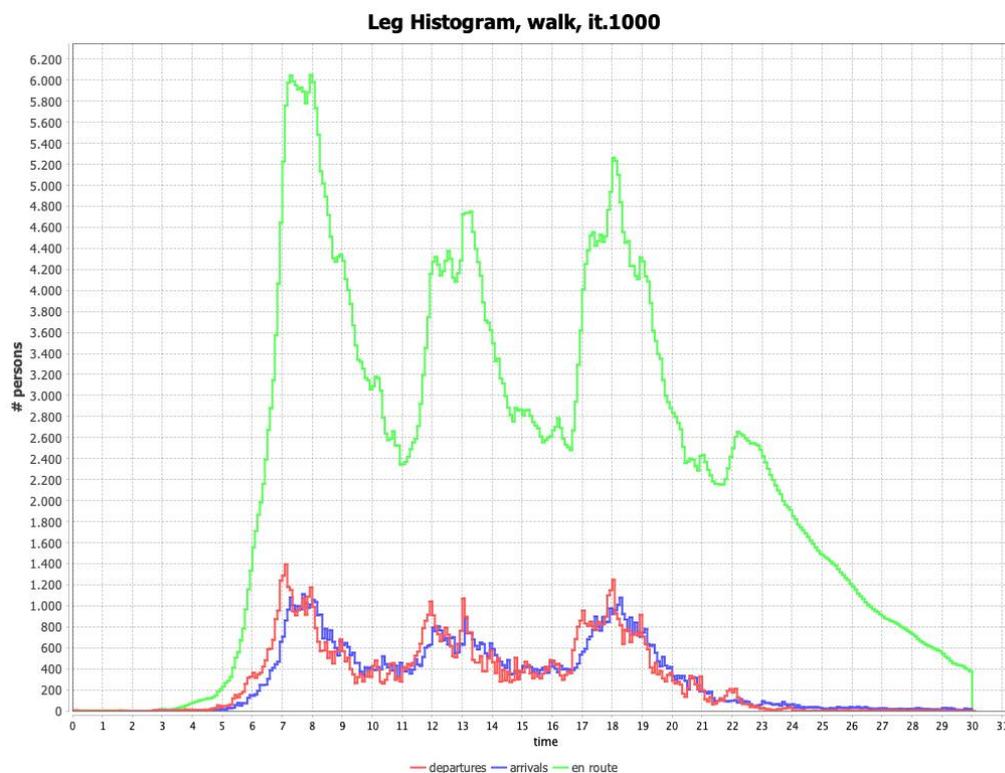


Figura 50: Histograma das viagens a pé realizadas no experimento com transporte coletivo



Como os agentes não foram permitidos a realizar a migração modal nesse experimento, a divisão modal se mantém constante ao longo de todas as iterações. No entanto, devido à necessidade de realização de viagens a pé para chegar até os pontos de parada, são realizados mais trechos de viagens a pé, chamadas *legs* no MATSim.

Comparando os histogramas de cada modo de transporte dos experimento sem e com a alocação do transporte coletivo na rede, é possível perceber que:

- o histograma das viagens de carro quase não foi alterado;
- o histograma das viagens de ônibus apresenta quantidade inferior de saídas e chegadas no experimento em que o transporte coletivo é alocado na rede;
- a quantidade de agentes presos nas viagens de ônibus passou de zero (no experimento 32) para 4.400 agentes no experimento em que o transporte coletivo é alocado na rede;
- o histograma das viagens a pé apresenta cerca de três vezes mais saídas, chegadas e pessoas em deslocamento no experimento em que o transporte coletivo é alocado na rede;

- o padrão de viagens a pé ao longo do dia é alterado no experimento em que o transporte coletivo é alocado na rede, passando a ter três períodos de pico, de diferentes intensidades, com o período do meio-dia sendo mais movimentado do que o período do final da tarde.

Sobre essas análises, é possível concluir que a falta de capacidade do transporte coletivo, causada pela modelagem de apenas 63,6% das linhas de ônibus, pode ter impedido alguns agentes de iniciarem as suas viagens no transporte coletivo, assim como de as finalizarem, causando os agentes presos e menor número de viagens do que o esperado.

Com a alocação do transporte coletivo na rede, a quantidade de trechos de viagens a pé aumentou muito, e o padrão de viagens ao longo do dia passou a se assemelhar menos ao histograma das viagens relatadas ao PLAMUS, ilustrado na Figura 43.

5.6 Inserção do módulo *ChangeMode* na simulação multimodal

Na simulação multimodal, a calibração da divisão modal é de extrema importância para as simulações de cenários futuros onde se deseja avaliar a influência na migração modal. Apesar da impossibilidade de calibração e validação do modelo multimodal de que trata esse estudo, essa seção visa apresentar os resultados de uma primeira simulação e analisar os resultados sob a ótica da calibração de um modelo multimodal.

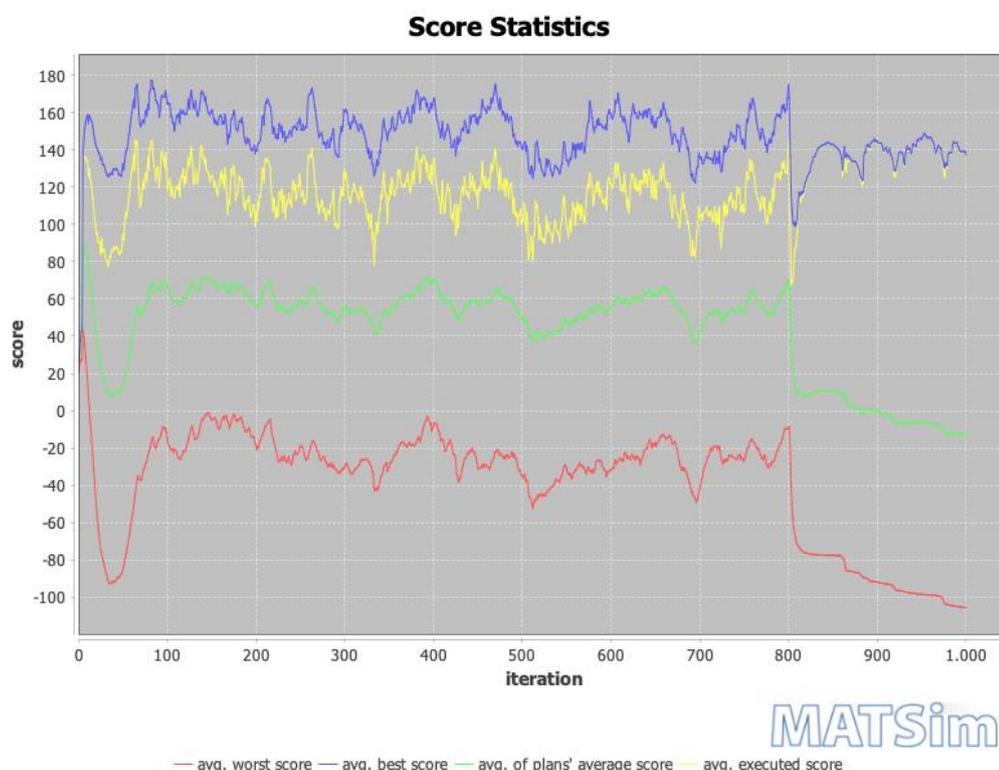
Para essa simulação, foram utilizados todos os mesmos parâmetros e valores do que a simulação apresentada na Seção 5.5, com a inclusão do parâmetro *ChangeSingleTripMode*, no módulo *strategy* do arquivo de configuração. O peso estabelecido para o parâmetro é de 0,1, com a respectiva redução do parâmetro *SelectExpBeta*, de 0,8 para 0,7.

Além disso, foi inserido o módulo *ChangeMode*, cujos valores selecionados para os parâmetros são:

- ignorar disponibilidade de carro (*ignoreCarAvailability*): verdadeiro (*true*);
- comportamento de migração modal (*modeSwitchBehavior*): de modos específicos para modos específicos (*fromSpecifiedModesToSpecifiedModes*), considerado o padrão do MATSim;
- modos em que é permitida a migração modal (*modes*): carros, transporte coletivo e motocicletas, os únicos modos alocados na malha viária.

A Figura 51 apresenta a evolução da pontuação média dos agentes.

Figura 51: Evolução da pontuação média dos agentes no experimento com transporte coletivo e migração modal



Com a inclusão de 10% dos agentes terem a possibilidade de realizar migração modal entre carros, motocicletas e ônibus, a evolução da pontuação apresentou maior instabilidade, e a diferença entre a melhor pontuação média e a pior pontuação média foram acentuadas em relação ao experimento anterior. O aumento da instabilidade está vinculado ao fato de que um agente agora pode ter em sua memória um plano em que usa o ônibus em local não atendido pelas linhas modeladas, causando viagem de alta duração, e um plano em que realiza esse trajeto de carro ou motocicleta, que tendem a pontuar melhor por serem mais rápidos.

Além disso, o próprio aumento da porcentagem de agentes que podem realizar inovações em seus planos em contraste com a redução na porcentagem de agentes que apenas escolhe o melhor plano entre os listados em sua memória tende a causar aumento na instabilidade da pontuação. Considerando que apenas os agentes que realizam viagens em carros e motocicletas estão sujeitos à pontuação do Cadyts, outro aspecto que pode ter impactado a instabilidade é o aumento de agentes que realizam viagens por esses modos e, portanto sofrem a influência de peso 100 do Cadyts.

A evolução da divisão modal ao longo das iterações é ilustrada na Figura 52, e a Tabela 15 apresenta a porcentagem de participação dos modos e a duração média de todas as viagens em

algumas iterações de destaque: iteração inicial (0), iteração antes da inicialização do Cadyts (5), iteração após o início da estabilização da pontuação (100), iteração mediana (500), iteração em que a inovação nos planos é desativada (800) e última iteração (1000).

Figura 52: Divisão modal das viagens no experimento com transporte coletivo e migração modal

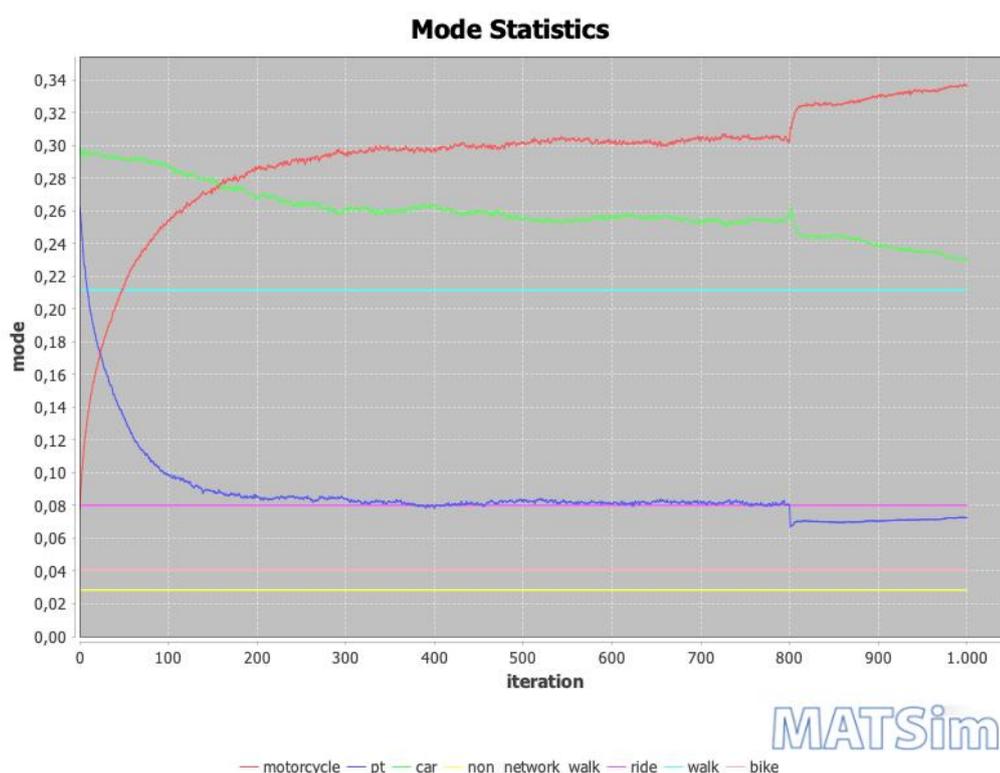


Tabela 15: Porcentagem de participação dos modos e duração média das viagens

Iter.	Carro	Motocicleta	TC	Bicicleta	A pé	Carona	Outros	Duração média
0	30%	8%	26%	4%	21%	8%	3%	00:53:02
5	30%	12%	23%	4%	21%	8%	3%	00:39:31
100	29%	25%	10%	4%	21%	8%	3%	00:36:47
500	26%	30%	8%	4%	21%	8%	3%	00:37:43
800	26%	30%	8%	4%	21%	8%	3%	00:36:51
1000	23%	34%	7%	4%	21%	8%	3%	00:31:02

Na evolução da divisão modal ao longo das iterações, houve redução não só nas viagens realizadas de ônibus (-19% de viagens totais), mas também nas viagens realizadas de carro (-7%), com aumento das viagens de motocicleta (+26%). A redução na quantidade de viagens de

transporte coletivo era esperada, dada a incapacidade de modelagem de todas as linhas ativas do sistema de transporte coletivo. Ainda, considerando apenas o custo de duração das viagens, e com igual valor para todos os modos, a pontuação das viagens de ônibus tendem a pontuar abaixo dos outros modos, influenciando na migração para modos individuais.

A migração modal dos agentes que usavam o carro e passam a usar a motocicleta ocorre de forma mais gradual do que as viagens realizadas de ônibus, estabilizando aos 26% (-4% do que o inicialmente modelado) e passam por uma nova redução de 3% após a 800ª iteração. A única diferença entre a modelagem do carro e da motocicleta é o fator de equivalência ao veículo passeio, ou unidade veículo padrão (UVP), igual a 1 para os carros e igual a 0,33 para as motocicletas (conforme apresentado na Seção 5.2).

Nesse sentido, a conclusão é que os agentes buscam por alternativas que geram melhorias nas condições viárias apesar de não haver nenhum ganho direto para o próprio agente. O ganho ocorre de forma indireta, ao demandar menos da capacidade viária e assim, causar menos congestionamentos e menor tempo de viagem nos *links* da rede. A redução no tempo médio de viagens entre a 800ª e a 1000ª iteração corrobora essa hipótese.

A Figura 53 ilustra o histograma de todas as viagens, e as Figuras 54, 55 e 56 ilustram o histograma das viagens de carro, de ônibus e a pé, respectivamente.

Figura 53: Histograma de todas as viagens realizadas no experimento com transporte coletivo com migração modal

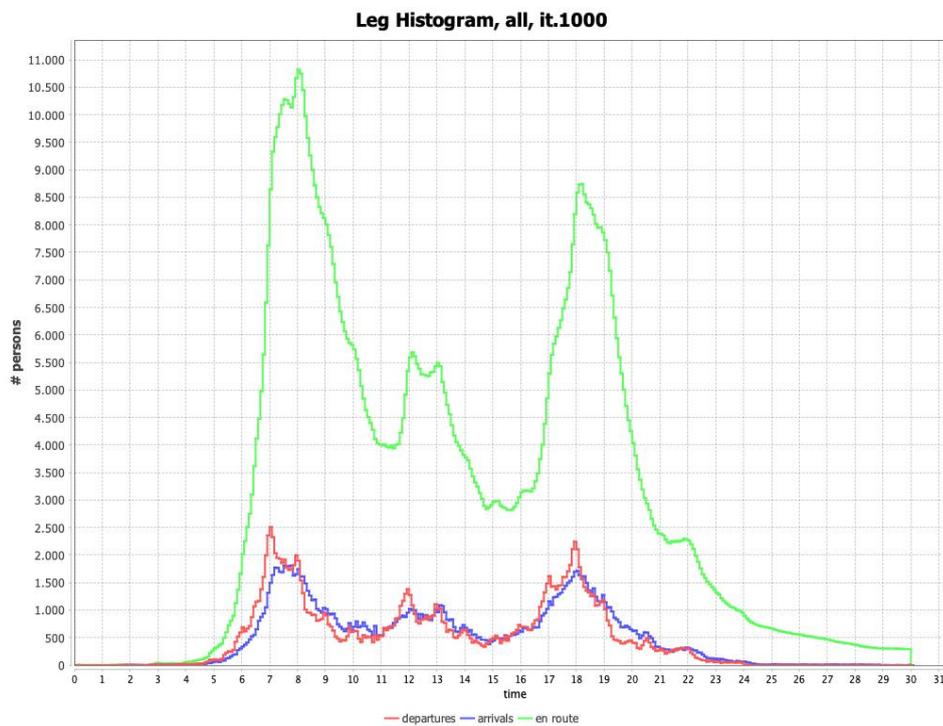


Figura 54: Histograma das viagens de carro realizadas no experimento com transporte coletivo com migração modal

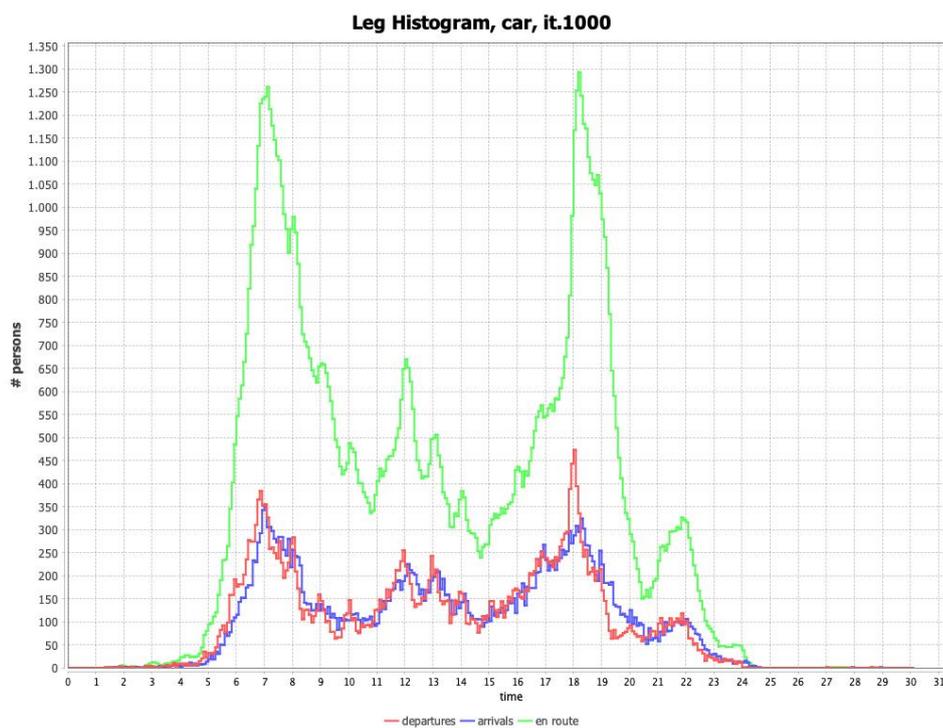


Figura 55: Histograma das viagens de ônibus realizadas no experimento com transporte coletivo com migração modal

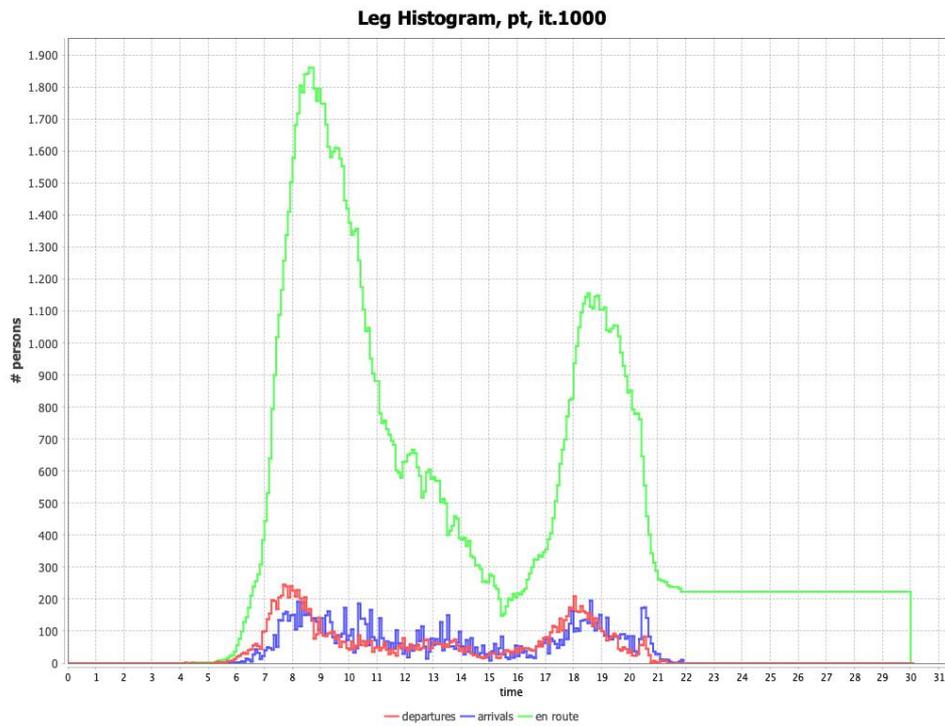
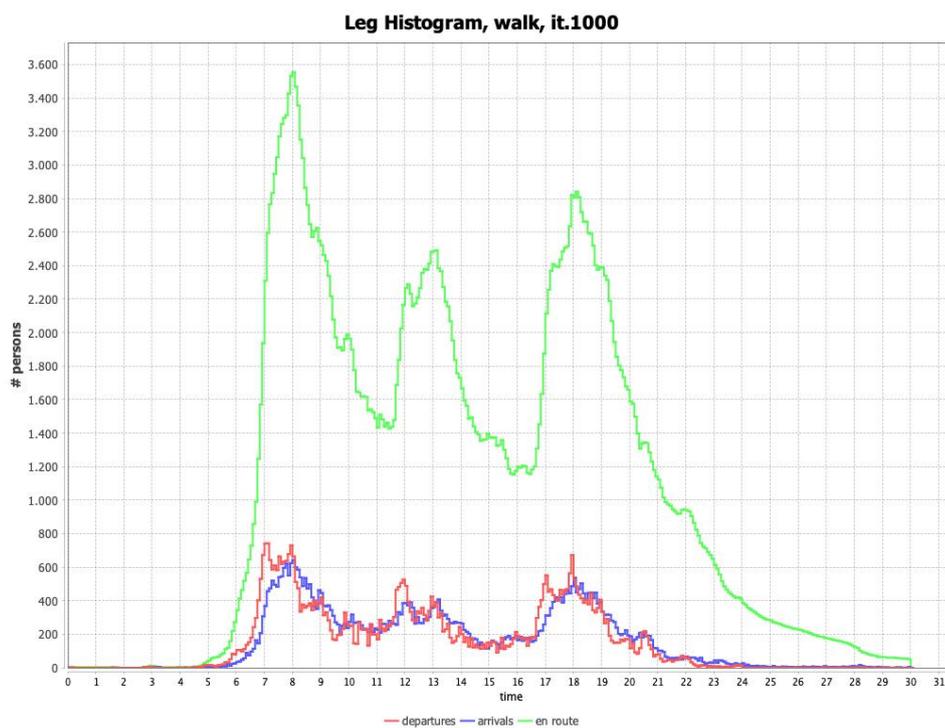


Figura 56: Histograma das viagens a pé realizadas no experimento com transporte coletivo com migração modal



A quantidade de agentes presos diminuiu, o padrão de viagens total permanece o mesmo do que nos experimentos 32 e com transporte coletivo. Já em relação ao histograma de viagens de carro, o padrão de viagens do experimento com migração modal possui picos mais acentuados, principalmente nos picos intermediários no meio-dia e entre as 21h e 22h, o que se assemelha mais ao padrão de viagens apresentado no PLAMUS (vide Figura 43).

O histograma das viagens de ônibus tiveram seu padrão alterado, passando a apresentar dois horários de pico, onde o pico da manhã apresenta mais viagens realizadas do que o pico da tarde, o oposto que acontece no experimento só com transporte coletivo, e com o mesmo padrão das viagens no experimento 32, com exceção dos picos do meio dia e noite. A quantidade de agentes presos diminuiu de cerca de 4.400 agentes presos no experimento com transporte coletivo, para 200 no experimento com migração modal.

A modelagem de apenas 63,6% das linhas de ônibus afeta a demanda em proporção maior do que esse valor. Isso pode ser ocasionado pela falta de linhas que, ao se integrarem a outras, eram capazes de realizar a conexão entre origem e destino dos agentes. A falta dessa integração ocasiona tempos de viagem maiores e, conseqüentemente, a migração para modos mais rápidos. Ainda, a utilização de linhas que não correspondem ao uso real dos agentes, ocasionam a superlotação dos veículos, o que leva a altos tempos de espera por veículos nos pontos de parada (já que veículos com capacidade de passageiros esgotada pela demanda não admitem mais embarques).

Caso houvesse a modelagem real dos dados, e os resultados obtidos fossem os mesmos apresentados nessa seção, os seguintes procedimentos deveriam ser realizados para calibração do modelo:

1. redução da constante de modo da motocicleta, para redução de sua participação modal;
2. aumento da constante de modo do transporte coletivo, para aumento de sua participação modal;
3. redução dos pesos das estratégias de inovação, e aumento do peso da estratégia de seleção de planos, para aumento da estabilidade da pontuação.

A partir dos resultados de uma simulação com essas alterações, o processo é replicado de forma a chegar em valores dos parâmetros dos modos de transporte que gerem a divisão modal modelada inicialmente.

6 Conclusões

A principal contribuição do trabalho é apresentar as possibilidades e dificuldades da análise do transporte de passageiros em áreas urbanas por meio da elaboração de um modelo de simulação de tráfego multimodal baseado em agentes utilizando apenas *softwares* livres e dados abertos ao público. Ao longo do estudo, foi possível concluir que o MATSim é uma ferramenta capaz de reproduzir sistemas de transporte de larga escala por meio da microssimulação, assim como é capaz de produzir resultados que auxiliem na avaliação dos impactos de medidas de incentivo à migração modal.

O *software* impõe alguns desafios ao modelador, como a grande quantidade de parâmetros que podem ser alterados no processo de calibração. Sem um modelo estatístico que defina previamente os parâmetros da pontuação, o processo de calibração segue um ciclo de tentativa e erro a partir dos parâmetros padrão indicados pelo MATSim. Conforme apresentado na seção 2.5, nem todos os estudos apresentam seu método de calibração e, quando apresentados, não há consenso entre os autores sobre o melhor método a ser aplicado. Nesse sentido, esta pode ser considerada a principal dificuldade no que concerne ao uso do MATSim.

Na elaboração do modelo, foi evidente a dependência da disponibilização de dados, que deve ser feita pelos órgãos públicos responsáveis pela gestão dos sistemas de transporte na área de estudo. A disponibilidade dos dados relacionados à rede viária, à demanda de viagens e à oferta do transporte coletivo por meio PLAMUS, dos dados das edificações pela Prefeitura de Florianópolis, assim como a indisponibilidade de dados sobre os pontos de parada e quadros de horários do transporte coletivo, moldaram o método desse estudo, como comumente ocorre nos modelos de planejamento de transportes urbanos.

Apesar da existência de dados de rede viária advindos de outras fontes, as simulações que utilizaram os dados disponibilizados pelo PLAMUS apresentaram melhor desempenho do que aquelas que utilizaram o OSM. Além disso, os resultados de pesquisas de campo, como a Pesquisa Domiciliar de Origem e Destino, contagens de tráfego e pesquisa de velocidade, não poderiam ser obtidos de outra forma senão por meio de órgãos públicos. Esses dados são imprescindíveis para a construção e validação do modelo.

Os dados do OSM possuem as informações mínimas para modelagem da rede viária. Considerando a facilidade de conversão desses dados para o formato requerido pelo MATSim, por meio

do JOSM, caso não houvesse incompatibilidade temporal entre os dados, essa alternativa poderia se tornar a mais interessante para o estudo em questão.

Como o modelo de simulação do PLAMUS foi aplicado no TransCAD, cuja demanda é inserida de forma agregada (por zonas de tráfego), foi evidenciada a necessidade de ajustes nos dados da demanda para entrada no MATSim. Nos testes com a utilização dos centroides como pontos de origem e destino, em comparação com o uso das edificações, o segundo apresentou r^2 de volume diária cerca de 0,4 a mais que o primeiro, e com menos de 0,05% de agentes presos. Portanto, é possível concluir que para a correta modelagem da demanda no MATSim é necessária a inserção de dados das edificações dentro da mesma ZT como origem e destino nos planos de viagem gerados a partir de cada pessoa entrevistada, assim como a atribuição de diferentes horários de saída (considerando um intervalo máximo de variação do horário de saída).

Apesar de não ter utilizado as restrições de movimento em cada interseção, ou os controles de tráfego existentes, o modelo foi capaz de representar a dinâmica de fluxo na área de estudo. O uso do Cadyts teve influência decisiva na adequação da alocação do tráfego na rede de acordo com as contagens de tráfego auferidas no sistema de análise.

Para o transporte coletivo, o nível de deficiência nos dados fez com que os procedimentos incluíssem a eliminação de uma quantidade tão grande de dados que não pode ser mais esperado que o modelo represente a realidade, além da utilização de parâmetros determinados de forma arbitrária. Esse estudo recorreu à modelagem de apenas 63,6% da oferta de linhas do transporte coletivo, com a imputação de pontos de ônibus e horários de saída de forma sintética segundo os critérios estabelecidos na Seção 5.5. Isto teve um impacto direto nos resultados de validação para o modelo multimodal. No entanto, foi realizada uma simulação não calibrada, cujos resultados apresentam cerca de 4.400 agentes presos nas viagens de transporte coletivo.

Para a avaliação da migração modal em cenários alternativos, é preciso integrar as etapas de divisão modal e alocação de viagens, permitindo que o agente decida entre os diferentes modos de transportes disponíveis. O último experimento apresentado nesse estudo contou com a possibilidade de 10% dos agentes realizarem migração modal entre carro, motocicleta e transporte coletivo em cada iteração, como forma de demonstração dos próximos passos a serem realizados na calibração de um modelo multimodal que vise à avaliação da migração modal.

Naturalmente, as viagens realizadas no transporte coletivo possuem maior distância e maior duração de viagem do que os modos individuais motorizados, causando maior interesse dos agentes

nos modos mais rápidos. Com a modelagem de apenas 63,6% das linhas, essa situação tende a se intensificar. Considerando pontuações iguais para todos os modos, o modelo se demonstrou descalibrado. Isso porque foi identificado que as viagens de motocicleta passaram a representar mais 26% das viagens totais do que o inicialmente modelado, cuja demanda foi transferida das viagens de transporte coletivo, que passou a representar -19% de viagens totais, e das viagens de carro, que passaram a representar -7% das viagens totais.

Mesmo que o modelo não esteja representando corretamente a realidade, já que não foi possível calibrá-lo, é interessante notar como os resultados da simulação tendem para um cenário que tem se agravado nos municípios brasileiros, em especial no norte e nordeste no Brasil: a tendência de aumento do uso das motocicletas. No entanto, as razões pelas quais o uso da motocicleta tem se intensificado são os baixos custos de aquisição e sua rapidez, já que consegue trafegar entre corredores de trânsito, não estando sujeitas à mesma capacidade viária que os carros. O modelo não representa nenhuma dessas questões, apenas considerando que a motocicleta ocupa menos espaço na via (com UVP de 0,33) e esse fator isolado é capaz de aumentar sua utilidade de forma a aumentar seu uso no modelo em detrimento dos demais modos.

Seguindo a lógica dos municípios brasileiros da priorização dos modos individuais motorizados, é possível concluir que a realização de simulações de tráfego com alocação apenas de automóveis e motocicletas apresenta maior facilidade em relação à disponibilidade de dados. É raro nos municípios brasileiros que existam cadastros completos e adequados, principalmente em relação à simulação dos transportes coletivos.

Por fim, conclui-se que o objetivo do estudo foi cumprido, com a exploração e descrição dos desafios encontrados na elaboração de um modelo de simulação de tráfego multimodal baseado em agentes com a utilização do MATSim. O MATSim se demonstrou desafiador pela quantidade de parâmetros a serem configurados, mas foi capaz de permitir um aprofundamento da análise da influência dos parâmetros nos resultados do modelo e uma exploração do processo de alocação de viagens com e sem integração dessa etapa com a divisão modal. A inclusão dessa integração é essencial para a análise da migração modal em cenários alternativos, e não é realizada de forma simples em modelos macroscópicos, como os modelos desenvolvidos no TransCAD. A utilização do MATSim em detrimento dos *softwares* de uso comercial permitiu demonstrar a possibilidade de uso de ferramentas gratuitas para microsimulações em áreas de larga escala. Dado sua licença aberta, com a sua utilização por mais pesquisadores, estão atualmente sendo desenvolvidas novas ferramen-

tas para calibração automatizada, para visualização dos resultados e espera-se um aprimoramento do *software* como um todo

Os esforços desse trabalho se concentraram na elaboração de um modelo que sirva como base para a análise do impacto de medidas na migração modal em cenários futuros. A partir do que foi apresentado, a próxima seção expõe as recomendações para efetivação dessa análise, além de outras análises adicionais e melhorias na construção do modelo.

7 Recomendações para trabalhos futuros

A partir do último experimento realizado e com a inserção da real oferta do transporte coletivo, deve ser realizada a calibração e validação do modelo. Caso os resultados se assemelhem aos indicados na Seção 5.6, para a correta calibração do modelo, os modos motorizados individuais devem ser penalizados em maior grau do que o transporte coletivo, em especial as motocicletas.

Com a calibração e validação do modelo multimodal, é possível elaborar cenários alternativos com alterações de infraestruturas viárias ou com alterações operacionais no transporte coletivo. A primeira, realizada no arquivo *network.xml*, pode incluir, por exemplo, a criação de vias exclusivas para ônibus, com a criação de novos *links* em que apenas os ônibus são permitidos, ou a restrição de carros e motocicletas no centro histórico do município, retirando esses modos como “permitidos” nos *links* desejados. A segunda, realizada no arquivo *transitSchedule.xml*, pode incluir novas linhas de ônibus, ou aumentar a frequência das viagens. Também podem ser testados veículos com maior capacidade, alterando o arquivo *transitVehicles.xml*, ou diferentes valores tarifários, por meio da indicação de um valor para o parâmetro *dailyMonetaryConstant* para tarifas mensais, ou para a constante do modo (β_{mode}) para tarifas únicas a cada viagem de ônibus.

Os resultados obtidos após a realização dessas alterações representaria o comportamento da população na ocorrência destas alterações na vida real. Portanto, seria possível identificar a porcentagem de migração modal para cada modo de transporte, os novos tempos de viagem, a existência de congestionamentos, e parâmetros de qualidade e produtividade do sistema como um todo, a fim de avaliar as novas alterações.

Por fim, outros procedimentos podem ser realizados para melhorar o modelo. São eles:

- utilizar modelo estatístico para calibração da função pontuação;
- implantar processo metodológico automatizado de calibração do modelo;
- considerar a bicicleta como possível modo de integração ao transporte coletivo, conforme Zannat et al. (2024);
- avaliar a inclusão do Cadyts para calibração das viagens no transporte coletivo; e
- permitir a escolha de horário de saída aos agentes, além da escolha da rota e do modo de transporte.

Referências

- ARENTZE, T. et al. Data needs, data collection, and data quality requirements of activity-based transport demand models. *Transportation research circular*, US National Academy of Sciences Transportation Research Board, n. E-C008, 2000.
- AXHAUSEN, K. W. Definition of movement and activity for transport modelling. In: HENSHER, D. A.; BUTTON, K. J. (Ed.). *Handbook of transport modelling*. [S.l.]: Emerald Group Publishing Limited, 2008.
- BALVEDI, G. A. *Microssimulação de tráfego aplicada na avaliação do desempenho da segurança viária em rotatórias: um estudo de caso na rótula da UFSC/Trindade*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Transportes e Gestão Territorial, 2018.
- BARCELÓ, J. (Ed.). *Fundamentals of Traffic Simulation*. [S.l.]: Springer, 2010.
- BATES, J. History of demand modelling. In: *Handbook of transport modelling*. [S.l.]: Emerald Group Publishing Limited, 2008.
- BAZZAN, A. L. C.; KLUGL, F. A review on agent-based technology for traffic and transportation. *The Knowledge Engineering Review*, 2013.
- BHAT, C. R.; KOPPELMAN, F. S. Activity-based modeling of travel demand. In: HALI, R. W. (Ed.). *Handbook of transportation science*. [S.l.]: Springer Science and Business Media, 1999.
- BICUDO, D. G. *Aplicação do Simulador de Tráfego MATSim à Cidade de Joinville/SC*. Dissertação (Trabalho de Conclusão de Curso (Graduação)) — Universidade Federal de Santa Catarina, 2015.
- BIGI, F.; SCHWEMMLE, N.; VITI, F. Evaluating the impact of free public transport using agent-based modeling: the case-study of luxembourg. In: *hEART 2023: 11th Symposium of the European Association for Research in Transportation, September 6-8, 2023*. [S.l.: s.n.], 2023.
- BORTOLAZZO, S. S. A. *Contribuição metodológica para planejamento de transporte urbano: uma proposta sob a ótica da sustentabilidade ambiental*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Transportes e Gestão Territorial, 2017.
- BRASIL. Lei nº 12.587, de 03 de janeiro de 2012. institui as diretrizes da política nacional de mobilidade urbana; revoga dispositivos dos decretos-leis nºs 3.326, de 3 de junho de 1941, e 5.405, de 13 de abril de 1943, da consolidação das leis do trabalho (clt), aprovada pelo decreto-lei nº 5.452, de 1º de maio de 1943, e das leis nºs 5.917, de 10 de setembro de 1973, e 6.261, de 14 de novembro de 1975; e dá outras providências. *Diário Oficial da República Federativa do Brasil*, Brasília, DF, 2012. Disponível em: <https://www.planalto.gov.br/ccivil_03/_ato2011-2014/2012/lei/l12587.htm>.
- BRAVO, M. D. *Contribuição metodológica para estimativa da demanda transferida do automóvel para o transporte coletivo em um polo gerador de viagens*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Transportes e Gestão Territorial, 2019.
- BRUTON, M. J. *Introdução ao planejamento dos transportes*. [S.l.]: Editora Interciência Ltda, 1979.
- BUISSON, C. et al. Calibration and validation principles. In: HOOGENDOORN, W. D. C. B. S. P. (Ed.). *Traffic Simulation and Data: Validation Methods and Applications*. [S.l.: s.n.], 2015.

- CAMPOS, V. B. G. *Planejamento de transportes: conceitos e modelos*. [S.l.]: Editora Interciência Ltda., 2013.
- CARVALHO, C. H. R. de. *Desafios da mobilidade urbana no Brasil*. Brasília, 2016.
- CARVALHO, C. H. R. de; PEREIRA, R. H. M. *Efeitos da variação da tarifa e da renda da população sobre a demanda de transporte público coletivo urbano no Brasil*. Brasília, 2011.
- CHARYPAR, D. et al. *A High-performance Traffic Flow Microsimulation for Large Problems*. Eidgenössische Technische Hochschule, Institut für Verkehrsplanung und Transportsysteme, 2008. Disponível em: <<https://books.google.com.br/books?id=lltMcgAACAAJ>>.
- DELLAROZA, L. F. G. *Modelo contínuo para desenho de rede de transporte público por ônibus: aplicação à cidade de Londrina/PR*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Transportes e Gestão Territorial, 2021.
- DENATRAN. *Manual Brasileiro de Sinalização de Trânsito: Volume V - Sinalização Semafórica*. [S.l.], 2014.
- DOWLING, R.; SKABARDONIS, A.; ALEXIADIS, V. *Traffic analysis toolbox, volume II: guidelines for applying traffic microsimulation software*. Fhwa-hop-18-036. Washington (EUA), 2019.
- EIBEN, J. S. A. *Introduction to Evolutionary Computing*. 2. ed. [S.l.]: Springer Berlin, Heidelberg, 2015. (Natural Computing Series).
- FARINHA, P. M. L. *Modelos de Simulação em MATSim aplicados à análise de Sistemas de Transportes*. Dissertação (Mestrado) — Área Departamental de Engenharia Civil - Instituto Superior de Engenharia de Lisboa, 2013.
- FERT, L. *Análise comparativa do uso de diferentes softwares para modelagem de tráfego em interseções urbanas*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Transportes e Gestão Territorial, 2017.
- FLÖTTERÖD, G. Cadyts—calibration of dynamic traffic simulations—version 1.1. 0 manual. *Transport and Mobility Laboratory, École Polytechnique Fédérale de Lausanne*, 2010.
- HENSHER, D. A.; BUTTON, K. J. *Handbook of transport modelling*. [S.l.]: Emerald Group Publishing Limited, 2008.
- HEYDECKER, B. G.; ADDISON, J. D. Analysis of dynamic traffic equilibrium with departure time choice. *Transportation Science*, v. 39, n. 1, p. 39–57, 2005.
- HORNI, A.; NAGEL, K.; AXHAUSEN, K. W. (Ed.). *The Multi-Agent Transport Simulation MATSim*. [S.l.]: Ubiquity Press, 2016.
- HORNI, A. et al. Location choice modeling for shopping and leisure activities with matsim: Combining microsimulation and time geography. *Transportation Research Record*, v. 2135, n. 1, p. 87–95, 2009. Disponível em: <<https://doi.org/10.3141/2135-11>>.
- KICKHÖFERA, B. et al. Creating an open matsim scenario from open data: The case of santiago de chile. *VSP Working Paper*, v. 16, n. 02, 2016. Disponível em: <<http://www.vsp.tu-berlin.de/publications>>.
- KLÜGL, F.; BAZZAN, A. L. C. Agent-based modeling and simulation. *AI Magazine*, v. 33, n. 3, p. 29–40, 2012.

- LAW, A. M.; KELTON, W. D. *Simulation Modeling and Analysis*. [S.l.]: McGraw-Hill Higher Education, 1999.
- LOPEZ, P. A. et al. Microscopic traffic simulation using sumo. In: *21st International Conference on Intelligent Transportation Systems (ITSC)*. [S.l.: s.n.], 2018. p. 2575–2582.
- MACHADO, S. D. *Mobilidade urbana sustentável: o caso do sistema de transporte público de ônibus da Região Metropolitana de Florianópolis*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Transportes e Gestão Territorial, 2022.
- MAIA, T. d. G. *Transporte público integrado ao planejamento urbano nos planos diretores: estudo multicaso do núcleo da Região Imediata de Itajaí - SC*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Transportes e Gestão Territorial, 2022.
- MARCON, A. F. *Using a multimodal database to assess the effect of the North-South line on the use of bicycle as a feeder mode to urban public transit in Amsterdam*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Transportes e Gestão Territorial, 2022.
- MARTENDAL, H. R. *Sistemas de transportes inteligentes e sua relação com a mobilidade urbana sustentável: proposta de framework para cidades brasileiras*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Transportes e Gestão Territorial, 2023.
- MCNALLY, M. G. The activity-based approach. *UC Irvine: Institute of Transportation Studies*, 2000.
- MELO, J. C. *Planejamento dos transportes urbanos*. Rio de Janeiro: Editora Campus, 1981.
- MIRANDA, D. F. *Metodologia de tratamento de dados para simulação de modelo baseado em atividades usando o software MATSim*. [S.l.]: Departamento de Engenharia Civil e Ambiental da Universidade de Brasília, 2017.
- NICOLAI, T. W. Using matsim as a travel model plug-in to urbansim. *SustainCity. Deliverable D*, v. 7, 2012.
- NOVAIS, A. G. *Modelos em planejamento urbano, regional e de transportes*. [S.l.]: Editora Edgard Blücher Ltda, 1981.
- OLIVEROS, M. M. *Calibration of Public Transit Routing for Multi-Agent Simulation*. Dissertação (Mestrado) — Technischen Universität Berlin, 2014.
- ORTÚZAR, J. de D.; WILLUMSEN, L. G. *Modelling Transport*. [S.l.]: John Wiley and Sons, 2011.
- PEREIRA, H. A. M. *Método de análise de sistemas de transporte público urbano em cidades de médio porte: estudo de caso em São José dos Pinhais/PR*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Transportes e Gestão Territorial, 2021.
- POPOVICI, E. et al. Coevolutionary principles. In: _____. *Handbook of Natural Computing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 987–1033. ISBN 978-3-540-92910-9. Disponível em: <https://doi.org/10.1007/978-3-540-92910-9_31>.
- SHIFTAN, Y.; BEN-AKIVA, M. A practical policy-sensitive, activity-based, travel-demand model. *Annals of Regional Science of the 3rd Israel-Netherlands Workshop in Regional Science*, 2010.

- SILVA, A. N. R. da; COSTA, M. da S.; MACÊDO, M. H. Planejamento integrado, organização espacial e mobilidade sustentável no contexto de cidades brasileiras. In: BALBIM, R. N. O.; KRAUSE, C. H. O.; LINKE, C. C. O. (Ed.). *Cidade e movimento: mobilidades e interações no desenvolvimento urbano*. [S.l.]: Instituto de Pesquisa Econômica Aplicada (IPEA) and Instituto de Políticas de Transporte e Desenvolvimento (ITDP), 2016.
- SILVA, B. S. d. *A utilização de subsídios e receitas extratarifárias ao transporte coletivo no Brasil e sua relação com a mobilidade urbana*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Transportes e Gestão Territorial, 2022.
- SIVAKUMAR, A. Modelling transport: A synthesis of transport modelling methodologies. In: . [S.l.: s.n.], 2007.
- SLOBODEN, J. et al. *Traffic Analysis Toolbox Volume II: Decision Support Methodology for Selecting Traffic Analysis Tools*. Fhwa-hrt-04-039. [S.l.], 2004.
- SLOBODEN, J. et al. *Traffic Analysis Toolbox Volume XIV: Guidebook on the Utilization of Dynamic Traffic Assignment in Modeling Traffic Analysis Toolbox Volume XIV: Guidebook on the Utilization of Dynamic Traffic Assignment in Modeling*. Fhwa-hop-13-015. [S.l.], 2012.
- TREIBER, M.; KESTING, A. *Traffic Flow Dynamics: Data, Models and Simulation*. [S.l.]: Springer, 2013.
- WISE, S.; CROOKS, A.; BATTY, M. Transportation in agent-based urban modelling. In: NAMAZI-RAD, M.-R. et al. (Ed.). *Agent Based Modelling of Urban Systems*. Cham: Springer International Publishing, 2017. p. 129–148. ISBN 978-3-319-51957-9.
- ZANNAT, K. E. et al. Developing an agent-based microsimulation for predicting the bus rapid transit (brt) demand in developing countries: A case study of dhaka, bangladesh. *Transport Policy*, v. 148, p. 92–106, 2024. ISSN 0967-070X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0967070X23003426>>.
- ZHANG, L.; LEVINSON, D. An agent-based approach to travel demand modeling: An exploratory analysis. *Transportation Research Record Journal of the Transportation Research Board*, v. 1898, 02 2004.
- ZIEMKE, D.; NAGEL, K. Development of a fully synthetic and open scenario for agent-based transport simulations – the matsim open berlin scenario. *VSP Working Paper*, v. 17, n. 12, 2017. Disponível em: <<http://www.vsp.tu-berlin.de/publications>>.

APÊNDICES

A Script de modelagem da rede viária

```
1 # Código para leitura da malha viária de Florianópolis
2 # Fernanda Beckhauser Mallon | baseado no código de Greg Macfarlane e
   Parsons Brinckerhoff
3 #=====
4 # IMPORTAÇÕES
5 #=====
6 import psycopg2
7 import lxml.etree as ET
8 from pathlib import Path
9 import subprocess
10 #=====
11 # CONFIGURAÇÕES
12 #=====
13 db_string = "host=localhost dbname=dissertacao user=fernandabmallon"
14 links_path = Path("/Users/fernandabmallon/Google_Drive/UFSC/Mestrado/
   Shapefile/Rede_Viaria_FLN/links.shp")
15 nodes_path = Path("/Users/fernandabmallon/Google_Drive/UFSC/Mestrado/
   Shapefile/Rede_Viaria_FLN/nodes.shp")
16 XY_redeviaria_path = Path("/Users/fernandabmallon/Google_Drive/UFSC/
   Mestrado/Shapefile/Rede_Viaria_FLN/XY_links_simpl.csv")
17 XY_node_path = Path("/Users/fernandabmallon/Google_Drive/UFSC/Mestrado/
   Shapefile/Rede_Viaria_FLN/XY_nodes_simpl.csv")
18 # =====
19 # INCLUSÃO DA REDE VIÁRIA NO BANCO DE DADOS
20 # =====
21 class DataToPGIS():
22     def __init__( self ):
23         con = psycopg2.connect(db_string)
24         con.set_isolation_level(psycopg2.extensions.
           ISOLATION_LEVEL_AUTOCOMMIT)
25         with con:
26             cur = con.cursor()
27 #Limpar base de dados entre iterações
28             cur.execute( "DROP TABLE IF EXISTS links;" )
29             cur.execute( "DROP TABLE IF EXISTS nodes;" )
30             cur.execute( "DROP TABLE IF EXISTS XY_redeviaria;" )
31             cur.execute( "DROP TABLE IF EXISTS XY_nodes;" )
32
33 #Inserir tabela coordenadas e roads na base de dados dissertacao
34
35             self.system_run( "shp2pgsql -S %s | psql -d dissertacao" %
           links_path)
36             self.system_run( "shp2pgsql -S %s | psql -d dissertacao" %
           nodes_path)
37
38             cur.execute( "CREATE TABLE XY_redeviaria ( gid_link text, ID
           integer, X1 text, Y1 text, X2 text, Y2 text);" )
39             with open( XY_redeviaria_path, "r" ) as dados:
40                 print("dados =", dados)
41                 #cur.copy_from( dados, "XY_redeviaria", sep="," )
42                 cur.copy_expert("""COPY XY_redeviaria FROM STDIN WITH (
           FORMAT CSV)""", dados)
```

```

43
44         cur.execute( "CREATE TABLE XY_nodes ( X text, Y text, id text,
45                     tipo integer);" )
46         with open( XY_node_path, "r" ) as dados1:
47             print("dados1 =", dados1)
48             next(dados1)
49             cur.copy_from( dados1, "XY_nodes", sep="," )
50             #cur.copy_expert( """COPY XY_nodes FROM STDIN WITH (FORMAT
51                             CSV)""", dados1)
52
53         cur.execute("ALTER TABLE nodes DROP COLUMN IF EXISTS x")
54         cur.execute("ALTER TABLE nodes DROP COLUMN IF EXISTS y")
55         cur.execute("ALTER TABLE links ADD COLUMN x1 text,ADD COLUMN y1
56                     text,ADD COLUMN x2 text,ADD COLUMN y2 text")
57         cur.execute("ALTER TABLE nodes ADD COLUMN x text,ADD COLUMN y
58                     text")
59         #cur.execute("ALTER TABLE XY_nodes ALTER COLUMN gid_node TYPE
60                     INT using gid_node::integer")
61         #cur.execute("ALTER TABLE nodes ADD COLUMN ID1 int)
62         #cur.execute("UPDATE nodes SET ID1 = gid USING gid::integer")
63
64         cur.execute("SELECT ID FROM XY_REDEVIARIA ORDER BY ID")
65         xy_gids = cur.fetchall()
66
67         for xy_gid in xy_gids:
68             print("xy_gid[0]=", xy_gid[0])
69             cur.execute("SELECT X1,Y1,X2,Y2 FROM XY_REDEVIARIA WHERE ID
70                         = '%s'" %xy_gid[0])
71             links_xy = cur.fetchall()
72             print("links_xy=", links_xy)
73             x_1 = links_xy[0][0]
74             y_1 = links_xy[0][1]
75             x_2 = links_xy[0][2]
76             y_2 = links_xy[0][3]
77
78             #gid_alt = int(str(xy_gid[0]).replace("_",""))
79             #print("gid_alt=", gid_alt)
80
81             cur.execute("UPDATE links SET x1 = %s, y1 = %s, x2 = %s, y2
82                         = %s WHERE gid = %d" %(x_1, y_1, x_2, y_2, xy_gid[0]))
83
84         def system_run( self, input ):
85             print( "-----" )
86             print( "comando: %s" % input )
87             result = subprocess.call( input, shell=True )
88
89         # =====
90         # MATSIM NETWORK BUILDER
91         # =====
92         class MATSimNetworkBuilder():
93             def __init__( self ):
94                 con = psycopg2.connect(db_string)
95                 con.set_isolation_level(psycopg2.extensions.
96                     ISOLATION_LEVEL_AUTOCOMMIT)
97                 with con:
98                     cur = con.cursor()

```

```

91     network = ET.Element("network",
92         attrib={'name':"MATSim network exported from PLAMUS
          shapefile."})
93
94     nodes = ET.SubElement(network, "nodes")
95
96     cur.execute("SELECT ID FROM links WHERE tipologia_ <= 98 ORDER
          BY ID")
97     links_gids = cur.fetchall()
98     cur.execute("SELECT id FROM XY_nodes ORDER BY id")
99     id_nodes = cur.fetchall()
100    #id_node = 0
101    mynodes = []
102    for links_gid in links_gids:
103        #gid_l_alt = str(links_gid[0]) + "_"
104        cur.execute("SELECT x1, y1, x2, y2 FROM xy_redeviaria WHERE
          ID = '%s'" %int(links_gid[0]))
105        xy_links = cur.fetchall()
106        x1 = xy_links[0][0]
107        y1 = xy_links[0][1]
108        x2 = xy_links[0][2]
109        y2 = xy_links[0][3]
110
111
112        print("SELECT id FROM XY_nodes WHERE x1 = '%s' AND y1 = '%s
          ' AND x2 = '%s' AND y2 = '%s'" %(x1,y1,x2,y2))
113        cur.execute("SELECT id FROM XY_nodes WHERE x = '%s' AND y =
          '%s'" %(x1,y1))
114        nodes_1 = cur.fetchall()[0][0]
115        node_1 = int(str(nodes_1).replace("a",""))-2
116        mynodes.append(nodes_1)
117        #print("SELECT gid FROM nodes WHERE x = '%s' AND y = '%s'"
          %(x2,y2))
118        cur.execute("SELECT id FROM XY_nodes WHERE x = '%s' AND y =
          '%s'" %(x2,y2))
119        nodes_2 = cur.fetchall()[0][0]
120        node_2 = int(str(nodes_2).replace("a",""))-2
121        mynodes.append(nodes_2)
122    mynodes = list(dict.fromkeys(mynodes))
123
124    for id_node in mynodes:
125        id_n_alt1 = int(str(id_node).replace("a",""))-2
126        #id_node = id_node + 1
127        print("id_node=",id_node)
128        cur.execute("SELECT X,Y FROM XY_nodes WHERE id = '%s'" %
          id_node)
129        xy_nodes = cur.fetchall()
130        print("xy_nodes=",xy_nodes)
131        x = xy_nodes[0][0]
132        y = xy_nodes[0][1]
133
134        #gid_n_alt = str(gid_node[0]).replace("_","")
135
136        #cur.execute("UPDATE nodes SET x = %s, y = %s WHERE gid = %
          d" %(x,y,int(gid_node[0])))
137

```

```

138
139         print("id_node =",id_node)
140
141         ET.SubElement(nodes, "node",
142             attrib={'id': str(id_n_alt1),
143                 'x':str(x),
144                 'y':str(y)})
145
146
147         links = ET.SubElement(network, "links",
148             attrib={'caperiod': "01:00:00",
149                 'effectivecellsize': "7.5",
150                 'effectivelanewidth': "3.75"})
151
152         #cur.execute("SELECT gid FROM links WHERE faixaefeti != 0.0 AND
153             faixaefet1 != 0.0 AND velfl_ba IS NOT NULL AND velfl_ab IS
154             NOT NULL AND cap_ab IS NOT NULL AND cap_ba IS NOT NULL
155             ORDER BY gid")
156
157         #print("links_gids=",links_gids)
158         for links_gid in links_gids:
159             #gid_l_alt = str(links_gid[0]) + "_"
160             cur.execute("SELECT x1, y1, x2, y2 FROM xy_redeviaria WHERE
161                 ID = '%s'" %int(links_gid[0]))
162             xy_links = cur.fetchall()
163             x1 = xy_links[0][0]
164             y1 = xy_links[0][1]
165             x2 = xy_links[0][2]
166             y2 = xy_links[0][3]
167
168             print("SELECT id FROM XY_nodes WHERE x1 = '%s' AND y1 = '%s'
169                 ' AND x2 = '%s' AND y2 = '%s'" %(x1,y1,x2,y2))
170             cur.execute("SELECT id FROM XY_nodes WHERE x = '%s' AND y =
171                 '%s'" %(x1,y1))
172             node_1 = int(str(cur.fetchall()[0][0]).replace("a",""))-2
173             #print("SELECT gid FROM nodes WHERE x = '%s' AND y = '%s'"
174                 %(x2,y2))
175             cur.execute("SELECT id FROM XY_nodes WHERE x = '%s' AND y =
176                 '%s'" %(x2,y2))
177             node_2 = int(str(cur.fetchall()[0][0]).replace("a",""))-2
178             print("node_1,node_2=",node_1,node_2)
179
180             cur.execute("SELECT length1,tipologia_ FROM links WHERE ID
181                 = %d" %int(links_gid[0]))
182             result = cur.fetchall()
183             length = result[0][0]
184             tipo = result[0][1]
185
186             cur.execute("SELECT dir FROM links WHERE ID = %d" %int(
187                 links_gid[0]))
188             direction = cur.fetchall()[0][0]
189
190             cur.execute("SELECT velfl_ab FROM links WHERE ID = %d" %int
191                 (links_gid[0]))
192             freespeed_AB = cur.fetchall()[0][0]/3.6

```

```

183
184         cur.execute("SELECT velfl_ba FROM links WHERE ID = %d" %int
                    (links_gid[0]))
185         freespeed_BA = cur.fetchall()[0][0]/3.6
186         print("freespeed_BA=",freespeed_BA)
187
188
189
190         cur.execute("SELECT faixaefeti FROM links WHERE ID = %d" %
                    int(links_gid[0]))
191         permlanes_AB = cur.fetchall()[0][0]
192
193         cur.execute("SELECT faixaefet1 FROM links WHERE ID = %d" %
                    int(links_gid[0]))
194         permlanes_BA = cur.fetchall()[0][0]
195
196         if tipo <90:
197
198             cur.execute("SELECT cap_ab FROM links WHERE ID = %d" %
                        int(links_gid[0]))
199             capacity_AB = cur.fetchall()[0][0]
200
201             cur.execute("SELECT cap_ba FROM links WHERE ID = %d" %
                        int(links_gid[0]))
202             capacity_BA = cur.fetchall()[0][0]
203
204             if direction == 1:
205                 id_link = str(int(links_gid[0])) + '01'
206                 ET.SubElement(links, "link", attrib={
207                     'id': str(id_link),
208                     'from': str(node_1),
209                     'to': str(node_2),
210                     'length': str(length),
211                     'freespeed': str(freespeed_AB),
212                     'capacity': str(capacity_AB),
213                     'permlanes': str(permlanes_AB),
214                     'oneway': str(1),
215                     'modes': "car,motorcycle,pt"})
216
217             if direction == -1:
218                 id_link = str(int(links_gid[0])) + '02'
219                 ET.SubElement(links, "link", attrib={
220                     'id': str(id_link),
221                     'from': str(node_2),
222                     'to': str(node_1),
223                     'length': str(length),
224                     'freespeed': str(freespeed_BA),
225                     'capacity': str(capacity_BA),
226                     'permlanes': str(permlanes_BA),
227                     'oneway': str(1),
228                     'modes': "car,motorcycle,pt"})
229
230             if direction == 0:
231                 id_link1 = str(int(links_gid[0])) + '01'
232                 id_link2 = str(int(links_gid[0])) + '02'
233                 ET.SubElement(links, "link", attrib={

```

```

234         'id': str(id_link1),
235         'from': str(node_1),
236         'to': str(node_2),
237         'length': str(length),
238         'freespeed': str(freespeed_AB),
239         'capacity': str(capacity_AB),
240         'permlanes': str(permlanes_AB),
241         'oneway': str(1),
242         'modes': "car,motorcycle,pt"})
243
244     ET.SubElement(links, "link", attrib={
245         'id': str(id_link2),
246         'from': str(node_2),
247         'to': str(node_1),
248         'length': str(length),
249         'freespeed': str(freespeed_BA),
250         'capacity': str(capacity_BA),
251         'permlanes': str(permlanes_BA),
252         'oneway': str(1),
253         'modes': "car,motorcycle,pt"})
254
255     elif tipo == 98:
256         if direction == 1:
257             id_link = str(int(links_gid[0])) + '01'
258             ET.SubElement(links, "link", attrib={
259                 'id': str(id_link),
260                 'from': str(node_1),
261                 'to': str(node_2),
262                 'length': str(length),
263                 'freespeed': str(freespeed_AB),
264                 'capacity': str('20000'),
265                 'permlanes': str(1),
266                 'oneway': str(1),
267                 'modes': "pt"})
268
269             if direction == -1:
270                 id_link = str(int(links_gid[0])) + '02'
271                 ET.SubElement(links, "link", attrib={
272                     'id': str(id_link),
273                     'from': str(node_2),
274                     'to': str(node_1),
275                     'length': str(length),
276                     'freespeed': str(freespeed_BA),
277                     'capacity': str('20000'),
278                     'permlanes': str(1),
279                     'oneway': str(1),
280                     'modes': "pt"})
281
282             if direction == 0:
283                 id_link1 = str(int(links_gid[0])) + '01'
284                 id_link2 = str(int(links_gid[0])) + '02'
285                 ET.SubElement(links, "link", attrib={
286                     'id': str(id_link1),
287                     'from': str(node_1),
288                     'to': str(node_2),
289                     'length': str(length),

```

```

290         'freespeed': str(freespeed_AB),
291         'capacity': str('20000.0'),
292         'permlanes': str(1),
293         'oneway': str(1),
294         'modes': "pt"})
295
296     ET.SubElement(links, "link", attrib={
297         'id': str(id_link2),
298         'from': str(node_2),
299         'to': str(node_1),
300         'length': str(length),
301         'freespeed': str(freespeed_BA),
302         'capacity': str('20000.0'),
303         'permlanes': str(1),
304         'oneway': str(1),
305         'modes': "pt"})
306
307     tree = ET.ElementTree(network)
308
309     tree.write("network_pt_v3.xml", pretty_print = True)
310
311
312
313
314
315
316 if __name__ == "__main__":
317
318     dataToPGIS = DataToPGIS()
319
320     MATSimNetworkBuilder = MATSimNetworkBuilder()

```

B Script de modelagem da demanda

```
1 # Codigo para leitura das informacoes da populacao na Matriz OD
2 # de Florianopolis
3 # Fernanda Beckhauser Mallon
4
5 #=====
6 #Importacoes
7 import lxml.etree as ET
8 import pandas as pd
9 import sqlite3 as lite
10 import psycopg2
11 from datetime import datetime
12 import math
13 from statistics import mean
14
15 #=====
16 #Dicionarios
17 fator_expansao_OD = {"12":"123",
18 "13":"123",
19 "14":"123",
20 "15":"123",
21 "16":"10",
22 ...}
23 leg_mode = {"a pe" : "walk",
24 "transporte escolar":"non_network_walk",
25 "dirigindo automovel":"car",
26 "motocicleta":"car",
27 "outros":"non_network_walk",
28 "passageiro de automovel":"car",
29 "bicicleta":"bike",
30 "onibus":"pt",
31 "transporte fretado":"non_network_walk",
32 "taxi":"car"
33 }
34 act_type = {"Residencia" : "home",
35 "Trabalho" : "work",
36 "Lazer" : "leisure",
37 "Escola/Educacao" : "education",
38 "Saude" : "all",
39 "Compras" : "shop",
40 "Outros / nao sabe" : "all"
41 }
42 posse_auto = {"Sim":"always",
43 "Nao":"never"}
44 #=====
45 #Configurações
46 db_string = "host=localhost dbname=dissertacao user=fernandabmallon"
47 #=====
48 #Procedimento
49
50 #Criação da estrutura xml
51
52 plans = ET.Element("population")
```

```

53
54 #Transferência dos dados da pesquisa OD para uma base de dados sqlite
55 con = psycopg2.connect(db_string)
56 con.set_isolation_level(psycopg2.extensions.ISOLATION_LEVEL_AUTOCOMMIT)
57 with con:
58     cur = con.cursor()
59
60     cur.execute("SELECT DISTINCT ID_Morador FROM viagens_PLAMUS ORDER BY
61 ID_Morador")
62 ID_Moradores = cur.fetchall()
63 cur.execute("SELECT DISTINCT ID_Morador FROM viagens_PLAMUS WHERE (
        ID_Morador IS NULL OR ETAPA != 1 OR Hora_Saída IS NULL OR
        Hora_Chegada IS NULL OR X_Zona_0 = '#N/D' OR Y_Zona_0 = '#N/D' OR
        X_Zona_D = '#N/D' OR Y_Zona_D = '#N/D')")
64 IDs_retirados = cur.fetchall()
65
66 for i in range(len(IDs_retirados)):
67     ID_a_ser_retirado = IDs_retirados[i]
68     print("ID_a_ser_retirado = %s" %ID_a_ser_retirado)
69     ID_Moradores.remove(ID_a_ser_retirado)
70
71
72 for ID_Morador0 in ID_Moradores:
73     print("ID_Morador=", ID_Morador0[0])
74     if ID_Morador0 is not None:
75         ID_Morador = ID_Morador0
76
77         cur.execute("SELECT Gênero FROM viagens_PLAMUS WHERE ID_Morador = %
78 d" %ID_Morador)
79 sexo = cur.fetchall()[0]
80 if sexo == '1':
81     sex_person = 'f'
82 else:
83     sex_person = 'm'
84
85 cur.execute("SELECT Idade FROM viagens_PLAMUS WHERE ID_Morador = %d
86 " %ID_Morador)
87 age_person = cur.fetchall()[0][0]
88
89 cur.execute("SELECT Posse_Auto FROM viagens_PLAMUS WHERE ID_Morador
90 = %d" %ID_Morador)
91 car_avail_person = cur.fetchall()[0][0]
92
93 cur.execute("SELECT Viagem FROM viagens_PLAMUS WHERE ID_Morador = %
94 d ORDER BY Viagem" %ID_Morador)
95 num_viagens = cur.fetchall()
96 cur.execute("SELECT Max(Viagem) FROM viagens_PLAMUS WHERE
97 ID_Morador = %d" %ID_Morador)
98 max_viagens = cur.fetchall()[0][0]
99
100 FE_Morador = math.ceil(int(fator_expansao_OD[str(int(ID_Morador0
101 [0])]))/10)
102 print("FE_Morador=", FE_Morador)
103 cur.execute("UPDATE viagens_PLAMUS SET FE_Morador = %s WHERE
104 ID_Morador = %s" %(FE_Morador, ID_Morador0[0]))

```

```

99     for i in range(int(FE_Morador)):
100         i_texto = str(i+1)
101         print("i_texto=",i_texto)
102         if i < 9:
103             str_00 = "00"
104             sufixo_ID = str_00 + i_texto
105         elif i < 99:
106             str_0 = "0"
107             sufixo_ID = str_0 + i_texto
108         else:
109             sufixo_ID = i_texto
110
111         ID_Morador_Exp = str(int(ID_Morador[0])) + sufixo_ID
112
113         person = ET.SubElement(plans, "person", attrib={
114             'id': str(int(ID_Morador_Exp))})
115
116         plan = ET.SubElement(person, "plan")
117         for num_viagem in num_viagens:
118             print("num_viagem =",num_viagem[0])
119             print("max_viagens =", max_viagens)
120             cur.execute("SELECT Motivo_0 FROM viagens_PLAMUS WHERE
121                 ID_Morador = %d AND Viagem = %f" %(ID_Morador[0],
122                 num_viagem[0]))
121             Type_act = cur.fetchall()[0][0]
122             print("act_type[Type_act]=",act_type[Type_act])
123
124             if Type_act == 'Residência':
125                 cur.execute("SELECT X_Residencia FROM viagens_PLAMUS
126                     WHERE ID_Morador = %d AND Viagem = %f" %(ID_Morador
127                     [0], num_viagem[0]))
126                 X = cur.fetchall()[0][0]
127                 cur.execute("SELECT Y_Residencia FROM viagens_PLAMUS
128                     WHERE ID_Morador = %d AND Viagem = %f" %(ID_Morador
129                     [0], num_viagem[0]))
128                 Y = cur.fetchall()[0][0]
129             else:
130                 cur.execute("SELECT X_Zona_0 FROM viagens_PLAMUS WHERE
131                     ID_Morador = %d AND Viagem = %f" %(ID_Morador[0],
132                     num_viagem[0]))
131                 X = cur.fetchall()[0][0]
132                 cur.execute("SELECT Y_Zona_0 FROM viagens_PLAMUS WHERE
133                     ID_Morador = %d AND Viagem = %f" %(ID_Morador[0],
134                     num_viagem[0]))
133                 Y = cur.fetchall()[0][0]
134
135             cur.execute("SELECT Motivo_D FROM viagens_PLAMUS WHERE
136                 ID_Morador = %d AND Viagem = %f" %(ID_Morador[0],
137                 num_viagem[0]))
136             Type_act_D = cur.fetchall()[0][0]
137             print("act_type[Type_act_D]=",act_type[Type_act_D])
138
139             if Type_act_D == 'Residência':
140                 cur.execute("SELECT X_Residencia FROM viagens_PLAMUS
141                     WHERE ID_Morador = %d AND Viagem = %f" %(ID_Morador
142                     [0], num_viagem[0]))

```

```

141         X_D = cur.fetchall()[0][0]
142         cur.execute("SELECT Y_Residencia FROM viagens_PLAMUS
                    WHERE ID_Morador = %d AND Viagem = %f" %(ID_Morador
                    [0], num_viagem[0]))
143         Y_D = cur.fetchall()[0][0]
144     else:
145         cur.execute("SELECT X_Zona_0 FROM viagens_PLAMUS WHERE
                    ID_Morador = %d AND Viagem = %f" %(ID_Morador[0],
                    num_viagem[0]))
146         X_D = cur.fetchall()[0][0]
147         cur.execute("SELECT Y_Zona_0 FROM viagens_PLAMUS WHERE
                    ID_Morador = %d AND Viagem = %f" %(ID_Morador[0],
                    num_viagem[0]))
148         Y_D = cur.fetchall()[0][0]
149
150
151     cur.execute("SELECT Hora_Saida FROM viagens_PLAMUS WHERE
                ID_Morador = %d AND Viagem = %f AND Hora_Saida IS NOT
                NULL" %(ID_Morador[0], num_viagem[0]))
152     result1 = cur.fetchall()
153     print(result1)
154     if len(result1) != 0:
155         End_time_act = "%s" %(result1[0][0])
156         Hora_Saida = datetime.strptime(End_time_act, '%H:%M:%S
                ')
157         Hora_Saida_hour = Hora_Saida.hour
158         Hora_Saida_minute = Hora_Saida.minute
159         if Hora_Saida_minute < 10 and Hora_Saida_hour < 10:
160             str_Hora_Saida = str("0%d:0%d" %(Hora_Saida_hour,
                Hora_Saida_minute))
161
162         elif Hora_Saida_minute < 10 and Hora_Saida_hour >= 10:
163             str_Hora_Saida = str("%d:0%d" %(Hora_Saida_hour,
                Hora_Saida_minute))
164
165         elif Hora_Saida_minute >= 10 and Hora_Saida_hour < 10:
166             str_Hora_Saida = str("0%d:%d" %(Hora_Saida_hour,
                Hora_Saida_minute))
167
168         else:
169             str_Hora_Saida = str("%d:%d" %(Hora_Saida_hour,
                Hora_Saida_minute))
170
171     else:
172         str_Hora_Saida = ""
173     print("str_Hora_Saida=", str_Hora_Saida)
174
175
176
177     if num_viagem[0] > 1.0:
178         id_viagem0 = num_viagem[0]-1
179         cur.execute("SELECT Hora_Chegada FROM viagens_PLAMUS
                    WHERE ID_Morador = %d AND Viagem = %f AND
                    Hora_Chegada IS NOT NULL" %(ID_Morador[0],
                    id_viagem0))
180         #print(cur.fetchall())

```

```

181     result = cur.fetchall()
182     print("result=",result)
183     if len(result) != 0:
184         Hora_Chegada = "%s" %(result[0][0])
185         Hora_Chegada = datetime.strptime(Hora_Chegada, '%H
           :%M:%S')
186         print("Hora_Chegada=",Hora_Chegada)
187
188         Duration_act = Hora_Saida- Hora_Chegada
189         print("Duration_act = ",Duration_act)
190
191         str_Duration_act_hour = math.floor((Duration_act.
           seconds)/(3600))
192         str_Duration_act_minute = int((Duration_act.seconds
           /60) - (str_Duration_act_hour*60))
193
194         if str_Duration_act_minute < 10 and
           str_Duration_act_hour < 10:
195             str_Duration_act = str("0%d:0%d" %(
           str_Duration_act_hour,
           str_Duration_act_minute))
196
197         elif str_Duration_act_minute < 10 and
           str_Duration_act_hour >= 10:
198             str_Duration_act = str("%d:0%d" %(
           str_Duration_act_hour,
           str_Duration_act_minute))
199
200         elif str_Duration_act_minute >= 10 and
           str_Duration_act_hour < 10:
201             str_Duration_act = str("0%d:%d" %(
           str_Duration_act_hour,
           str_Duration_act_minute))
202
203         else:
204             str_Duration_act = str("%d:%d" %(
           str_Duration_act_hour,
           str_Duration_act_minute))
205
206
207     else:
208         str_Duration_act = ""
209     else:
210         str_Duration_act = ""
211     print("str_Duration_act=",str_Duration_act)
212
213
214     if num_viagem[0] == 1:
215         act = ET.SubElement(plan, "activity",
216                             attrib={'type': str(act_type[Type_act]),
217                                     'x':str(X),
218                                     'y':str(Y),
219                                     'end_time': str(str_Hora_Saida)})
220

```

```

221         cur.execute("SELECT MeioTransporte FROM viagens_PLAMUS
                WHERE ID_Morador = %d AND Viagem = %d AND Etapa =
                1" %(ID_Morador[0], num_viagem[0]))
222         Mode_leg = cur.fetchall()[0][0]
223
224         leg = ET.SubElement(plan, "leg",
225                             attrib={'mode': str(leg_mode[Mode_leg])})
226
227         cur.execute("SELECT osm_id_o FROM viagens_plamus WHERE
                ID_Morador = %d AND Viagem = %f" %(ID_Morador[0],
                num_viagem[0]))
228         osm_id_o = cur.fetchall()[0][0]
229
230
231     elif num_viagem[0] > 1.0:
232         act = ET.SubElement(plan, "activity",
233                             attrib={'type': str(act_type[Type_act]),
234                                     'x':str(X),
235                                     'y':str(Y),
236                                     'end_time': str(str_Hora_Saida),
237                                     'max_dur': str(str_Duration_act)})
238         cur.execute("SELECT MeioTransporte FROM viagens_PLAMUS
                WHERE ID_Morador = %d AND Viagem = %d AND Etapa =
                1" %(ID_Morador[0], num_viagem[0]))
239         Mode_leg = cur.fetchall()[0][0]
240
241         leg = ET.SubElement(plan, "leg",
242                             attrib={'mode': str(leg_mode[Mode_leg])})
243
244         cur.execute("SELECT osm_id_o FROM viagens_plamus WHERE
                ID_Morador = %d AND Viagem = %f" %(ID_Morador[0],
                num_viagem[0]))
245         osm_id_o = cur.fetchall()[0][0]
246
247
248     if num_viagem[0] > 1.0 and num_viagem[0] == max_viagens:
249         act = ET.SubElement(plan, "activity",
250                             attrib={'type': str(act_type[Type_act_D]),
251                                     'x':str(X_D),
252                                     'y':str(Y_D)})
253
254
255
256
257     '''
258     average_duracao_home = mean(lista_Duracao_home)
259     average_duracao_work = mean(lista_Duracao_work)
260     average_duracao_education = mean(lista_Duracao_education)
261     average_duracao_leisure = mean(lista_Duracao_leisure)
262     average_duracao_shop = mean(lista_Duracao_shop)
263     average_duracao_all = mean(lista_Duracao_all)
264     print("average_duracao_home = ",average_duracao_home)
265     print("average_duracao_work = ",average_duracao_work)
266     print("average_duracao_education = ",average_duracao_education)
267     print("average_duracao_leisure = ",average_duracao_leisure)
268     print("average_duracao_shop = ",average_duracao_shop)

```

```
269     print("average_duracao_all = ", average_duracao_all)
270     '''
271
272
273     tree = ET.ElementTree(plans)
274
275     tree.write("plans.xml", pretty_print = True)
276     con.close()
```

C Parser de conversão dos dados de contagem de tráfego

```
1 # Código para estruturação dos dados de contagens de tráfego em arquivo xml
2 # Fernanda Beckhauser Mallon
3
4 #=====
5 #Importações
6 from pathlib import Path
7 import lxml.etree as ET
8 import pandas as pd
9 import psycopg2
10 from datetime import datetime
11 import math
12 from statistics import mean
13
14 #=====
15 #Dicionários
16
17 #=====
18 #Configurações
19 db_string = "host=localhost dbname=dissertacao user=fernandabmallon"
20 dados_contagem = Path("/Users/fernandabmallon/Library/CloudStorage/
    GoogleDrive-feemallon@gmail.com/Meu Drive/UFSC/Dissertacao/
    Processamento_Dados/Dados_Entrada_MATSim/Contagens_Trafego/
    postos_contagem_LIMPO_completo3.csv")
21 #=====
22 #Sumário
23 #1 Upar arquivo csv das contagens e arquivo kml dos postos de contagem para
    o banco de dados 'dissertacao'
24 #2 Identificar volume horário por posto de contagem
25 #3 Identificar link_id correspondente ao posto de contagem
26 #4 Montar estrutura xml
27
28
29 #=====
30 #Procedimento
31
32 #Criação da estrutura xml (counts>count>volume)
33
34 plans = ET.Element("counts",
35     attrib={'name':"MATSim counts exported from PLAMUS counts results
        related to PLAMUS's network."})
36
37 #Transferência dos dados da pesquisa OD para uma base de dados sqlite
38 con = psycopg2.connect(db_string)
39 con.set_isolation_level(psycopg2.extensions.ISOLATION_LEVEL_AUTOCOMMIT)
40 with con:
41     cur = con.cursor()
42
43     cur.execute("DROP TABLE IF EXISTS contagens_fase2;")
44     cur.execute("CREATE TABLE contagens_fase2 ( ID_Posto TEXT, ID_Link
        TEXT, Vol_6_7 REAL, Vol_7_8 REAL, Vol_8_9 REAL, Vol_9_10
        REAL, Vol_10_11 REAL, Vol_11_12 REAL, Vol_12_13 REAL, Vol_13_14
        REAL, Vol_14_15 REAL, Vol_15_16 REAL, Vol_16_17 REAL, Vol_17_18
```

```

REAL, Vol_18_19 REAL, Vol_19_20 REAL, Vol_20_21 REAL, Vol_21_22
REAL);" )
45 print("dados =")
46 with open( dados_contagem, "r" ) as dados:
47     print("dados =", dados)
48     cur.copy_expert("""COPY contagens_fase2 FROM STDIN WITH (FORMAT CSV
        )""", dados)
49
50
51
52 counts = ET.Element("counts")
53
54 cur.execute("SELECT ID_Posto FROM contagens_fase2 ORDER BY ID_Posto")
55 postos = cur.fetchall()
56
57 for posto in postos:
58
59     id_posto = posto[0]
60     cur.execute("SELECT ID_Link, vol_6_7,vol_7_8,vol_8_9,vol_9_10,
        vol_10_11,vol_11_12,vol_12_13,vol_13_14,vol_14_15,vol_15_16,
        vol_16_17,vol_17_18,vol_18_19,vol_19_20,vol_20_21,vol_21_22
        FROM contagens_fase2 WHERE ID_Posto='%s'" %id_posto)
61     result_posto = cur.fetchall()
62     id_link = result_posto[0][0]
63     vol_6_7 = result_posto[0][1]
64     vol_7_8 = result_posto[0][2]
65     vol_8_9 = result_posto[0][3]
66     vol_9_10 = result_posto[0][4]
67     vol_10_11 = result_posto[0][5]
68     vol_11_12 = result_posto[0][6]
69     vol_12_13 = result_posto[0][7]
70     vol_13_14 = result_posto[0][8]
71     vol_14_15 = result_posto[0][9]
72     vol_15_16 = result_posto[0][10]
73     vol_16_17 = result_posto[0][11]
74     vol_17_18 = result_posto[0][12]
75     vol_18_19 = result_posto[0][13]
76     vol_19_20 = result_posto[0][14]
77     vol_20_21 = result_posto[0][15]
78     vol_21_22 = result_posto[0][16]
79
80     count_ = ET.SubElement(counts, "count", attrib={
81         'loc_id': str(id_link),
82         'cs_id': str(id_posto)})
83     if vol_6_7 is not None:
84         volume6 = ET.SubElement(count_, "volume", attrib={
85             'h': str(6),
86             'val': str(vol_6_7)})
87     if vol_7_8 is not None:
88         volume7 = ET.SubElement(count_, "volume", attrib={
89             'h': str(7),
90             'val': str(vol_7_8)})
91     if vol_8_9 is not None:
92         volume8 = ET.SubElement(count_, "volume", attrib={
93             'h': str(8),
94             'val': str(vol_8_9)})

```

```

95     if vol_9_10 is not None:
96         volume9 = ET.SubElement(count_, "volume", attrib={
97             'h': str(9),
98             'val': str(vol_9_10)})
99     if vol_10_11 is not None:
100        volume10 = ET.SubElement(count_, "volume", attrib={
101            'h': str(10),
102            'val': str(vol_10_11)})
103    if vol_11_12 is not None:
104        volume11 = ET.SubElement(count_, "volume", attrib={
105            'h': str(11),
106            'val': str(vol_11_12)})
107    if vol_12_13 is not None:
108        volume12 = ET.SubElement(count_, "volume", attrib={
109            'h': str(12),
110            'val': str(vol_12_13)})
111    if vol_13_14 is not None:
112        volume13 = ET.SubElement(count_, "volume", attrib={
113            'h': str(13),
114            'val': str(vol_13_14)})
115    if vol_14_15 is not None:
116        volume14 = ET.SubElement(count_, "volume", attrib={
117            'h': str(14),
118            'val': str(vol_14_15)})
119    if vol_15_16 is not None:
120        volume15 = ET.SubElement(count_, "volume", attrib={
121            'h': str(15),
122            'val': str(vol_15_16)})
123    if vol_16_17 is not None:
124        volume16 = ET.SubElement(count_, "volume", attrib={
125            'h': str(16),
126            'val': str(vol_16_17)})
127    if vol_17_18 is not None:
128        volume17 = ET.SubElement(count_, "volume", attrib={
129            'h': str(17),
130            'val': str(vol_17_18)})
131    if vol_18_19 is not None:
132        volume18 = ET.SubElement(count_, "volume", attrib={
133            'h': str(18),
134            'val': str(vol_18_19)})
135    if vol_19_20 is not None:
136        volume19 = ET.SubElement(count_, "volume", attrib={
137            'h': str(19),
138            'val': str(vol_19_20)})
139    if vol_20_21 is not None:
140        volume20 = ET.SubElement(count_, "volume", attrib={
141            'h': str(20),
142            'val': str(vol_20_21)})
143    if vol_21_22 is not None:
144        volume21 = ET.SubElement(count_, "volume", attrib={
145            'h': str(21),
146            'val': str(vol_21_22)})
147
148    tree = ET.ElementTree(counts)
149
150    tree.write("counts2.xml", pretty_print = True)

```

151 | `con.close()`

D Script para calibração das proporções de capacidade viária

```
1 #
   =====
2 # Execução iterativa do MATSim com diferentes arquivos de configuração, e
   elaboração de materiais gráficos para representação dos resultados
3 # Fernanda Beckhauser Mallon
4 # 10 jun. 2022
5 #
   =====
6 # Importações
7
8 import subprocess #para execução do MATSim na linha de comando (terminal)
9 import xml.etree.ElementTree as ET #para modificação dos arquivos 'config.
   xml'
10 #
   =====
11 # Procedimento
12
13 # modificação dos arquivos 'config.xml'
14 # alterar capacidade viária para 10% a 30%
15 for cap_viaria1 in range(10,31):
16
17     cap_viaria = cap_viaria1/100
18
19     mytree = ET.parse('/Users/fernandabmallon/Documents/Mestrado/
       Dissertacao/Processamento_Dados/MATSim_Cadyts/matsim-example -
       project-master/scenarios/PLAMUS/config_19-09-23.xml')
20     myroot = mytree.getroot()
21
22
23     flowCapacityFactor = myroot[1][6]
24     storageCapacityFactor = myroot[1][7]
25     flowCapacityFactor.set('value', '%s' %cap_viaria)
26     storageCapacityFactor.set('value', '%s' %(cap_viaria**0.75))
27
28     outputDirectory = myroot[2][2]
29     runId = myroot[2][5]
30     outputDirectory.set('value', './output_%sC_PLAMUS_19-09" %cap_viaria1)
31     runId.set('value', './%sC_PLAMUS_19-09" %cap_viaria1)
32
33
34     header = """<?xml version="1.0" encoding="UTF-8"?>
35 <!DOCTYPE config SYSTEM "http://www.matsim.org/files/dtd/config_v2.dtd">
36 """ .encode()
37
38     # We can now wrap the entire element tree and write it to an xml file
   with the
```

```

39 # same headers that came in the original file.
40
41 with open('/Users/fernandabmallon/Documents/Mestrado/Dissertacao/
    Processamento_Dados/MATSim_Cadyts/matsim-example-project-master/
    escenarios/PLAMUS/config_%sC.xml' %cap_viaria1, 'wb') as f:
42     f.write(header)
43     mytree.write(f)
44 with open('/Users/fernandabmallon/Documents/Mestrado/Dissertacao/
    Processamento_Dados/MATSim_Cadyts/matsim-example-project-master/
    escenarios/PLAMUS/config.xml', 'wb') as f:
45     f.write(header)
46     mytree.write(f)
47
48
49 p = subprocess.Popen("java -jar /Users/fernandabmallon/Documents/
    Mestrado/Dissertacao/Processamento_Dados/MATSim_Cadyts/matsim-
    example-project-master/RunMATSim2.jar", stdout=subprocess.PIPE,
    shell=True)
50
51 print(p.communicate())

```

E Script de elaboração do *transitSchedule.xml*

```
1 # Código para criação dos arquivos de entrada do transporte coletivo de
   Florianópolis
2 # Fernanda Beckhauser Mallon
3 #=====
4 # IMPORTAÇÕES
5 #=====
6 import psycopg2
7 import lxml.etree as ET
8 from pathlib import Path
9 import subprocess
10 import datetime
11 import numpy
12 import sys
13 import csv
14 import resource
15 import pandas as pd
16 #=====
17 # CONFIGURAÇÕES
18 #=====
19 db_string = "host=localhost dbname=dissertacao user=fernandabmallon"
20 rotas_path = Path("/Users/fernandabmallon/Google_Drive/UFSC/Mestrado/
   Shapefile/rotas.shp")
21 pontos_path = Path("/Users/fernandabmallon/Google_Drive/UFSC/Mestrado/
   Shapefile/pontos_onibus.shp")
22 network_path = Path("/Users/fernandabmallon/Google_Drive/UFSC/Mestrado/
   Shapefile/Rede_Viaria_FLN/network_PLAMUS_links.shp")
23 nodes_path = Path("/Users/fernandabmallon/Google_Drive/UFSC/Mestrado/
   Shapefile/Rede_Viaria_FLN/network_PLAMUS_nodes.shp")
24 # =====
25 # INCLUSÃO DA REDE VIÁRIA NO BANCO DE DADOS
26 # =====
27 class DataToPGIS():
28     def __init__( self ):
29         con = psycopg2.connect(db_string)
30         con.set_isolation_level(psycopg2.extensions.
   ISOLATION_LEVEL_AUTOCOMMIT)
31         with con:
32             cur = con.cursor()
33 #Limpar base de dados entre iterações
34             cur.execute( "DROP TABLE IF EXISTS rotas;" )
35             cur.execute( "DROP TABLE IF EXISTS pontos_onibus;" )
36             cur.execute( "DROP TABLE IF EXISTS network_PLAMUS_links;" )
37             cur.execute( "DROP TABLE IF EXISTS network_PLAMUS_nodes;" )
38
39 #Inserir tabela coordenadas e roads na base de dados dissertacao
40
41             self.system_run( "shp2pgsql -S %s | psql -d dissertacao" %
   rotas_path)
42             self.system_run( "shp2pgsql -S %s | psql -d dissertacao" %
   pontos_path)
43             self.system_run( "shp2pgsql -S %s | psql -d dissertacao" %
   network_path)
```

```

44         self.system_run( "shp2pgsql -S %s | psql -d dissertacao" %
                           nodes_path)
45
46
47     def system_run( self, input ):
48         print( "-----" )
49         print( "comando: %s" % input )
50         result = subprocess.call( input, shell=True )
51
52 # =====
53 # MATSIM TRANSIT SCHEDULE BUILDER
54 # =====
55
56 class MATSimTransitScheduleBuilder():
57     def __init__( self ):
58         con = psycopg2.connect(db_string)
59         con.set_isolation_level(psycopg2.extensions.
60                                ISOLATION_LEVEL_AUTOCOMMIT)
61         with con:
62             inicio = datetime.datetime.now()
63
64             cur = con.cursor()
65
66             transitSchedule = ET.Element("transitSchedule")
67
68             transitStops = ET.SubElement(transitSchedule, "transitStops")
69
70             # Selecionar pontos de ônibus que fazem parte das rotas
71             cur.execute("SELECT ST_AsText(ST_Collect(geom)) FROM rotas
72                         WHERE FREQHPT IS NOT NULL AND FREQHPT IS NOT NULL")
73
74             geom_rotas = cur.fetchall()
75
76             print("SELECT fid FROM pontos_onibus p WHERE ST_Within(
77                   ST_SetSRID(p.geom,31982),ST_SetSRID(ST_Buffer(
78                   ST_GeomFromText('%s'),20),31982))" %(geom_rotas[0][0]))
79             cur.execute("SELECT fid,ST_x(geom),ST_Y(geom) FROM
80                   pontos_onibus p WHERE ST_Within(ST_SetSRID(p.geom,31982),
81                   ST_SetSRID(ST_Buffer(ST_GeomFromText('%s'),15),31982))" %(
82                   geom_rotas[0][0]))
83             pontos_rota = cur.fetchall()
84             print("pontos_rota=",pontos_rota)
85
86             cur.execute("SELECT ID FROM rede_viaria_fln l WHERE tipologia_
87                   < 90 and ST_Within(ST_SetSRID(l.geom,31982),ST_SetSRID(
88                   ST_Buffer(ST_GeomFromText('%s'),5),31982))" %(geom_rotas
89                   [0][0]))
90             links_rotas = cur.fetchall()
91             print("links_rotas=",links_rotas)
92             id_links_tc = []
93             for link_rota in links_rotas:
94                 id_links_tc.append(int(link_rota[0]))
95             print("id_links_tc=",id_links_tc,"=id_links_tc")
96
97             # Selecionar os link da malha viária que representam os pontos
98             de ônibus

```

```

88     dict_links_pontos = {}
89     lista_link = []
90     for ponto_rota in pontos_rota:
91
92         cur.execute("SELECT ST_AsText(geom) FROM pontos_onibus
93             WHERE fid = '%s'" % ponto_rota[0])
94         geom_ponto = cur.fetchall()
95
96         cur.execute("SELECT id,dir,ST_X(ST_LineInterpolatePoint(
97             geom, 0.5)),ST_Y(ST_LineInterpolatePoint(geom, 0.5)),
98             ST_Distance(ST_SETSRID(ST_GeomFromText('%s'),31982),
99             ST_SetSRID(l.geom,31982)) FROM rede_viaria_fln l WHERE
100             ID IN %s ORDER BY 5 LIMIT 1" %(geom_ponto[0][0],tuple(
101             id_links_tc)))
102         #cur.execute("SELECT id,dir,ST_Distance(ST_SETSRID(
103             ST_GeomFromText('%s'),31982),ST_SetSRID(l.geom,31982))
104             FROM rede_viaria_fln l WHERE tipologia_ < 90 ORDER BY 3
105             LIMIT 1" %(geom_ponto[0][0]))
106         id_link_ponto = cur.fetchall()
107         print("id_link_ponto=",id_link_ponto)
108
109         if id_link_ponto[0][0] in lista_link:
110
111             print("link repetido")
112
113         elif id_link_ponto[0][1] == 1:
114
115             lista_link.append(id_link_ponto[0][0])
116             id_link1 = str(int(id_link_ponto[0][0])) + '01'
117
118             print("ponto ",ponto_rota[0],": link ",id_link1)
119
120             ponto = str(int(ponto_rota[0]))+'01'
121             dict_links_pontos[ponto] = int(id_link1)
122
123             stopFacility = ET.SubElement(transitStops, "
124                 stopFacility",
125                 attrib={'id':str(id_link1)+'_pt',
126                     'x':str(id_link_ponto[0][2]),
127                     'y':str(id_link_ponto[0][3]),
128                     'linkRefId':id_link1})
129
130         elif id_link_ponto[0][1] == -1:
131
132             lista_link.append(id_link_ponto[0][0])
133             id_link2 = str(int(id_link_ponto[0][0])) + '02'
134
135             print("ponto ",ponto_rota[0],": link ",id_link2)
136
137             ponto = str(int(ponto_rota[0]))+'02'
138             dict_links_pontos[ponto] = int(id_link2)
139
140             stopFacility = ET.SubElement(transitStops, "
141                 stopFacility",
142                 attrib={'id':str(id_link2)+'_pt',
143                     'x':str(id_link_ponto[0][2]),

```

```

133         'y':str(id_link_ponto[0][3]),
134         'linkRefId':id_link2})
135
136     else:
137
138         lista_link.append(id_link_ponto[0][0])
139         id_link1 = str(int(id_link_ponto[0][0])) + '01'
140         id_link2 = str(int(id_link_ponto[0][0])) + '02'
141
142         print("ponto ",ponto_rota[0],": link ",id_link1)
143         print("ponto ",ponto_rota[0],": link ",id_link2)
144
145         ponto = str(int(ponto_rota[0]))+'01'
146         dict_links_pontos[ponto] = int(id_link1)
147         ponto = str(int(ponto_rota[0]))+'02'
148         dict_links_pontos[ponto] = int(id_link2)
149
150         stopFacility = ET.SubElement(transitStops, "
151             stopFacility",
152             attrib={'id':str(id_link1)+'_pt',
153                 'x':str(id_link_ponto[0][2]),
154                 'y':str(id_link_ponto[0][3]),
155                 'linkRefId':id_link1})
156
157         stopFacility = ET.SubElement(transitStops, "
158             stopFacility",
159             attrib={'id': str(id_link2)+'_pt',
160                 'x':str(id_link_ponto[0][2]),
161                 'y':str(id_link_ponto[0][3]),
162                 'linkRefId':id_link2})
163
164     # Seleção das rotas
165     qtdd_rotas_retiradas = 0
166     rotas_retiradas = []
167     rotas_simuladas = []
168     id_veiculo = 1
169     cur.execute("SELECT NOME_ROTA FROM rotas WHERE nome_rota != '
170         BI0043102_IV' and nome_rota != 'BI0044801_MV' and nome_rota
171         != 'BI0066900_II' and nome_rota != 'BI0066900_IV' and
172         nome_rota != 'J00000635_MI' and nome_rota != 'TR0000110_MI'
173         and nome_rota != 'CA0000250_MI' and nome_rota != '
174         CA0000272_MI' and nome_rota != 'J00000003_MV' and nome_rota
175         != 'J00000019_MV' and nome_rota != 'J00000632_MV' and
176         nome_rota != 'TR0000134_MI' and FREQHPM IS NOT NULL AND
177         FREQHPT IS NOT NULL ORDER BY ROUTE_ID DESC")
178     #cur.execute("SELECT NOME_ROTA FROM rotas WHERE NOME_ROTA = '
179         BI0011300_IV' ")
180     rotas = cur.fetchall()
181     num = 0
182     for rota in rotas:
183         num = num + 1
184
185     # Seleção da geometria da rota

```

```

177 cur.execute("SELECT ST_AsText(ST_Collect(geom)) FROM rotas
178 WHERE NOME_ROTA = '%s'" %rota[0])
179 geom_rota = cur.fetchall()
180
181
182 # Criação do grafo de rede da rota de TC para roteamento no
183 pgrouting
184 cur.execute( "DROP TABLE IF EXISTS links_pgrouting;" )
185 cur.execute( "CREATE TABLE links_pgrouting (id integer,
186 source integer, target integer, cost real, reverse_cost
187 real);" )
188
189 cur.execute("SELECT id,dir FROM rede_viaria_fln l WHERE
190 tipologia_ < 90 and ST_Within(ST_SetSRID(l.geom,31982),
191 ST_SetSRID(ST_Buffer(ST_GeomFromText('%s'),5),31982))"
192 %(geom_rota[0][0]))
193 id_links_rota = cur.fetchall()
194 links_rota = []
195 links_undirected_rota = []
196
197 for id_link_rota in id_links_rota:
198 links_undirected_rota.append(id_link_rota[0])
199
200 if id_link_rota[1] == 0:
201
202     id_link1 = str(int(id_link_rota[0])) + '01'
203     id_link2 = str(int(id_link_rota[0])) + '02'
204     cur.execute("SELECT id, fromid, toid, length,
205 freespeed FROM network_PLAMUS_links l WHERE id
206 = '%s'" %(id_link1))
207 dados_links = cur.fetchall()
208
209     id1 = int(dados_links[0][0])
210     fromid = int(dados_links[0][1])
211     toid = int(dados_links[0][2])
212     length = float(dados_links[0][3])
213     freespeed = float(dados_links[0][4]) #velocidade em
214     m/s
215     tempo = numpy.maximum(float(length/freespeed), 1)
216     reverse_tempo = 0 - tempo
217     print("fromid:",fromid,"toid:",toid,"tempo:",tempo)
218     links_rota.append(dados_links[0][0])
219     cur.execute("INSERT INTO links_pgrouting (id,
220 source, target, cost, reverse_cost) VALUES (%d
221 ,%d,%d,%d,%d)" %(id1,fromid,toid,tempo,
222 reverse_tempo))
223
224     cur.execute("SELECT id, fromid, toid, length,
225 freespeed FROM network_PLAMUS_links l WHERE id
226 = '%s'" %(id_link2))
227 dados_links = cur.fetchall()
228
229     id2 = int(dados_links[0][0])
230     fromid = int(dados_links[0][1])
231     toid = int(dados_links[0][2])

```

```

218         length = float(dados_links[0][3])
219         freespeed = float(dados_links[0][4]) #velocidade em
           m/s
220         tempo = numpy.maximum(float(length/freespeed), 1)
221         reverse_tempo = 0 - tempo
222         print("fromid:",fromid,"toid:",toid,"tempo:",tempo)
223         links_rota.append(dados_links[0][0])
224         cur.execute("INSERT INTO links_pgrouting (id,
           source, target, cost, reverse_cost) VALUES (%d
           ,%d,%d,%d,%d)" %(id2,fromid,toid,tempo,
           reverse_tempo))

225
226     elif id_link_rota[1] == 1:
227
228         id_link = str(int(id_link_rota[0])) + '01'
229         print("id_link=",id_link)
230
231         cur.execute("SELECT id, fromid, toid, length,
           freespeed FROM network_PLAMUS_links l WHERE id
           = '%s'" %(id_link))
232         dados_links = cur.fetchall()
233
234         id1 = int(dados_links[0][0])
235         fromid = int(dados_links[0][1])
236         toid = int(dados_links[0][2])
237         length = float(dados_links[0][3])
238         freespeed = float(dados_links[0][4]) #velocidade em
           m/s
239         tempo = numpy.maximum(float(length/freespeed), 1)
240         reverse_tempo = 0 - tempo
241         print("fromid:",fromid,"toid:",toid,"tempo:",tempo)
242         links_rota.append(dados_links[0][0])
243         cur.execute("INSERT INTO links_pgrouting (id,
           source, target, cost, reverse_cost) VALUES (%d
           ,%d,%d,%d,%d)" %(id1,fromid,toid,tempo,
           reverse_tempo))

244
245     elif id_link_rota[1] == -1:
246         id_link = str(int(id_link_rota[0])) + '02'
247         cur.execute("SELECT id, fromid, toid, length,
           freespeed FROM network_PLAMUS_links l WHERE id
           = '%s'" %(id_link))
248         dados_links = cur.fetchall()
249
250         id2 = int(dados_links[0][0])
251         fromid = int(dados_links[0][1])
252         toid = int(dados_links[0][2])
253         length = float(dados_links[0][3])
254         freespeed = float(dados_links[0][4]) #velocidade em
           m/s
255         tempo = numpy.maximum(float(length/freespeed), 1)
256         reverse_tempo = 0 - tempo
257         print("fromid:",fromid,"toid:",toid,"tempo:",tempo)
258         links_rota.append(dados_links[0][0])
259         cur.execute("INSERT INTO links_pgrouting (id,
           source, target, cost, reverse_cost) VALUES (%d

```

```

                                ,%d,%d,%d,%d)" %(id2,fromid,toid,tempo,
                                reverse_tempo))
260
261
262
263     links_rota2 = list( dict.fromkeys(links_undirected_rota) )
264     links_rota1 = list( dict.fromkeys(links_rota) )
265     print("links_rota1=",links_rota1)
266     cur.execute("SELECT pgr_createTopology('links_pgrouting',
                0.001, 'geom', clean := TRUE)")
267     cur.execute("SELECT pgr_analyzegraph('links_pgrouting',
                0.001, 'geom')")
268     cur.execute("SELECT pgr_nodeNetwork('links_pgrouting',
                0.001, id := 'id', the_geom:='geom')")
269     # Seleção das coordenadas de início e fim da rota
270     cur.execute("SELECT ST_AsText(ST_StartPoint(geom)) FROM
                rotas WHERE NOME_ROTA = '%s'" %rota[0])
271     ponto_inicio_rota = cur.fetchall()[0][0]
272     print("ponto_inicio_rota=",ponto_inicio_rota)
273
274     cur.execute("SELECT ST_AsText(ST_EndPoint(geom)) FROM rotas
                WHERE NOME_ROTA = '%s'" %rota[0])
275     ponto_final_rota = cur.fetchall()[0][0]
276     print("ponto_final_rota=",ponto_final_rota)
277
278
279
280
281     # Selecionar nodes que fazem parte da rota
282     nodes_pontos = []
283     nodes_pontos1 = []
284
285     # Criar dicionário para armazenar relação dos pontos de ô
                nibus X vértices
286     dict_ponto_node = {}
287
288     # Selecionar os nós da malha viária que representam os
                pontos de ônibus
289     for id_link_rota in id_links_rota:
290
291         if id_link_rota[1] == 0:
292
293             id_link1 = str(int(id_link_rota[0])) + '01'
294             id_link2 = str(int(id_link_rota[0])) + '02'
295
296             cur.execute("SELECT toid FROM network_plamus_links
                WHERE id = '%s'" %id_link1)
297             id_node_link = cur.fetchall()
298
299             nodes_pontos.append(int(id_node_link[0][0]))
300             nodes_pontos1.append(id_node_link[0][0])
301
302             cur.execute("SELECT toid FROM network_plamus_links
                WHERE id = '%s'" %id_link2)
303             id_node_link = cur.fetchall()
304

```

```

305         nodes_pontos.append(int(id_node_link[0][0]))
306         nodes_pontos1.append(id_node_link[0][0])
307
308         elif id_link_rota[1] == 1:
309
310             id_link1 = str(int(id_link_rota[0])) + '01'
311
312             cur.execute("SELECT toid FROM network_plamus_links
313                 WHERE id = '%s'" %id_link1)
314             id_node_link = cur.fetchall()
315
316             nodes_pontos.append(int(id_node_link[0][0]))
317             nodes_pontos1.append(id_node_link[0][0])
318
319         elif id_link_rota[1] == -1:
320
321             id_link2 = str(int(id_link_rota[0])) + '02'
322
323             cur.execute("SELECT toid FROM network_plamus_links
324                 WHERE id = '%s'" %id_link2)
325             id_node_link = cur.fetchall()
326
327             nodes_pontos.append(int(id_node_link[0][0]))
328             nodes_pontos1.append(id_node_link[0][0])
329
330     # Remover duplicatas da lista dos nodes que compõem a rota,
331     # desordenados
332     nodes_rotas = list( dict.fromkeys(nodes_pontos) )
333     nodes_rotas1 = list( dict.fromkeys(nodes_pontos1) )
334
335     print("nodes_rotas=", nodes_rotas)
336     print("rota=", rota[0], "num=", num, "/", len(rotas))
337
338     # Selecionar primeiro node da rota
339     cur.execute("SELECT id, ST_Distance(ST_SetSRID(
340         ST_GeomFromText('%s'), 31982), ST_SetSRID(n.geom, 31982))
341         FROM network_plamus_nodes n WHERE id IN %s ORDER BY 2
342         LIMIT 1"%(ponto_inicio_rota, tuple(nodes_rotas1)))
343     node_inicio_rota = int(cur.fetchall()[0][0])
344
345     # Organizar listas para ordenação dos nodes e links que
346     # compõem a rota
347     nodes_rotas.remove(node_inicio_rota)
348     node_origem = node_inicio_rota
349     nodes_rotas_orden = []
350     nodes_rotas_orden.append(node_inicio_rota)
351     nodes_rota = numpy.array(nodes_rotas)
352     links_rota = []
353     id_links_rota = []
354     # Ordenar lista de nodes
355     for node_rota in nodes_rotas[:]:
356
357         print("rota=", rota[0], "num=", num, "/", len(rotas))

```

```

352     print("SELECT * FROM pgr_Dijkstra('select id, source,
        target, cost, reverse_cost from links_pgrounting', %
        d, ARRAY%s, false)" %(node_origem,nodes_rotas))
353     cur.execute("SELECT * FROM pgr_Dijkstra('select id,
        source, target, cost, reverse_cost from
        links_pgrounting', %d, ARRAY%s, true)" %(node_origem
        ,nodes_rotas))
354     result_dijkstra = cur.fetchall()
355     print("result_dijkstra=",result_dijkstra)
356
357     if len(result_dijkstra) == 0:
358
359         nodes_rotas = []
360         nodes_rotas_orden = []
361         links_rota = []
362         break
363     else:
364
365         filtered_nodes = [x for x in result_dijkstra if x
        [4] == -1]
366         next_node_info = min(filtered_nodes, key=lambda x:
        x[6])
367         print("next_node_info=",next_node_info)
368
369
370         # Ajustar lista de nodes
371         node_destino = next_node_info[2]
372         print("node_destino=",node_destino)
373         nodes_rotas.remove(node_destino)
374         nodes_rotas_orden.append(node_destino)
375         print("nodes_rotas=",nodes_rotas)
376         print("nodes_rotas_orden=",nodes_rotas_orden)
377
378         # Inserir lista de links
379
380         path_seq_final = next_node_info[1]
381         path_seq_list = list(range(1, path_seq_final))
382         seq_final = next_node_info[0]
383         print("seq_final=",seq_final)
384         seq_list = list(range(seq_final-path_seq_final+1,
        seq_final))
385         filtered_links = [x[4] for x in result_dijkstra if
        x[1] in path_seq_list and x[0] in seq_list]
386         links_rota.extend(filtered_links)
387         print("SELECT id FROM network_plamus_links l WHERE
        fromid = '%s' and toid = '%s' and ID IN %s" %(
        str(node_origem),str(node_destino),tuple(
        links_rota1)))
388         cur.execute("SELECT id FROM network_plamus_links l
        WHERE fromid = '%s' and toid = '%s' and ID IN %
        s" %(str(node_origem),str(node_destino),tuple(
        links_rota1)))
389         id_link = cur.fetchall()
390
391         if len(id_link) == 0:
392

```

```

393         nodes_rotas_orden = []
394     else:
395
396         node_origem = node_destino
397         id_links_rota.append(id_link[0][0])
398
399     links_rota = list(set(links_rota)) # Lista de links com
        direção definida por 01 ou 02 no final do ID, possuem
        tonode fromnode
400     print("links_rota=", links_rota)
401     print("nodes_rotas=", nodes_rotas)
402     print("nodes_rotas_orden=", nodes_rotas_orden)
403
404
405     #sys.exit()
        #####
406
407     #Seleção dos pontos da rota
408     dict_links_pontos1 = {}
409     lista_link = []
410     links_rota2 = links_rota2
411
412     cur.execute("SELECT fid FROM pontos_onibus p WHERE
        ST_Within(ST_SetSRID(p.geom, 31982), ST_SetSRID(ST_Buffer
        (ST_GeomFromText('%s'), 15), 31982))" %(geom_rota[0][0]))
413     pontos_rota = cur.fetchall()
414
415     for ponto_rota in pontos_rota:
416
417         cur.execute("SELECT ST_AsText(geom) FROM pontos_onibus
        WHERE fid = '%s'" %ponto_rota[0])
418         geom_ponto = cur.fetchall()
419
420         print("SELECT id, dir, ST_Distance(ST_SetSRID(
        ST_GeomFromText('%s'), 31982), ST_SetSRID(l.geom
        , 31982)) FROM rede_viaria_fln l WHERE ID IN %s
        ORDER BY 3 LIMIT 1" %(geom_ponto[0][0], tuple(
        links_rota2)))
421     cur.execute("SELECT id, dir, ST_Distance(ST_SetSRID(
        ST_GeomFromText('%s'), 31982), ST_SetSRID(l.geom
        , 31982)) FROM rede_viaria_fln l WHERE ID IN %s
        ORDER BY 3 LIMIT 1" %(geom_ponto[0][0], tuple(
        links_rota2)))
422     id_link_ponto = cur.fetchall()
423     print("id_link_ponto[0][0]=", int(id_link_ponto[0][0]))
424     print("list(dict_links_pontos.values())=", list(
        dict_links_pontos.values()))
425     id_link_1 = int(id_link_ponto[0][0])*100 + 1
426     id_link_2 = int(id_link_ponto[0][0])*100 + 2
427     if id_link_1 in list(dict_links_pontos.values()) or
        id_link_2 in list(dict_links_pontos.values()):
428
429
430         if id_link_ponto[0][0] in lista_link:
431
432             id_link = str(int(id_link_ponto[0][0]))

```

```

433
434         elif id_link_ponto[0][1] == 1:
435
436             lista_link.append(id_link_ponto[0][0])
437             id_link1 = str(int(id_link_ponto[0][0])) + '01'
438
439             print("ponto ", ponto_rota[0], ": link ", id_link1
440                   )
441
442             ponto = str(int(ponto_rota[0]))+'01'
443             dict_links_pontos1[ponto] = int(id_link1)
444
445         elif id_link_ponto[0][1] == -1:
446
447             lista_link.append(id_link_ponto[0][0])
448             id_link2 = str(int(id_link_ponto[0][0])) + '02'
449
450             print("ponto ", ponto_rota[0], ": link ", id_link2
451                   )
452
453             ponto = str(int(ponto_rota[0]))+'02'
454             dict_links_pontos1[ponto] = int(id_link2)
455
456         else:
457
458             lista_link.append(id_link_ponto[0][0])
459             id_link1 = int(id_link_ponto[0][0])*100 + 1
460             id_link2 = int(id_link_ponto[0][0])*100 + 2
461
462             #Verificar se está na lista de links da rota
463             #direcionada
464             if id_link1 in links_rota:
465
466                 print("ponto ", ponto_rota[0], ": link ",
467                       id_link1)
468
469                 ponto = str(int(ponto_rota[0]))+'01'
470                 dict_links_pontos1[ponto] = int(id_link1)
471
472             if id_link2 in links_rota:
473
474                 print("ponto ", ponto_rota[0], ": link ",
475                       id_link2)
476
477                 ponto = str(int(ponto_rota[0]))+'02'
478                 dict_links_pontos1[ponto] = int(id_link2)
479
480
481         if len(nodes_rotas_orden) == 0:
482
483             qtdd_rotas_retiradas = qtdd_rotas_retiradas + 1
484             rotas_retiradas.append(rota[0])
485
486         else:
487
488             rotas_simuladas.append(rota[0])

```

```

484
485 # Indicação dos dados básicos da rota
486 transitLine = ET.SubElement(transitSchedule, "
487     transitLine",
488     attrib={'id':str(rota[0])})
489
490 cur.execute("SELECT ID FROM rotas WHERE NOME_ROTA= '%s
491     '" %rota[0])
492 id_rota = cur.fetchall()
493
494 transitRoute = ET.SubElement(transitLine, "transitRoute
495     ",
496     attrib={'id':str(id_rota[0][0])})
497
498 transportMode = ET.SubElement(transitRoute, "
499     transportMode").text="pt"
500
501 routeProfile = ET.SubElement(transitRoute, "
502     routeProfile")
503
504 # Indicação dos pontos de parada
505
506 qtdd_pontos = sum(links_rota.count(item) for item in
507     list(dict_links_pontos1.values()))
508 print("qtdd_pontos =",qtdd_pontos)
509 cur.execute("SELECT TEMPO_COLE FROM rotas WHERE
510     NOME_ROTA = '%s'" %rota[0])
511 tempo_rota = cur.fetchall()[0][0]
512 if qtdd_pontos > 1:
513     tempo_pontos = int((tempo_rota*60)/qtdd_pontos)
514 else:
515     tempo_pontos = 0
516
517 print("tempo_pontos=",tempo_pontos)
518 departureOffset = 0
519
520 i = 1
521 for node_rota_orden in nodes_rotas_orden:
522
523     cur.execute("SELECT id FROM network_plamus_links l
524         WHERE toid = '%s' and ID IN %s" %(str(
525             node_rota_orden),tuple(links_rota1)))
526     id_link = cur.fetchall()[0][0] #string
527
528     if int(id_link) not in list(dict_links_pontos1.
529         values()):
530         print("id_link=",id_link)
531         #print("list(dict_links_pontos1.values())=",
532             list(dict_links_pontos1.values()))
533         print("link no itinerário, mas sem ponto de
534             parada:",id_link)
535
536     elif i == 1:
537         arrivalOffset = departureOffset + 30

```

```

527         stop = ET.SubElement(routeProfile, "stop",
528                               attrib={'refId':str(id_link)+'_pt',
529                                       'departureOffset':str(datetime.timedelta(
530                                           seconds = departureOffset))})
531
532         departureOffset = departureOffset +
533                             tempo_pontos
534         i = i + 1
535
536     elif i == qtdd_pontos:
537         arrivalOffset = departureOffset + 30
538
539         stop = ET.SubElement(routeProfile, "stop",
540                               attrib={'refId':str(id_link)+'_pt',
541                                       'arrivalOffset':str(datetime.timedelta(
542                                           seconds = arrivalOffset)),
543                                       'departureOffset':str(datetime.timedelta(
544                                           seconds = departureOffset)),
545                                       'awaitDeparture':"false"})
546
547         departureOffset = departureOffset +
548                             tempo_pontos
549         i = i + 1
550
551     else:
552         arrivalOffset = departureOffset + 30
553
554         stop = ET.SubElement(routeProfile, "stop",
555                               attrib={'refId':str(id_link)+'_pt',
556                                       'departureOffset':str(datetime.timedelta(
557                                           seconds = departureOffset)),
558                                       'arrivalOffset':str(datetime.timedelta(
559                                           seconds = arrivalOffset)),
560                                       'awaitDeparture':"false"})
561
562         departureOffset = departureOffset +
563                             tempo_pontos
564         i = i + 1
565
566     route = ET.SubElement(transitRoute, "route")
567
568     primeiro_node = nodes_rotas_orden[0]
569     ultimo_node = nodes_rotas_orden[-1]
570     print("primeiro_node=", primeiro_node, "ultimo_node=",
571           ultimo_node)
572
573     for id_link_rota in id_links_rota:
574
575         link = ET.SubElement(route, "link",
576                               attrib={'refId':str(int(id_link_rota))})
577
578     departures = ET.SubElement(transitRoute, "departures")

```

```

574 cur.execute("SELECT FREQHPM,FREQHPT FROM rotas WHERE
      NOME_ROTA = '%s'" %rota[0])
575 freq_HP = cur.fetchall()
576 print("freq_HP=",freq_HP)
577 if freq_HP[0][0] is not None and freq_HP[0][1] is not
      None:
578     freq_vale = int((freq_HP[0][0]+freq_HP[0][1])/0.2 -
      freq_HP[0][0] - freq_HP[0][1])/15
579     viagens_pico = (freq_HP[0][0]+freq_HP[0][1])
580     qtdd_viag_diarias = int((freq_HP[0][0]+freq_HP
      [0][1])/0.2)
581     print("qtdd_viag_diarias=",qtdd_viag_diarias)
582 elif freq_HP[0][0] is None and freq_HP[0][1] is not
      None:
583     freq_vale = int(freq_HP[0][1]/0.1 - freq_HP[0][1])
      /15
584     viagens_pico = 2*(freq_HP[0][1])
585     qtdd_viag_diarias = int(freq_HP[0][1]/0.1)
586 elif freq_HP[0][1] is None and freq_HP[0][0] is not
      None:
587     freq_vale = int(freq_HP[0][0]/0.1 - freq_HP[0][0])
      /15
588     viagens_pico = 2*(freq_HP[0][0])
589     qtdd_viag_diarias = int(freq_HP[0][0]/0.1)
590 else:
591     freq_vale = 0
592     qtdd_viag_diarias = 0
593
594
595 print("qtdd_viag_diarias=",qtdd_viag_diarias)
596 cur.execute("SELECT INTHPM,INTHPT FROM rotas WHERE
      NOME_ROTA = '%s'" %rota[0])
597 interv_HP = cur.fetchall()
598 if interv_HP[0][0] is None and interv_HP[0][1] is not
      None:
599     int_HPM = 60*interv_HP[0][1]
600 elif interv_HP[0][1] is None and interv_HP[0][0] is not
      None:
601     int_HPT = 60*interv_HP[0][0]
602 elif interv_HP[0][0] is not None and interv_HP[0][1] is
      not None:
603     int_HPM = 60*interv_HP[0][0]
604     int_HPT = 60*interv_HP[0][1]
605 else:
606     int_HPM = 0
607     int_HPT = 0
608 if qtdd_viag_diarias != 0:
609
610     int_vale = (15/(qtdd_viag_diarias-viagens_pico))
      *3600 #Previendo funcionamento de 15h/dia: das 5
      h às 23h00min, retirando as quatro horas de
      pico (uma na manhã e outra na tarde)
611 else:
612     int_vale = 0
613 print("int_HPM=",int_HPM)
614 print("int_vale=",int_vale)

```

```

615         cur.execute("SELECT TIPOVEICUL FROM rotas WHERE
616             NOME_ROTA = '%s'" %rota[0])
617         tipo_veic = cur.fetchall()
618
619         hora_saida = 19800
620         intervalo = 0
621         id_viagem = 1
622         mu, sigma = 0, 1
623         samples = numpy.random.normal(mu, sigma, 1)
624         ajuste = 900 * samples[0]
625         hora_saida = hora_saida + ajuste
626
627         for viagem in range(qtdd_viag_diarias):
628
629             if hora_saida > 25200 and hora_saida < 32400:
630                 departure = ET.SubElement(departures, "
631                     departure",
632                     attrib={'id':str(id_viagem),
633                         'departureTime':str(datetime.timedelta(
634                             seconds = int(hora_saida))),
635                         'vehicleRefId':str(tipo_veic[0][0]) + str
636                             (" ") + str(id_veiculo)})
637
638                 hora_saida = hora_saida + int_HPM
639                 id_veiculo = id_veiculo + 1
640                 id_viagem = id_viagem + 1
641
642             elif hora_saida > 61200 and hora_saida < 68400:
643                 departure = ET.SubElement(departures, "
644                     departure",
645                     attrib={'id':str(id_viagem),
646                         'departureTime':str(datetime.timedelta(
647                             seconds = int(hora_saida))),
648                         'vehicleRefId':str(tipo_veic[0][0]) + str
649                             (" ") + str(id_veiculo)})
650
651                 hora_saida = hora_saida + int_HPT
652                 id_veiculo = id_veiculo + 1
653                 id_viagem = id_viagem + 1
654             else:
655                 departure = ET.SubElement(departures, "
656                     departure",
657                     attrib={'id':str(id_viagem),
658                         'departureTime':str(datetime.timedelta(
659                             seconds = int(hora_saida))),
660                         'vehicleRefId':str(tipo_veic[0][0]) + str
661                             (" ") + str(id_veiculo)})
662
663                 hora_saida = hora_saida + int_vale
664                 id_veiculo = id_veiculo + 1
665                 id_viagem = id_viagem + 1

```

```

661         tree = ET.ElementTree(transitSchedule)
662
663         tree.write("transitSchedule.xml", pretty_print = True)
664         print("qtdd_rotas_retiradas=", qtdd_rotas_retiradas)
665         print("rotas_retiradas=", rotas_retiradas)
666         print("rotas_simuladas=", rotas_simuladas)
667
668         fim = datetime.datetime.now()
669         tempo_decorrido = fim - inicio
670         print(f"Tempo de processamento: {tempo_decorrido}")
671
672
673 if __name__ == "__main__":
674
675     #dataToPGIS = DataToPGIS()
676
677     MATSimTransitScheduleBuilder = MATSimTransitScheduleBuilder()

```