
Controle de Semáforos para Veículos de Emergência

Rodrigo Gonçalves de Branco

SERVIÇO DE PÓS-GRADUAÇÃO DA FACOM - UFMS

Data de Depósito:

Assinatura:_____

Controle de Semáforos para Veículos de Emergência

Rodrigo Gonçalves de Branco

Orientador: *Prof. Dr. Edson Norberto Cáceres*

Coorientador: *Prof. Dr. Ronaldo Alves Ferreira*

Tese apresentada à Faculdade de Computação - UFMS como parte dos requisitos necessários à obtenção do título de Doutor em Ciência da Computação.

UFMS - Campo Grande
Dezembro/2025

*À minha esposa,
Juliana,*

*Aos meus pais,
Nei e Leda,*

*Ao meu irmão,
Leandro,*

*Aos meus sobrinhos,
Maria, Enzo, Mateus e Bento, e*

Ao Blue.

Agradecimentos

Agradeço primeiramente ao meu orientador, Prof. Dr. Edson Norberto Cáceres, que me acompanhou por toda essa jornada, desde o início até este momento, contribuindo com sua experiência, avaliação e corrigindo os rumos sempre que necessário.

Agradeço ao meu Coorientador, Prof. Dr. Ronaldo Alves Ferreira, que, além de servir de inspiração no que diz respeito à pesquisa e à área acadêmica, ainda colaborou sobremaneira na forma e no conteúdo desta Tese de Doutorado.

Agradeço à Faculdade de Computação - FACOM e à UFMS por contribuírem com uma formação acadêmica sólida que permitiu culminar neste documento.

Esta pesquisa é parte do INCT da Internet do Futuro para Cidades Inteligentes, financiado por CNPq (proc. 465446/2014-0), Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001 e FAPESP (procs. 14/50937-1 e 15/24485-9). Portanto, meus agradecimentos à CAPES, à FAPESP e ao INCT.

Agradeço também aos meus órgãos laborais, cujo apoio foi fundamental ao longo desta jornada. Aos que estiveram presentes antes mesmo do início da escrita, como o TJMS, o IFMS e o TRT14, e àqueles que me acompanharam durante o processo: o TRT24, o Ibict e, mais recentemente, a Câmara dos Deputados. É difícil conceber produzir um trabalho com exigências intelectuais e de tempo dispendido sem o apoio necessário para que isto seja feito.

Agradeço aos membros da banca examinadora pela leitura atenta e pelas valiosas sugestões que, sem dúvida, enriquecem a versão final deste trabalho, demonstrando a importância do seu papel no processo de pesquisa.

O agradecimento mais profundo é para minha esposa, Juliana, e minha família. O tempo dedicado a analisar códigos, refletir sobre os problemas e escrever esta tese foi, muitas vezes, um tempo subtraído do nosso convívio. A compreensão e o apoio incondicional de vocês foram o alicerce para que este trabalho pudesse ser concluído.

Abstract

The high volume of vehicles in large metropolitan areas creates challenges for first responders, such as firefighters and paramedics, as it delays emergency vehicles due to traffic congestion. A standard way to reduce congestion is traffic light preemption on emergency vehicle (EV) routes. In this context, we propose an innovative traffic-light control system for emergency vehicles using smart city middleware that centralizes information and manages traffic lights along the whole EV route. Our primary contribution is the design and evaluation of an efficient preemption algorithm to reduce EV delays during emergencies. The algorithm utilizes a Timed Petri Net to coordinate concurrent events, thereby avoiding simultaneous actions at traffic lights along the EV's route. Complementarily, the time required for vehicle queue dissipation is calculated based on the shockwave principle, allowing for an accurate estimation of road clearance. Extensive simulations in both real and synthetic scenarios demonstrate that our solution reduces EV delays and outperforms three competing algorithms.

Keywords: Traffic Light Control, Intelligent Traffic Management, and Traffic Light Preemption.

Resumo

O alto volume de veículos em grandes áreas metropolitanas cria desafios para os serviços de emergência, como bombeiros e paramédicos, pois atrasa os veículos de emergência (EVs) devido ao congestionamento no trânsito. Uma abordagem padrão para reduzir o congestionamento é a preempção de semáforos nas rotas dos EVs. Neste contexto, propomos um sistema inovador de controle de semáforos para veículos de emergência utilizando um middleware de cidade inteligente que centraliza informações e gerencia os semáforos ao longo de toda a rota do EV. Nossa principal contribuição é o projeto e a avaliação de um algoritmo de preempção eficiente para reduzir os atrasos dos EVs durante emergências. O algoritmo utiliza uma Rede de Petri Temporizada (Timed Petri Net) para coordenar eventos concorrentes, evitando assim ações simultâneas nos semáforos ao longo da rota do EV. Complementarmente, o tempo necessário para a dissipação da fila de veículos é calculado com base no princípio da onda de choque, permitindo uma estimativa precisa da liberação da via. Simulações extensivas em cenários reais e sintéticos demonstram que nossa solução reduz os atrasos dos EVs e supera o desempenho de três algoritmos concorrentes.

Palavras-chave: Controle de Semáforos, Gerenciamento de Tráfego Inteligente e Preempção de Semáforos.

Sumário

Sumário	xvi
Lista de Figuras	xxii
Lista de Tabelas	xxx
Glossário	xxxix
Glossário	xxxix
Lista de Algoritmos	xxxix
1 Introdução	1
1.1 Motivação e Objetivos	5
1.2 Visão Geral da Solução	7
1.3 Principais Contribuições	9
1.4 Delimitação de Escopo e Premissas Básicas	10
2 Fundamentação Teórica e Trabalhos Relacionados	13
2.1 Descrição do Problema	14
2.2 Critérios de Avaliação e Requisitos Desejáveis	15
2.2.1 Requisitos Funcionais e de Segurança	16
2.2.2 Requisitos Metodológicos e de Validação	17
2.3 Sistemas de Detecção Baseados em Infraestrutura (Sem Comunicação Direta)	17
2.3.1 Sistemas sonoros	18
2.3.2 Sistemas ópticos	18
2.3.3 Sistemas baseados em Rádio	19
2.3.4 Sistemas baseados em Sensores	20
2.3.5 Sistemas baseados em GPS	21
2.3.6 Comparação dos sistemas de Notificação	21
2.3.7 Conclusão	21
2.4 Sistemas Cooperativos com Comunicação Direta (V2X)	22
2.5 Aplicações Avançadas: Semáforos Virtuais e Gestão Distribuída	23
2.6 Cidades Inteligentes	24
2.7 Análise de Trabalhos Relacionados	26
2.7.1 Soluções Baseadas em Detecção Local e Pontual	27
2.7.2 Soluções Baseadas em Conectividade Veicular (V2V/V2I)	27

2.7.3	Soluções Baseadas em IoT e Arquiteturas Centralizadas . .	27
2.7.4	Soluções Baseadas em IA e Métodos Formais	28
2.7.5	Síntese das Lacunas	28
2.8	Conclusão	31
3	Soluções Selecionadas para Comparação e Modelo de Simulação	33
3.1	Modelo do Sistema	34
3.2	Soluções Selecionadas Para Comparação	35
3.2.1	Soluções baseadas em RFID	36
3.2.2	Soluções baseadas em Fuzzy	37
3.2.3	Soluções baseadas em Filas	39
3.2.4	Soluções baseadas em Preempção Imediata - Onda Verde .	40
3.3	Simulação e Modelos de Tráfego	42
3.3.1	Macroscópico	42
3.3.2	Mesoscópico	42
3.3.3	Microscópico	43
3.3.4	SUMO	43
3.4	Cenários	44
3.4.1	Cenários Sintético de São Paulo e Nova Iorque	44
3.4.2	Cenário De Uma interseção	47
3.4.3	Cenários de Turim e Colônia	48
3.4.4	Cenário do Centro Expandido de São Paulo - Metrô OD SP 2017	49
3.4.5	Considerações sobre a Simulação	52
3.5	Metodologia de Avaliação	53
3.5.1	Framework de Simulação	53
3.5.2	Detalhes de Implementação	55
3.5.3	Roteamento dos Veículos na Simulação	56
3.5.4	Métricas de desempenho	57
3.5.5	Recursos Computacionais	60
3.5.6	Adaptações no Simulador	61
3.6	Conclusão	61
4	Primeira Solução - TPN'	63
4.1	Rede Petri Temporizada	63
4.2	Modelo da Solução	65
4.3	Modelo de Controle	66
4.4	Algoritmo de Controle	68
4.4.1	Detalhes de Implementação	72
4.4.2	Complexidade do Algoritmo	73
4.5	Resultados	74

4.5.1	Cenários São Paulo e Nova Iorque - Sintético	74
4.6	Conclusão	78
5	Segunda Solução - Onda de Choque	81
5.1	Motivação	81
5.2	Princípio da Onda de Choque	82
5.3	Algoritmo Onda de Choque	84
5.3.1	Complexidade do Algoritmo	86
5.4	Resultados	86
5.4.1	Cenário Uma interseção	87
5.4.2	Cenário Turim	91
5.4.3	Cenário Colônia	96
5.5	Estudo das Anomalias	101
5.6	Conclusão	107
6	Terceira Solução - Onda de Choque + TPN*	109
6.1	Visão Geral da Solução	109
6.2	Modelo de Controle	110
6.3	Propriedades da TPN* Proposta	119
6.3.1	Boundedness	120
6.3.2	Grafo de Alcançabilidade	123
6.4	Algoritmo Onda de Choque + TPN*	125
6.4.1	Complexidade do Algoritmo	128
6.5	Implementação	128
6.6	Resultados	129
6.6.1	Cenário Turim	130
6.6.2	Cenário Colônia	136
6.6.3	Cenário do Centro Expandido de São Paulo - Metrô OD SP 2017	142
6.7	Conclusão	150
7	Conclusão	153
7.1	Contribuições	154
7.2	Limitações e Trabalhos Futuros	156
7.2.1	Representatividade do Modelo de Car-Following e Compor- tamento Humano	156
7.2.2	Inteligência Artificial e Aprendizado de Máquina	156
7.2.3	Escalonamento e Concorrência de Múltiplos EVs	157
7.2.4	Granularidade da Preempção e Impacto no Entorno	157
7.2.5	Validação em Infraestrutura Real (V2I/V2V/VTL)	157
7.2.6	Fidelidade dos Cenários e Atualização da Base de Dados	158

7.3 Produção Bibliográfica	158
Referências	170
A Apêndice	171

Lista de Figuras

1.1	No dia 23 de maio de 2014, o congestionamento apurado pela Companhia de Engenharia de Tráfego (CET) na cidade de São Paulo foi de 344 km (G1 São Paulo, 2014).	2
1.2	Tráfego do Veículo de Emergência (EV) dificultado pelo congestionamento (Portal Unibus, 2013).	3
1.3	Visão simplificada do problema, com informações de posicionamento do EV e comunicação entre os semáforos e a central de controle.	7
2.1	Descrição do problema. TL_i representa o i -ésimo semáforo da rota do EV.	15
2.2	Ilustração do bloqueio de uma interseção por congestionamento a frente. Mesmo com o sinal verde favorável (seta roxa), o veículo líder (círculo superior) não avança por falta de espaço no trecho seguinte, obstruindo toda a fila. Consequentemente, veículos com rotas alternativas livres (círculo inferior, com intenção de virar à direita) também são impedidos de prosseguir.	16
2.3	Exemplo de sistema de detecção de <i>loop</i> indutivo (marcas no asfalto) (METROCOUNT, 2020).	20
2.4	Tipos de comunicação entre veículos, infraestrutura e pessoas. Fonte da imagem: < https://www.thalesgroup.com/en/markets/digital-identity-and-security/iot/industries/automotive/use-cases/v2x >.	23
2.5	Representação do <i>Semáforo Virtual</i> (VTL) no para-brisa do veículo apresentado em Olaverri-Monreal et al. (2012).	24
2.6	<i>Middleware</i> de Cidades Inteligentes	26
3.1	Distância de Detecção (<i>Read range</i>) do EV. Fonte da imagem: < https://www.nedapidentification.com/insights/understanding-rfid-read-ranges-for-automatic-vehicle-identification/ >	36
3.2	Sistema de Controle para Preempção de Semáforos durante o atendimento a uma emergência.	45

3.3	Rota do EV dentro da área do experimento do cenário de São Paulo - Sintético.	46
3.4	Rota do EV dentro da área do experimento do cenário de Nova Iorque - Sintético.	46
3.5	Cenário de uma interseção, baseado nas características do experimento descrito em Kapusta et al. (2017)	47
3.6	Rotas dos EVs dentro da área do experimento do Cenário Turim	48
3.7	Rotas dos EVs dentro da área do experimento do Cenário Colônia	50
3.8	Área Metropolitana de São Paulo e suas zonas de pesquisa (obtido de ">https://maposmatic.osm-baustelle.de/new/#>).	51
3.9	Rotas dos EVs dentro da área do experimento do cenário Centro Expandido de São Paulo (Companhia Metropolitana de São Paulo (METRÔ), 2017).	52
3.10	Framework desenvolvido para este trabalho.	54
3.11	Relação entre as métricas tl , btt e ttt	58
4.1	Blocos de Controle dos semáforos $TL_1 \dots TL_N$ para as interseções na rota do EV.	66
4.2	Modelo <i>Timed Petri Net</i> ' (TPN') proposto. Essa figura representa uma expansão de cada um dos blocos de controle da Figura 4.1. A região vermelha é compartilhada por todos os blocos de controle de todos os TL_i , e as regiões amarelas e verdes são replicadas para cada TL_i ($1 \leq i \leq N$).	67
4.3	Tempo Perdido (s) - Versão Sem Preempção - Cenário São Paulo - Sintético.	74
4.4	Tempo Perdido (s) - Versão Sem Preempção - Cenário Nova Iorque - Sintético.	75
4.5	Tempo Perdido sobre o Tempo Total de Viagem (%) - Versão Sem Preempção - Cenário São Paulo - Sintético.	76
4.6	Tempo Perdido sobre o Tempo Total de Viagem (%) - Versão Sem Preempção - Cenário Nova Iorque - Sintético.	76
4.7	Melhoria do Tempo Perdido (%) - Algoritmos - Cenário São Paulo - Sintético.	77
4.8	Melhoria do Tempo Perdido (%) - Algoritmos - Cenário Nova Iorque - Sintético.	78
5.1	Relação linear entre velocidade e densidade do tráfego (GREENSHIELDS et al., 1935)	83
5.2	Curva de concentração de veículos <i>vs</i> fluxo (adaptada) de Lighthill e Whitham (1955).	83

5.3	Tempo Perdido (s) - Versão Sem Preempção - Cenário Uma interseção.	87
5.4	Tempo Perdido sobre o Tempo Total de Viagem (%) - Versão Sem Preempção - Cenário Uma interseção	88
5.5	Melhoria do Tempo Perdido (vezes) - Cenário Uma Interseção .	88
5.6	Melhoria do Tempo Perdido (%) - Cenário Uma Interseção . . .	89
5.7	Tempo Perdido (s) - Algoritmos - Cenário Uma Interseção . . .	90
5.8	Tempo de Preempção Médio (s) - Cenário Uma Interseção . . .	90
5.9	Tempo Perdido (s) - Versão Sem Preempção - Cenário Turim .	92
5.10	Tempo Perdido sobre o Tempo Total de Viagem (%) - Versão Sem Preempção - Cenário Turim	93
5.11	Melhoria do Tempo Perdido (%) - Cenário Turim	94
5.12	Tempo de Preempção Médio (s) - Cenário Turim	94
5.13	Melhoria do Tempo Perdido (vezes) - Cenário Turim	95
5.14	Tempo Perdido (s) - Algoritmos - Cenário Turim	95
5.15	Tempo Perdido (s) - Versão Sem Preempção - Cenário Colônia	97
5.16	Tempo Perdido sobre o Tempo Total de Viagem (%) - Cenário Colônia	98
5.17	Melhoria do Tempo Perdido (%) - Cenário Colônia	98
5.18	Tempo de Preempção Médio (s) - Cenário Colônia	99
5.19	Melhoria do Tempo Perdido (vezes) - Cenário Colônia	100
5.20	Tempo Perdido (s) - Algoritmos - Cenário Colônia.	100
5.21	Interseção onde o veículo líder só pode prosseguir por um caminho. Se este não é o caminho desejado, o mesmo deve aguardar a fase correta do semáforo para que o seu caminho seja retomado.	102
5.22	Melhoria do Tempo Perdido (%) - Cenário Alto Tempo de Teletransporte	104
5.23	Tempo Perdido Médio (%) - Todos os Veículos - Cenário Alto Tempo de Teletransporte	105
5.24	Velocidade Média das Viagens (%) - Todos os Veículos - Cenário Alto Tempo de Teletransporte	106
5.25	Veículos Teletransportados (%) - Cenário Alto Tempo de Teletransporte	106
6.1	Exemplo de uma rota e o respectivo Bloco de Controle Global. O número de semáforos ao longo da rota do EV equivale ao número de Controles de interseção no Bloco de Controle Global.111	

- 6.2 O modelo *Timed Petri Net** (TPN*) proposto. A região vermelha (Bloco de Cancelamento) é compartilhada por todos os blocos de controle de todos os TL_i , o restante é replicado para cada TL_i ($1 \leq i \leq N$). 112
- 6.3 O disparo da transição t_0 remove o *token* da posição P_0 e o coloca na posição P_1 , iniciando a preempção para este semáforo. 113
- 6.4 O disparo da transição t_1 remove o *token* da posição P_1 e o coloca na posição P_2 . Neste ponto, para que o fluxo da TPN* continue, é necessário ou que o EV cruze o semáforo ou que a preempção seja cancelada. 114
- 6.5 Quando o EV cruza o semáforo, a transição t_3 é disparada, colocando tokens nas posições P_3 e P_7 . Esse evento indica que o semáforo será ressincronizado, mas isso acontecerá quando o *token* estiver na posição P_4 114
- 6.6 Com *tokens* nas posições P_2 e P_7 , a transição t_2 é habilitada. A transição t_2 é então disparada, o que remove os *tokens* das posições P_2 e P_7 e coloca um *token* na posição P_4 . Neste momento, o semáforo é ressincronizado. 115
- 6.7 O disparo da transição t_4 remove o *token* da posição P_4 e o coloca na posição P_5 , finalizando a preempção para este semáforo. 116
- 6.8 Quando a preempção é cancelada, a transição global t_{Cancel} é disparada, colocando um *token* nas posições P_{Cancel} e P_6 . Nesta Figura, há um *token* na posição P_0 , mas a preempção pode ser cancelada a qualquer momento, significando que o *token* pode estar nas posições P_0 , P_1 , P_2 , P_4 ou P_5 quando isso acontecer. Para o caso desta Figura, a posição P_{Cancel} inibe a transição t_0 , significando que a preempção não iniciará para este semáforo. 116
- 6.9 A preempção foi cancelada, então a transição global t_{Cancel} é disparada, colocando um *token* nas posições P_{Cancel} e P_6 . Contudo, a preempção já havia sido iniciada, então a transição t_0 já havia disparado, e o *token* passou pela posição P_1 e está na posição P_2 117
- 6.10 O *token* que estava na posição P_6 é colocado na posição P_7 pelo disparo da transição t_5 . O *token* nas posições P_7 e P_2 fazem disparar a transição t_2 , que os retira das respectivas posições e coloca o *token* na posição P_4 118

6.11	A preempção iniciou (disparo da transição t_0) e o EV cruzou o semáforo (disparo da transição t_3 , colocando <i>tokens</i> nas posições P_3 e P_7). Neste momento, a preempção é cancelada (disparo da transição t_{Cancel} , colocando <i>tokens</i> nas posições P_{Cancel} e P_6). Neste caso, a posição P_3 inibe a transição t_5 , repressando o <i>token</i> na posição P_6 . Contudo, já há um <i>token</i> na posição P_7 , que é usado para concluir a preempção.	118
6.12	A preempção foi finalizada com a informação de cruzamento do EV (disparo da transição t_3), mesmo que o cancelamento tenha ocorrido durante este processo (utilização dos <i>tokens</i> nas posições P_2 e P_7 para habilitar a transição t_2 e colocar o <i>token</i> na posição P_4).	119
6.13	Exemplo de uma Rede Petri e a respectiva matriz de incidência (MURATA, 1989). As linhas representam as posições e as colunas, as transições.	121
6.14	Limite de <i>tokens</i> nas posições na TPN* proposta. É importante notar que todas as posições são $[0, 1]$ -bounded.	123
6.15	Grafo de Alcançabilidade da <i>Timed Petri Net</i> (TPN) proposta na Figura 6.2.	124
6.16	Tempo Perdido (s) - Versão Sem Preempção - Cenário Turim .	131
6.17	Tempo Perdido sobre o Tempo Total de Viagem (%) - Versão Sem Preempção - Cenário Turim	131
6.18	Melhoria do Tempo Perdido (vezes) - Cenário Turim	132
6.19	interseção Complexa para o Cenário de Turim - Visão SUMO .	133
6.20	Interseção Complexa para o Cenário de Turim - Visões de <i>Layers</i> do Google	133
6.21	Melhoria do Tempo Perdido (%) - Algoritmos - Cenário Turim .	134
6.22	Tempo Perdido (s) - Algoritmos - Cenário Turim	135
6.23	Tempo de Preempção Médio (s) - Algoritmos - Cenário Turim .	135
6.24	Tempo Perdido (s) - Versão Sem Preempção - Cenário Colônia .	137
6.25	Tempo Perdido sobre o Tempo Total de Viagem (%) - Cenário Colônia	138
6.26	Tempo Perdido (s) - Algoritmos - Cenário Colônia	138
6.27	Melhoria do Tempo Perdido (%) - Algoritmos - Cenário Colônia .	139
6.28	Uma das interseções Complexas para o Cenário de Colônia - Visão SUMO	140
6.29	Uma das interseções Complexas para o Cenário Colônia - Visões de <i>Layers</i> do Google	140
6.30	Mais uma interseção complexa para o Cenário de Colônia - Visão SUMO	141

6.31	Mais uma interseção complexa para o Cenário Colônia - Visões de <i>Layers</i> do Google	141
6.32	Tempo Perdido (s) - Algoritmos - Cenário Colônia	142
6.33	Tempo de Preempção Médio (s) - Algoritmos - Cenário Colônia	143
6.34	Trecho da rota do EV1 no simulador SUMO para o Cenário do Centro Expandido de São Paulo. Cruzamento da Av. Pedroso de Moraes e da R. Ferreira de Araújo.	145
6.35	Trecho da rota do EV1 no Google Maps para o Cenário do Centro Expandido de São Paulo. Cruzamento da Av. Pedroso de Moraes e da R. Ferreira de Araújo.	146
6.36	Trecho da rota do EV1 no Google Earth para o Cenário do Centro Expandido de São Paulo. Cruzamento da Av. Pedroso de Moraes e da R. Ferreira de Araújo.	147
6.37	Tempo Perdido (s) - Versão Sem Preempção - Cenário Centro Expandido - Metrô OD SP 2017.	147
6.38	Tempo Perdido sobre o Tempo Total de Viagem (%) - Cenário Centro Expandido - Metrô OD SP 2017.	148
6.39	Melhoria de Tempo Perdido (vezes) - Algoritmos - Cenário Centro Expandido - Metrô OD SP 2017.	148
6.40	Tempo Perdido (%) - Algoritmos - Cenário Centro Expandido - Metrô OD SP 2017.	149
6.41	Tempo Perdido (s) - Algoritmos - Cenário Centro Expandido - Metrô OD SP 2017.	150
6.42	Tempo de Preempção Médio (s) - Algoritmos - Centro Expandido - Metrô OD SP 2017.	151
A.1	Área Metropolitana de São Paulo obtido diretamente do OpenStreetMap em < https://www.openstreetmap.org/relation/2661855#map=10/-23.5756/-46.1604 >	174
A.2	Representação da Amostra da Pesquisa OD Metrô 2017 (Companhia Metropolitana de São Paulo (METRÔ), 2017).	176
A.3	Centro Expandido de São Paulo - Área do Rodízio - Minianel Viário.	177

Lista de Tabelas

2.1	Comparação entre os sistemas de notificação	22
2.2	Análise de limitações nas soluções da Literatura	30
3.1	Número de veículos para os diferentes cenários.	45
3.2	Parâmetros das rotas dos EVs - cenários São Paulo e Nova Iorque - Sintético	46
3.3	Parâmetros das rotas dos EVs - Cenário Turim	49
3.4	Parâmetros das rotas dos EVs - Cenário Colônia	49
3.5	Parâmetros das rotas dos EVs - cenário Centro Expandido de São Paulo	51
3.6	Métodos da classe base <i>PreemptionStrategy</i>	55
3.7	Recursos computacionais usados na execução dos experimentos deste trabalho	61
6.1	Linhas de código da implementação dos Algoritmos	129
A.1	Tempo Perdido (s) - Primeira Solução - Cenário São Paulo (Sintético) - Sem Preempção	178
A.2	Tempo Perdido (s) - Primeira Solução - Cenário Nova Iorque (Sintético) - Sem Preempção	178
A.3	Tempo Perdido sobre o Tempo Total de Viagem (%) - Primeira Solução - Cenário São Paulo (Sintético) - Sem Preempção	179
A.4	Tempo Perdido sobre o Tempo Total de Viagem (%) - Primeira Solução - Cenário Nova Iorque (Sintético) - Sem Preempção . . .	179
A.5	Melhoria do Tempo Perdido (%) - Primeira Solução - Cenário São Paulo (Sintético) - <i>Radio-Frequency Identification</i> (RFID) . .	179
A.6	Melhoria do Tempo Perdido (%) - Primeira Solução - Cenário São Paulo (Sintético) - Fuzzy	179
A.7	Melhoria do Tempo Perdido (%) - Primeira Solução - Cenário São Paulo (Sintético) - Filas	179
A.8	Melhoria do Tempo Perdido (%) - Primeira Solução - Cenário São Paulo (Sintético) - TPN'	180
A.9	Melhoria do Tempo Perdido (%) - Primeira Solução - Cenário São Paulo (Sintético) - Onda Verde	180

A.10	Melhoria do Tempo Perdido (%) - Primeira Solução - Cenário Nova Iorque (Sintético) - RFID	180
A.11	Melhoria do Tempo Perdido (%) - Primeira Solução - Cenário Nova Iorque (Sintético) - Fuzzy	180
A.12	Melhoria do Tempo Perdido (%) - Primeira Solução - Cenário Nova Iorque (Sintético) - Filas	180
A.13	Melhoria do Tempo Perdido (%) - Primeira Solução - Cenário Nova Iorque (Sintético) - TPN'	181
A.14	Melhoria do Tempo Perdido (%) - Primeira Solução - Cenário Nova Iorque (Sintético) - Onda Verde	181
A.15	Tempo Perdido (s) - Segunda Solução - Cenário Uma Interseção - Sem Preempção	181
A.16	Tempo Perdido sobre o Tempo Total de Viagem (%) - Segunda Solução - Cenário Uma Interseção - Sem Preempção	181
A.17	Melhoria do Tempo Perdido (vezes) - Segunda Solução - Cenário Uma Interseção	181
A.18	Melhoria do Tempo Perdido (%) - Segunda Solução - Cenário Uma Interseção	182
A.19	Tempo de Preempção Médio (s) - Segunda Solução - Cenário Uma Interseção	182
A.20	Tempo Perdido (s) - Segunda Solução - Cenário Turim - Sem Preempção	182
A.21	Tempo Perdido sobre o Tempo Total de Viagem (%) - Segunda Solução - Cenário Turim - Sem Preempção	182
A.22	Melhoria do Tempo Perdido (%) - Segunda Solução - Cenário Turim - RFID	182
A.23	Melhoria do Tempo Perdido (%) - Segunda Solução - Cenário Turim - Fuzzy	183
A.24	Melhoria do Tempo Perdido (%) - Segunda Solução - Cenário Turim - Filas	183
A.25	Melhoria do Tempo Perdido (%) - Segunda Solução - Cenário Turim - Onda de Choque	183
A.26	Melhoria do Tempo Perdido (%) - Segunda Solução - Cenário Turim - Onda Verde	183
A.27	Melhoria do Tempo Perdido (vezes) - Segunda Solução - Cenário Turim - RFID	183
A.28	Melhoria do Tempo Perdido (vezes) - Segunda Solução - Cenário Turim - Fuzzy	184
A.29	Melhoria do Tempo Perdido (vezes) - Segunda Solução - Cenário Turim - Filas	184

A.30	Melhoria do Tempo Perdido (vezes) - Segunda Solução - Cenário Turim - Onda de Choque	184
A.31	Melhoria do Tempo Perdido (vezes) - Segunda Solução - Cenário Turim - Onda Verde	184
A.32	Tempo Perdido (s) - Segunda Solução - Cenário Uma Interseção	184
A.33	Tempo de Preempção Médio (s) - Segunda Solução - Cenário Turim - RFID	185
A.34	Tempo de Preempção Médio (s) - Segunda Solução - Cenário Turim - Fuzzy	185
A.35	Tempo de Preempção Médio (s) - Segunda Solução - Cenário Turim - Filas	185
A.36	Tempo de Preempção Médio (s) - Segunda Solução - Cenário Turim - Onda de Choque	185
A.37	Tempo de Preempção Médio (s) - Segunda Solução - Cenário Turim - Onda Verde	185
A.38	Tempo Perdido (s) - Segunda Solução - Cenário Turim - RFID .	186
A.39	Tempo Perdido (s) - Segunda Solução - Cenário Turim - Fuzzy .	186
A.40	Tempo Perdido (s) - Segunda Solução - Cenário Turim - Filas . .	186
A.41	Tempo Perdido (s) - Segunda Solução - Cenário Turim - Onda de Choque	186
A.42	Tempo Perdido (s) - Segunda Solução - Cenário Turim - Onda Verde	186
A.43	Tempo Perdido (s) - Segunda Solução - Cenário Colônia - Sem Preempção	187
A.44	Tempo Perdido sobre o Tempo Total de Viagem (%) - Segunda Solução - Cenário Colônia - Sem Preempção	187
A.45	Melhoria do Tempo Perdido (%) - Segunda Solução - Cenário Colônia - RFID	187
A.46	Melhoria do Tempo Perdido (%) - Segunda Solução - Cenário Colônia - Fuzzy	187
A.47	Melhoria do Tempo Perdido (%) - Segunda Solução - Cenário Colônia - Filas	187
A.48	Melhoria do Tempo Perdido (%) - Segunda Solução - Cenário Colônia - Onda de Choque	188
A.49	Melhoria do Tempo Perdido (%) - Segunda Solução - Cenário Colônia - Onda Verde	188
A.50	Tempo de Preempção Médio (s) - Segunda Solução - Cenário Colônia - RFID	188
A.51	Tempo de Preempção Médio (s) - Segunda Solução - Cenário Colônia - Fuzzy	188

A.52	Tempo de Preempção Médio (s) - Segunda Solução - Cenário Colônia - Filas	188
A.53	Tempo de Preempção Médio (s) - Segunda Solução - Cenário Colônia - Onda de Choque	189
A.54	Tempo de Preempção Médio (s) - Segunda Solução - Cenário Colônia - Onda Verde	189
A.55	Melhoria do Tempo Perdido (vezes) - Segunda Solução - Cenário Colônia - RFID	189
A.56	Melhoria do Tempo Perdido (vezes) - Segunda Solução - Cenário Colônia - Fuzzy	189
A.57	Melhoria do Tempo Perdido (vezes) - Segunda Solução - Cenário Colônia - Filas	189
A.58	Melhoria do Tempo Perdido (vezes) - Segunda Solução - Cenário Colônia - Onda de Choque	190
A.59	Melhoria do Tempo Perdido (vezes) - Segunda Solução - Cenário Colônia - Onda Verde	190
A.60	Tempo Perdido (s) - Segunda Solução - Cenário Colônia - RFID .	190
A.61	Tempo Perdido (s) - Segunda Solução - Cenário Colônia - Fuzzy	190
A.62	Tempo Perdido (s) - Segunda Solução - Cenário Colônia - Filas .	190
A.63	Tempo Perdido (s) - Segunda Solução - Cenário Colônia - Onda de Choque	191
A.64	Tempo Perdido (s) - Segunda Solução - Cenário Colônia - Onda Verde	191
A.65	Tempo Perdido sobre o Tempo Total de Viagem (%) - Cenário Alto Tempo de Teletransporte - Onda de Choque	191
A.66	Tempo Perdido Médio (%) - Todos os Veículos - Cenário Alto Tempo de Teletransporte - Onda de Choque	191
A.67	Velocidade Média das Viagens (%) - Todos os Veículos - Cenário Alto Tempo de Teletransporte - Onda de Choque	191
A.68	Veículos Teletransportados (%) - Cenário Alto Tempo de Teletransporte - Onda de Choque	191
A.69	Tempo Perdido (s) - Terceira Solução - Cenário Turim - Sem Preempção	192
A.70	Tempo Perdido sobre o Tempo Total de Viagem (%) - Terceira Solução - Cenário Turim - Sem Preempção	192
A.71	Melhoria do Tempo Perdido (vezes) - Terceira Solução - Cenário Turim - RFID	192
A.72	Melhoria do Tempo Perdido (vezes) - Terceira Solução - Cenário Turim - Fuzzy	192

A.73	Melhoria do Tempo Perdido (vezes) - Terceira Solução - Cenário Turim - Filas	192
A.74	Melhoria do Tempo Perdido (vezes) - Terceira Solução - Cenário Turim - Onda de Choque	193
A.75	Melhoria do Tempo Perdido (vezes) - Terceira Solução - Cenário Turim - Onda de Choque + TPN*	193
A.76	Melhoria do Tempo Perdido (vezes) - Terceira Solução - Cenário Turim - Onda Verde	193
A.77	Melhoria do Tempo Perdido (%) - Terceira Solução - Cenário Turim - RFID	193
A.78	Melhoria do Tempo Perdido (%) - Terceira Solução - Cenário Turim - Fuzzy	193
A.79	Melhoria do Tempo Perdido (%) - Terceira Solução - Cenário Turim - Filas	194
A.80	Melhoria do Tempo Perdido (%) - Terceira Solução - Cenário Turim - Onda de Choque	194
A.81	Melhoria do Tempo Perdido (%) - Terceira Solução - Cenário Turim - Onda de Choque + TPN*	194
A.82	Melhoria do Tempo Perdido (%) - Terceira Solução - Cenário Turim - Onda Verde	194
A.83	Tempo Perdido (s) - Terceira Solução - Cenário Turim - RFID . .	194
A.84	Tempo Perdido (s) - Terceira Solução - Cenário Turim - Fuzzy .	195
A.85	Tempo Perdido (s) - Terceira Solução - Cenário Turim - Filas . .	195
A.86	Tempo Perdido (s) - Terceira Solução - Cenário Turim - Onda de Choque	195
A.87	Tempo Perdido (s) - Terceira Solução - Cenário Turim - Onda de Choque + TPN*	195
A.88	Tempo Perdido (s) - Terceira Solução - Cenário Turim - Onda Verde	195
A.89	Tempo de Preempção Médio (s) - Terceira Solução - Cenário Turim - RFID	196
A.90	Tempo de Preempção Médio (s) - Terceira Solução - Cenário Turim - Fuzzy	196
A.91	Tempo de Preempção Médio (s) - Terceira Solução - Cenário Turim - Filas	196
A.92	Tempo de Preempção Médio (s) - Terceira Solução - Cenário Turim - Onda de Choque	196
A.93	Tempo de Preempção Médio (s) - Terceira Solução - Cenário Turim - Onda de Choque + TPN*	196

A.94	Tempo de Preempção Médio (s) - Terceira Solução - Cenário Turim - Onda Verde	197
A.95	Tempo Perdido (s) - Terceira Solução - Cenário Colônia - Sem Preempção	197
A.96	Tempo Perdido sobre o Tempo Total de Viagem (%) - Terceira Solução - Cenário Colônia - Sem Preempção	197
A.97	Melhoria do Tempo Perdido (vezes) - Terceira Solução - Cenário Colônia - RFID	197
A.98	Melhoria do Tempo Perdido (vezes) - Terceira Solução - Cenário Colônia - Fuzzy	197
A.99	Melhoria do Tempo Perdido (vezes) - Terceira Solução - Cenário Colônia - Filas	198
A.100	Melhoria do Tempo Perdido (vezes) - Terceira Solução - Cenário Colônia - Onda de Choque	198
A.101	Melhoria do Tempo Perdido (vezes) - Terceira Solução - Cenário Colônia - Onda de Choque + TPN*	198
A.102	Melhoria do Tempo Perdido (vezes) - Terceira Solução - Cenário Colônia - Onda Verde	198
A.103	Melhoria do Tempo Perdido (%) - Terceira Solução - Cenário Colônia - RFID	198
A.104	Melhoria do Tempo Perdido (%) - Terceira Solução - Cenário Colônia - Fuzzy	199
A.105	Melhoria do Tempo Perdido (%) - Terceira Solução - Cenário Colônia - Filas	199
A.106	Melhoria do Tempo Perdido (%) - Terceira Solução - Cenário Colônia - Onda de Choque	199
A.107	Melhoria do Tempo Perdido (%) - Terceira Solução - Cenário Colônia - Onda de Choque + TPN*	199
A.108	Melhoria do Tempo Perdido (%) - Terceira Solução - Cenário Colônia - Onda Verde	199
A.109	Tempo Perdido (s) - Terceira Solução - Cenário Colônia - RFID .	200
A.110	Tempo Perdido (s) - Terceira Solução - Cenário Colônia - Fuzzy .	200
A.111	Tempo Perdido (s) - Terceira Solução - Cenário Colônia - Filas .	200
A.112	Tempo Perdido (s) - Terceira Solução - Cenário Colônia - Onda de Choque	200
A.113	Tempo Perdido (s) - Terceira Solução - Cenário Colônia - Onda de Choque + TPN*	200
A.114	Tempo Perdido (s) - Terceira Solução - Cenário Colônia - Onda Verde	201

A.115 Tempo de Preempção Médio (s) - Terceira Solução - Cenário Colônia - RFID	201
A.116 Tempo de Preempção Médio (s) - Terceira Solução - Cenário Colônia - Fuzzy	201
A.117 Tempo de Preempção Médio (s) - Terceira Solução - Cenário Colônia - Filas	201
A.118 Tempo de Preempção Médio (s) - Terceira Solução - Cenário Colônia - Onda de Choque	201
A.119 Tempo de Preempção Médio (s) - Terceira Solução - Cenário Colônia - Onda de Choque + TPN*	202
A.120 Tempo de Preempção Médio (s) - Terceira Solução - Cenário Colônia - Onda Verde	202
A.121 Tempo Perdido (s) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Sem Preempção	202
A.122 Tempo Perdido sobre o Tempo Total de Viagem (%) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017)	202
A.123 Melhoria de Tempo Perdido (vezes) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - RFID	202
A.124 Melhoria de Tempo Perdido (vezes) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Fuzzy	203
A.125 Melhoria de Tempo Perdido (vezes) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Filas	203
A.126 Melhoria de Tempo Perdido (vezes) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Onda de Choque	203
A.127 Melhoria de Tempo Perdido (vezes) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Onda de Choque + TPN*203	
A.128 Melhoria de Tempo Perdido (vezes) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Onda Verde	203
A.129 Melhoria de Tempo Perdido (%) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - RFID	203
A.130 Melhoria de Tempo Perdido (%) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Fuzzy	204
A.131 Melhoria de Tempo Perdido (%) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Filas	204
A.132 Melhoria de Tempo Perdido (%) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Onda de Choque	204
A.133 Melhoria de Tempo Perdido (%) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Onda de Choque + TPN*204	
A.134 Melhoria de Tempo Perdido (%) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Onda Verde	204

A.135 Tempo Perdido (s) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - RFID	204
A.136 Tempo Perdido (s) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Fuzzy	205
A.137 Tempo Perdido (s) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Filas	205
A.138 Tempo Perdido (s) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Onda de Choque	205
A.139 Tempo Perdido (s) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Onda de Choque + TPN*	205
A.140 Tempo Perdido (s) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Onda Verde	205
A.141 Tempo de Preempção Médio (s) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - RFID	205
A.142 Tempo de Preempção Médio (s) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Fuzzy	206
A.143 Tempo de Preempção Médio (s) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Filas	206
A.144 Tempo de Preempção Médio (s) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Onda de Choque	206
A.145 Tempo de Preempção Médio (s) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Onda de Choque + TPN*206	206
A.146 Tempo de Preempção Médio (s) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Onda Verde	206

Glossário

API *Application Programming Interface.* 24, 53, 54

AVS *Average Vehicle Speed.* 39

CAM *Cooperative Awareness Messages.* 27

CET *Companhia de Engenharia de Tráfego.* xvii, 2

CL *Congestion Level.* 39

COG *Center of Gravity.* 37

CONTRAN *Conselho Nacional de Trânsito.* 2

CSV *Comma Separated Value.* 175, 176

DBF *dBase Database File.* 176

DSRC *Dedicated Short-Range Communication.* 22, 27

ERP *Emergency Response Plan.* 39

EV *Veículo de Emergência.* xvii-xxiii, 2-11, 13-27, 34-37, 39-61, 63-70, 72-75, 78, 81, 82, 84-87, 89-93, 95-97, 99, 101, 104, 105, 110, 111, 113-115, 117-119, 124-128, 130, 132-137, 139-147, 149, 153, 154, 156, 157, 181-206

GPS *Global Positioning System.* 4, 21, 25, 27

GPX *GPS eXchange Format.* 177

GSM *Global System for Mobile Communications.* 27

IoT *Internet das Coisas.* 25

IoV *Internet of Vehicles.* 22

IQR Intervalo Interquartil. 75, 104, 105, 142

IR *Infrared*. 18, 25, 27

LOC *Linhas de Código*. 129

MIRT *Mobile Infrared Transmitter*. 19

NHTSA *National Highway Traffic Safety Administration*. 3

OL *Occupancy Level*. 39

OSM *OpenStreetMap*. 175

P2P *Peer-to-Peer*. 19

PDO *Property Damage Only* - Apenas Danos à Propriedade. 4

PN *Petri Net*. 63

RFID *Radio-Frequency Identification*. xxiii–xxx, 4, 19, 20, 22, 25, 27, 35–37, 55, 99, 129, 179, 180, 183, 186, 189, 190, 192–194, 196–198, 200–205

RSU *Road-Side Unit*. 4, 23, 27, 35

TPN *Timed Petri Net*. xxi, 10, 61, 64–66, 110, 120, 123–125, 134, 145, 149, 150

TPN' *Timed Petri Net'*. xviii, xxiii, xxiv, xxxiii, 10, 65–68, 70, 71, 73, 77–79, 81, 110, 128, 153, 154, 180, 181

TPN* *Timed Petri Net**. xx, xxi, xxvii–xxx, 10, 109–112, 114, 115, 117, 119, 120, 122, 123, 125, 128, 129, 132–134, 137, 139, 141, 142, 149, 150, 153–155, 172, 193–196, 198–200, 202–206

TraCI *Traffic Control Interface*. 53

V2I *Vehicle-to-Infrastructure Communication*. 11, 14, 19, 22–25, 31, 37, 157

V2V *Vehicle-to-Vehicle Communication*. 11, 14, 19, 22–25, 27, 31, 157

VANET *Vehicular Ad Hoc Network*. 11, 19, 22

VTL *Semáforo Virtual*. xvii, 24, 27, 157

WAVE *Wireless Access in Vehicular Environments*. 19, 22, 27

WSN *Rede de Sensores sem Fio*. 27

XML *eXtensible Markup Language*. 175

Lista de Algoritmos

3.1	Preempção com RFID	37
3.2	Soluções baseadas em Lógica Fuzzy	38
3.3	Soluções baseadas em filas	41
3.4	Onda Verde	41
4.1	Preempção com <i>Timed Petri Net'</i> (TPN')	71
4.2	Cancelamento Global	72
5.1	Onda de Choque	85
6.1	Onda de Choque + TPN*	126

Introdução

A mobilidade urbana tornou-se um grande problema nas grandes áreas metropolitanas. Em cidades como Londres, Chicago, Paris e Bogotá, congestionamentos são frequentes, e os motoristas gastam diversas horas ao dia dirigindo para se deslocarem de casa para o trabalho (PISHUE, 2023). Essa situação é sensivelmente agravada em grandes cidades de países em desenvolvimento, entre eles o Brasil, principalmente as grandes capitais como São Paulo, Rio de Janeiro, Belo Horizonte, entre outras. A Figura 1.1 apresenta o retrato dessa situação na cidade de São Paulo.

O planejamento da infraestrutura em tais cidades, por si só, não necessariamente resulta em uma solução para o problema do trânsito a longo prazo. Brasília, por exemplo, foi planejada para ter 500 mil habitantes até os anos 2000. Contudo, esse valor foi atingido ainda nos anos 60 (BURSZTYN; ARAÚJO, 1997) e já contava com mais de 2,8 milhões de habitantes em 2022 e com uma estimativa de quase 3 milhões de habitantes em 2024 (Instituto Brasileiro de Geografia e Estatística, 2025). Neste cenário, a situação é agravada pois, devido à sua característica, a população se locomove utilizando veículos próprios ou transporte público, enfrentando invariavelmente o trânsito, e nos horários de grande volume de tráfego (direção do trânsito sentido cidades satélites → Plano Piloto no começo da manhã e Plano Piloto → cidades satélites no fim da tarde), as vias ficam completamente congestionadas (G1 DF, 2016). Observamos ainda que os investimentos em infraestrutura também possuem um efeito limitado na melhora do fluxo de veículos automotores, e que dificilmente eliminarão por completo os congestionamentos existentes nas grandes cidades.

Além de causar atrasos e poluição, os grandes congestionamentos dificultam



Figura 1.1: No dia 23 de maio de 2014, o congestionamento apurado pela Companhia de Engenharia de Tráfego (CET) na cidade de São Paulo foi de 344 km (G1 São Paulo, 2014).

tam a chegada de veículos de emergência (EV) ao local de uma ocorrência no tempo necessário.

O Código de Trânsito Brasileiro define, no inciso VII do Art. 29, que um Veículo de Emergência (EV) é um “veículo destinado a socorro de incêndio e salvamento, de polícia, de fiscalização e operação de trânsito e a ambulância” (BRASIL, 1997). Além disso, o Conselho Nacional de Trânsito (CONTRAN) normatiza, no § 4º do Art. 5º da Resolução nº 970/22, que “prestação de serviço de urgência [são] os deslocamentos realizados pelos veículos de emergência, em circunstâncias que necessitem de brevidade para o atendimento, sem a qual haverá risco concreto à vida de terceiros ou grande prejuízo à incolumidade pública” (Conselho Nacional do Trânsito (CONTRAN), 2022). Por este motivo, espera-se que o tráfego de um EV seja prejudicado por congestionamentos, como mostra a Figura 1.2.

Para chegar no menor tempo possível no local desejado, em geral os EVs usam dois artifícios para contornar os obstáculos apresentados pelo trânsito, simultaneamente ou não: avisos sonoros e ultrapassagem de sinais vermelhos. O primeiro artifício é utilizar um sinal sonoro (normalmente utilizando uma sirene) para solicitar que a sua passagem seja concedida. Ressalta-se que, em tal situação, o motorista incumbido de ceder a passagem deve atentar para a direção e o sentido do EV (ele pode estar em outra via, sendo ouvido por todos na redondeza), verificando se é possível ceder espaço ou permitir



Figura 1.2: Tráfego do EV dificultado pelo congestionamento (Portal Unibus, 2013).

a ultrapassagem, pois a via pode estar completamente congestionada, impedindo o deslocamento físico dos veículos entre as faixas.

Ainda assim, resta o problema das interseções. As interseções podem ou não possuir um semáforo, mas na prática o mesmo procedimento é aplicado. Ao detectar um sinal sonoro de um EV, o motorista primeiro aplica a verificação descrita anteriormente, cedendo a passagem se necessário. Caso contrário, a probabilidade é que o EV precise de prioridade na interseção, e os veículos de vias distintas devem interromper o trânsito para que isso aconteça (inclusive os veículos cuja via esteja liberada com o sinal verde). Esse procedimento é empírico e manual, e os motoristas, tanto do EV quanto dos outros veículos nas vias, não se sentem seguros o suficiente sobre o que realmente deve ser feito, e essa situação pode resultar em acidentes graves (G1 Presidente Prudente, 2017) e (G1 Amazonas, 2017).

Nos Estados Unidos, acidentes envolvendo EVs resultaram em 198 mortes em 2021 (National Safety Council, 2023). O risco de acidente aumenta de 4,6 para 5,5 por 100 mil atendimentos quando o EV está atendendo a uma emergência utilizando sirenes e *giroflex*. Esse risco é ainda maior quando o EV está transportando uma vítima, saltando de 7,0 para 16,5 por 100 mil atendimentos (WATANABE et al., 2019). O *National Highway Traffic Safety Administration* (NHTSA) coletou dados referentes a acidentes de ambulâncias

nos Estados Unidos entre 2012 e 2018 (GRAHAM et al., 2023). Nesse período, ocorreram 33.811 acidentes sem vítimas (classificados como *Property Damage Only* - Apenas Danos à Propriedade (PDO)). Já para acidentes com pessoas machucadas, 9.922 foram computados. O total de pessoas machucadas por estes acidentes foi estimado em 17.530. Neste período, ocorreram 173 acidentes fatais, e entre esses, 45,7% ocorreram durante o atendimento a uma emergência, e 28% ocorreram enquanto o EV utilizava sirenes e *giroflex*.

Em função disso, obter uma solução algorítmica segura para controlar a temporização dos semáforos existentes na rota de um veículo de emergência pode diminuir o tempo perdido por um EV nos congestionamentos, além de evitar acidentes.

A maneira típica de se reduzir os efeitos do congestionamento no trajeto de um EV é utilizar um mecanismo de preempção para alterar a temporização dos semáforos ao longo de sua rota, para garantir que estes estejam com as luzes verdes para a via onde o EV está trafegando e impondo luzes vermelhas para as outras vias da interseção (PANIATI; AMONI, 2006).

Para que o mecanismo de preempção funcione, os semáforos devem detectar a aproximação do EV. Há várias maneiras de se detectar a aproximação do EV (por exemplo, utilizando sirenes, *Radio-Frequency Identification* (RFID) ou *Global Positioning System* (GPS), que serão detalhadas no Capítulo 2) e de se construir algoritmos que decidam o que deve ser feito uma vez que o EV tenha sido detectado. Contudo, as propostas da literatura que analisamos – apresentadas no Capítulo 2 – não apresentam melhorias significativas e consistentes no tempo perdido pelo EV no tráfego. Ademais, alguns desses trabalhos não podem ser facilmente generalizados para grupos grandes de interseções, e nestes casos um único semáforo pode não acarretar nenhum benefício no tempo perdido pelo EV, enquanto outros exigem infraestrutura cuja implantação é difícil e custosa (por exemplo, *loops* indutivos ou *Road-Side Unit* (RSU)). Um outro ponto que também deve ser considerado é o impacto que a preempção dos semáforos na rota do EV terá na circulação dos demais veículos, bem como a garantia do funcionamento correto do algoritmo de preempção.

Entre os trabalhos apresentados na literatura, não encontramos propostas que cobrissem a rota completa do EV. Também não encontramos nenhuma solução com garantias formais do funcionamento dos algoritmos. Neste trabalho, apresentamos soluções competitivas para diminuir o tempo perdido por EVs em grandes cidades, onde a ocorrência de congestionamentos é frequente. As soluções consideram a rota completa do EV além de analisar as propriedades formais dos algoritmos apresentados.

A próxima Seção apresenta a motivação e os objetivos deste trabalho.

1.1 Motivação e Objetivos

Como observamos anteriormente, o atraso de veículos de emergência pode resultar em perdas irreparáveis de bens materiais e até mesmo de vidas humanas. Por exemplo, alguns estudos sugerem que a taxa de sobrevivência de pacientes com ataques cardíacos cai entre 7 e 10% a cada minuto sem o atendimento apropriado (PANIATI; AMONI, 2006). Em 2018, o número anual de mortes decorrentes de acidentes de trânsito alcançou a marca de 1,35 milhão, liderando o ranking de mortes em pessoas entre 5 e 29 anos (Organização Mundial da Saúde (OMS), 2018), sendo que as interseções apresentam um risco elevado de acidentes (CHOI, 2010), e um mau planejamento nas características das vias e temporização dos semáforos pode levar ao aumento de acidentes (AGBELIE; ROSHANDEH, 2015). É imperativo que boas soluções controlem os semáforos nas interseções e possam reduzir o tempo perdido pelo EV durante um atendimento de emergência.

Considerando que o EV deve perder o menor tempo possível durante o atendimento de uma ocorrência, os veículos à sua frente devem liberar o caminho, mas semáforos fechados atrapalham esse procedimento. Para se fazer isso, deve-se realizar a preempção dos semáforos na rota do EV, isto é, fornecer luz verde para que os veículos à frente e o próprio EV possam trafegar livremente. Informalmente, pode-se pensar em uma “onda verde”, na qual os semáforos são liberados um a um de forma sincronizada e ordenada. Contudo, isso nem sempre é possível, caso os semáforos mais à frente possuam filas veiculares mais longas de forma que pode ser necessário liberá-los antes dos semáforos mais próximos do EV. Mais do que isso, é importante que a solução seja “satisfatória”, ou seja, os veículos das redondezas sofram o menor impacto possível para que o EV tenha o menor atraso possível. Portanto, temos a Pergunta de Pesquisa: **como podemos reduzir o atraso do EV no trânsito perturbando o mínimo possível seu entorno?**

Como resposta a esta questão, formulamos a seguinte **Hipótese**: Uma estratégia de preempção semaforica baseada em coordenação global, combinando modelos físicos de tráfego e modelagem formal de concorrência, reduz de forma mais eficiente o tempo perdido de veículos de emergência, quando comparada a abordagens locais ou não coordenadas.

Neste contexto, é fundamental qualificar os critérios de êxito propostos. Por *Segurança*, entende-se uma abordagem dual: refere-se tanto à segurança viária (mitigação de riscos de acidentes nas interseções causados por interferências abruptas) quanto à segurança computacional (garantia formal, via Redes de Petri, de que o sistema de controle não atingirá estados inválidos ou de travamento/*deadlock*).

Já a *Eficiência* não se resume apenas ao desempenho computacional do algoritmo, mas define-se pela efetividade operacional: a capacidade de maximizar a redução do tempo de viagem do Veículo de Emergência (EV) enquanto reduz o impacto negativo (como filas e atrasos) no fluxo dos demais veículos da rede.

O objetivo principal deste trabalho é, portanto, apresentar soluções seguras, isto é, que não permitam estados inválidos no sistema, para diminuir o tempo perdido pelo EV no trânsito. Para isso, propomos a utilização de preempção dos semáforos de forma a liberar antecipadamente o trajeto do EV. As soluções aqui propostas discutem políticas de preempção que diminuem o atraso do EV quando ele se aproxima de uma interseção semaforizada. Em uma situação ideal, o tempo perdido pelo EV no trânsito é igual a zero, significando que o tempo total de viagem equivale ao tempo de deslocamento do veículo trafegando à máxima velocidade.

A eficiência e segurança de uma solução de preempção de semáforos dependem da antecedência em que o EV é detectado (tipo de sensor usado), dos dados de telemetria, tais como posição, velocidade e aceleração do EV, bem como da distância entre o EV e o semáforo, e de como tais dados são utilizados para que a luz verde do semáforo seja fornecida em um momento adequado, além da correção do algoritmo utilizado. Contudo, do momento em que tais dados são coletados até o momento futuro em que a preempção ocorra, é possível que todos os valores utilizados no algoritmo de preempção devam ser recalculados. Isso porque, pela natureza complexa do sistema, modelagens matemáticas de predição apresentam comportamento não contínuo e não linear, devido ao seu comportamento caótico intrínseco (DISBRO; FRAME, 1989). Esse comportamento caótico deve estar materializado nas soluções.

Além disso, a solução proposta deve ser “satisfatória”: ela não deve iniciar a preempção muito antecipadamente, o que acarretaria uma parada dos veículos nas vias concorrentes quando estes poderiam estar trafegando, nem deve iniciar tardiamente, e neste caso o EV estaria parado aguardando a troca das luzes do semáforo.

Nas soluções de preempção, que alterem o funcionamento regular de dispositivos como semáforos, a segurança no trânsito deve ser sempre levada em consideração, de modo que os elementos envolvidos na dinâmica do trânsito (tais como veículos, pedestres, ciclistas) tenham tempo de reagir a tais alterações de forma segura. Mais do que isso, uma solução deste tipo deve ser robusta o suficiente para garantir que comandos inválidos ou erros em leitura dos sensores não impactem de forma negativa todo o sistema.

Neste trabalho discutimos o problema do tempo perdido por um EV e propomos soluções que levam em conta os pontos anteriormente citados. Os

resultados obtidos, tanto com dados sintéticos quanto com dados reais, são competitivos e podem ser avaliados para serem incorporados em *frameworks* de cidades inteligentes. A próxima seção descreve as atividades que foram desenvolvidas neste trabalho.

1.2 Visão Geral da Solução

Nosso objeto de estudo é a diminuição do tempo perdido por EVs num dado trajeto em grandes metrópoles. O problema estudado considera uma rota fixa que será percorrida pelo EV. A rota possui um conjunto de semáforos que, juntamente com os demais semáforos da cidade, são controlados por uma central de controle. Os semáforos conseguem detectar a aproximação de um EV e informar a central de controle. Além disso, a central de controle tem como avaliar o fluxo de veículos na cidade e na via em que o EV está trafegando. De posse dessas informações, a central de controle define a política de preempção dos semáforos na rota do EV. O cenário descrito é compatível com os *frameworks* utilizados no estudo de cidades inteligentes, e uma visão simplificada é apresentada na Figura 1.3.

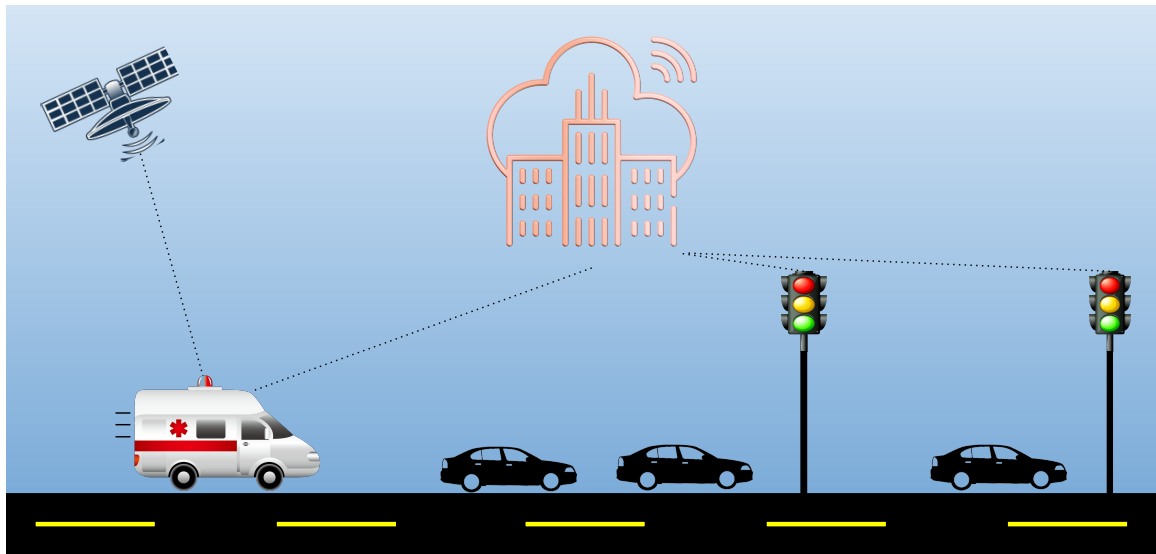


Figura 1.3: Visão simplificada do problema, com informações de posicionamento do EV e comunicação entre os semáforos e a central de controle.

Na busca por soluções para diminuir o tempo perdido por um EV usando esse cenário, vários desafios surgem. O primeiro deles é o de como garantir que a política de preempção proposta crie uma “janela” para o EV e não bloqueie desnecessariamente as interseções com semáforos. A política de preempção também deve ser capaz de se autoajustar em função do congestionamento na rota do EV, além de ser correta e segura. Um outro desafio é o de como tratar os diferentes tipos de interseções existentes nas grandes cidades,

onde a interseção envolve mais de duas vias e números diferentes de faixas de rolamento. Outro desafio que surgiu durante o trabalho foi o da avaliação dos algoritmos, principalmente quando utilizamos dados reais de circulação de veículos nas cidades estudadas.

Podemos destacar que o elemento-chave para a análise e o desempenho dos algoritmos é a avaliação do tempo necessário para que as filas de veículos que estão paradas no semáforo, em virtude de um eventual sinal vermelho, sejam dissipadas. Contudo, para semáforos distantes do EV, pode ser que a fila de veículos já esteja grande a ponto de atrapalhar preempções em andamento. Na revisão bibliográfica efetuada, não foi encontrada nenhuma outra solução que utilize essa estratégia. A preempção antecipada dos semáforos ao longo do trajeto facilita o tráfego do EV e é uma das contribuições deste trabalho.

Dos trabalhos avaliados na literatura, também não constam soluções que abordem de uma forma dinâmica o ingresso de veículos na rede. Rede, nesse contexto, é uma representação da interligação das vias rodoviárias e interseções de uma área geográfica de um bairro, cidade ou outra região e pode ser diretamente representada por um grafo. As soluções propostas neste trabalho consideram que uma parcela de veículos pode ingressar nestas redes, entre o tempo em que o cálculo da preempção foi efetuado e o momento em que a preempção foi efetivamente iniciada.

De acordo com os pontos levantados acima, temos que o problema de diminuir o tempo perdido EV num dado trajeto é bastante complexo. A indisponibilidade dos dados dos outros veículos, bem como a natureza não determinística, dificulta a modelagem e a solução do problema. A segurança do tráfego exige ainda que as soluções sejam estáveis e reproduzíveis.

Para analisar os algoritmos propostos para a solução desse problema, necessitamos de um modelo formal. Para isso, utilizamos um modelo de Rede Petri Temporizada para representar o sistema estudado. Com esse modelo, podemos demonstrar que a ordem dos eventos ocorre de forma controlada, o que possibilita modelar e analisar a rota completa do EV, sendo esse um dos principais diferenciais do trabalho, uma vez que os trabalhos descritos na literatura tratam apenas de uma ou poucas interseções com semáforos percorridas pelo EV.

Os algoritmos propostos também foram avaliados e comparados com os existentes na nossa revisão bibliográfica. No modelo proposto neste trabalho, o comportamento e a interação de cada um dos elementos (veículos, vias etc.) são analisados individualmente.

Pela característica da interação dos elementos dos cenários, um simulador microscópico foi utilizado para fazer as avaliações. Os resultados obtidos foram comparados com as soluções da literatura que puderam ser implemen-

tadas. Em nosso trabalho, utilizamos o simulador de tráfego microscópico *Simulation of Urban Mobility* (SUMO). As simulações foram efetuadas com dados sintéticos e reais. Dependendo do volume de dados, o tempo da simulação é muito grande. A simulação da Região Metropolitana da Cidade de São Paulo, para uma janela de 24 horas, por exemplo, executou por aproximadamente 112 dias numa máquina com 32 processadores e consumiu 83 GB dos 396 GB de memória RAM disponível, sem finalizar a execução. O experimento precisou ser encerrado para atualizações de segurança do equipamento pelos administradores do ambiente. Portanto, alguns ajustes nas janelas de tempo da simulação foram efetuados. Os resultados obtidos mostram que os algoritmos propostos neste trabalho são competitivos.

A próxima Seção apresenta algumas das contribuições fornecidas por este trabalho.

1.3 Principais Contribuições

As contribuições desta tese foram desenvolvidas e avaliadas sob a ótica de três pilares fundamentais para o controle semafórico em sistemas críticos: Segurança, Eficiência e Robustez. No contexto deste trabalho, define-se Segurança não apenas como a prevenção de acidentes viários, mas também sob a perspectiva computacional, garantindo formalmente — via Redes de Petri — a ausência de *deadlocks* e estados inconsistentes nos controladores. A Eficiência refere-se à capacidade operacional de reduzir o tempo de viagem dos EVs através da dissipação antecipada de filas (*queue clearance*). Por fim, a Robustez é entendida como a resiliência do sistema para lidar com eventos estocásticos, falhas de comunicação e o cancelamento súbito de preempção sem colapsar a malha viária.

Dessa forma, as principais contribuições estão organizadas cronologicamente conforme o desenvolvimento das soluções propostas e a estruturação do texto:

- **Framework de Avaliação e Cenários Reais (Capítulo 3):** Desenvolvimento de um *framework* de simulação integrado ao simulador SUMO e compatível com um *middlewares* de cidade inteligente real (BATISTA et al., 2016).
- **Construção de Cenários Reais (Capítulo 3):** Construção de dois cenários com demandas reais utilizando os dados da Pesquisa Origem e Destino do Metrô de 2017, que abrange a região metropolitana da Cidade de São Paulo. O primeiro contempla toda a região metropolitana e o tráfego dentro de uma janela de 24 horas, e o segundo abrange apenas o

Centro Expandido da cidade de São Paulo. Dos trabalhos analisados, estes são os maiores cenários disponíveis para utilização em experimentos deste tipo, tanto em tamanho da rede quanto em volume de veículos.

- **1ª Solução - Foco na Segurança (Capítulo 4):** Proposta do modelo baseado em Redes de Petri Temporizadas estendidas (TPN') para o controle de rota completa. Esta abordagem prioriza a verificação formal do sistema, assegurando que a priorização do EV ocorra sem violações de segurança nos estados semafóricos.
- **2ª Solução - Foco na Eficiência (Capítulo 5):** Implementação de um algoritmo baseado no princípio físico de Onda de Choque (*Shockwave*). Esta solução foca na fluidez, calculando o tempo exato para a dissipação física das filas à frente do EV, permitindo o cruzamento de interseções com redução significativa de desacelerações.
- **3ª Solução - Abordagem Híbrida e Robusta (Capítulo 6):** Desenvolvimento da solução unificada *Timed Petri Net** (TPN*). Esta contribuição combina a eficiência de limpeza de filas da Onda de Choque com as garantias de segurança da *Timed Petri Net* (TPN), introduzindo mecanismos de cancelamento global e recuperação de falhas, oferecendo a solução mais resiliente entre as avaliadas. Uma das três soluções reduz o tempo perdido pelo EV em até 98,96%, quando comparado à versão sem preempção.
- **Conclusão (Capítulo 7):** Apresenta a síntese dos resultados obtidos, limitações do estudo e direcionamentos para trabalhos futuros.

A próxima Seção apresenta a delimitação do escopo deste trabalho, bem como as premissas básicas usadas durante o seu desenvolvimento.

1.4 Delimitação de Escopo e Premissas Básicas

Em nosso escopo, consideramos que o EV já está usando a rota mais curta e rápida quando inicia o atendimento da emergência. Portanto, o roteamento do EV não está sendo considerado. Na nossa proposta, a solução deve reduzir o tempo perdido pelo EV independentemente da escolha de rota.

Consideramos o fator de estabilidade e reprodutibilidade da solução como um aspecto importante, inclusive no fator de segurança. Algoritmos que utilizam técnicas de Inteligência Artificial e suas variantes como Algoritmos Genéticos, Aprendizado de Máquina, entre outros, não necessariamente atendem a esses requisitos. Por exemplo, Li et al. (2020) e Xiao et al. (2022), que não são específicos para redução de tempo perdido de EVs, tentam reduzir o tempo de

espera médio de forma geral, sem garantias, apesar de os autores afirmarem que suas soluções são confiáveis.

O tamanho dos cenários varia bastante. Para os cenários sintéticos, por exemplo, o cenário de São Paulo possui uma rede com 33.367 arestas e 25.882 veículos para uma janela de 1 hora na sua maior instância, enquanto o cenário de Nova Iorque possui uma rede com 23.854 arestas e 16.454 veículos para a mesma situação. Já para o cenário de São Paulo criado neste trabalho, utilizando dados reais, a rede do Centro Expandido possui 126.295 arestas com 345.926 veículos em uma janela de meia hora, enquanto o cenário completo possui 3.052.896 arestas e 13.215.558 veículos para uma janela de 24 horas.

Apenas um EV de cada vez é considerado neste trabalho, apesar do arcabouço da solução permitir mais de um. Contudo, se for este o caso, havendo conflito de rota, a última ordem é a acatada. Por este motivo, os resultados são apresentados em termos de cada um dos EVs individualmente.

Ainda dentro das premissas consideradas, consideramos que o sistema se comunica com os semáforos, sensores e veículos de emergência, mas não com os outros veículos presentes na rede. Isso porque, embora tecnologias como *Vehicle-to-Vehicle Communication* (V2V), *Vehicle-to-Infrastructure Communication* (V2I) e *Vehicular Ad Hoc Network* (VANET) possam resolver essa comunicação intraveicular e situações de veículos autônomos, tal infraestrutura não está predominantemente disponível. No Brasil, por exemplo, a implantação da Rede 5G, que pode prover internet e comunicação rápida para as rodovias federais, está sendo viabilizada no momento da escrita deste documento (BRASIL, 2021). Há ainda o agravante da idade média da frota dos veículos brasileiros girar em torno de 10,2 anos, a maior nos últimos 25 anos (EXAME, 2021). Portanto, esperar a conectividade desses veículos não é factível.

Fundamentação Teórica e Trabalhos Relacionados

Este capítulo estabelece as bases para a solução proposta, partindo das abordagens clássicas de detecção passiva — onde a infraestrutura reage à presença física do veículo — para os sistemas modernos de comunicação cooperativa. Analisamos como cada tecnologia tenta mitigar o tempo perdido e identificamos, através de uma revisão crítica da literatura recente, por que as abordagens atuais ainda falham em garantir segurança e eficiência em rotas completas dentro de malhas urbanas congestionadas.

Um veículo perde tempo no trânsito quando este, por algum motivo, precisa reduzir sua velocidade durante seu trajeto. Ao final do percurso, é possível calcular quanto tempo o veículo perdeu em virtude dessas desacelerações, e este é denominado o tempo perdido pelo veículo no trânsito. Para um EV, esse tempo perdido pode ocasionar problemas consideráveis e, portanto, deve ser evitado. De forma geral, um veículo precisa desacelerar quando o veículo à sua frente o está impedindo de prosseguir, ou em função de outros tipos de bloqueio, por exemplo, pelo sinal vermelho do semáforo. Quando um EV se utiliza de sirenes (emitindo avisos sonoros e luminosos na proximidade), por exemplo, ele pretende alertar os veículos à sua frente para darem passagem ao mesmo quando possível; contudo, isso não resolve o bloqueio do trânsito devido à existência de semáforos fechados.

Dada a contextualização e os objetivos definidos no capítulo anterior, torna-se fundamental compreender as variáveis que influenciam esse atraso e examinar as tecnologias existentes para mitigá-lo. Neste capítulo, estabelecemos a fundamentação teórica e o estado da arte que alicerçam esta pesquisa. Inici-

almente, apresentamos a descrição formal do problema (Seção 2.1) e definimos os critérios de avaliação e requisitos desejáveis para soluções de preempção (Seção 2.2). Em seguida, detalhamos os mecanismos de detecção de EVs (Seção 2.2.2) e discutimos as tecnologias de comunicação V2V/V2I (Seção 2.4) e semáforos virtuais (Seção 2.5). Posteriormente, situamos a discussão no contexto de Cidades Inteligentes (Seção 2.6). Por fim, realizamos uma análise crítica dos trabalhos relacionados (Seção 2.7), destacando as lacunas na literatura que este trabalho busca preencher.

2.1 Descrição do Problema

Considere um instante de tempo $t(s)$, onde um EV ingressa em uma rede. Como dito anteriormente, uma rede é uma representação da interligação de vias rodoviárias em uma determinada área geográfica pré-determinada, podendo representar uma área de um bairro, de uma cidade e assim por diante. A interligação de tais vias, em geral, pode ser representada por um grafo, onde as vias são as arestas e as interseções são os nós. Contudo, essa não é a única representação possível. Podemos usar as vias como nós da rede, e as arestas podem ser representadas como as trocas de tráfego entre as vias. Há situações em que é primordial que se conheça o modelo da rede, como, por exemplo, quando é necessário interagir com um simulador de trânsito. Para a descrição do problema, contudo, o modelo da rede é irrelevante.

O EV mencionado possui uma direção $\rightarrow$, velocidade $v(m/s)$, aceleração em metros por segundo ao quadrado $a(m/s^2)$, comprimento $l(m)$ e posição em latitude e longitude (lat, lon) , bem como uma rota pré-definida. Este veículo deve trafegar por um conjunto de vias, passando por interseções (que podem ou não ser semaforizadas). À sua frente, o EV pode encontrar veículos, cada um possuindo também sua direção, velocidade, aceleração, comprimento, posição e rota, parâmetros que não são previamente conhecidos, e todos os veículos devem manter um espaço mínimo um do outro, denominado $gap(m)$. Periodicamente, há um incremento no tempo t em $\Delta t(s)$, ($\Delta t > 0$), onde os parâmetros dos veículos podem sofrer alterações na velocidade em $\Delta v(m/s)$ e aceleração em $\Delta a(m/s^2)$, bem como a variação do seu deslocamento em $\Delta d(m)$. Ainda, novos veículos podem ingressar em qualquer lugar da rede ou podem sair (no caso de fim de rota, por exemplo).

Um veículo desacelera quando o trânsito à frente está lento ou quando encontra um bloqueio, como um sinal vermelho. Há outras situações que impedem o avanço de um veículo, por exemplo, quando este está tentando cruzar uma via preferencial. Em todos esses casos, o conjunto de veículos parados nesta situação forma uma fila $Q_i(m)$, que é dissipada quando o ca-

minho estiver novamente livre. A Figura 2.1 apresenta uma representação do problema.

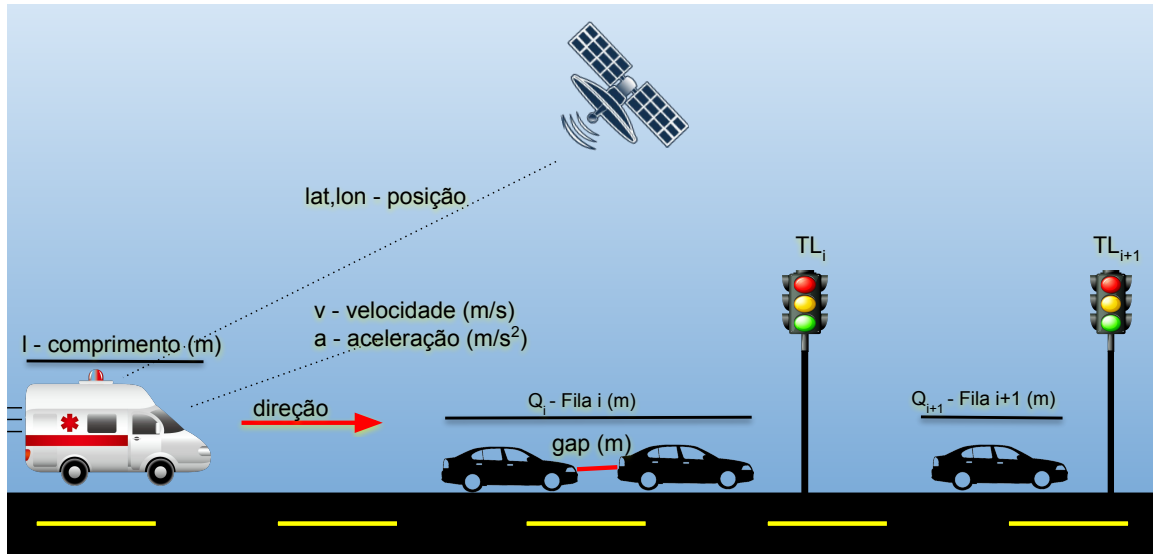


Figura 2.1: Descrição do problema. TL_i representa o i -ésimo semáforo da rota do EV.

O objetivo do problema é, portanto, evitar que o EV precise reduzir sua velocidade mantendo a distância de segurança (gap) dos veículos à sua frente. Se as rotas de todos os veículos da rede fossem conhecidas e nenhum veículo ingressasse na rede, seria possível prever com razoável exatidão o estado da rede após n passos ($t + n\Delta t$). Contudo, nenhuma dessas duas premissas é verdadeira, pois o trânsito é um sistema caótico (DISBRO; FRAME, 1989). Há ainda um complicador relevante: a rota de alguns veículos líderes (veículos que estão mais à frente da fila de carros) pode estar bloqueada por algum motivo (por exemplo, na situação em que o próximo segmento de via para onde o líder deveria ir estiver congestionado). Neste caso, mesmo que o caminho esteja "liberado" (por exemplo, o sinal verde está ativo para esta via), o veículo líder irá bloquear o restante da fila, já que ele não encontrou o espaço mínimo $l + gap$ para se posicionar. A Figura 2.2 mostra essa situação. Por conseguinte, há a possibilidade de o EV não conseguir prosseguir em seu caminho.

2.2 Critérios de Avaliação e Requisitos Desejáveis

Para que uma solução de priorização de veículos de emergência seja considerada eficaz e viável em cenários urbanos reais, ela não deve apenas focar na redução do tempo de viagem do EV, mas também garantir a segurança e a fluidez do tráfego circundante. Com base na complexidade do problema apresentada, estabelecemos um conjunto de critérios fundamentais que serão utilizados neste trabalho para analisar as propostas da literatura e validar as soluções desenvolvidas.

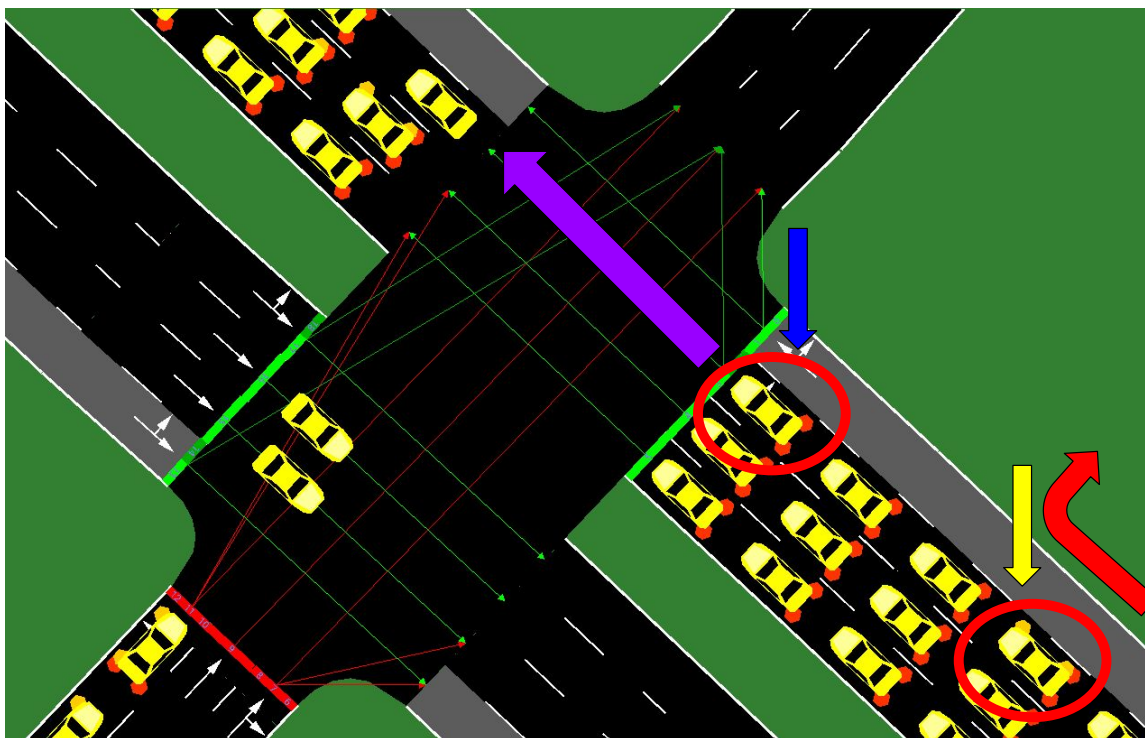


Figura 2.2: Ilustração do bloqueio de uma interseção por congestionamento a frente. Mesmo com o sinal verde favorável (seta roxa), o veículo líder (círculo superior) não avança por falta de espaço no trecho seguinte, obstruindo toda a fila. Consequentemente, veículos com rotas alternativas livres (círculo inferior, com intenção de virar à direita) também são impedidos de prosseguir.

Estes critérios foram divididos em Requisitos Funcionais, que dizem respeito ao comportamento do algoritmo de controle, e Requisitos Metodológicos, que avaliam a qualidade e a reprodutibilidade da pesquisa.

2.2.1 Requisitos Funcionais e de Segurança

Uma solução robusta de preempção semafórica deve atender aos seguintes aspectos operacionais:

1. **Redução do Tempo Perdido:** O critério primário é a capacidade efetiva de diminuir o atraso do EV em comparação a um cenário sem preempção.
2. **Atuação em Rota Completa:** A solução deve ser capaz de gerenciar todo o trajeto do EV, e não apenas atuar isoladamente em uma ou poucas interseções, garantindo uma “onda verde” contínua.
3. **Segurança e Conciliação de Conflitos:** O sistema deve ser capaz de conciliar ordens conflitantes (ex: múltiplos EVs) e evitar impasses (*deadlocks*) que travem o trânsito.
4. **Transição Segura de Fases:** A alteração dos tempos semafóricos deve respeitar os tempos de segurança (amarelo/vermelho total) para evitar

acidentes, jamais forçando uma mudança abrupta de verde para vermelho.

5. **Eficiência no Encerramento e Cancelamento:** A preempção deve ser encerrada imediatamente após a passagem do EV para minimizar o impacto no trânsito. Além disso, o sistema deve permitir o cancelamento da prioridade caso a emergência seja abortada ou a rota alterada.
6. **Mitigação de Impactos Colaterais:** É fundamental evitar a inanição (*starvation*) dos demais veículos, garantindo que o tráfego transversal não fique bloqueado indefinidamente.
7. **Robustez e Segurança da Informação:** A solução deve lidar adequadamente com falhas de sensores/hardware e garantir a autenticidade das comunicações para evitar acionamentos maliciosos ou não autorizados.

2.2.2 *Requisitos Metodológicos e de Validação*

Além dos aspectos técnicos, a análise do estado da arte considerará a qualidade da validação proposta pelos autores, buscando evitar lacunas comuns na literatura. Serão valorizados os seguintes pontos:

- **Cenários Realistas:** Utilização de topologias de redes reais e complexas, em oposição a grades sintéticas simplificadas (*grids*) ou cenários com poucas interseções.
- **Avaliação de Desempenho e Métricas Adequadas:** Apresentação de resultados quantitativos claros, utilizando métricas que reflitam o ganho real (como tempo perdido ou tempo de viagem) em vez de métricas indiretas (como número de mensagens trocadas).
- **Reprodutibilidade e Comparabilidade:** Disponibilidade de códigos ou detalhamento suficiente que permita a reprodução dos experimentos e a comparação direta com outras soluções.
- **Escalabilidade e Generalização:** Capacidade do modelo de ser aplicado a diferentes configurações urbanas sem exigir infraestrutura proprietária de difícil implantação ou custos proibitivos.

2.3 *Sistemas de Detecção Baseados em Infraestrutura (Sem Comunicação Direta)*

A eficácia de qualquer sistema de priorização semaforica depende primariamente da sua capacidade de detectar a aproximação do EV com antecedência

e precisão suficientes. Na literatura e na prática comercial, diversas tecnologias têm sido empregadas para esse fim, cada uma com implicações distintas em termos de latência, alcance, custo de infraestrutura e suscetibilidade a interferências. Nesta seção, analisamos criticamente os principais mecanismos de detecção, classificando-os pela natureza física do sinal utilizado.

2.3.1 *Sistemas sonoros*

Este é talvez um dos mais conhecidos métodos de detecção de EV no trânsito. Em sua versão mais básica, uma sirene alerta os veículos da redondeza de que um EV está se aproximando. Contudo, tais sistemas, desde que adaptados para tal, também podem ser usados para preempção de semáforos de forma automatizada. Tais sistemas podem utilizar sons de sirenes e/ou geradores de ultrassom. Em seu modo puro, apenas um alto-falante (sirene) e um microfone no semáforo seriam suficientes para o processamento do pedido de preempção. Neste caso, um emissor dedicado não é necessário, porém, interferências de sinais analógicos e ruídos alheios podem inviabilizar o sistema. Quando se utiliza uma unidade emissora de ultrassom dedicada, tais interferências podem ser amenizadas, mas não eliminadas. Condições climáticas que não ofereçam risco às frequências sonoras empregadas não são um problema nesta abordagem. Porém, a característica não direcional de tal comunicação não impede a preempção de outros semáforos e, consequentemente, não impede o controle de dispositivos não autorizados caso nenhum controle adicional de segurança seja utilizado. A característica sonora implica sua alta latência e perturbação ambiental, mesmo usando mecanismos de ultrassom, pois alguns animais são sensíveis a tais estímulos. Um exemplo de um sistema comercial que utiliza tal método é o *SONEM 2000* (RGH Pacific Inc., 2025).

Esse mecanismo sofre da limitação da velocidade do som, que trafega a $\approx 340m/s$. Filas de congestionamentos de 1 km implicam em 3 segundos apenas para que o sensor próximo ao semáforo detecte o EV, e ao se considerar várias interseções, tais soluções podem acumular bastante tempo perdido no final da emergência.

2.3.2 *Sistemas ópticos*

Assim como seu método sonoro análogo, luzes de alerta em um giroflex alertam os veículos da proximidade da aproximação do EV, de modo que estes veículos liberem a passagem. Com sensores específicos, é possível automatizar a preempção de semáforos usando sistemas ópticos. Neste caso, tais sistemas podem utilizar comunicação infravermelha (*Infrared* (IR)) ou trans-

missões visadas do EV para o semáforo. Tal sistema prevê a instalação de um dispositivo dedicado para transmissão e recepção dos dados, tanto no EV quanto em cada semáforo. Interferências nos sinais analógicos não atrapalham sua utilização, tampouco ruídos eletrônicos. Porém, essa solução exige uma linha de visão entre o transmissor e o receptor. Mais ainda, condições climáticas e obstáculos podem dificultar a utilização do sistema. Como a comunicação é *Peer-to-Peer* (P2P), a preempção a ser executada pelo semáforo só ocorrerá a um veículo por vez. O disparo ilegal da preempção por dispositivos não autorizados pode ser um problema nesta abordagem. Excluindo o *overhead* de processamento, tal abordagem possui uma baixa latência, sendo que sua comunicação ocorre na velocidade da luz na atmosfera. Este sistema não oferece perturbação ambiental. Exemplos de alguns sistemas comerciais conhecidos deste tipo são *OPTICOM* (Miovision Technologies Incorporated, 2025) e *Mobile Infrared Transmitter* (MIRT).

Esses sistemas possuem a limitação de distância entre o transmissor e o receptor. Dos sensores comerciais existentes, a distância máxima observada foi de 150 m, o que é um valor muito baixo para situações de alto congestionamento. Neste caso, a preempção não é iniciada a tempo.

2.3.3 *Sistemas baseados em Rádio*

Nestes sistemas, transmissores e receptores utilizam tecnologias de radio-frequência para efetuar a comunicação entre o EV e o semáforo. É importante notar que, na classificação aqui proposta, a tecnologia RFID, apesar de utilizar rádio-frequência, tem por finalidade oferecer a função de sensor e, portanto, não é considerada nesta seção.

Os EVs em geral não possuem uma unidade de rádio disponível para notificação do EV e, portanto, uma unidade dedicada deve ser instalada. Este sistema não sofre interferências de sinais analógicos, mas sim de ruídos eletrônicos. Linha de visão e problemas de clima não são uma preocupação, mas transmissões simultâneas podem causar problemas, caso nenhum protocolo de acesso ao meio seja usado. Como consequência, o disparo ilegal de preempção também pode ser um problema caso mecanismos adicionais de segurança não sejam adicionados às soluções. A latência nesses casos é tão baixa quanto a desejada, e tal sistema não causa perturbação ambiental. Há uma grande quantidade de soluções comerciais que utilizam esse método, entre elas o *XBee* (Digi, 2025), *ZigBee* (Connectivity Standards Alliance (CSA), 2025), ou mesmo protocolos tradicionais como o Wi-Fi, englobando o padrão IEEE 802.11 e suas variações, como o IEEE 802.11p *Wireless Access in Vehicular Environments* (WAVE).

Redes que utilizam V2V, V2I e VANETs exigem ainda que a infraestrutura

das vias ou da frota utilize tais tecnologias, premissa não atendida em muitas cidades.

2.3.4 Sistemas baseados em Sensores

Tais sistemas utilizam sensores, isto é, *transponders* de baixa frequência instalados nos veículos, e estes são detectados por receptores instalados em pontos de interesse específicos, como interseções com semáforos. Tal proposta exige um transmissor e receptor aptos para este fim tanto no emissor quanto no receptor. As características são semelhantes aos sistemas que utilizam rádio, considerando as seguintes exceções: 1) os sensores forçam a análise do dispositivo receptor, portanto, ordens simultâneas são escalonadas; e 2) a possibilidade de disparo ilegal de preempção é reduzida devido ao critério de identificação do sensor, apesar de não ser nula, exigindo assim que mecanismos adicionais de segurança sejam considerados.

Entre os sensores utilizados, atualmente o mais comum é o RFID (Identificação por Radiofrequência). Contudo, de acordo com as versões disponíveis no mercado (IMPINJ, 2025), a distância de detecção é um fator limitador importante deste método. Uma vez que um sensor é detectado, não é possível verificar a variação da distância percorrida pelo EV apenas com um par transmissor/receptor, a menos que vários receptores sejam instalados em sequência.

Um outro tipo de sensor comumente encontrado são os *loops* indutivos, que detectam a passagem de carros conforme apresentado na Figura 2.3. Contudo, tais sensores são fixos e exigem alterações estruturais nas vias, o que dificulta a implantação.



Figura 2.3: Exemplo de sistema de detecção de *loop* indutivo (marcas no asfalto) (METROCOUNT, 2020).

2.3.5 *Sistemas baseados em GPS*

Nesta modalidade, o diferencial é a utilização do sistema GPS, onde as coordenadas do EV e/ou do semáforo são utilizadas pelos algoritmos para determinação da preempção. Tais sistemas exigem um módulo GPS, e qualquer *smartphone* pode ser usado para esse fim. Este método não sofre interferências de sinais analógicos e ruídos eletrônicos, e tampouco exige uma linha de visão ou é afetado pelo clima. O sistema por si só não garante a possibilidade de disparo ilegal de preempção, já que é relativamente fácil *hackear* as coordenadas nativas do módulo GPS, forçando um comportamento anormal do sistema.

2.3.6 *Comparação dos sistemas de Notificação*

Apresentamos uma adaptação e atualização da Tabela de comparação apresentada em Collura et al. (2001), acrescida da coluna de notificação por sensores da linha Ciência dos veículos, considerações e atualizações (Tabela 2.1). Não há pretensões de apresentar as condições finais de propostas utilizando tais sistemas de notificação, uma vez que os sistemas podem ser combinados e mecanismos adicionais para superação das limitações podem ser utilizados, mas visa clarificar a utilização dos sistemas e suas limitações físicas, conceituais e estruturais.

2.3.7 *Conclusão*

Embora os mecanismos de detecção baseados em sensores e avisos sonoros/ópticos sejam amplamente utilizados e de baixo custo inicial, eles sofrem de limitações físicas inerentes, como alcance reduzido e falhas de detecção em condições climáticas adversas ou oclusão visual. Mais criticamente, eles operam de forma reativa e local, incapazes de negociar rotas ou otimizar o fluxo de tráfego antes que o veículo de emergência atinja o congestionamento crítico. Até o momento, as soluções apresentadas levam em consideração apenas as informações enviadas pelo EV ou informações das proximidades dos semáforos e interseções obtidas de maneiras indiretas, como, por exemplo, por meio de câmeras de monitoramento. Nestes casos, não há comunicação direta entre veículos comuns (quaisquer veículos que não sejam o EV e que não sejam monitorados pela solução). Veículos e infraestrutura de cidades mais modernas podem disponibilizar novas formas de comunicação e protocolos, habilitando, assim, uma variedade de novas soluções. As próximas seções apresentam o que pode ser feito considerando que existem meios de comunicar-se com os veículos comuns.

Tabela 2.1: Comparação entre os sistemas de notificação

Consideração	Sistema				
	Acústico	Óptico	GPS	Rádio	Sensores ^a
Emissor dedicado	Não ^b	Sim	Não ^c	Sim	Sim
Interferência de sinais analógicos	Sim ^d	Não	Não	Não	Não
Interferência de ruídos eletrônicos	Não	Não	Não	Sim	Sim
Linha de visão requerida	Não	Sim	Não	Não	Não
Afetado pelo tempo/clima	Não	Sim	Não	Não	Não
Preempção por aproximação de outros veículos	Sim	Não	Não	Sim ^e	Não ^f
Disparo ilegal de preempção	Alta	Alta	Média ^g	Alto	Baixa ^h
Latência	Alta	Baixa	Média	Baixa	Baixa
Perturbação ambiental	Sim	Não	Não	Não	Não
Ciência dos veículos vizinhos	Sim	Sim ⁱ	Não	Não ^j	Não

^aConsiderando sistemas utilizando sensores RFID

^bCaso em que apenas a sirene do veículo é usada. Pode-se usar um emissor dedicado de ultrassom para este fim.

^cClassificação diferente da proposta em Sahana et al. (2016)

^dCaso em que apenas a sirene do veículo é usada. Pode-se usar um emissor ultrassom para diferenciação, mas sem garantias de sua efetividade

^eConsiderando transmissão e recepção pura dos sinais

^fOs sinais são processados um por vez

^gConsiderando possibilidades de *hacking* e simulação de coordenadas nos dispositivos

^hA falsificação da identidade dos sensores é possível, porém improvável, e mecanismos extras de segurança podem ser adicionados

ⁱCaso o objetivo seja notificar os vizinhos utilizando avisos luminosos

^jCom exceção da comunicação explícita usando V2V (Comunicação entre veículos) e/ou V2I (Comunicação entre veículos e a Infraestrutura)

2.4 Sistemas Cooperativos com Comunicação Direta (V2X)

Caso a frota de veículos possua equipamentos capazes de trocar informações entre si (V2V - Comunicação entre veículos) ou com a infraestrutura de trânsito (V2I - Comunicação entre veículos e a Infraestrutura) de forma bidirecional, é possível solicitar que tais veículos troquem de faixa, deem passagem ou até mesmo mudem de rota, para assim permitir a passagem do EV. Os termos dos protocolos e tecnologias envolvidas em comunicação veicular podem variar dependendo do contexto, tecnologia ou país, podendo haver interseções entre si, sendo possível citar *Dedicated Short-Range Communication* (DSRC) (Society of Automotive Engineers, 2009), WAVE (COHEN, 2006), VANET (TOH, 2001), *Internet of Vehicles* (IoV) (GERLA et al., 2014), 802.11p (IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks-Specific Requirements, 2020) e ITS-G5 (ETSI, 2017). Assim, estas tecnologias permitem comunicações, tais quais apresentadas na Figura 2.4.

Dentre os trabalhos que utilizam este tipo de abordagem, é possível citar Unibaso et al. (2010), Mohamed et al. (2015). Já em Djahel et al. (2015), o

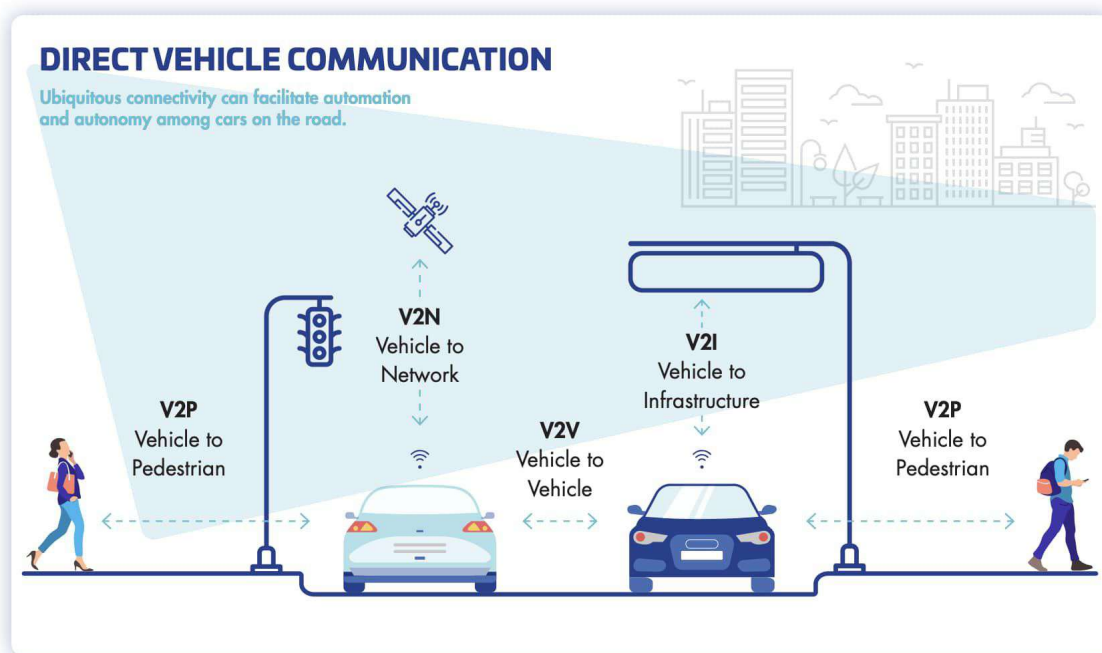


Figura 2.4: Tipos de comunicação entre veículos, infraestrutura e pessoas.
Fonte da imagem: <<https://www.thalesgroup.com/en/markets/digital-identity-and-security/iot/industries/automotive/use-cases/v2x>>.

trabalho não deixa explícita a utilização de comunicação V2V/V2I. Mas como os autores afirmam que a solução solicita aos veículos da redondeza que librem a faixa e até mesmo que estes sejam reroteados durante o atendimento da emergência, é seguro assumir que este trabalho também utiliza algum tipo de comunicação V2V/V2I.

Portanto, a introdução da comunicação direta do tipo V2I envolve a necessidade de instalar hardware de comunicação dedicado (RSUs) em cada interseção semaforizada, o que eleva drasticamente o custo de implantação (CAPEX). Em cidades de médio/grande porte, cobrir 100% das interseções torna-se financeiramente inviável, criando “zonas de sombra” onde o benefício da preempção é perdido. Para as soluções que usam V2V, é necessário que uma porcentagem crítica da frota circulante possua a tecnologia instalada. Em cenários de baixa penetração (fase de transição tecnológica), a “corrente” de comunicação é quebrada, isolando o EV.

2.5 Aplicações Avançadas: Semáforos Virtuais e Gestão Distribuída

Proposto em Ferreira et al. (2010), este método utiliza comunicação V2V para, em teoria, substituir a infraestrutura semaforica atual. Por meio da comunicação entre seus pares, os veículos obtêm os parâmetros telemétricos

e chegam a um consenso sobre qual via deve ter prioridade de fluxo. Assim, todas as interseções podem ser gerenciadas. Há propostas que apresentam a representação do *Semáforo Virtual* (VTL) no para-brisa do veículo, como o apresentado na Figura 2.5.



Figura 2.5: Representação do VTL no para-brisa do veículo apresentado em Olaverri-Monreal et al. (2012).

Há vários trabalhos que utilizam ou estudam VTL, geralmente visando reduzir o congestionamento de forma global. Dentre os trabalhos analisados, é possível citar Chou, Tseng e Yang (2013), Tonguz (2011), Younes e Boukerche (2017), Humagain e Sinha (2020), Sharma et al. (2013), Ferreira (2016).

Apesar da variedade de possibilidades que tecnologias como V2V/V2I e VTL proporcionam, isto não é uma realidade próxima para muitos lugares, devido à falta dessa infraestrutura básica necessária, em especial na questão da frota e da idade dos veículos. A próxima seção apresenta uma solução intermediária, que não exige todos os recursos tecnológicos exigidos por V2V/V2I e VTL.

2.6 Cidades Inteligentes

Com o objetivo de não ter que se preocupar com os detalhes de comunicação e obtenção dos dados dos componentes da preempção (por exemplo, do EV e do semáforo), podemos lançar mão de um elemento que lida com essa questão de uma forma conveniente. Conforme exposto em Borja e Gama (2014), há iniciativas para prover plataformas que fornecem dados de serviços de cidades oriundos das mais diversas fontes, em formato aberto, entregues e expostos em *Application Programming Interface* (API)s por meio de um *Middleware*, com o intuito de centralizar o controle e monitoramento, promover a

gestão eficiente e o apoio à tomada de decisão.

Entre elas, Batista et al. (2016) oferecem uma oportunidade única de abordar problemas urbanos complexos, tais como a preempção de semáforos para veículos de emergência, entre outros. Tais componentes simplificam a coleta de dados de sensores e a comunicação com atuadores, centralizam o controle de acesso para os dispositivos e implementam políticas de segurança definidas pelo operador da rede, reduzindo, portanto, os custos de implantação e gerenciamento. Neste caso, os sistemas que utilizam tal serviço focam na solução algorítmica, enquanto a comunicação de baixo nível é de responsabilidade do *Middleware*.

Especialmente quando não se possui conhecimento de todos os parâmetros da rede a cada instante, afinal, as rotas dos outros veículos são desconhecidas e eles podem ingressar e deixar a rede a todo instante, não é possível estabelecer o estado do sistema nos próximos passos apenas com os dados conhecidos, tornando-o um problema não determinístico. Isso reforça a necessidade de soluções estáveis e reproduzíveis, e a utilização de uma Rede Petri, como a utilizada neste trabalho, auxilia na garantia de certas propriedades de segurança no tráfego. A utilização de simulações extensivas é essencial nesse caso. Contudo, o nível de granularidade da simulação altera a precisão e o tempo de execução de forma significativa, e isso deve ser considerado na hora da preparação dos experimentos.

Enquanto é possível encontrar diversas propostas de *Smart Cities* na literatura, destacamos os trabalhos de (NOGRADY, 2017), (Digital Dubai, 2025), (Escola São Paulo de Ciência Avançada, 2017) e (BUENO, 2021). A arquitetura de *Middleware* projetada para soluções de preempção de semáforos assemelha-se àquelas apresentadas na Figura 2.6a, juntamente a Figura 2.6b apresenta uma implementação chamada InterSCity para tal arquitetura.

Há uma variedade de trabalhos focados em soluções baseadas em Cidades Inteligentes, desde propostas de Simuladores até trabalhos de gerenciamento do trânsito de forma geral. Entre estes, pode haver interseções entre os métodos de detecção previamente apresentados (como RFID, GPS, IR, V2V/V2I, entre outros), evidenciando que são tópicos não disjuntos, além do número de oportunidades de abordagem para o problema do tempo perdido pelo EV no trânsito.

A quantidade e disponibilidade de dados relacionados à infraestrutura básica de comunicação V2V ou V2I, optamos por utilizar o método de detecção GPS, devido à importância da obtenção de informações relativas à posição, velocidade e aceleração do EV a qualquer instante. Além disso, o *middleware* para Cidades Inteligentes trata o EV e os semáforos como dispositivos de *Internet das Coisas* (IoT), enviando e recebendo os dados e enviando os comandos

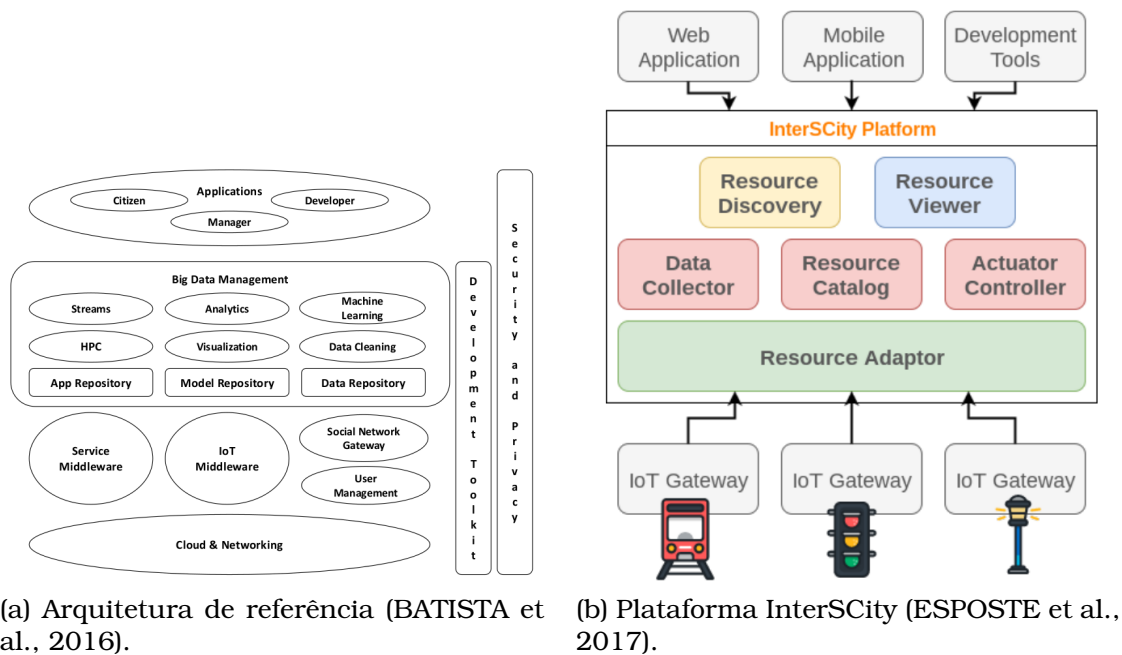


Figura 2.6: *Middleware* de Cidades Inteligentes

para que a preempção ocorra no momento correto.

Assim como outros autores, pretendemos abordar o problema do tempo perdido pelo EV no trânsito, sendo necessário verificar o que já foi feito, quais as qualidades e possibilidades de melhoria das soluções disponíveis. A próxima seção apresenta uma breve análise de trabalhos diretamente relacionados ao Problema de Redução do Tempo Perdido pelo EV no trânsito e o porquê de alguns deles não apresentarem resultados satisfatórios.

A integração em arquiteturas de Cidades Inteligentes oferece a camada de orquestração necessária para superar as limitações locais dos sensores e o custo proibitivo da infraestrutura dedicada em cada esquina ou de exigir uma frota 100% conectada. É neste contexto de dados centralizados e gestão holística que as lacunas das soluções atuais, analisadas na seção a seguir, tornam-se evidentes e onde a proposta deste trabalho se situa.

2.7 Análise de Trabalhos Relacionados

A literatura apresenta uma vasta gama de propostas para mitigar o atraso de veículos de emergência. No entanto, ao submeter esses trabalhos aos Critérios de Avaliação definidos na Seção 2.2 — especialmente quanto à capacidade de atuação em rota completa, segurança e escalabilidade —, notam-se lacunas recorrentes que comprometem sua aplicabilidade em cenários urbanos complexos e congestionados.

Para fins de análise crítica, agrupamos as abordagens identificadas em quatro categorias principais.

2.7.1 Soluções Baseadas em Detecção Local e Pontual

Um grupo significativo de trabalhos e patentes foca na detecção do EV na iminência da interseção, utilizando sensores IR, sonoros ou de radiofrequência simples, utilizando tecnologias como *ZigBee* ou RFID para acionamento local (HUSSIN et al., 2012; SUNDAR; HEBBAR; GOLLA, 2015; ALBAGUL; HRAIRI; HIDAYATHULLAH, 2006; HASHIM et al., 2013; MIYAWAKI; YAMASHIRO; YOSHIDA, 1999). Há ainda propostas que utilizam uma rede de sensores sem fio (*Rede de Sensores sem Fio* (WSN)) e nós sensores de tráfego para verificar a densidade e aplicar um algoritmo de despacho (CHAKRABORTY; TIWARI; SINHA, 2015).

Como discutido na Seção 2.3, essas soluções falham no critério de Alcance. A detecção ocorre muito próxima ao semáforo, muitas vezes quando o EV já está retido pela fila de congestionamento. Embora existam trabalhos que tentam mitigar essa situação com a utilização de RFID para a contagem de veículos (SHARMA et al., 2013), a limitação física da leitura da *tag* impede a criação de uma “Onda Verde” coordenada para múltiplas interseções à frente, resultando em uma atuação puramente reativa e local.

2.7.2 Soluções Baseadas em Conectividade Veicular (V2V/V2I)

Avançando para abordagens distribuídas, há vários trabalhos que propõem o uso de comunicação veicular (DSRC, WAVE, VTL) para que os próprios carros negociem a prioridade ou formem semáforos virtuais (CHOU; TSENG; YANG, 2013; HUMAGAIN; SINHA, 2020; ZLATKOVIC et al., 2023; MOROI; TAKAMI, 2015). Há trabalhos específicos que utilizam o padrão IEEE 802.11p WAVE e mensagens *Cooperative Awareness Messages* (CAM) do padrão europeu para realizar a preempção (UNIBASO et al., 2010). Neste caso, o algoritmo pondera a importância das mensagens baseada na distância e ângulo do veículo. Contudo, o foco da avaliação é diminuir a probabilidade de erros na troca de mensagens, e não explicitamente reduzir o tempo de atraso do EV.

Embora tecnicamente promissoras para a limpeza de filas (*lane clearing*), essas soluções esbarram no critério de Custo e Infraestrutura. Elas exigem uma taxa de penetração de tecnologia de quase 100% na frota circulante ou a instalação massiva de RSUs. Em cenários de transição tecnológica, como o brasileiro, onde a frota é antiga, a dependência exclusiva de V2V torna a solução inaplicável a curto e médio prazo.

2.7.3 Soluções Baseadas em IoT e Arquiteturas Centralizadas

Estes trabalhos aproximam-se da abordagem de Cidades Inteligentes, utilizando GPS e redes celulares (*Global System for Mobile Communications* (GSM)/4G)

para controlar semáforos remotamente (SAHANA et al., 2016; MISBAHUDDIN et al., 2015; SIDDIQI et al., 2023; KHAN et al., 2018; ALMURAYKHI; AKHLAQ, 2019). Há abordagens que propõem um sistema que combina módulos GPS e microcontroladores Atmel (AVR) para calcular a posição do veículo e enviar solicitações de preempção via mensagens SMS (rede GSM) (ELTAYEB; ALMUBARAK; ATTIA, 2013). Há ainda abordagens que focam no planejamento dinâmico e controle adaptativo de rota (GEDAWY; DIAS; HARRAS, 2008). É possível encontrar soluções de preempção baseadas na rota completa (ZHAO et al., 2018; MU; SONG; LIU, 2018), enquanto há trabalhos que sugerem métodos adaptativos para lidar com situações de emergência em cidades inteligentes (HAJIEBRAHIMI; IRANMANESH, 2018; PARUCHURI, 2017).

Muitas dessas propostas falham em Robustez e Segurança. Soluções que dependem de acionamento manual por aplicativo de *smartphone* pela equipe de socorro introduzem carga cognitiva perigosa em momentos críticos. Além disso, a falta de algoritmos de otimização de rota completa faz com que muitos desses sistemas atuem apenas na próxima interseção, sem visão global do tráfego.

2.7.4 Soluções Baseadas em IA e Métodos Formais

Recentemente, abordagens utilizando Aprendizado de Máquina (CHENG et al., 2025; ALZAMZAMI et al., 2025) e Métodos Formais como Redes de Petri (PEI et al., 2020; BANIARDALANI; SOLEIMANI, 2025; HUANG; WENG; ZHOU, 2015; ROW; LEE; CHENG, 2024) ganharam destaque.

Há ainda modelos analíticos baseados em teoria das filas e onda de choque, que buscam quantificar o impacto da preempção no tráfego transversal (BAI et al., 2022; HAGHANI; KLUGER, 2025).

As soluções de IA, embora eficientes em média, carecem de determinismo e garantias formais de segurança, essenciais para sistemas críticos. Por outro lado, os trabalhos que usam Redes de Petri focam excessivamente na verificação de propriedades de segurança de uma única interseção, ignorando a dinâmica da rede e o impacto do tráfego em larga escala (escalabilidade).

2.7.5 Síntese das Lacunas

A análise transversal da literatura revela deficiências sistêmicas que resumimos na Tabela 2.2. As principais lacunas identificadas são:

- **Falta de Atuação em Rota Completa:** A vasta maioria dos trabalhos foca em resolver o problema de uma interseção isolada, ignorando que o atraso do EV é cumulativo ao longo de todo o trajeto.

- **Cenários Irreais:** O uso de *grids* sintéticos ou simulações com poucos veículos mascara o comportamento caótico do trânsito real, impedindo a generalização dos resultados.
- **Métricas Inadequadas:** Muitos trabalhos avaliam o “tempo médio de espera” ou “número de mensagens trocadas”, métricas que não refletem diretamente o objetivo final: a redução do tempo total de viagem do EV.
- **Indisponibilidade de Código:** A falta de código aberto impede a reprodutibilidade e a comparação justa entre os algoritmos.

Tabela 2.2: Análise de limitações nas soluções da Literatura

Trabalho	Sem Avaliação de De- sempenho	Limitações Tecnológicas	Métricas inade- quadas	Escopo Limitado (Interse- ções)	Modelo Não Ge- neralizável	Cenário Real / Sinté- tico	Resultados não comparáveis	Código Indispo- nível	Difícil Implan- tação
Patentes - Smith, Davidson e Pfister (1988), etc	✓	✓					✓	✓	✓
Albagul, Hrairi e Hidayathullah (2006)	✓			✓		✓	✓	✓	
Hussin et al. (2012)	✓	✓		✓		✓	✓	✓	
Hashim et al. (2013)	✓			✓		✓	✓	✓	
Sundar, Hebbar e Golla (2015)	✓	✓				✓	✓	✓	✓
Sharma et al. (2013)	✓						✓	✓	
Sahana et al. (2016)	✓						✓	✓	
Misbahuddin et al. (2015)	✓						✓	✓	
Unbaso et al. (2010)			✓			✓	✓	✓	✓
Eltayeb, Almubarak e Attia (2013)			✓	✓		✓	✓	✓	
Gedawy, Dias e Harras (2008)			✓	✓		✓	✓	✓	
Chou, Tseng e Yang (2013)			✓	✓		✓	✓	✓	✓
Chakraborty, Tiwari e Sinha (2015)			✓	✓		✓	✓	✓	
Miyawaki, Yamashiro e Yoshida (1999)	✓	✓	✓	✓		✓	✓	✓	
Almuraykhi e Akhlaq (2019)			✓	✓		✓	✓	✓	
Shankar, Gautham e Vedaprakashvarma (2015)			✓	✓		✓	✓	✓	
Qin e Khan (2012)			✓	✓		✓	✓	✓	
Hajjebrahimi e Iranmanesh (2018)			✓	✓	✓	✓	✓	✓	
Huang, Weng e Zhou (2015)			✓	✓	✓	✓	✓	✓	
Zhao et al. (2018)			✓	✓	✓	✓	✓	✓	
Mu, Song e Liu (2018)			✓	✓	✓	✓	✓	✓	
Khan et al. (2018)			✓	✓	✓	✓	✓	✓	
Paruchuri (2017)			✓	✓	✓	✓	✓	✓	
Morol e Takami (2015)			✓	✓	✓	✓	✓	✓	
Djahel et al. (2015)			✓	✓	✓	✓	✓	✓	
Kapusta et al. (2017)			✓	✓	✓	✓	✓	✓	
Baniardalani e Soleimani (2025)	✓		✓	✓	✓	✓	✓	✓	✓
Row, Lee e Cheng (2024)			✓	✓	✓	✓	✓	✓	✓
Haghani e Kluger (2025)			✓	✓	✓	✓	✓	✓	✓
Cheng et al. (2025)			✓	✓	✓	✓	✓	✓	✓
Alzanzami et al. (2025)			✓	✓	✓	✓	✓	✓	✓
Zlatkovic et al. (2023)			✓	✓	✓	✓	✓	✓	✓
Humagaim e Sinha (2020)	✓		✓	✓	✓	✓	✓	✓	✓
Siddiqi et al. (2023)	✓		✓	✓	✓	✓	✓	✓	✓
Pet et al. (2020)				✓	✓	✓	✓	✓	✓
Bai et al. (2022)			✓	✓	✓	✓	✓	✓	✓

Para a correta interpretação da tabela, deve-se notar que algumas células foram intencionalmente deixadas em branco. Isso ocorre quando uma coluna se torna logicamente inconsistente com outra, como em “Métricas inadequadas” para um trabalho “Sem Avaliação de Desempenho”. De forma mais ampla, a frequente indisponibilidade do código-fonte (“Código Indisponível”) limitou a comparação entre os resultados (“Resultados não comparáveis”). Fundamentalmente, um campo vazio não significa que a condição inversa é verdadeira. Muitas vezes, a ausência de dados, como na avaliação de generalização do modelo (“Modelo Não Generalizável”), reflete a insuficiência de informações no trabalho original para realizar tal análise.

Esta tese busca preencher exatamente essas lacunas, propondo uma solução de rota completa, validada formalmente (Redes de Petri) e testada em cenários de larga escala com dados reais.

2.8 Conclusão

A revisão do estado da arte apresentada neste capítulo permitiu uma análise aprofundada das tecnologias e metodologias empregadas para a priorização de veículos de emergência. A evolução histórica demonstra uma clara tendência de migração de soluções puramente sensoriais e locais (detecção por som ou luz na esquina) para arquiteturas conectadas e distribuídas (V2V/V2I), culminando na integração com ecossistemas de Cidades Inteligentes.

Apesar dos avanços significativos, a aplicação dos critérios de avaliação definidos na Seção 2.2 revelou que a literatura ainda carece de uma solução que atenda simultaneamente aos requisitos de atuação em rota completa, robustez de segurança e escalabilidade em cenários realistas. Observou-se que:

1. Soluções de baixo custo (sensores locais) falham no alcance, sendo incapazes de limpar o trânsito com a antecedência necessária para evitar a parada do EV.
2. Soluções baseadas em V2V são boas para alertar motoristas, mas dependem de uma taxa de adoção massiva que ainda não é realidade, além de não controlarem diretamente a infraestrutura semafórica de forma coordenada.
3. A maioria das propostas acadêmicas falha na validação: utilizam cenários simplificados (*grids* artificiais) e métricas que mascaram o impacto real no tempo de resposta da emergência.

Diante deste cenário, constata-se a necessidade de uma abordagem que una a visão global das Cidades Inteligentes (para gerenciamento de rota e semáforos) com métodos rigorosos de validação.

Esta tese propõe preencher essas lacunas através do desenvolvimento de uma Arquitetura de Controle Semafórico Inteligente Baseada em Redes de Petri e Otimização de Rota Completa. Diferente dos trabalhos correlatos, nossa proposta não se limita a abrir o sinal na chegada do veículo; ela orquestra o tráfego preventivamente ao longo de todo o trajeto, garantindo fluidez contínua. Além disso, a solução inova ao utilizar métodos formais para garantir a segurança da lógica de controle (prevenção de *deadlocks* e acidentes) e validação em cenários de larga escala extraídos de dados reais de tráfego.

Dessa forma, o presente trabalho avança o estado da arte ao oferecer uma solução que é, ao mesmo tempo, tecnicamente robusta, segura e viável para as metrópoles modernas. O próximo capítulo apresenta a primeira solução desenvolvida neste trabalho.

Soluções Seleccionadas para Comparação e Modelo de Simulação

Dentre as soluções apresentadas no capítulo anterior, é necessário estabelecer quais podem ser usadas como bases de comparação para as soluções que serão apresentadas neste trabalho. Neste capítulo, apresentamos a base metodológica para a avaliação experimental das soluções propostas neste trabalho. Inicialmente, apresentamos o **modelo do sistema** (Seção 3.1), que define a arquitetura e os componentes comuns a todas as abordagens. Em seguida, detalhamos as **soluções da literatura** que foram seleccionadas como base de comparação (Seção 3.2), cujos desempenhos servirão de *baseline* para as propostas deste trabalho. Posteriormente, discutimos a infraestrutura de simulação, justificando a escolha do **modelo microscópico e do simulador SUMO** (Seção 3.3) e descrevendo em detalhes os múltiplos cenários sintéticos e reais que compõem o ambiente experimental (Seção 3.4). Por fim, definimos a **metodologia de avaliação** na Seção 3.5, incluindo o *framework* de software desenvolvido e as métricas de desempenho que serão utilizadas para aferir a eficácia dos algoritmos. Dessa forma, este capítulo consolida todos os elementos necessários para a contextualização e análise dos resultados que serão apresentados nos capítulos subsequentes.

3.1 *Modelo do Sistema*

O sistema considera as ruas das cidades como arestas da rede e as interseções como os respectivos nós. Para fins de clareza na exposição teórica, a formulação apresentada nesta seção assume, por simplificação, que há apenas uma faixa em cada uma das arestas. No entanto, deve-se ressaltar que esta é apenas uma abstração conceitual: o modelo desenvolvido é totalmente compatível com a geometria real de vias com múltiplas faixas. A implementação e os experimentos, detalhados posteriormente, não sofrem desta restrição e operam sobre o cenário real de múltiplas vias. Diferente de outras propostas, a modelagem do sistema não é limitada a interseções de quatro vias, como as encontradas em trabalhos similares. No modelo utilizado neste trabalho, o EV cruza um sem-número de interseções semaforicas, pertencentes à rota deste veículo. Os semáforos neste modelo obedecem a uma programação e temporização, definidas como a sincronização dos respectivos semáforos. As interseções que não são controladas por semáforos não são abordadas neste trabalho.

Cada uma das vias, assim como no mundo real, possui uma velocidade máxima a ser obedecida pelos veículos. Embora a velocidade do EV não seja limitada a velocidade da via, ela precisa ser conhecida para a correta operação do sistema.

Ao iniciar, o sistema possui as informações da rota do EV, origem e destino, e as informações de posição e fases de todos os semáforos presentes na rota do EV. A informação de fase necessária é a programação completa do semáforo: quais das luzes do semáforo devem estar acesas em qualquer instante de tempo t da execução do sistema. Neste modelo, as informações no sistema são atualizadas a cada segundo. Nas atualizações, são coletadas as informações do EV (posição, velocidade e aceleração) e informações do fluxo da via, incluindo o tamanho das filas formadas pelos demais veículos nos semáforos. O sistema deve ser capaz de processar as informações e enviar comandos para os semáforos, tais como trocar suas fases e encerrar a fase atual, por exemplo.

Baseado no processamento efetuado, o sistema deve, dentre as fases dos semáforos, decidir qual é a fase que será ativada quando o EV se aproximar do semáforo, de sorte que a luz esteja verde para ele. Como tratamos todos os semáforos existentes na rota do EV, o sistema deve ser capaz de, dada qualquer fase dos demais semáforos da rota, determinar qual é a antecedência necessária para que os semáforos alterem para a fase desejada com segurança, em um dado instante de tempo t , passando assim pelas fases de luzes amarela e vermelha para os outros veículos, garantindo assim a segurança do trânsito de forma geral. Para isso, o sistema realiza uma transição segura para a

luz verde, diminuindo o intervalo de tempo em que as luzes ficam vermelhas para os demais semáforos. Essa operação quebra a sincronização original dos semáforos, que deve ser restabelecida assim que o EV passa pelo semáforo, retornando-o à sua atividade normal, já que o sistema conhece qual a fase desse determinado semáforo deveria estar naquele instante de tempo t ou posterior. Assim, com base no restante de tempo da fase atual, ele consegue aguardar uma fase vermelha intermediária para então alterar para a fase correta de sua programação, ajustando o restante deste tempo para o correto da tabela de programação do semáforo.

Observe que neste modelo não há necessidade de RSUs (*Road Side Unit*) já que apenas o EV envia dados para o sistema.

A seguir, apresentamos uma seleção de trabalhos que se adequam ao modelo descrito e para o qual foi possível fornecer as respectivas implementações. Isso nos permitirá comparar tais trabalhos com as soluções que propusemos.

3.2 Soluções Seleccionadas Para Comparação

A seleção dos trabalhos que serão utilizados para comparação implica na escolha de trabalhos representativos, que possam ter critérios claros de comparação e, em especial, dos que possam ser adaptados ao modelo proposto. Portanto, uma condição estabelecida era que a solução deveria ser factível e estar disponível para análise. Contudo, isso não se mostrou uma tarefa fácil. Foram realizadas tentativas de contato com os autores dos trabalhos para obter o código-fonte de suas implementações. Nenhuma foi bem-sucedida. Diante disso, dos trabalhos passíveis de serem implementados apenas com as informações constantes nos trabalhos, foi possível derivar uma solução baseada em Lógica Fuzzy (DJAHEL et al., 2015) e outra baseada em Filas nos semáforos (KAPUSTA et al., 2017). As implementações foram feitas de acordo com o descrito nos artigos. De qualquer forma, não é possível garantir que o desempenho apresentado por essas implementações seja maior ou menor do que o obtido por este trabalho.

Analisando os trabalhos relacionados, notamos ainda que alguns usaram tecnologias recorrentes como, por exemplo, a utilização de sensores RFID para detecção do EV para diminuição de seu tempo perdido. Embora tais trabalhos não possuam algoritmos explícitos em suas propostas, foi possível entender o comportamento das soluções e fornecer uma implementação que pôde ser usada para fins comparativos com as outras propostas. Também fornecemos uma solução que mantém todo o trajeto do EV com luzes verdes desde o início da preempção até o final.

3.2.1 Soluções baseadas em RFID

Soluções que utilizam *tags* RFID representam uma grande classe de algoritmos baseados em sensores. A solução apresentada neste trabalho não está diretamente relacionada a nenhum dos trabalhos prévios que usam RFID (HEGDE; SALI; INDIRA, 2013; SHARMA et al., 2013; HASHIM et al., 2013; MITTAL; BHANDARI, 2013). Contudo, o objetivo de se usar *tags* RFID é detectar quando o EV está próximo do semáforo para que sua passagem seja facilitada. Portanto, seu comportamento é simples o suficiente para que possamos propor uma solução semelhante para fins comparativos neste trabalho.

Nessa solução, há uma *tag* RFID afixada no para-brisa do EV, que está se aproximando de um semáforo TL_i . A solução identifica quando o EV está à distância de detecção $dd(m)$ do TL_i e inicia a preempção de TL_i , deixando a luz verde para o EV. A Figura 3.1 apresenta um exemplo de distância de detecção utilizando uma *tag* RFID.

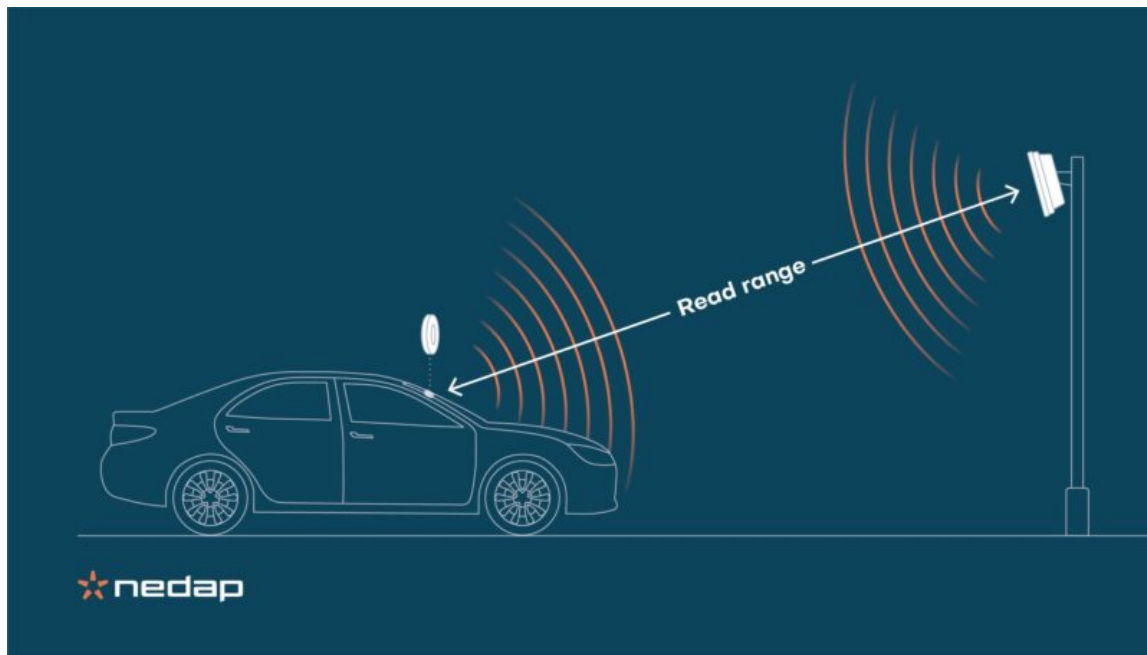


Figura 3.1: Distância de Detecção (*Read range*) do EV. Fonte da imagem: <https://www.nedapidentification.com/insights/understanding-rfid-read-ranges-for-automatic-vehicle-identification/>

A luz verde permanece ativa por um número fixo de ciclos igual a n_c (ou seja, um tempo fixo para a preempção ocorrer).

Como mencionado, os trabalhos analisados não descrevem explicitamente um algoritmo e não disponibilizam os códigos associados aos experimentos. Com base nas ideias de tais trabalhos, apresentamos o Algoritmo 3.1, que soluciona o problema de redução de tempo perdido pelo EV no trânsito.

Embora não tenhamos encontrado uma solução de tal natureza nos trabalhos relacionados, sua simplicidade sugere que alguma já alguma possa ter

sido descrita anteriormente.

Este Algoritmo é uma abstração: na situação real, a distância do EV até o semáforo não é calculada. O sensor instalado na infraestrutura da via lê a *tag* RFID e envia a informação da detecção para a central, e o Algoritmo se encarrega de abrir o semáforo. O cálculo da distância apresentado no Algoritmo também é uma simplificação. Para precisão, a distância geodésica deveria ser calculada. Contudo, devido às pequenas distâncias em relação às distâncias reais, as diferenças nos cálculos não são significativas. Essa simplificação é adequada para representar a detecção do EV utilizando o sensor RFID no nosso modelo.

Algoritmo 3.1: Preempção com RFID

```

1 enquanto EV está atendendo uma emergência faça
2   para TL ∈ proximo_conjunto_TLs faça
3      $d \leftarrow \sqrt{(x_{EV} - x_{TL})^2 + (y_{EV} - y_{TL})^2}$ ;
4     se  $d \leq dd$  então
5       abrir_TL_por_ciclos(TL,nc);
6     fim
7   fim
8 fim

```

3.2.2 Soluções baseadas em Fuzzy

A solução proposta por Djahel et al. (2015) utiliza informações globais das vias e Lógica Fuzzy (ZADEH, 1965) para inferir os níveis de congestionamento usando dados de ocupação das vias e prevê a utilização de V2I para solicitar a cooperação dos outros veículos na passagem do EV. Como foi possível derivar uma implementação a partir do trabalho original (com as devidas considerações que serão feitas a seguir), esta foi a escolha para a representação dessa classe de soluções.

Os autores utilizam a taxa de ocupação OL_{faixa_i} (Equação 3.1) e diferentes níveis de emergência (*EL*) para o EV (baixo, médio e alto) para determinar se deve iniciar a preempção de TL_i . Como os autores não forneceram o código-fonte, tampouco há um algoritmo explicitamente descrito no trabalho, foi necessário deduzi-lo e preencher algumas lacunas e parâmetros com inferências, resultando no Algoritmo 3.2.

$$OL_{faixa_i} = \frac{N_{faixa_i} \times (\overline{Veh_{comprimento}} + Veh_{gap})}{comprimento_{faixa_i}} \times 100\% \quad (3.1)$$

O trabalho menciona que a função *defuzzify* usa o método *Center of Gravity* (COG) e nada mais é dito sobre os outros parâmetros. Por este motivo, com

Algoritmo 3.2: Soluções baseadas em Lógica Fuzzy

```
1 para  $aresta \in arestas\_da\_rota$  faça
2    $velocidade_{aresta} \leftarrow velocidade\_media\_da\_aresta(e);$ 
3   para  $faixa_i \in faixas\_da\_aresta(e)$  faça
4      $OL_{faixa_i} \leftarrow taxa\_ocupacao$  /* conforme Equação 3.1 */;
5   fim
6 fim
7  $ERP \leftarrow defuzzify()$  /* usando o Anexo A(media( $velocidade_{aresta}$ ),
   media( $OL_{faixa_i}$ ,EL)) */;
8 enquanto  $EV$  está atendendo uma emergência faça
9   se  $ERP \geq 1$  E  $aresta(TL) == aresta(EV)$  então
10    abrir o próximo TL por 1 ciclo;
11  fim
12  se  $ERP \geq 2$  então
13    para  $aresta \in arestas\_da\_rota$  faça
14      aumentar o limite de velocidade da aresta em em SL%;
15    fim
16  fim
17  se  $ERP \geq 3$  então
18    para  $veículo \in veiculos(faixa_{EV})$  faça
19      solicitar ao veículo para mudar de faixa;
20    fim
21  fim
22  se  $ERP \geq 4$  então
23    para  $veículo \in veiculos(rota_{EV})$  faça
24      solicitar ao veículo para alterar sua rota;
25    fim
26  fim
27  se  $ERP == 3$  OU  $ERP == 5$  então
28    /*SEM AÇÃO*/
29  fim
30 fim
```

base na lógica *fuzzy*, foram criados triângulos não sobrepostos para representar as classes *negligible*, *low*, *medium*, *high* e *critical* para gerar o *Congestion Level* (CL) (Anexo A).

O trabalho também não descreve detalhes sobre a frequência de atualização dos parâmetros *Average Vehicle Speed* (AVS) e *Occupancy Level* (OL), portanto, eles estão sendo calculados quando o EV entra na rede.

Ainda sobre a OL, o trabalho informa que tal valor é obtido para cada uma das faixas das vias, mas não diz como esse valor é consolidado para ser usado em suas equações e, portanto, foi utilizada a média desses valores.

Após o cálculo de *Emergency Response Plan* (ERP), as ações a serem executadas não estão totalmente definidas; portanto, neste trabalho foi considerado que:

- Mudança do Semáforo: a preempção ocorre quando o EV chega no segmento da rota do semáforo, alterando (ou estendendo) seu ciclo por 1 ciclo;
- Alteração do Limite de Velocidade: Não foi definido qual o percentual da mudança. Neste trabalho, portanto, o Limite de Velocidade (*SL*) foi aumentado em 50%;
- Reroteamento: uma nova aresta, escolhida aleatoriamente obedecendo a uma distribuição uniforme, é escolhida para o veículo alvo. Após o fim da emergência, o veículo é roteado novamente para o destino original; e
- A nossa descrição do algoritmo não utiliza vias reservadas. Portanto, quando o Algoritmo indicar que o EV utilize faixas reservadas para outros veículos, nada é feito.

No nosso entendimento, as ações tomadas na nossa implementação não comprometem o desempenho do algoritmo, mas não é possível afirmar que o algoritmo implementado é precisamente o mesmo descrito no trabalho original.

3.2.3 Soluções baseadas em Filas

Há uma variedade de trabalhos que analisam as filas de veículos causadas pelos semáforos no trânsito. Dos trabalhos que utilizam essa informação para fornecer mecanismos de preempção para veículos de emergência, foi possível implementar uma solução usando as informações de Kapusta et al. (2017), que representa, portanto, essa classe de soluções.

No trabalho, a proposta de solução do problema de redução de tempo perdido pelo EV no trânsito utiliza apenas uma interseção da via. Também não

foi possível obter o código-fonte do referido trabalho. Considerando que no nosso modelo utilizamos um conjunto de interseções, utilizando os mesmos parâmetros do trabalho Kapusta et al. (2017), apresentamos o Algoritmo 3.3, que estende a solução original. A solução deve conhecer os valores da distância $d(m)$ do EV para o semáforo e da velocidade do EV ($v_{EV}(m/s)$). Quando o tempo de chegada do EV até o semáforo ($chegada_{EV}$ (s)) atinge um certo limiar, a preempção é iniciada. Esse limiar depende de constantes definidas pelos autores, no caso, T_{Beta} e $T_{Alpha}(s)$, e do tempo de duração do ciclo do semáforo $ciclo_{duracao}(s)$. O limiar é calculado conforme a Equação 3.2.

$$T_{Beta} * ciclo_{duracao} \leq chegada_{EV} \leq T_{Alpha} * ciclo_{duracao} \quad (3.2)$$

Quando a preempção é iniciada, a solução fica alternando as luzes do semáforo, variando entre aumentar o tempo de luz verde e diminuir o tempo de luz vermelha, e essa ação depende do tamanho da fila de veículos ($Q(m)$) e de outras constantes definidas pelos autores, s_{Alpha} e $s_{Beta}(m)$. Quando um segundo limiar, apresentado na Equação 3.3, é atingido, o semáforo fica verde indefinidamente, até que o EV o cruze.

$$chegada_{EV} < T_{Beta} * ciclo_{duracao} \quad (3.3)$$

Os valores T_{Alpha} , T_{Beta} , s_{Alpha} e s_{Beta} são valores definidos arbitrariamente pelos autores, e os mesmos valores são usados e detalhados na Seção de Implementação 3.5.2.3.

Há três problemas com este algoritmo. Primeiro, ele é projetado para atuar em uma interseção individual, não há troca de dados entre as interseções na rota. Segundo, em cenários de grande congestionamento, esta ação poderia acontecer tarde demais, já que o algoritmo é dependente do tamanho do ciclo. E por último, uma vez que a preempção esteja ativa, ele nunca a encerrará até que o EV cruze a interseção. No mesmo cenário de alto congestionamento, esta falta de movimento poderia ser um efeito colateral da preempção (isto é, se uma via perpendicular de alta prioridade estiver vermelha por causa da preempção, mas a via do EV está verde sem movimento dos veículos). Esta é uma das situações em que manter a preempção é pior do que deixar o tráfego fluir de acordo com seu estado original sem preempção.

3.2.4 Soluções baseadas em Preempção Imediata - Onda Verde

A solução ótima para este problema é quando o EV não perde tempo nenhum no seu trajeto, ou seja, ele percorre toda sua rota na velocidade máxima permitida. Embora não se possa garantir que não haverá outros veículos atrapalhando o trajeto do EV, para fins de comparações, é possível ativar a pre-

Algoritmo 3.3: Soluções baseadas em filas

```

1 enquanto EV está atendendo uma emergência faça
2   para  $i \in \text{proximo\_conjunto\_TLs}$  faça
3      $\text{chegada}_{EV} \leftarrow \frac{d}{v_{EV}};$ 
4     se  $T_{Beta} \cdot \text{ciclo}_{duracao} \leq \text{chegada}_{EV} \leq T_{Alpha} \cdot \text{ciclo}_{duracao}$  então
5       se  $Q > s_{Alpha}$  então
6         Aumenta a duração da fase verde;
7       fim
8       se  $Q > s_{Beta}$  então
9         Diminui a duração da fase vermelha;
10      fim
11    fim
12    se  $\text{chegada}_{EV} < T_{Beta} \cdot \text{ciclo}_{duracao}$  então
13      se  $\text{fase\_atual} == \text{verde}$  então
14        Mantenha a luz verde;
15      fim
16      se  $\text{fase\_atual} == \text{vermelho}$  então
17        Altere para a luz verde;
18      fim
19    fim
20  fim
21 fim

```

empção para todos os semáforos da rota do EV assim que ele entra na rede. Assim, o Algoritmo 3.4 foi projetado neste trabalho como *baseline*, mas tal qual o Algoritmo 3.1, é perfeitamente plausível supor que outros trabalhos tenham descrito algo parecido. Este algoritmo pode ser pensado como um bom limite superior para algoritmos baseados apenas em preempção de semáforos, que é o caso de todos os algoritmos apresentados até o momento. Contudo, este Algoritmo sofre do mesmo problema descrito na Subseção 3.2.3, pois ele jamais cancela uma preempção mal-sucedida. Além disso, este algoritmo pode deixar a rede ociosa, já que o EV pode estar longe e o semáforo já está aberto, criando filas nas proximidades.

Algoritmo 3.4: Onda Verde

```

1 para  $TL_i \in \text{TLs\_na\_rota}$  faça
2   abrir  $TL_i$  indefinidamente;
3 fim
4 aguarde até que o EV finalize a emergência;
5 para  $TL_i \in \text{TLs\_na\_rota}$  faça
6   sincronize  $TL_i$ ;
7 fim

```

Com os algoritmos apresentados, a próxima seção descreve o modelo de simulação. Este modelo é utilizado pela maioria dos trabalhos estudados e

também é utilizado por este trabalho.

3.3 *Simulação e Modelos de Tráfego*

Na impossibilidade de se testar em uma infraestrutura de trânsito real, os simuladores são utilizados na ampla maioria dos trabalhos verificados na literatura. A cidade (principalmente suas vias) é modelada como uma matriz ou digrafo, e o nível de detalhes do modelo pode alcançar veículos, pessoas, semáforos, entre outros elementos. A granularidade de detalhes especifica qual modelo é adotado. Há basicamente três tipos de modelos (BARCELÓ, 2010), que são apresentados nas Subseções 3.3.3, 3.3.2 e 3.3.1, respectivamente: Microscópico, Mesoscópico e Macroscópico. O modelo de tráfego apresentado na Seção 3.1 é implementado em um modelo microscópico.

3.3.1 *Macroscópico*

Em Modelos Macroscópicos, o trânsito é modelado como fluxos na rede e, portanto, os elementos não são individualmente modelados, e as velocidades são inferidas observando o tamanho do fluxo em um dado período. Devido a essas características, simuladores macroscópicos são capazes de modelar trânsitos de países inteiros. Como exemplo de simulador que utiliza tal modelo, podemos citar o VISUM ¹.

3.3.2 *Mesoscópico*

Embora neste modelo cada elemento seja individualmente modelado, a atualização da posição e velocidade dos elementos obedece a uma função de densidade que geralmente considera o comprimento da via e o número de veículos presentes na rua para calcular sua capacidade. Por este motivo, este modelo consegue executar simulações de grandes áreas metropolitanas ou até cidades completas. O próprio SUMO possui uma versão de execução mesoscópica, mas outros simuladores podem ser encontrados como InterSCSimulator (SANTANA et al., 2018), MATSim (AXHAUSEN; HORNI; NAGEL, 2016), DEUS (PICONE; AMORETTI; ZANICHELLI, 2012), entre outros. Contudo, a posição de um elemento da rede não pode ser obtida com precisão, condição necessária para os algoritmos que abordam o problema do tempo perdido pelo EV no trânsito, que precisam calcular a distância entre o EV e os semáforos da rota.

¹<https://www.ptvgroup.com/en/solutions/products/ptv-visum/>

3.3.3 Microscópico

Neste tipo de modelo, o comportamento dos elementos da rede é individualmente calculado baseado nos demais componentes, e a velocidade e posição do veículo são calculadas usando modelos matemáticos chamados *Car Following* (que altera a velocidade baseado no veículo à frente) e *Lane Change* (que altera a faixa de viagem baseado em fatores como urgência ou lentidão do veículo à frente). De modo geral, quando implementado em cenários maiores, o tempo de execução pode inviabilizar a solução. Contudo, este é o único dentre os modelos que permite a verificação precisa da velocidade e posição de algum elemento na rede, que é o caso do nosso objeto de estudo. Exemplos de simuladores que utilizam este modelo são o SUMO (LOPEZ et al., 2018) e o VISSIM ².

3.3.4 SUMO

Considerando a necessidade de se utilizar um simulador microscópico, já que a velocidade e posição do EV devem ser precisamente obtidas, optamos por utilizar o simulador SUMO³ (LOPEZ et al., 2018), inicialmente desenvolvido pelo Instituto de Sistemas de Transporte do Centro Aeroespacial Alemão e agora mantido pela Eclipse Foundation. Este simulador pode ser configurado para utilizar o modelo de simulação microscópica e, portanto, cada veículo reage por meio de modelos de *Car Following* e *Lane Change* ditos anteriormente.

O fato de o SUMO ter o código aberto favoreceu a escolha, já que assim é possível fazer intervenções no código-fonte, o que se mostrou necessário neste trabalho e que propomos posteriormente. Cabe salientar que o simulador VISSIM, concorrente direto do SUMO, não pôde ser utilizado, pois este é um *software* proprietário e possui algumas restrições em sua licença. A sua versão de Estudante possui severas limitações para o tamanho da rede e tempo de simulação, e a versão “Tese” é limitada a no máximo seis meses de utilização⁴. Além disso, a grande maioria dos trabalhos avaliados também utiliza o SUMO como simulador, o que facilita a comparação dos resultados obtidos tanto por este trabalho quanto pelos trabalhos avaliados. Mais especificamente, para os resultados avaliados neste trabalho que tratam de redução do tempo perdido pelo EV no trânsito e que usam simulação, a grande maioria deles opta pelo SUMO, tal como (DJAHEL et al., 2015). As exceções são explicitamente indicadas no texto deste trabalho.

A rede no SUMO é representada por um digrafo, aderente ao Modelo de Trá-

²<https://www.ptvgroup.com/en/solutions/products/ptv-vissim/>

³<https://eclipse.dev/sumo/>

⁴https://company.ptvgroup.com/en/about/academia/licenses?_ga=2.193426669.1572819530.1642127570-467923805.1642127570

fego apresentado na Seção 3.1, onde as interseções são os nós e as vias são as arestas. O SUMO possui diversas ferramentas para facilitar a sua utilização, como, por exemplo, o OSMWebWizard ⁵, um script Python que baixa uma área livremente selecionada do OpenStreetMap (OpenStreetMap contributors, 2025) e gera demandas a partir de parâmetros informados em sua interface.

Como há um espaço de tempo entre as construções e execuções dos experimentos deste trabalho, várias versões do SUMO foram usadas, sendo elas 1.4, 1.8 e 1.16. Quando for o caso, essas versões ficarão explícitas nas descrições dos experimentos.

A próxima Seção apresenta os cenários que são utilizados ao longo deste trabalho e suas particularidades.

3.4 Cenários

Para a avaliação dos cenários, partimos da premissa de que é possível obter os dados do EV e, como a frota dos veículos de emergência é muito menor que a frota dos veículos particulares, é razoável imaginar que um *smartphone* autorizado pode fornecer tais informações. É razoável também presumir que a coleta de informações de semáforos, bem como o comando remoto deles, pode ser feita sem grandes complicações e, por este motivo, supomos que um *Middleware* de cidades inteligentes esteja disponível. A Figura 3.2 mostra a arquitetura deste Sistema. Além disso, precisamos coletar o tamanho das filas nos semáforos (que pode ser feito utilizando câmeras de monitoramento/*loops* indutivos). Após essas considerações, os cenários que serão usados para a avaliação dos algoritmos propostos neste trabalho são apresentados a seguir. A seguir, apresentamos os quatro cenários, que foram construídos à medida do desenvolvimento deste trabalho.

3.4.1 Cenários Sintético de São Paulo e Nova Iorque

As cidades de São Paulo e Nova Iorque são duas que possuem problemas de congestionamento amplamente conhecidos e, por este motivo, tais cidades foram escolhidas. Dois cenários sintéticos para regiões destas cidades foram gerados usando a ferramenta OSMWebWizard⁶. Esta ferramenta obtém dos dados de mapa e redes do Projeto OpenStreetMap (OpenStreetMap contributors, 2025) e é capaz de gerar demandas de tráfego aleatórias e com várias modalidades de transporte. A demanda segue uma distribuição de probabilidade que obedece a alguns parâmetros fornecidos pelo usuário, como *Through*

⁵<https://sumo.dlr.de/docs/Tutorials/OSMWebWizard.html>

⁶<https://sumo.dlr.de/docs/Tutorials/OSMWebWizard.html>

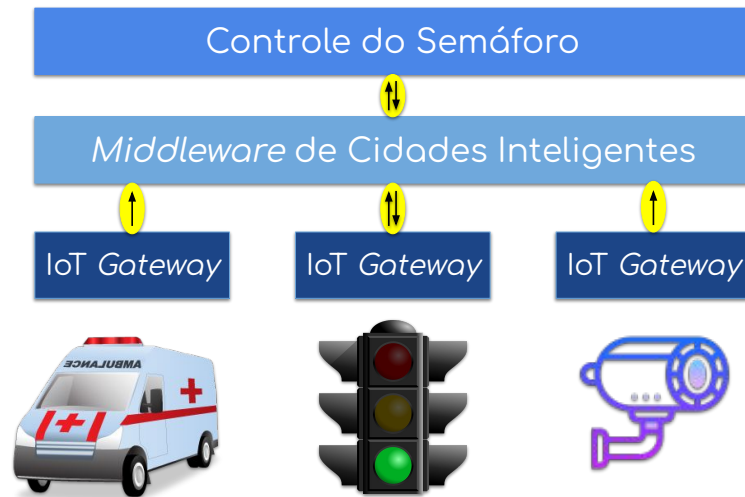


Figura 3.2: Sistema de Controle para Preempção de Semáforos durante o atendimento a uma emergência.

Traffic Factor, que indica a probabilidade de a aresta de partida de um veículo ser uma aresta próxima das bordas da rede e o parâmetro *Count*, que indica quantos veículos serão gerados por hora e por faixa/quilômetro. Os cenários gerados resultaram em 25.882 veículos para a cidade de São Paulo e 16.454 veículos para a cidade de Nova Iorque. Quatro subcenários de cada cidade foram criados, removendo veículos para simular diferentes níveis de congestionamento. A Tabela 3.1 apresenta o número de veículos para cada cenário.

Tabela 3.1: Número de veículos para os diferentes cenários.

(a) São Paulo

Cenário	Carros	Caminhões	Motocicletas	Total
1	3.494	1.167	2.355	7.016
2	5.826	2.322	4.694	12.842
3	8.136	3.509	5.855	17.500
4	10.457	4.656	7.041	22.154
5	12.258	5.624	8.000	25.882

(b) Nova Iorque

Cenário	Carros	Caminhões	Motocicletas	Total
1	2.128	706	1.431	4.265
2	3.544	1.412	2.833	7.789
3	5.009	2.151	3.639	10.799
4	6.422	2.847	4.353	13.622
5	7.830	3.579	5.045	16.454

Para cada execução da simulação, cada EV é sempre o mesmo, conforme dados da Tabela 3.2. As rotas dos veículos são apresentadas nas Figuras 3.3 e

3.4. O EV escolhido em cada cenário foi, dentre os carros, aquele que cruzava o maior número de semáforos. Essa escolha tem como objetivo verificar as situações em que o veículo ficava mais tempo na rede.

Tabela 3.2: Parâmetros das rotas dos EVs - cenários São Paulo e Nova Iorque - Sintético

EV	Distância	Semáforos
São Paulo	10.945,24m	65
Nova Iorque	8.887,04m	63



Figura 3.3: Rota do EV dentro da área do experimento do cenário de São Paulo - Sintético.



Figura 3.4: Rota do EV dentro da área do experimento do cenário de Nova Iorque - Sintético.

3.4.2 Cenário De Uma interseção

Em Kapusta et al. (2017), os autores criaram um cenário usando os dados presentes no trabalho de Vujić (2013), utilizando o simulador VISSIM, e se referem a uma área da cidade de Zagreb, Croácia. Como os dados não estavam disponíveis, geramos um conjunto de dados que se assemelha ao apresentado no trabalho. O nosso cenário é composto de uma rede contendo uma interseção com um semáforo ao centro e oito vias em formato de cruz (uma via de origem e uma via de destino em cada uma das posições *Norte*, *Sul*, *Leste* e *Oeste*).

Cada via de origem está ligada a três outras vias de destino. Portanto, um fluxo de veículos é definido para todos os pares de origem-destino possíveis (12 no total). Os veículos de cada fluxo são gerados com probabilidade de 15% a cada segundo. Esta configuração é suficiente para saturar a rede no momento da entrada do EV, mas sem criar um *backlog* de inserção de veículos (quando o simulador não consegue inserir os veículos aptos na rede por falta de espaço físico disponível na via). Dessa forma, no momento da partida do EV, ele é instantaneamente inserido na rede. Além do EV, são inseridos, em média, aproximadamente 3532 veículos em cada execução do algoritmo. Nesse cenário, o EV percorre seis quilômetros entre a origem e o destino. A Figura 3.5 apresenta a interseção deste cenário.

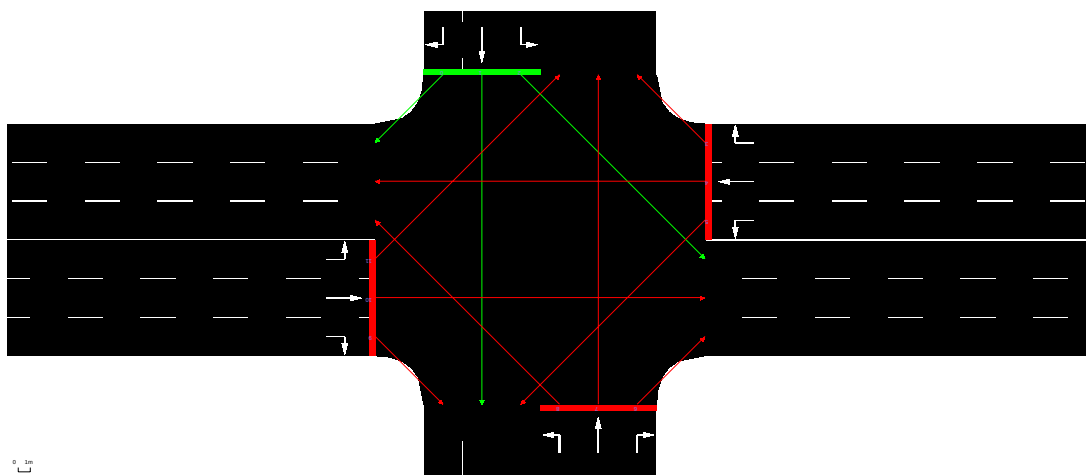


Figura 3.5: Cenário de uma interseção, baseado nas características do experimento descrito em Kapusta et al. (2017)

Este cenário tem o objetivo de avaliar o algoritmo proposto em (KAPUSTA et al., 2017) em condições parecidas com as que os autores fizeram, incluindo o número de veículos e saturação da rede. Esse cenário ainda é útil para modelar uma avenida de grande fluxo dos grandes centros metropolitanos, e como este cenário foi criado utilizando a ferramenta visual *netedit* do SUMO, ele pode ser usado para entender a representação dos elementos da rede neste simulador.

3.4.3 Cenários de Turim e Colônia

Estes cenários utilizam demandas reais e são derivados de experimentos presentes na literatura. São cenários expressivos, considerando as limitações de escala das simulações microscópicas. Para isto, foram utilizadas informações das cidades de Turim - Itália e Colônia - Alemanha, que são detalhadas nas Subseções a seguir. O objetivo original destes trabalhos não tem relação com o problema da redução do tempo perdido pelo EV no trânsito, mas sim com modelar o trânsito de maneira representativa em áreas de interesse para essas cidades. As informações desses cenários possibilitaram testar o problema do tempo perdido pelo EV nas cidades de Turim e Colônia, análise esta que não está presente nos trabalhos originais.

3.4.3.1 Cenário Turim

Este cenário foi adaptado de Rapelli, Casetti e Gagliardi (2022), uma área de 600 km^2 da região da cidade de Turim - Itália, com dados reais do trânsito para um período de 24 horas. Do conjunto de veículos, 3 EVs foram escolhidos. Não há nada especial nas rotas dos EVs, exceto pelo fato de não tomarem vias de baixa prioridade, para evitar que o EV espere nessas ruas indefinidamente, pois não é possível diferenciar o tempo perdido oriundo dos algoritmos de preempção do tempo perdido de espera em vias de baixa prioridade. A Figura 3.6 apresenta a área do experimento e as rotas dos EVs.

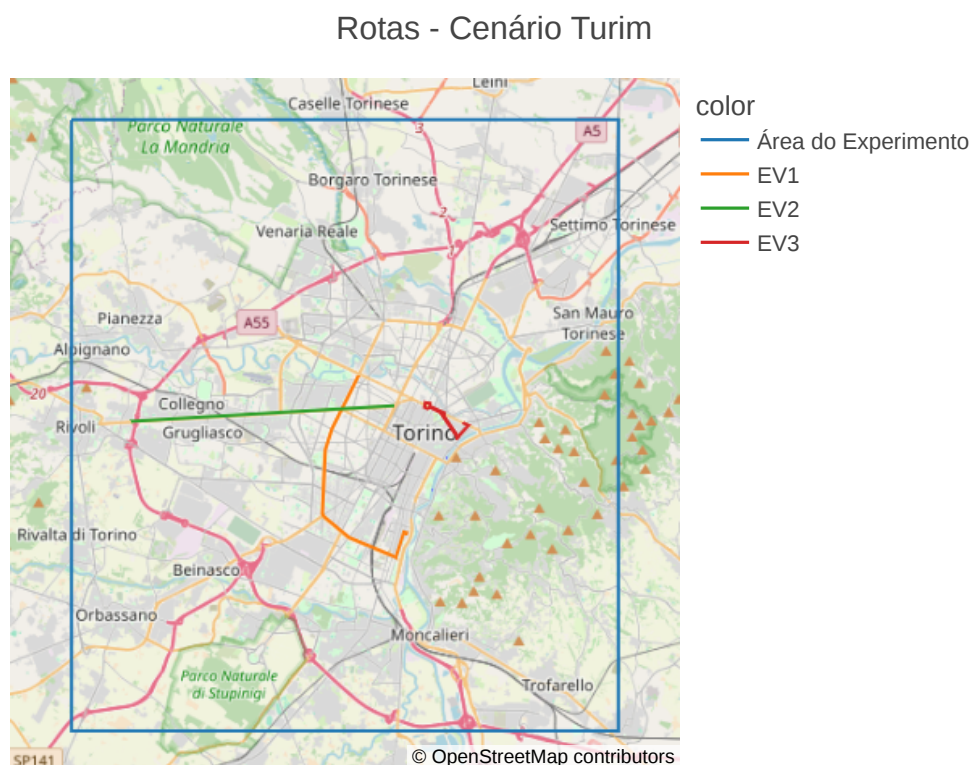


Figura 3.6: Rotas dos EVs dentro da área do experimento do Cenário Turim

O cenário original usa simulação mesoscópica, onde 2.202.814 viagens são inseridas na janela de 24 horas da simulação completa.

No nosso trabalho, utilizamos as informações do trânsito e as adaptamos para o modelo microscópico, pelo fato da simulação mesoscópica não ser adequada para a resolução do problema da redução do tempo perdido pelo EV no trânsito. Para isso, uma janela de tempo entre 6h e 8h da manhã para a inserção dos EVs na rede foi escolhida, o que resultou em 175.873 viagens oriundas do trabalho original.

A distância percorrida e a quantidade de semáforos cruzados por cada EV no Cenário Turim são apresentadas na Tabela 3.3.

Tabela 3.3: Parâmetros das rotas dos EVs - Cenário Turim

EV	Distância	Semáforos
EV1	9.961,68m	35
EV2	9.952,10m	19
EV3	5.300,60m	33

3.4.3.2 Cenário Colônia

Este cenário é baseado em Varschen e Wagner (2006), um registro de tráfego de 24 horas da cidade de Colônia - Alemanha. As considerações para este cenário são as mesmas que as da Subseção 3.4.3.1. O cenário original possui 1.549.612 viagens, e para a janela de 6h às 8h da manhã, o número de viagens é de 252.754.

Tanto no Cenário Turim quanto no Cenário Colônia, os mapas representam as rotas dos EV que foram escolhidas de maneira aleatória.

A distância percorrida e a quantidade de semáforos cruzados por cada EV no Cenário Colônia são apresentadas na Tabela 3.4.

Tabela 3.4: Parâmetros das rotas dos EVs - Cenário Colônia

EV	Distância	Semáforos
EV1	8.351,42m	26
EV2	5.044,34m	27
EV3	6.344,31m	24

3.4.4 Cenário do Centro Expandido de São Paulo - Metrô OD SP 2017

Esta Subseção apresenta contribuições de dois novos cenários com demandas reais: um abrangendo uma grande área, que pode ser usada em simulações mesoscópicas e macroscópicas, e outro com uma área reduzida da

Rotas - Cenário Colônia

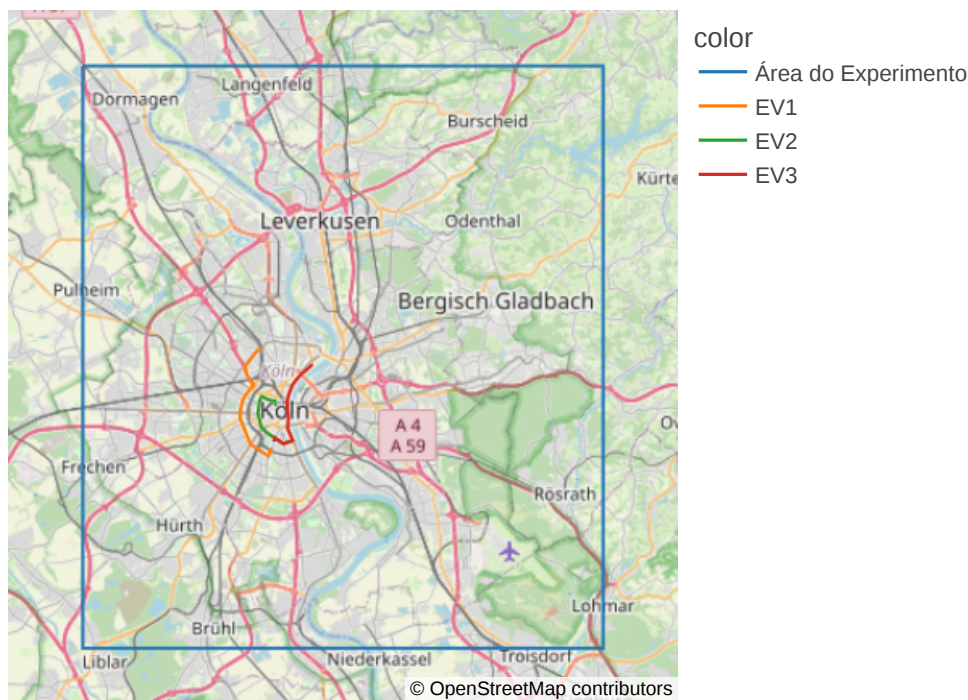


Figura 3.7: Rotas dos EVs dentro da área do experimento do Cenário Colônia

área maior, que pode ser executado em simuladores microscópicos, o que é o caso deste trabalho. O cenário maior foi gerado com base nos dados levantados pela CPTM (Companhia Metropolitana de São Paulo (METRÔ), 2017). A área de cobertura destes dados é a Região Metropolitana de São Paulo, com 7.946km^2 , que inclui 39 cidades compondo um total de 517 zonas de pesquisa (Figura 3.8). Uma viagem nesta pesquisa contém uma Zona de Origem, uma Zona de Destino, uma modalidade de transporte e um tempo de partida (dentro da janela de tempo de 24 horas). Os dados coletados são uma amostra de aproximadamente 90 mil viagens diárias, que correspondem a 28 milhões de viagens motorizadas (incluindo as compartilhadas, onde mais de uma pessoa que respondeu à pesquisa utiliza o mesmo veículo de transporte que outros passageiros).

Estes dados foram utilizados para gerar 13 milhões de viagens diárias, incluindo automóveis de passeio, motocicletas e ônibus (para as viagens compartilhadas, foi considerada uma média de 40 pessoas por veículo).

Executamos uma simulação para este cenário, que executou por aproximadamente 112 dias numa máquina com 32 processadores e consumiu 83GB dos 396GB de memória RAM disponível, sem finalizar. Ficou claro que o SUMO tem dificuldades em lidar com essa quantidade de dados em um intervalo de tempo razoável. Em função desse motivo, o experimento foi limitado à área conhecida como Centro Expandido de São Paulo, que abrange uma área de $189,60\text{km}^2$. Além disso, a janela de tempo utilizada nos outros dois cená-

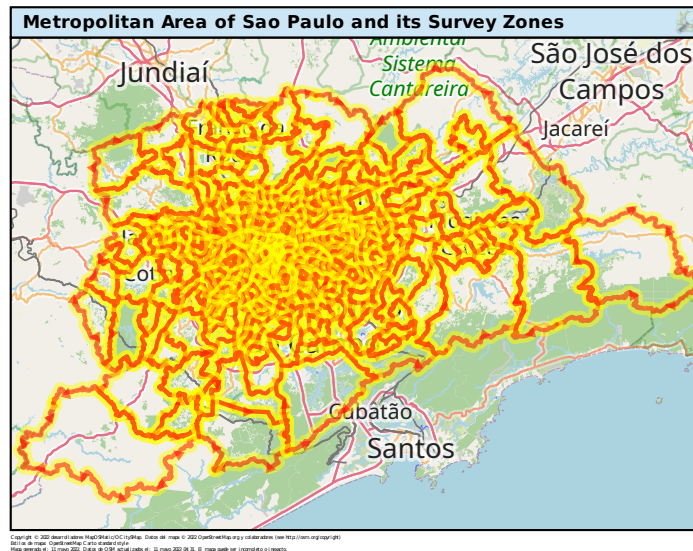


Figura 3.8: Área Metropolitana de São Paulo e suas zonas de pesquisa (obtido de <https://maposmatic.osm-baustelle.de/new/#>).

rios anteriores (6h às 8h) ainda é muito grande para ser tratada pelo SUMO. Nesta situação, utilizando simulação microscópica, temos aproximadamente 1 milhão de viagens nesta janela. Portanto, a janela de partida do EV, neste cenário específico, foi definida como das 6h45 às 7h15 da manhã. Como resultado, o cenário original de 13.215.558 viagens foi reduzido para uma versão limitada com 345.926 viagens. A demanda deste cenário ainda é maior do que a demanda dos cenários de Turim e Colônia, mesmo com uma janela de tempo menor. Estas restrições não prejudicam a qualidade da simulação, continuando a ser um cenário expressivo dentre os disponíveis para a execução de nossos experimentos.

Neste caso, a distância percorrida e a quantidade de semáforos cruzados por cada EV são apresentadas na Tabela 3.5. A Figura 3.9 apresenta a área do Centro Expandido usada no experimento, seus limites e as rotas dos EVs. Neste cenário, os EVs foram escolhidos e fixados aleatoriamente.

Tabela 3.5: Parâmetros das rotas dos EVs - cenário Centro Expandido de São Paulo

EV	Distância	Semáforos
EV1	12.140,19 m	55
EV2	6.914,45 m	15
EV3	9.922,56 m	23
EV4	8.602,20 m	43

Os detalhes e o passo a passo da criação destes cenários podem ser vistos no Apêndice A.

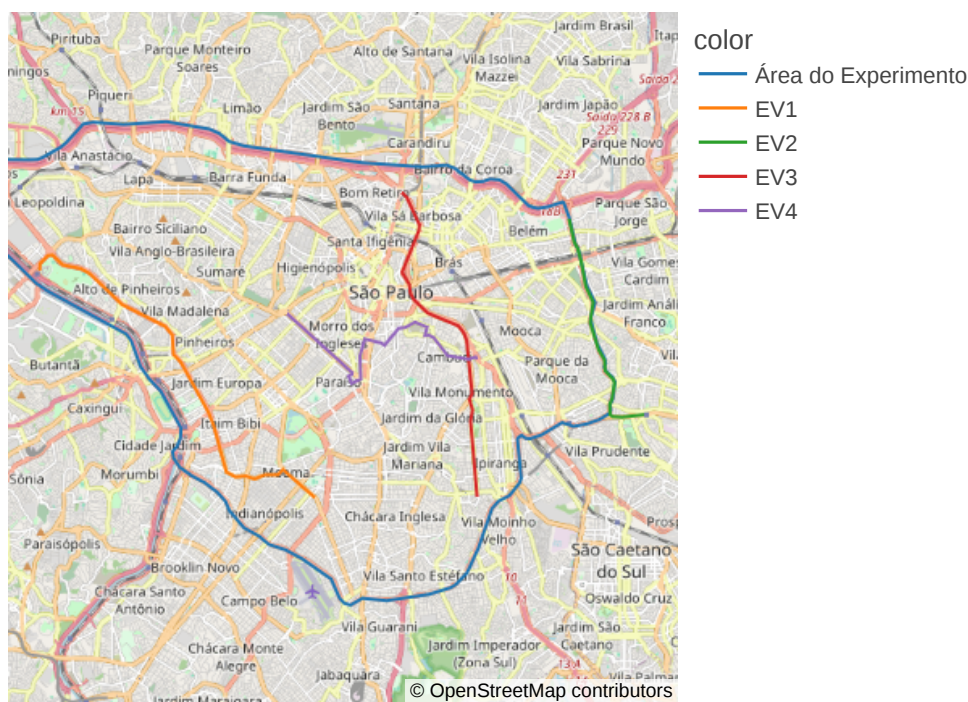


Figura 3.9: Rotas dos EVs dentro da área do experimento do cenário Centro Expandido de São Paulo (Companhia Metropolitana de São Paulo (METRÔ), 2017).

3.4.5 Considerações sobre a Simulação

Os cenários de Turim e Colônia possuem experimentos com 3 EVs que trafegam pelas vias com demandas reais. Durante a execução dos experimentos, notamos que havia algumas inconsistências nos resultados, que depois foram identificadas como problemas na rede do mapa e na simulação. Naquele momento, indagamos, dada a rede obtida dos trabalhos de Rapelli, Casetti e Gagliardi (2022) e Varschen e Wagner (2006), se o SUMO criaria demandas artificiais que não tratariam os problemas observados.

Por este motivo, foram geradas viagens sintéticas para fins comparativos, para confrontar com os dados da demanda real, com as arestas de origem e destino sendo escolhidas aleatoriamente. A versão sintética possui 180.001 viagens na janela de tempo de 6h às 8h da manhã, para o mapa de Turim. Da mesma forma, criamos uma versão sintética que possui 240.002 viagens para o mapa de Colônia.

Contudo, notamos que a hipótese de que os cenários artificiais não enfrentariam os problemas observados nos cenários reais não se confirmou; a execução dos cenários com dados sintéticos esbarrou nos mesmos problemas dos cenários originais.

Na análise dos resultados dos cenários sintéticos, observamos que as dificuldades encontradas eram, na verdade, devidas às limitações do SUMO.

Por exemplo, em interseções sem semáforos, na vida real, há uma espécie de cortesia entre os motoristas e o impasse acaba sendo resolvido. No SUMO, a interação entre os elementos só ocorre com veículos nas proximidades, com o intuito de preparar para o próximo passo de simulação. Nas situações em que a densidade do tráfego é muito alta, o veículo de emergência pode entrar em inanição ou ser teleportado. Dessa forma, os veículos não estão atentos à aproximação de veículos de emergência.

Assim, considerando todos os problemas observados acima e o tempo de execução do cenário do Centro Expandido de São Paulo, percebemos que o esforço que seria despendido para a criação de cenários sintéticos não resultaria em resultados significativos, e optamos por não o fazer. Em vez disso, para enrobustecer os resultados sem prejudicar o tempo de execução dos experimentos, preferimos selecionar mais um EV adicional, escolhido aleatoriamente.

3.5 Metodologia de Avaliação

Esta Seção apresenta a metodologia de avaliação e considerações sobre as avaliações dos experimentos. A metodologia de avaliação foi definida para avaliar não apenas eficiência, mas também a robustez e impacto da preempção no tráfego geral.

3.5.1 Framework de Simulação

As soluções devem ser capazes de alterar o comportamento dos elementos na simulação, por exemplo, solicitando a alteração do ciclo de um semáforo ou obter informações da rota de um EV. Cada simulador pode fornecer formas diferentes de se alcançar tais resultados, mas é comum que os simuladores forneçam uma API para isso. No caso do SUMO, o mesmo apresenta uma API para comunicação chamada *Traffic Control Interface* (TraCI)⁷. Essa API é extensa, e a programação direta dos algoritmos na API para algoritmos diferentes não dá uniformidade de tratamento para os diferentes algoritmos.

Com o intuito de evitar essas diferenças e com a motivação de padronizar as ações entre os experimentos, uma contribuição importante deste trabalho é a criação de um *framework* para garantir que os comandos fossem uniformes entre as soluções e, portanto, que os resultados fossem comparáveis.

A Figura 3.10 apresenta a arquitetura do *framework* usada neste trabalho. As soluções agora se comunicam apenas com o *framework*; os detalhes do simulador e de sua estrutura interna ficam encapsulados, o que simplifica sobremaneira a interface de comunicação do algoritmo com o simulador.

⁷<https://sumo.dlr.de/docs/TraCI.html>

Por exemplo, se as soluções quiserem alterar o ciclo de um semáforo por um número pré-determinado de ciclos, elas o fazem da mesma maneira.

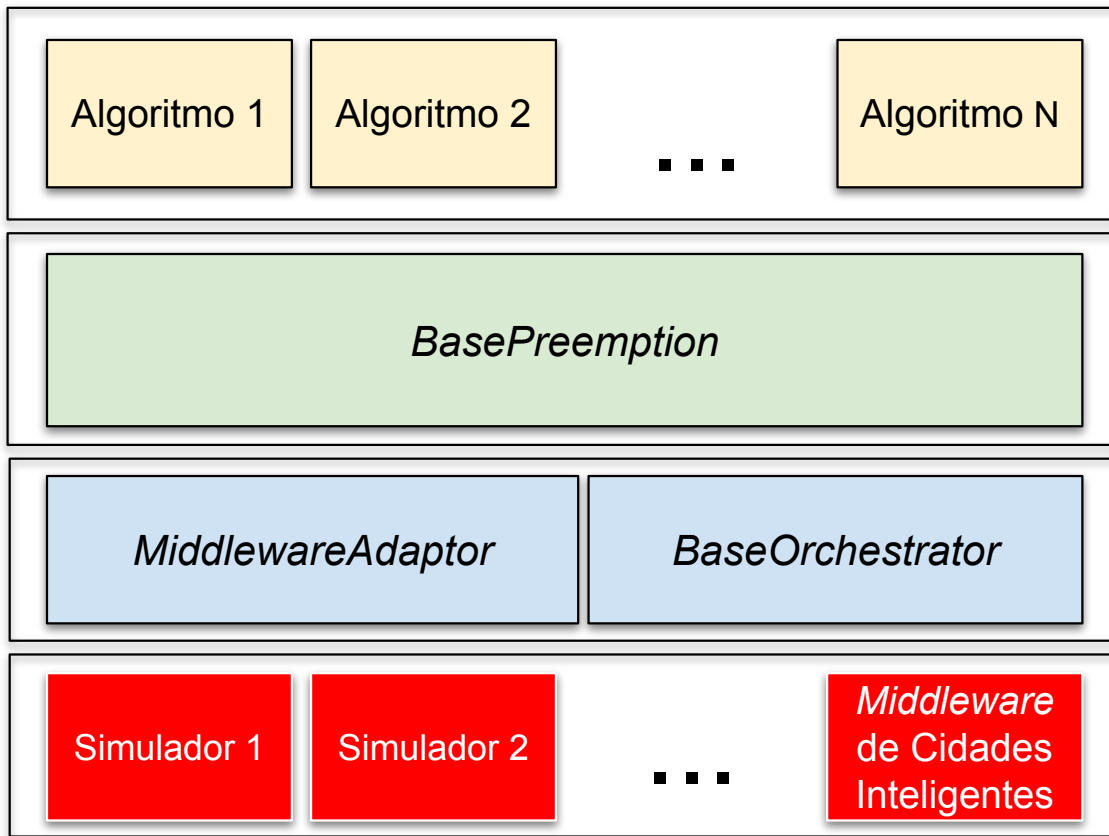


Figura 3.10: Framework desenvolvido para este trabalho.

Também projetamos o componente *Middleware Adaptor* para que este possa ser ajustado para utilizar outros simuladores, além do SUMO, inclusive *middlewares* de cidades inteligentes. Além disso, criamos o componente *BaseOrchestrator* para implementar as políticas necessárias quando mais de um EV está presente na rede, conciliando pedidos de preempções concorrentes.

Neste trabalho, projetamos soluções para controlar os trajetos dos EVs, um de cada vez, e, nesse caso, o *orquestrador* se preocupa apenas em executar a preempção sem validar conflitos de ordens, sempre executando a última ordem recebida.

O *Framework*, *Middleware Adaptor* e o *BaseOrchestrator* foram escritos em *Python*, bem como as implementações das soluções aqui apresentadas.

Para o *Framework*, implementamos uma classe base chamada *PreemptionStrategy*, da qual os Algoritmos de preempção devem herdar, para assim uniformizarem suas requisições à API do SUMO. Esta classe fornece métodos, *helpers* e *placeholders* para auxiliar as classes especialistas a executarem suas tarefas (ver Tabela 3.6). Tais classes podem ou não sobrescrever os métodos fornecidos; por exemplo, o Algoritmo 3.4 (Onda Verde), visto anteriormente, sobrescreve apenas dois métodos: *configure* e *execute_step*.

Tabela 3.6: Métodos da classe base *PreemptionStrategy*

Método	Descrição
setup	define as variáveis externas à classe
configure	configura o estado interno da classe
execute_before_step_simulation	ação executada antes do passo de simulação
get_colors	retorna as cores da fase atual e da fase alvo do semáforo
open_adj_tls	abre os semáforos adjacentes a um semáforo específico
clear_adj_tls	restaura os semáforos adjacentes a um semáforo específico
open_tl_at_time_by_cycles	abre o semáforo no tempo futuro determinado pela quantidade de ciclos especificada
open_tl_at_time_by_cycles_by_time	abre o semáforo no tempo futuro determinado pela quantidade de ciclos e duração especificada
execute_step	executado no passo de simulação
finish	informa que a emergência foi finalizada, restaurando os estados dos objetos
ev_is_in_simulation	informa se o EV está presente na simulação
instance_name	retorna o nome do Algoritmo e seus parâmetros

3.5.2 Detalhes de Implementação

Esta subseção descreve os detalhes de implementação omitidos na descrição dos algoritmos. Nem todos os trabalhos que serão usados na comparação explicitam todos os parâmetros necessários e, para os omitidos, usamos aproximações baseadas em dados compatíveis com os descritos nos artigos e, para os casos em que tais dados não estavam disponíveis, executamos experimentos empíricos, a fim de se obter o melhor conjunto de parâmetros para cada caso. O intuito é comparar o melhor resultado de cada uma das soluções selecionadas com a solução proposta neste trabalho.

3.5.2.1 RFID

Os parâmetros `dd` (distância de detecção, em metros) e `nc` (número de ciclos, sem escala) são parâmetros fixos do Algoritmo RFID (3.2.1), isto é, eles não são alterados durante a viagem do EV. Considerando que este Algoritmo é sensível apenas a informações de contexto local da interseção, os parâmetros `dd=100` foram fixados de forma a ficarem consistentes com os sensores disponíveis no mercado (IMPINJ, 2025). Já para o parâmetro `nc`, é importante ressaltar que esse parâmetro indica por quantos ciclos o semáforo permanece verde durante o início da preempção. Quanto maior este valor, mais tempo o algoritmo tolera aguardar que o EV cruze pelo semáforo. Assim, valores grandes demais indicam desperdício de tempo para os veículos que estão parados

no semáforo com a luz vermelha. Executamos alguns experimentos preliminares utilizando valores na faixa de 1 a 25, e o valor 5 foi um dos valores competitivos quando comparado aos outros valores testados.

3.5.2.2 Fuzzy

O Algoritmo *Fuzzy* utiliza o parâmetro e_l (nível de emergência), que pode assumir os valores *low*, *medium* e *high*, que indicam baixa, média e alta urgência no atendimento da emergência. No caso do parâmetro *high*, este algoritmo habilita mais funcionalidades para facilitar a passagem do EV; portanto, este foi o valor utilizado neste algoritmo.

3.5.2.3 Algoritmo Filas

O Algoritmo Filas (3.2.3) menciona os parâmetros T_{Alpha} e T_{Beta} (número de ponto flutuante), e em conjunto com os valores $ciclo_{duracao}$ e $chegada_{EV}$ (segundos), são usados para analisar quando o algoritmo é iniciado ($T_{Beta} * ciclo_{duracao} \leq chegada_{EV} \leq T_{Alpha} * ciclo_{duracao}$). No trabalho original, os autores fixaram os valores em $T_{Alpha} = 3$, $T_{Beta} = 0.5$ e $ciclo_{duracao} = 90$, e esses valores foram mantidos neste trabalho. Além disso, os autores definiram os parâmetros s_{Alpha} e s_{Beta} (metros) para analisar o tamanho da fila Q e iniciar uma fase de alternância de luzes verdes e vermelhas no semáforo. Assim, se $Q > s_{Alpha}$, o tempo de luz verde é incrementado e se $Q > s_{Beta}$, o tempo de luz vermelha é reduzido. Os autores definiram $s_{Alpha} = 10$ e $s_{Beta} = 15$, e esses valores também foram mantidos neste trabalho. Contudo, o incremento ou decremento do tempo das luzes vermelha e verde não é descrito no trabalho. Na nossa implementação, utilizamos a taxa de 10% para redução ou aumento do ciclo atual que consideramos adequada para a análise. Isso porque, com tal valor, rapidamente o algoritmo convergia para a situação onde a luz verde era acionada e permanecia assim até que o EV cruzasse a interseção.

3.5.2.4 Algoritmo Onda Verde

No Algoritmo Onda Verde (3.2.4), todos os semáforos são mantidos com a luz verde até que o EV percorra todo o trajeto e, nesse caso, não há necessidade de fixar parâmetros de preempção nem os ciclos dos semáforos.

3.5.3 Roteamento dos Veículos na Simulação

Para cada execução da simulação, a rota do EV é sempre pré-definida, não sendo alterada durante a execução e pode ser observada nas Figuras dos cenários sintéticos e reais apresentados nas Seções 3.4.1, 3.4.2, 3.4.3 e 3.4.4. Já para os outros veículos, suas rotas são definidas dinamicamente,

ou seja, antes do veículo entrar na rede, são conhecidos apenas os pontos de origem e destino, e assim que eles ingressam na rede, a rota é definida baseada nos algoritmos de roteamento do próprio simulador. Caso algum dos outros veículos fique parado por mais de 300 segundos (especialmente nos casos de *deadlock*), o simulador remove esses veículos da rede, gerando um evento denominado teletransporte (*teleport*)⁸. O EV nunca é removido da simulação, mesmo que fique parado por mais de 300 segundos.

O SUMO contempla uma heurística que tenta prevenir veículos de entrar em uma interseção se ela já está congestionada, evitando que este veículo bloqueie as vias dessa forma. Contudo, essa heurística nem sempre é bem-sucedida, e o comportamento padrão do simulador é teletransportar o veículo depois de algum tempo. Para evitar esse comportamento, o parâmetro *-ignore-junction-blocker* foi adicionado, onde, depois de um determinado tempo, os outros veículos “ignoram” o veículo que está bloqueando o trânsito. A documentação do simulador informa que esse parâmetro pode modelar o comportamento da vida real, já que os motoristas podem encontrar maneiras de contornar o veículo infrator. Em nossas simulações, o valor definido para esse parâmetro é de 50 segundos.

3.5.4 Métricas de desempenho

Esta Subseção apresenta métricas de desempenho e métricas auxiliares que foram utilizadas para o entendimento e comparação das situações e algoritmos avaliados. Para comparar o desempenho dos algoritmos, primeiro é definido o *Melhor Tempo de Viagem* (btt, em segundos) como o tempo de viagem do EV quando ele trafega utilizando a velocidade máxima possível e não para em nenhum semáforo ou interseção. Definimos também o *Tempo Total de Viagem* (ttt, em segundos) como sendo o tempo de viagem efetivamente aferido. A seguir, descreveremos as métricas utilizadas para a avaliação dos desempenhos dos algoritmos neste trabalho.

A primeira métrica que foi utilizada é o *Tempo Perdido*, sendo definida conforme a Equação 3.4 (s). Quanto menor, melhor. Os resultados dessa métrica são comparados com o tempo perdido da versão sem preempção que é considerado o *baseline* para avaliação dos algoritmos. Essa métrica é importante porque, quanto menor o tempo perdido pelo EV na versão *baseline*, menor é a possibilidade de melhoria alcançada pelos algoritmos de preempção. Essa

⁸Há três razões para que um veículo seja teletransportado no SUMO: **jam** - o veículo está preso em uma via de alta prioridade e não há espaço no próximo trecho - **yield** - o veículo está preso em uma via de baixa prioridade e não encontrou um espaço no tráfego priorizado - e **wrong lane** - o veículo está preso em uma faixa que não possui conexões para a próxima aresta de sua rota. A alteração efetuada no código previne os primeiros dois casos, mas o terceiro ainda pode ocorrer.

métrica também é importante na comparação de rotas de tamanhos variados e com diferentes números de semáforos no percurso dos EVs.

$$tl = ttt - btt \quad (3.4)$$

O entendimento da relação entre as métricas tl , btt e ttt é fundamental para compreender por que a maioria dos trabalhos na literatura falha ao utilizar métricas inadequadas. Essa relação é ilustrada na Figura 3.11.

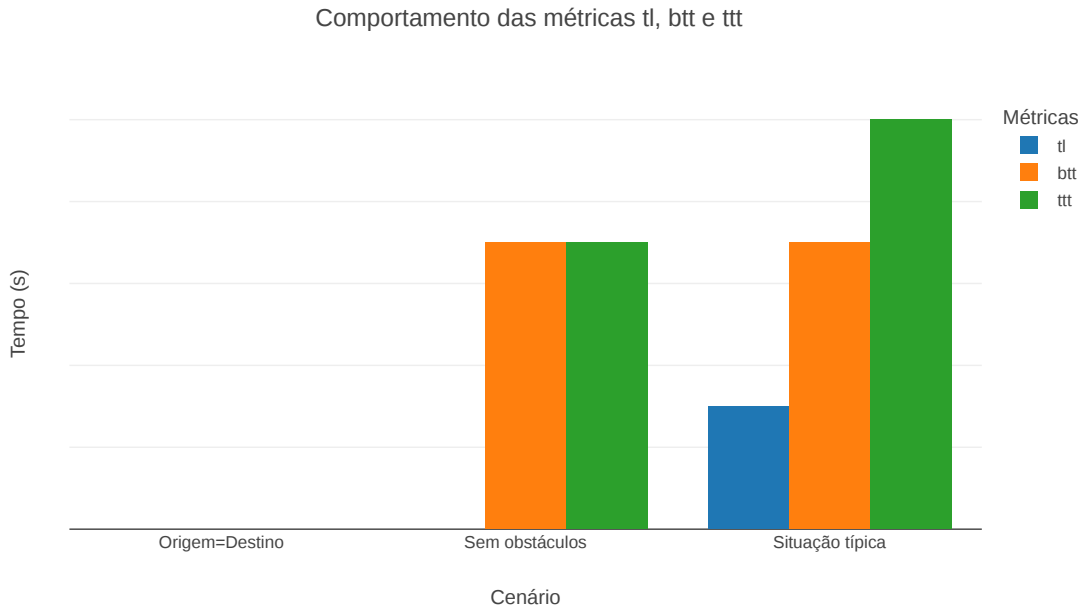


Figura 3.11: Relação entre as métricas tl , btt e ttt

A referida figura não apresenta valores no eixo ordenado (y) propositalmente, uma vez que o objetivo é demonstrar a proporcionalidade entre as métricas, independentemente de seus valores absolutos. Nela, observam-se três padrões distintos de comportamento:

- $tl = btt = ttt = 0$: Este cenário ocorre quando os pontos de origem e destino coincidem (ausência de deslocamento) ou na hipótese teórica de que o EV seja deslocado instantaneamente para o destino, mesmo quando $origem \neq destino$. Trata-se, portanto, de um "teleporte", padrão que pode ser descartado em análises de deslocamentos veiculares reais contemporâneos.
- $tl = 0$ e $btt = ttt$: Indica que o EV não encontrou obstáculos e trafegou sob condições ideais. Neste caso, a aplicação de algoritmos de preempção de semáforos oferece ganho marginal ou nulo, visto que o tempo de viagem já é o mínimo possível.

- $tl \neq 0$ (Cenário Realista): Neste padrão, o ttt é composto pela soma de btt e tl . O potencial de otimização de um algoritmo de preempção reside justamente na redução do tl , especialmente em cenários nos quais o tempo perdido representa a parcela preponderante do tempo total de viagem (ttt).

Ressalta-se que a literatura costuma reportar apenas a média do ttt dos experimentos. Essa abordagem é insuficiente, pois omite o fator de participação do tl na composição do ttt . Consequentemente, torna-se inviável dimensionar com precisão a eficácia do algoritmo proposto, dificultando a comparação entre rotas distintas ou cenários com números variados de intersecções.

Prosseguindo na apresentação das métricas de avaliação, a segunda métrica apresenta o Tempo Perdido sobre o Tempo Total de Viagem da versão sem preempção, definido conforme a Equação 3.5 (%). Quanto menor, melhor. É uma métrica que indica a retenção do trânsito na rota do EV, para a versão sem preempção. Para algumas situações, essa métrica supera 90%, indicando que a maior parte do tempo que o EV permanece na simulação, na realidade, sua velocidade média está muito aquém do ideal.

$$ptl = \frac{tl}{ttt} \quad (3.5)$$

A terceira métrica é a Melhoria de Tempo perdido (tl_{IMP}), na escala de quantas vezes o algoritmo objeto de análise é melhor (ou pior) que a versão sem preempção (sem escala). Para isto, é necessário obter o Tempo Perdido do Algoritmo tl_{ALG} (s) comparado ao Tempo Perdido da Versão sem Preempção tl_{SP} (s). Essa métrica é definida na Equação 3.6. Quanto maior, melhor. É uma métrica útil para se observar a diferença entre as melhorias obtidas em função do algoritmo.

$$tl_{IMP} = \begin{cases} \frac{tl_{ALG}}{tl_{SP}}, & \text{se } tl_{SP} \geq tl_{ALG} \\ -\frac{tl_{ALG}}{tl_{SP}} & \text{caso contrário} \end{cases} \quad (3.6)$$

A quarta métrica apresenta o Percentual de Melhoria p_{IMP} (%), computado em relação ao tempo perdido (tl_{ALG} e tl_{SP}), definido pela Equação 3.7. Quanto maior, melhor. Essa métrica possui o inconveniente de apresentar um limite superior de 100%, enquanto pode alcançar valores negativos indefinidamente. Por este motivo, é útil analisar essa métrica e a anterior em conjunto. Alguns trabalhos calculam o percentual de melhoria utilizando apenas o tempo total de viagem (segunda métrica). Contudo, o tempo total de viagem é uma composição do melhor tempo de viagem e do tempo perdido ($ttt = tl + btt$). Isso significa que esse valor pode esconder a melhoria efetiva do algoritmo, já que o que importa é saber a redução efetiva do tempo perdido. Por este motivo,

esta quarta métrica foi proposta neste trabalho, com o objetivo de diminuir o efeito de outras variáveis na comparação do desempenho dos algoritmos.

$$p_{IMP} = 100 \times \begin{cases} 1 - \frac{tl_{ALG}}{tl_{SP}}, & \text{se } tl_{ALG} < tl_{SP} \\ -1 \times \left(\frac{tl_{ALG}}{tl_{SP}} - 1 \right) & \text{caso contrário} \end{cases} \quad (3.7)$$

A quinta métrica apresenta o Tempo de Preempção Médio (tpm (s)), ou seja, quanto tempo os semáforos ficaram com a preempção ativa enquanto o EV estava na simulação, em média (s). Quanto menor, melhor. A análise dessa métrica deve ser efetuada em conjunto com as demais métricas, para analisar o *trade-off* da melhoria alcançada com relação ao tempo médio que a preempção ficou ativa para obter tal melhoria, isso porque a preempção, como efeito adverso, prejudica os veículos do entorno.

Os resultados da simulação são apresentados na próxima seção e foram obtidos por meio de 25 execuções independentes. Com esse número de execuções, considerando a distribuição desconhecida dos dados, o método de Thompson (THOMPSON, 1936) aponta que os resultados possuem um nível de confiança de 95% para os dados entre os percentis 15° e 85°. Os gráficos são apresentados em formato *boxplot* e são gerados em *Python* com a biblioteca *Plotly*, obedecendo à seguinte regra: a mediana representa o percentil 50°, e os percentis Q1 e Q3 são obtidos por meio do método da interpolação do ponto médio para os percentis 25° e 75°, respectivamente. A variação interquartil (IQR) é obtida como $IQR = Q3 - Q1$. O (*lower fence*) é calculado observando $Q1 - 1,5I * IQR$, enquanto para o (*upper fence*) o cálculo correspondente é $Q3 + 1,5 * IQR$. O (*lower fence*) é, portanto, o menor valor da amostra que seja maior ou igual a $Q1 - 1,5 * IQR$. Valores da amostra inferiores ao (*lower fence*) são considerados e apresentados como *outliers*. O mesmo ocorre para o (*upper fence*): o maior valor da amostra que seja menor ou igual a $Q3 + 1,5 * IQR$ é definido como o (*upper fence*). Valores da amostra maiores que o (*upper fence*) são igualmente considerados e apresentados como *outliers*.

No fim deste processo, o resultado é um conjunto de valores: menor *outlier* (se existir), (*lower fence*), Q1, mediana, Q3, (*upper fence*) e maior *outlier* (se existir). Os resultados são apresentados obedecendo a essa estrutura.

3.5.5 Recursos Computacionais

Neste trabalho, descrevemos os resultados obtidos no cálculo das métricas descritas acima utilizando o simulador SUMO, inclusive os tempos totais de execução dos experimentos obtidos na simulação. Os códigos-fonte utilizados nos experimentos estão disponíveis no Github⁹. Devido à alta demanda de

⁹<https://gitlab.com/rodrigo.g.branco/docker-sumo-intercity-spres-ev>

recursos computacionais, utilizamos três equipamentos diferentes, com componentes distintos de software e hardware, para que os experimentos fossem executados em um tempo adequado. Considerando os altos tempos de execução que, em alguns casos, levaram semanas, não foi possível garantir que os experimentos aqui descritos foram executados de modo exclusivo nos equipamentos. De qualquer forma, como um mesmo experimento foi executado várias vezes, não observamos diferenças expressivas nos tempos de execução. As tarefas foram distribuídas de acordo com os recursos de processador e memória disponíveis no início de sua execução. Os recursos computacionais utilizados estão descritos na Tabela 3.7.

Tabela 3.7: Recursos computacionais usados na execução dos experimentos deste trabalho

ID	SO	Processador	#Cores	#Threads	Cache	Memória	Python
Jau	Debian 8.11 (jessie)	2x Xeon E5-2630 v3 @ 2.40GHz	2x 8	2x 16	20MB	396G	3.5.3
Ranger01	Ubuntu 18.04.5 LTS	2x Xeon Silver 4114 @ 2.20GHz	2x 10	2x 20	14MB	131GB	3.6.9
Ranger02	Ubuntu 18.04.5 LTS	2x Xeon Silver 4114 @ 2.20GHz	2x 10	2x 20	14MB	131GB	3.6.9
Ranger03	Ubuntu 18.04.2 LTS	2x Xeon Silver 4114 @ 2.20GHz	2x 10	2x 20	14MB	131GB	3.6.9

Em análises preliminares, observamos que o tempo de execução em si dos algoritmos não é tão impactante quanto o processamento da interação dos elementos da rede (veículos e arestas/vias). Quanto mais tempo o EV passa na rede e quanto maior o tamanho do cenário, mais tempo de execução é dispendido e mais memória é utilizada.

3.5.6 Adaptações no Simulador

O comportamento padrão do SUMO é que, quando um veículo não se move, seu tempo de espera é incrementado. Esse tempo de espera é usado para verificar se um veículo aguardou muito e, neste caso, ele é teletransportado. Contudo, é necessário que o EV chegue ao destino, não importa o que aconteça ou quanto tempo demore, ou quanto tempo ele fique parado no mesmo lugar. Se o EV for removido da simulação antes de terminar sua rota, as estatísticas dessa execução não podem ser consideradas. Por este motivo, uma pequena alteração foi feita no código-fonte do sistema, para que o EV não seja removido da rede no caso de ele ficar parado muito tempo no mesmo lugar.

3.6 Conclusão

Este Capítulo apresentou os algoritmos e estabeleceu as métricas, cenários e ferramentas de simulação para que as soluções possam ser comparadas. O próximo capítulo apresenta a primeira solução apresentada neste trabalho, um algoritmo de controle que usa uma TPN modificada que trata eventos externos.

Primeira Solução - TPN'

Este capítulo apresenta a primeira solução proposta neste trabalho. Inicialmente, descrevemos a fundamentação teórica baseada em Redes de Petri Temporizadas (Seção 4.1). Em seguida, detalhamos o modelo da solução (Seção 4.2), o modelo de controle (Seção 4.3) e o algoritmo desenvolvido para sua implementação (Seção 4.4). Por fim, apresentamos os resultados obtidos nos experimentos (Seção 4.5) e as conclusões pertinentes a esta abordagem (Seção 4.6).

4.1 Rede Petri Temporizada

Uma Rede Petri – *Petri Net* (PN) – é um grafo dirigido com três tipos de componentes: *posições (places)*, *transições (transitions)* e *arestas dirigidas (directed edges)* que conectam as posições às transições e as transições às posições. O *token* é um elemento que é movido entre as transições e posições da PN quando o estado da rede é alterado, sendo o estado definido pelo posicionamento de tokens em lugares na rede. Uma transição pode ser *disparada* quando esta transição é *habilitada*, o que ocorre quando as posições de entrada possuem um número suficiente de *tokens*. Neste caso, os *tokens* são movimentados das posições de entrada, e os *tokens* resultantes são colocados em cada uma das posições de saída da transição. As PNs são comumente utilizadas para modelar comportamento concorrente desde os anos 60 (MURATA, 1989).

O algoritmo proposto reage a eventos temporizados, que são os momentos futuros em que os semáforos ficarão verdes para que o EV possa progredir. Como uma PN regular não provê mecanismos para lidar com eventos temporizados em suas transições, utilizamos neste trabalho uma Rede Petri Tempo-

rizada – TPN (MERLIN; FARBER, 1976) – para modelar e analisar nossos algoritmos. Uma TPN é formalmente definida pela tupla (P, T, F, I, D, w, M_0) , onde:

- $P = \{P_0, P_1, \dots, P_m\}$ é um conjunto finito de posições;
- $T = T^{Imm} \cup T^{Timed} = \{T_0, T_1, \dots, T_n\}$ é a união dos conjuntos finitos e disjuntos de transições *imediatas* (T^{Imm}) e *temporizadas* (T^{Timed});
- $F \subseteq (P \times T) \cup (T \times P)$ é um conjunto de arestas dirigidas que partem de posições para transições ou de transições para posições, representando as funções de entrada e saída, respectivamente;
- $I \subseteq (P \times T)$, $I \cap F = \emptyset$ é o conjunto de *arestas inibidoras*, isto é, arestas que previnem o disparo de uma transição quando um *token* está presente em uma das posições de entrada;
- $D : T^{Timed} \rightarrow \mathbb{R}^+$ é o tempo de disparo das transições temporizadas;
- $w : F \rightarrow \{1\}$ é a função de peso que move um *token* entre posições e transições (no modelo utilizado neste trabalho, somente um *token* é movido em cada aresta); e
- $M_0 \subseteq P$ é o conjunto de posições iniciais, ou seja, posições que contêm *tokens* quando a TPN é iniciada.

Usualmente em TPNs é comum encontrar transições cujos tempos de disparo sejam aleatórios obedecendo a uma *taxa de disparo que segue uma distribuição exponencial* (Λ). Contudo, como neste trabalho tais transições não são utilizadas, por simplicidade, optamos por omitir o formalismo relacionado a este componente.

Entre as muitas possíveis *marcas (estados)* M da TPN, M_0 é uma delas e representa o conjunto de posições iniciais da TPN. Uma marca associa um inteiro não negativo a uma posição, representando quantos *tokens* existem naquela posição no momento da análise. As marcas podem ser representadas como m – *vetores*, associando cada $P_x \in P, 0 \leq x \leq m$ a uma posição em um vetor com m posições.

Contudo, as TPN tradicionais, suas extensões e ferramentas gráficas de trabalho não fornecem apoio para eventos externos, como o evento de cancelamento e cruzamento do EV pelo semáforo, eventos esses essenciais que devem ser tratados pelo modelo que propomos neste trabalho.

Os dois eventos externos que não podem ser modelados usando uma TPN tradicional são o cruzamento do EV e o comando de cancelamento. Isso porque essas transições não possuem nenhuma posição de entrada, e pelo formalismo adotado, essas transições são consideradas imediatas, significando

dizer que tais transições podem disparar a qualquer momento e em qualquer ordem. Obviamente, esse não é o comportamento esperado, pois para estas transições específicas, o disparo delas é condicionado a eventos externos. Por este motivo, estendemos a TPN para que possamos diferenciar entre as transições acionadas por eventos externos e as transições imediatas normais. Visualmente, não há diferenças entre essas transições de eventos externos e as outras transições imediatas dos modelos apresentados a seguir. Contudo, como essas transições possuem comportamento diferenciado nas implementações e só são disparadas quando tais eventos ocorrem, definimos essa nova TPN como TPN', e ela será tratada dessa forma nas próximas seções.

4.2 Modelo da Solução

Para esta primeira solução, um *Middleware* para cidades inteligentes recebe informações sobre localização, velocidade e aceleração do EV, bem como volume de tráfego das ruas e estado e localização dos semáforos ao longo da rota do EV. Os cenários da Subseção 3.4.1 são usados para esta avaliação. O algoritmo proposto usa a preempção para alterar a temporização dos semáforos, garantindo assim luzes verdes para o EV quando este alcança as interseções com semáforos ao longo das rotas. Como apresentado na Seção 2.7, uma consideração em qualquer solução de controle de tráfego é a segurança de todos os veículos presentes na rota e nas proximidades do EV, de forma que uma preempção não faça alterações bruscas nas fases dos semáforos, de forma a não fechar o semáforo imediatamente se ele estiver verde e sem passar pelo ciclo amarelo, por exemplo. Não só isso, deve evitar que ordens maliciosas sejam enviadas ao sistema e ainda garantir que os outros veículos não sofram *starvation* por esperar demais a preempção. Portanto, a solução precisa considerar quando eventos inesperados acontecem e a preempção deve ser cancelada, mantendo o estado da solução consistente.

A solução é modelada utilizando uma TPN (*Timed Petri Net* - Rede Petri Temporizada) modificada, definida a seguir na Seção 4.1, garantindo que o algoritmo de preempção não atinja configurações conflitantes e concorrentes para os semáforos, mantendo a consistência do estado da solução.

A robustez deste formalismo matemático é corroborada por sua aplicação bem-sucedida em cenários de segurança crítica (*safety-critical*), onde a confiabilidade é mandatória. Estudos recentes demonstram a eficácia das Redes de Petri tanto na validação de sistemas de *cockpits* interativos tolerantes a falhas na aviação comercial (FAYOLLAS et al., 2012) quanto na análise de confiabilidade de instrumentação digital e controle em usinas nucleares (YAN; DUNNETT; ANDREWS, 2023). Tais aplicações evidenciam a capacidade dessa mo-

delagem em gerenciar concorrência e prevenir estados inválidos, justificando sua adoção para assegurar a integridade do controle de tráfego em situações de emergência.

Usar uma TPN especialmente desenvolvida para este fim é uma das formas de se atender a muitas das características desejáveis das soluções apresentadas na Seção 2.7. A TPN construída cobre completamente a rota do EV enquanto outros trabalhos modelam uma ou poucas interseções (HUANG; WENG; ZHOU, 2015; KAPUSTA et al., 2017), sem garantir a finalização correta da solução. A Figura 4.1 apresenta a TPN para a rota completa, onde cada TL_i representa todos os semáforos da interseção i . Cada bloco de controle recebe notificações do *Middleware* e, possivelmente, sensores conectados para detectar a ocupação das vias e, podendo determinar se o EV cruzou ou não a interseção. O modelo da TPN para os blocos de controle é apresentado na Seção 4.1.

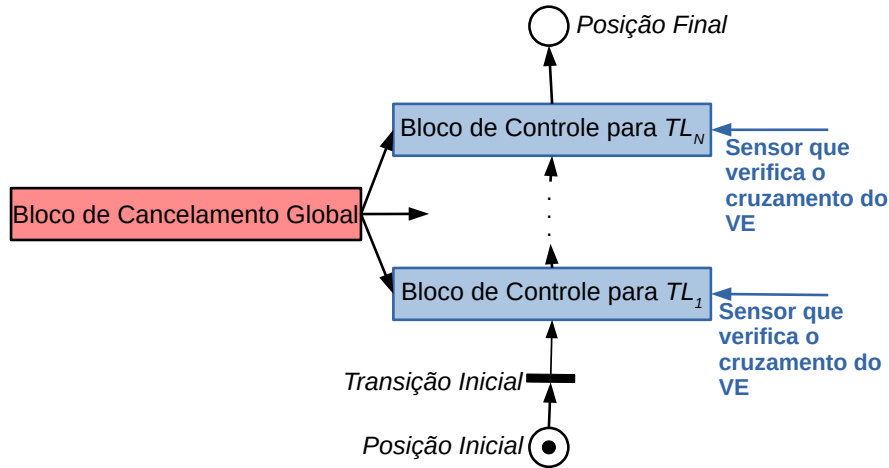


Figura 4.1: Blocos de Controle dos semáforos $TL_1 \dots TL_N$ para as interseções na rota do EV.

4.3 Modelo de Controle

Para modelar cada um dos cruzamentos da rota do EV, utilizamos uma TPN. A Figura 4.1 apresenta a visão global dos blocos de controle para o modelo proposto. Cada TL_i representa um bloco de controle para o semáforo de um cruzamento, representando uma expansão de uma das caixas azuis da Figura 4.2. A *transição inicial* na Figura 4.1 coloca um *token* na posição P_0 do bloco de controle TL_1 no início da emergência, e a transição T_1 do bloco TL_N coloca um *token* na *posição final*. As posições P_3 e P_6 (destacadas na Figura 4.2) são associadas a ações no modelo utilizado. A posição P_3 inicia a preempção de TL_i (isto é, garante a luz verde para o EV para a interseção

i) e a posição P_6 restaura TL_i para sua operação prevista (isto é, desabilita a preempção). Em caso de mudança de rota do EV, o comando de cancelamento deve ser acionado e, dessa forma, a TPN' é destruída e outra que considera a nova rota é criada.

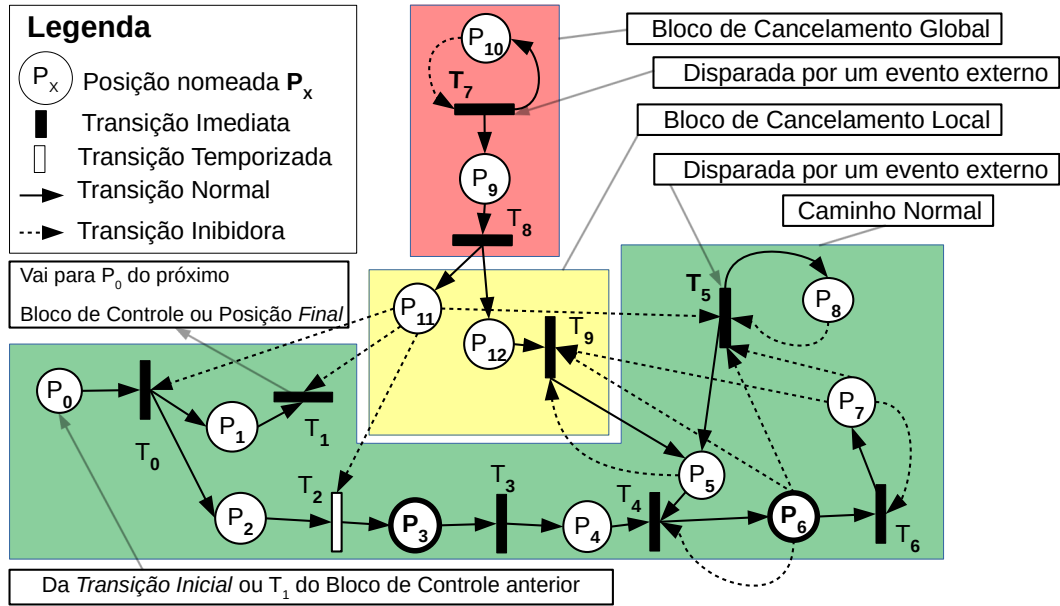


Figura 4.2: Modelo TPN' proposto. Essa figura representa uma expansão de cada um dos blocos de controle da Figura 4.1. A região vermelha é compartilhada por todos os blocos de controle de todos os TL_i , e as regiões amarelas e verdes são replicadas para cada TL_i ($1 \leq i \leq N$).

A área verde da Figura 4.2 apresenta as posições e transições para a operação prevista, isto é, quando TL_i inicia a preempção no momento certo e nenhum cancelamento da preempção é agendado. Neste caso, a *transição inicial* ou a transição T_1 de TL_{i-1} coloca um *token* na posição P_0 . A transição T_0 imediatamente coloca um *token* na posição P_1 e um na posição P_2 . Quando a transição temporizada T_2 dispara, ela coloca um *token* na posição P_3 para iniciar a preempção de TL_i . A Seção 4.4 apresenta o algoritmo para calcular o tempo para disparar a transição T_2 . Um *token* na posição P_3 dispara a transição T_3 , que imediatamente coloca um *token* na posição P_4 . Este último *token* não dispara a transição T_4 , já que essa transição também precisa de um *token* oriundo da posição P_5 .

Quando o EV cruza TL_i , a solução dispara a transição T_5 do bloco de controle i , que coloca um *token* na posição P_5 e, consequentemente, dispara a transição T_4 , colocando um *token* na posição P_6 para desabilitar a preempção de TL_i e restaurá-lo para sua operação prevista. A transição T_5 também coloca um *token* na posição P_8 para desabilitar (por meio da aresta inibidora relacionada) disparos subsequentes da transição T_5 . A posição P_6 então dispara a transição T_6 , que coloca um *token* na posição P_7 , sinalizando o fim da preempção de TL_i .

A solução pode precisar cancelar a preempção a qualquer momento porque o EV ou os veículos à sua frente estão muito lentos ou mesmo parados. Manter a preempção nesta situação pode criar caos nas ruas da vizinhança com longas filas de veículos parados com sinais vermelhos em ruas concorrentes. Quando a solução determina que é necessário cancelar a preempção (usando o Procedimento *Cancelamento Global* da Seção 4.4), ela dispara a transição T_7 , que coloca *tokens* nas posições P_9 e P_{10} (para desabilitar disparos futuros da transição T_7 por meio de sua aresta inibidora). Estas transições e posições são compartilhadas entre todos os blocos de controle (região vermelha na Figura 4.2). A transição T_8 coloca *tokens* nas posições P_{11} e P_{12} do bloco de controle de cada TL_i .

Um *token* na posição P_{11} de TL_i inibe as transições T_0 e T_2 prevenindo *tokens* de avançar por meio do caminho da operação prevista (a região verde na Figura 4.2) se a preempção já foi agendada para TL_i , mas a transição T_2 ainda não disparou para assim colocar um *token* na posição P_3 e habilitar a preempção em TL_i . Este *token* também inibe a transição T_1 , prevenindo um *token* de avançar para a posição P_0 de TL_{i+1} e, conseqüentemente, inibindo a preempção em $TL_{i+1} \dots TL_N$.

Se a preempção já iniciou em TL_i , é necessário restaurá-lo para sua programação padrão, garantindo assim que um *token* chegue até a posição P_6 . É necessário considerar se o EV já cruzou TL_i ou não. No primeiro caso, a solução ainda não disparou a transição T_5 , então a posição P_5 não recebeu um *token* da transição T_5 , inibindo assim a transição T_4 . Contudo, devido ao fato de a posição P_{12} ter habilitado a transição T_9 , um *token* é colocado na posição P_5 , habilitando a transição T_4 e colocando um *token* na posição P_6 para restaurar TL_i para sua programação padrão. Além disso, a posição P_{11} inibe a transição T_5 de colocar um *token* na posição P_5 e, conseqüentemente, restaura TL_i duas vezes. O segundo caso acontece quando o EV já cruzou TL_i , e assim nada precisa ser feito, já que a transição T_5 disparou e colocou um *token* na posição P_6 por meio da posição P_5 e da transição T_4 . As arestas inibidoras nas posições P_5 , P_6 e P_7 também inibem as transições T_5 e T_9 de colocar *tokens* na posição P_6 , mas do que uma vez.

4.4 Algoritmo de Controle

O objetivo de se usar uma TPN é a garantia de que os estados possíveis da solução sejam bem conhecidos e evitar que estados inválidos aconteçam, colocando em risco os seus usuários. E uma das garantias necessárias para que o modelo proposto forneça esse nível de segurança almejado é que a solução apresentada na Seção 4.3 não execute ações concorrentes.

Para que isso ocorra, cada bloco de controle deve satisfazer as seguintes propriedades: (i) executar no máximo uma ação de preempção; (ii) não se deve restaurar o estado de qualquer TL_i antes de iniciar a preempção; e (iii) executar no máximo uma ação de restauração.

Para avaliar se as condições anteriores são alcançadas, devemos verificar o fluxo de *tokens* pela rede. O *token* que chega à posição P_3 o faz apenas por meio da posição P_0 , que recebe um *token* somente uma vez da *transição inicial* para a transição TL_1 ou da transição T_1 oriundo de TL_{i-1} para todo TL_i , $2 \leq i \leq N$. Pelo fato de a ação de preempção acontecer quando o *token* chega à posição P_3 , um bloco de controle executa no máximo uma ação de preempção. Para as propriedades (ii) e (iii), o *token* precisa alcançar a posição P_6 por meio da transição T_4 . Como a transição T_4 depende das posições P_4 e P_3 , um bloco de controle não restaura o estado de TL_i antes da posição P_3 iniciar a preempção. Da mesma forma, devido ao fato de a posição P_4 receber apenas um *token*, a ação de restauração acontece no máximo uma vez.

Assim, o modelo de controle previne que a solução não fique em um estado inconsistente. Contudo, o modelo não fornece melhorias no desempenho para o tempo de viagem do EV por si só. A melhoria de desempenho vem de iniciar a preempção dos semáforos no momento exato para permitir que o EV e os veículos à sua frente cruzem os semáforos sem perder tempo nas interseções. Portanto, é necessário calcular as transições temporizadas T_2 para cada bloco de controle na rota do EV quando este inicia o atendimento de uma emergência.

A solução considera alguns parâmetros, como distância, velocidade e aceleração do EV, dados do ciclo dos semáforos e informações de densidade das vias.

Seja $d_{TL_i}(m)$ a distância entre o EV e TL_i , $ASLpath_i(m/s)$ é a média do limite de velocidade no caminho do EV para TL_i e $t_{flush_i}(s)$ é o tempo necessário para que TL_i altere suas luzes de verde para os outros veículos (passando pela luz amarela e vermelha) para possibilitar a passagem do EV. Considere ainda que a ocupação desta mesma via é $Opath_i$ (valores no intervalo $[0, 1]$) e indica o quão congestionadas estão as vias até TL_i . Os valores ϵ (s) e $\delta(m/s)$ são constantes pequenas que capturam quão perto o EV está de TL_i e o quão lentos os veículos estão se movendo, respectivamente.

O tempo da transição temporizada T_{2_i} é calculado conforme a Equação 4.1.

$$T_{2_i} = \begin{cases} 0, \text{ se } \frac{d_{TL_i}}{ASLpath_i} \leq \epsilon \text{ ou } ASpath_i \leq \delta \\ \max\{(\frac{d_{TL_i}}{ASLpath_i} - t_{flush_i}) \times (1 - Opath_i), 0\}, \text{ caso contrário} \end{cases} \quad (4.1)$$

Se o EV precisar de ϵ segundos ou menos para atingir TL_i ou os veículos

estão se movendo com velocidade δ ou inferior, a preempção precisa iniciar imediatamente para TL_i . Caso contrário, o tempo T_{2_i} é computado dividindo a distância de EV para o semáforo pela velocidade máxima que este veículo pode atingir até alcançar TL_i menos o tempo de segurança (t_{flush_i}), permitindo que TL_i altere para a luz verde para o EV. A Equação 4.1 também usa um fator de escala $1 - Opath_i$ para reduzir T_{2_i} quando a ocupação do segmento entre o EV e TL_i está alta.

A solução monitora continuamente o EV e o tráfego para detectar comportamentos inesperados. Caso ocorram, a preempção é cancelada, recalculada e reagendada.

O Algoritmo 4.1 descreve os passos necessários a serem executados enquanto na passagem do EV pelos semáforos. O algoritmo inicializa a TPN' das Figuras 4.1 e 4.2 usando a Equação 4.1 (linha 9), dispara as transições imediatas assim que possível (linha 10) e executa as ações associadas com as posições P_3 e P_6 quando *tokens* os alcançam (linha 11). Por exemplo, quando o *token* alcança a posição P_3 de TL_i , a solução precisa iniciar a preempção para TL_i . Quando o EV cruza TL_i , a solução precisa disparar a transição T_5 (linha 13). O último passo completa a operação normal habilitando a posição P_5 e, na sequência, a posição P_6 . A ação associada com a posição P_6 desabilita a preempção de TL_i e o restaura para sua programação normal.

Caso seja necessário cancelar a preempção, a solução dispara a transição T_7 nos blocos de controle de todos os TL_i para os quais o EV ainda não cruzou (linha 17). Se o EV consegue avançar na sua rota, a solução detecta quando ele cruza algum TL_i (linha 14 do Algoritmo 4.1) e reconstrói a TPN' para agendar uma nova preempção (linha 18 do Algoritmo 4.1). Mesmo que o EV não cruze alguma interseção, o agendamento de uma nova preempção pode ajudar, portanto, a solução cria um *timer* para realizar uma nova tentativa de reconstruir a TPN' e agendar uma nova preempção (linhas 22, 23 e 24).

A solução então executa o Procedimento *Cancelamento Global* (linha 27) para decidir se ele deve cancelar uma preempção em andamento. O procedimento inicialmente verifica se o próximo TL_i à frente do EV está verde, bem como se os veículos estão parados (linha 2). Se essas condições forem verdadeiras, uma abordagem conservadora é usada, e ao invés de cancelar a preempção imediatamente, a solução inicia um *timer* antes de cancelar a preempção (linha 4). Se as condições não forem alteradas, então a preempção é cancelada (linha 7). Outra possibilidade para cancelar a preempção é quando o EV não está se movendo (linha 13), e a velocidade dos veículos no mesmo caminho é menor que $\beta(\frac{m}{s})$, um parâmetro da solução que indica que a preempção não está ocorrendo como o desejado, e os veículos estão se movendo muito devagar (linha 17). Neste caso, o mesmo *timer* é definido (linha 15) e

Algoritmo 4.1: Preempção com TPN'

```
1 TPN  $\leftarrow$  NULO;  
2 cancelar  $\leftarrow$  FALSO;  
3 enquanto ev está atendendo a uma emergência faça  
4   curTL  $\leftarrow$  próximo TL;  
5   se temporizador de novas tentativas expirou então  
6     TPN  $\leftarrow$  NULO;  
7   fim  
8   se TPN == NULO então  
9     limpe o temporizador de novas tentativas;  
10    TPN  $\leftarrow$  construa a TPN' usando a Equação 4.1;  
11  fim  
12  Dispare as transições habilitadas;  
13  Execute as ações das posições habilitadas;  
14  se EV cruzou curTL então  
15    Dispare a transição  $T_5$  de curTL, se habilitada;  
16    se cancelar então  
17      cancelar  $\leftarrow$  FALSO;  
18      TPN  $\leftarrow$  reconstrua a TPN' usando a Equação 4.1;  
19    fim  
20  fim  
21  senão se curTL está verde E ev está parado E a faixa tem veículos E  
    os veículos estão parados então  
22    Dispare a transição  $T_7$ , se habilitada;  
23    cancelar  $\leftarrow$  VERDADEIRO;  
24    defina o temporizador de novas tentativas;  
25  fim  
26  se !cancelar então  
27    executar o Procedimento Cancelamento Global;  
28  fim  
29 fim
```

o cancelamento é feito na linha 18. Esse comportamento do Procedimento *Cancelamento Global* garante que a preempção não é cancelada quando as condições do trânsito mudam em intervalos pequenos. O restante das linhas dispara as transições necessárias para o cancelamento e redefine os *timers* se necessário.

Algoritmo 4.2: Cancelamento Global

```

1  podeCancelar ← FALSO;
2  se curTL está verde E os veículos da faixa de curTL estão parados então
3    se temporizador para curTL NÃO está definido então
4      defina o temporizador para curTL;
5    fim
6    senão se temporizador para curTL expirou então
7      podeCancelar ← VERDADEIRO;
8    fim
9    senão
10     limpe o temporizador para curTL;
11   fim
12 fim
13 senão se ev está parado então
14   se temporizador para curTL NÃO está definido então
15     defina o temporizador para curTL;
16   fim
17   se faixa está ocupada E velocidade médias dos veículos  $\leq \beta$  então
18     podeCancelar ← VERDADEIRO;
19   fim
20   senão se temporizador para curTL está definido então
21     limpe o temporizador para curTL;
22   fim
23 fim
24 se podeCancelar OU o temporizador para curTL expirou então
25   Dispare a transição  $T_7$ , se habilitada;
26   cancelar ← VERDADEIRO;
27   defina o temporizador de retentativas;
28 fim

```

4.4.1 Detalhes de Implementação

Como dito anteriormente, a linha 10 do Algoritmo 4.1 utiliza a Equação 4.1, e o valor δ representa o limiar de velocidade dos outros veículos para considerar que o trânsito está congestionado. Em experimentos empíricos, definimos o valor para $0,1m/s$. Na mesma equação, o valor ϵ representa a distância tolerável do EV para o semáforo, em segundos. Quando a previsão de chegada do EV no semáforo se torna inferior a ϵ , a preempção deve iniciar imediatamente. Empiricamente, definimos o valor de ϵ para 3 segundos.

O Procedimento *Cancelamento Global* (Procedimento 4.2), em sua linha 17, faz referência à constante β . Este valor representa o limiar de velocidade do EV para considerar o trânsito congestionado, para que, assim, o cancelamento da preempção seja disparado. Este parâmetro também foi definido empiricamente em experimentos com o valor de $0,25 \frac{m}{s}$. Estes valores são conservadores; isto é, eles adiam o início da preempção o máximo possível, evitando, assim, perturbações desnecessárias no trânsito no entorno do EV.

4.4.2 Complexidade do Algoritmo

É incomum encontrar análises sobre a complexidade de algoritmos de preempção para veículos de emergência, quando analisando pela perspectiva da análise proposta por Knuth (1973). Tal fato ocorre porque, ao se realizar essa modalidade de análise, geralmente se considera que o algoritmo executa do início ao fim e, na tomada de tempo, são desconsiderados o tempo gasto pelo sistema operacional, as trocas de contexto e o carregamento dos dados do disco para a memória. No caso de soluções para a redução do tempo perdido pelo EV no trânsito, em especial, os eventos que acionam e cancelam a preempção, em geral, são assíncronos e não dependem diretamente do algoritmo, e o tempo intrínseco ao deslocamento do EV no trânsito não está relacionado com o tempo de execução da solução.

Entretanto, isso não inviabiliza completamente a análise. As duas primeiras linhas do Algoritmo 4.2 executam em $\mathcal{O}(1)$. Depois, na linha 3, há um laço com fim indeterminado. Excluindo algoritmos que acionam a preempção para todos os semáforos de uma vez, a maioria dos algoritmos de preempção tem uma estrutura de laço similar, ou seja, o algoritmo executará enquanto o EV não tiver cruzado todos os semáforos.

Agora, algumas simplificações precisam ser feitas. Consideraremos que os comandos de obtenção de parâmetros externos ao algoritmo (como parâmetros de telemetria do EV ou de dados do semáforo) executam em $\mathcal{O}(1)$. Então, das linhas 4 a 7, o algoritmo executa em $\mathcal{O}(1)$.

As linhas 8, 9 e 11 também executam em $\mathcal{O}(1)$. Já na linha 10, observamos que a TPN' é construída, utilizando a Equação 4.1. Supondo que haja $|TL|$ semáforos, a complexidade dessa construção depende de $|TL|$, do número de posições $|P|$ e $|T|$ transições da TPN' e do número de arestas que relacionam esses componentes $|\rightarrow|$. Portanto, a complexidade de construção da TPN' é $\mathcal{O}(|TL| \cdot (|P| + |T| + |\rightarrow|))$. O valor $|P| + |T| + |\rightarrow|$ é constante, pois cada bloco possui 11 posições, 8 transições e 30 arestas de relacionamento. Ainda deve-se considerar as duas posições, duas transições e 4 arestas de relacionamento do bloco de cancelamento global, conforme apresentado na Figura 4.2.

As linhas 12 e 13 dependem da implementação. Se o algoritmo for re-

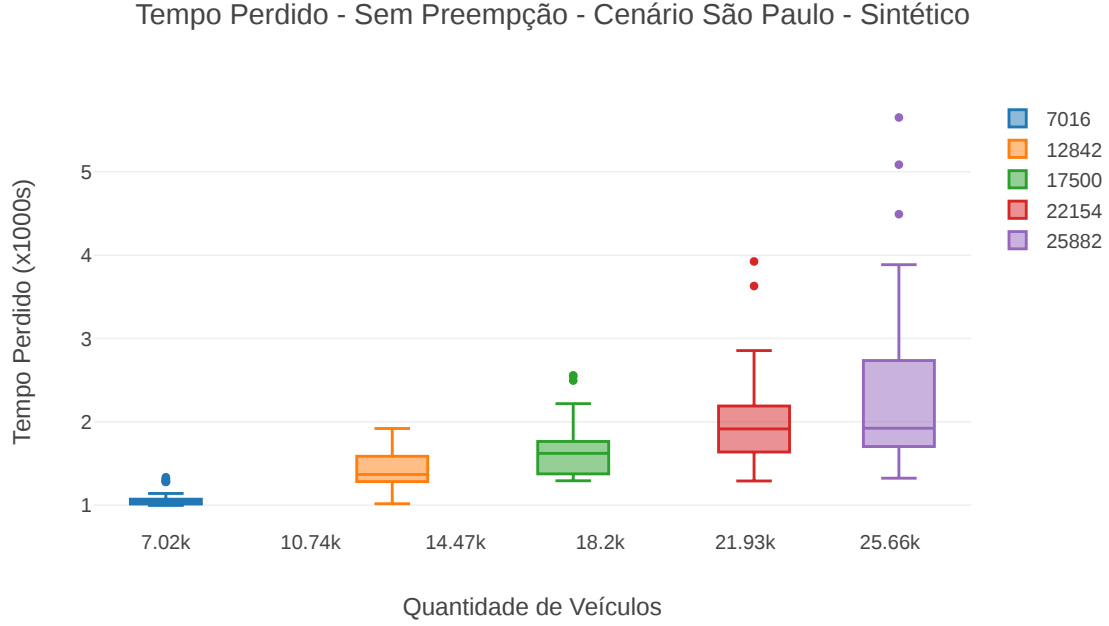


Figura 4.3: Tempo Perdido (s) - Versão Sem Preempção - Cenário São Paulo - Sintético.

ativo, apenas as transições habilitadas são disparadas e apenas as ações das posições habilitadas são executadas. Isso pode ser feito usando funções de *callback* ou *webhooks*, por exemplo. Contudo, consideraremos aqui que o algoritmo precisa percorrer a lista de $|T|$ e $|P|$, para cada $|TL|$. Portanto $\mathcal{O}(|TL| \cdot (|P| + |T|))$.

As linhas 14 a 20, com exceção da linha 18, executam em $\mathcal{O}(1)$. A linha 18 também executa em $\mathcal{O}(|TL| \cdot (|P| + |T| + |\rightarrow|))$, tal qual a linha 10.

O restante das linhas, de 21 a 29, executam em $\mathcal{O}(1)$, com exceção da linha 27. A linha 27 invoca o procedimento de cancelamento global. Contudo, esse procedimento, das linhas 1 a 28, apenas testa e define algumas variáveis, então todas as linhas desse procedimento executam em $\mathcal{O}(1)$.

4.5 Resultados

Para estes experimentos, a versão do SUMO utilizada foi a versão 1.4.

4.5.1 Cenários São Paulo e Nova Iorque - Sintético

Os primeiros gráficos apresentam o impacto do aumento do congestionamento para o EV, na versão Sem Preempção. As Figuras 4.3 e 4.4 apresentam o tempo perdido pelo EV nos dois cenários.

É possível notar que o aumento do congestionamento aumenta o tempo perdido e também a distância das medições para a mediana. Por exemplo,

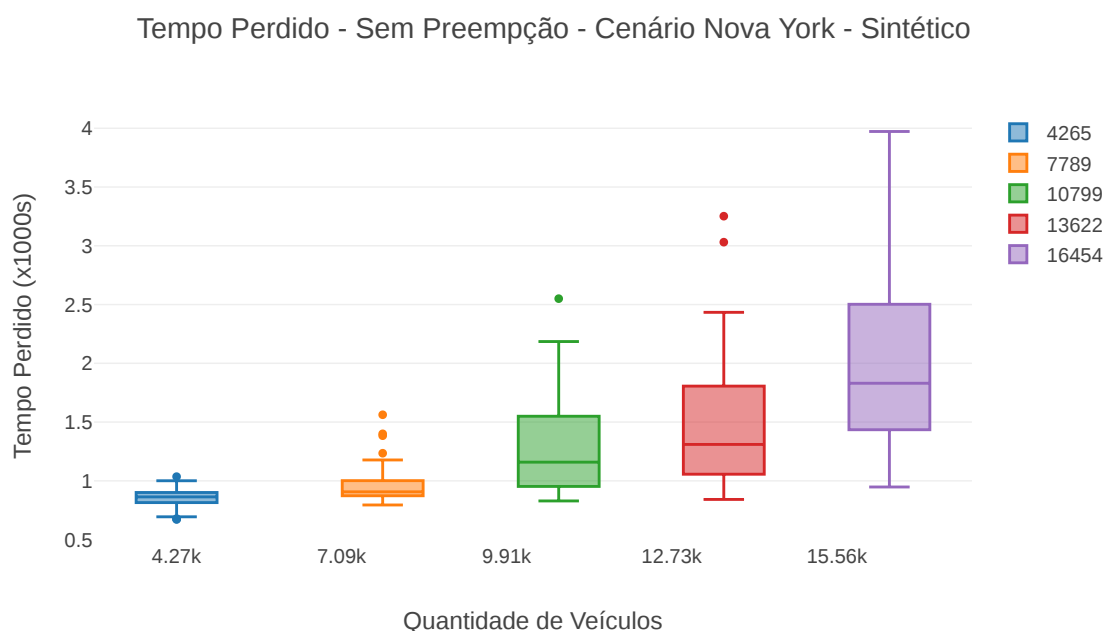


Figura 4.4: Tempo Perdido (s) - Versão Sem Preempção - Cenário Nova Iorque - Sintético.

para 7.016 veículos no cenário sintético de São Paulo, o EV apresenta tempo perdido com mediana de 1.037,14s, com o Intervalo Interquartil (IQR) de apenas 57,74s, enquanto que para 25.882 veículos a mediana é 1.923,27s com o IQR de 1.032,15s.

Da mesma forma, isso é verificado no cenário de Nova Iorque, para 4.265 veículos, a mediana é 863,42s com o IQR de 85,92s, enquanto que para 16.454 veículos, a mediana é 1.830,27s com o IQR igual a 1.067,36s.

As Tabelas A.1 e A.2 apresentam os dados completos do experimento.

As Figuras 4.5 e 4.6 apresentam, dentre o tempo total de viagem do EV, qual é o percentual referente ao tempo perdido.

Notamos que boa parte do tempo gasto pelo EV no trânsito é tempo perdido, com medianas apresentando valores de 58,95% para 7.016 veículos e 73,36% para 25.882 veículos, no caso do cenário de São Paulo, e 68,97% para 4.265 veículos e 83,76% para 16.454 veículos, no caso do cenário de Nova Iorque.

Da mesma forma, o IQR também aumenta conforme o congestionamento aumenta, sendo igual a 3,24% para 7.016 veículos e 8,72% para 25.882 veículos, para o cenário de São Paulo, e 2,92% para 4.265 veículos e 6,61% para 16.454 veículos, para o cenário de Nova Iorque.

As Tabelas A.3 e A.4 apresentam os dados relacionados às Figuras 4.5 e 4.6.

A Figura 4.7 apresenta a melhoria do tempo perdido pelos algoritmos para São Paulo.

O Algoritmo Onda Verde é superior em todos os casos, como esperado, e o

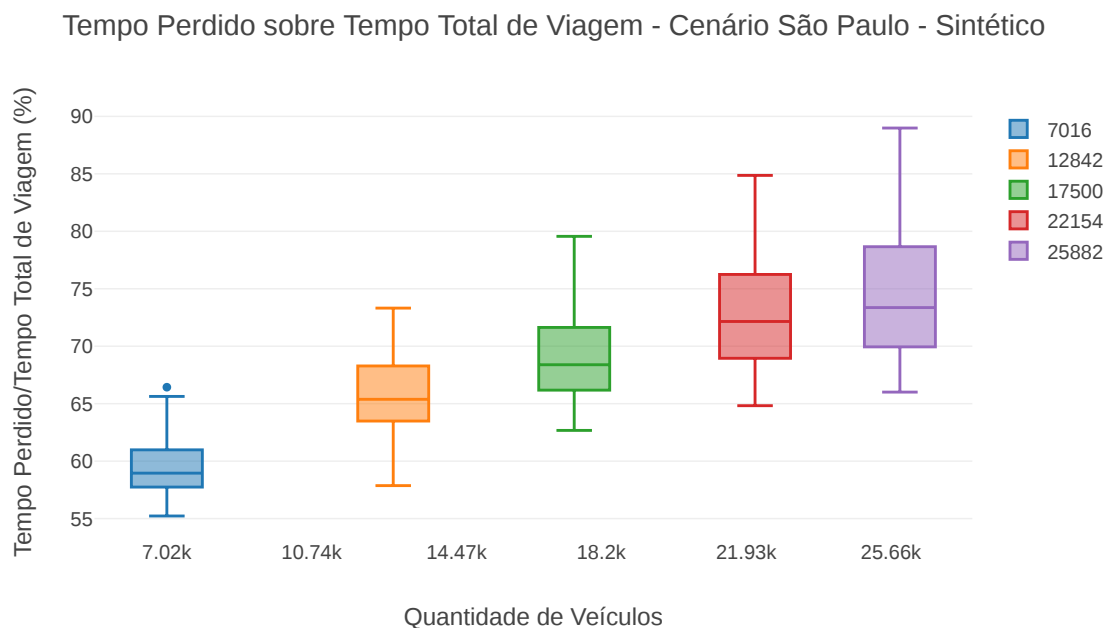


Figura 4.5: Tempo Perdido sobre o Tempo Total de Viagem (%) - Versão Sem Preempção - Cenário São Paulo - Sintético.

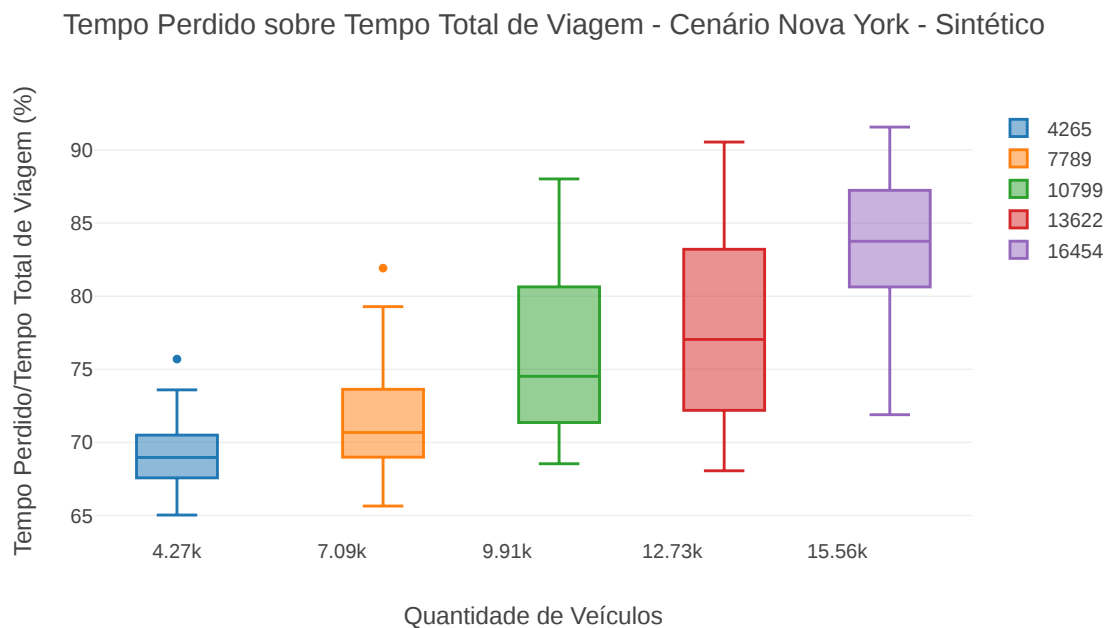


Figura 4.6: Tempo Perdido sobre o Tempo Total de Viagem (%) - Versão Sem Preempção - Cenário Nova Iorque - Sintético.

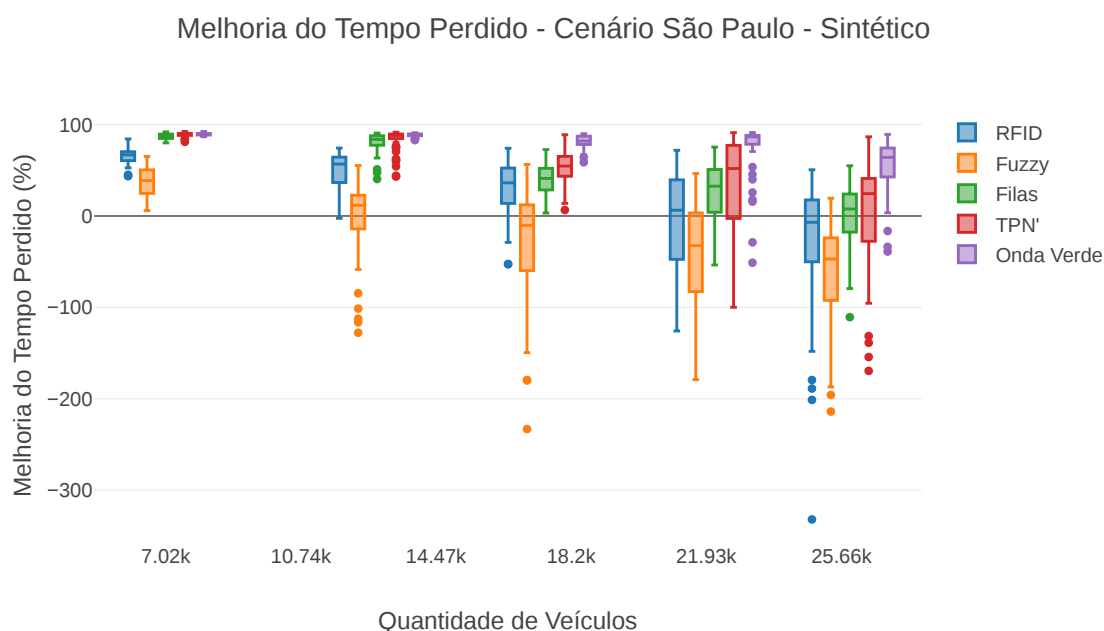


Figura 4.7: Melhoria do Tempo Perdido (%) - Algoritmos - Cenário São Paulo - Sintético.

Algoritmo TPN', proposto neste trabalho, apresenta resultados melhores que os outros nas três primeiras instâncias, para cenários com 7.016, 12.842 e 17.500 veículos.

Apesar de ser superior em sua mediana à versão Baseada em Filas, a quarta instância, com 22.154 veículos, apresenta sobreposição da região interquartil, além de contar com 25% dos resultados negativos. O mesmo ocorre na quinta instância, para 25.882 veículos, mas com mais resultados negativos.

É possível observar que o Algoritmo Onda Verde, para esta última instância, também apresenta alguns *outliers* ruins, com 25% dos resultados ficando abaixo de 42,71%, indicando que mesmo este Algoritmo Teórico encontra alguns desafios nesta instância.

Os dados completos estão disponíveis nas Tabelas A.5, A.6, A.7, A.7, A.8 e A.9 que apresentam os dados completos para avaliação.

Já para o cenário de Nova Iorque, conforme apresentado pela Figura 4.8, as medianas do Algoritmo TPN' permanecem maiores apenas nas duas primeiras instâncias, sendo que o Algoritmo Filas é superior nos outros casos.

Apesar de possuir alguns *outliers* para o Algoritmo Onda Verde e de que 25% dos resultados do Algoritmo Filas apresentam resultados negativos para esta instância, o Algoritmo TPN' é ainda pior, com mais de 50% dos resultados apresentando resultados negativos.

As Tabelas A.10, A.11, A.12, A.13 e A.14 apresentam os dados para este cenário. É possível navegar interativamente nos gráficos apresentados para

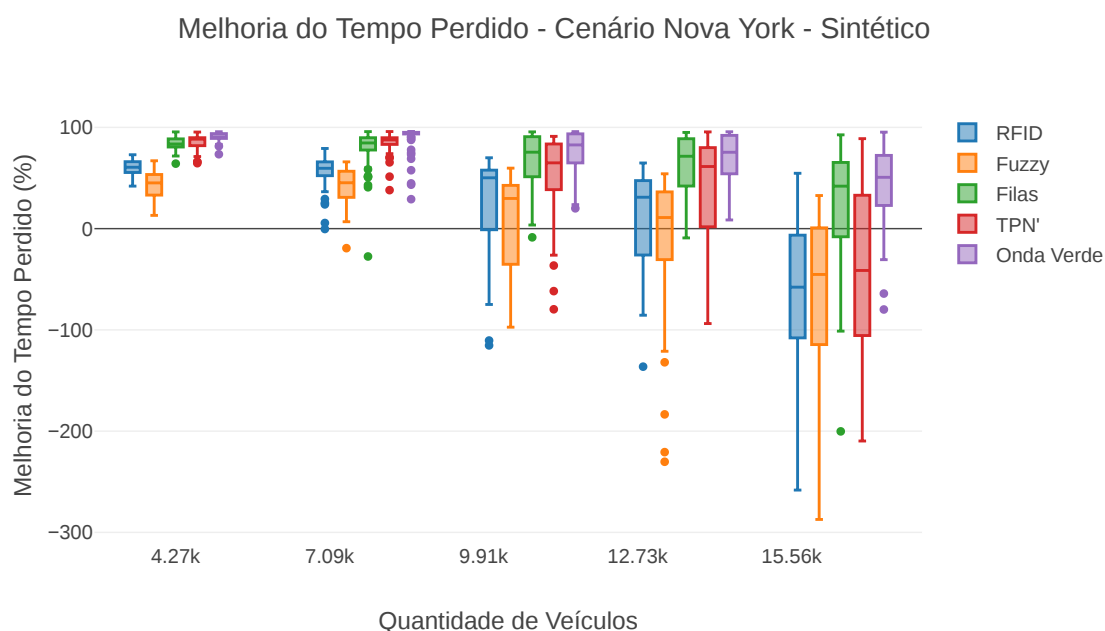


Figura 4.8: Melhoria do Tempo Perdido (%) - Algoritmos - Cenário Nova Iorque - Sintético.

verificação dos resultados¹.

4.6 Conclusão

Os resultados obtidos mostram que o Algoritmo TPN' é competitivo frente aos outros, mas há situações em que seu desempenho foi inferior ao que é observado nos algoritmos Filas e Onda Verde.

Observando a Figura 4.8, é possível notar que o Algoritmo Filas possui quase os mesmos valores de mediana que o Algoritmo Onda Verde, seguidos do Algoritmo TPN'. O fato é que o Algoritmo TPN' possui um mecanismo de cancelamento, quando o mesmo percebe que a preempção não está surtindo o efeito desejado. O mesmo não ocorre com os Algoritmos Filas e Onda Verde, e uma vez que esses algoritmos iniciam a preempção para um determinado semáforo, o mesmo permanece dessa forma até que o EV cruze esse semáforo. Em cenários muito congestionados, então, os dois algoritmos se comportam de forma muito parecida, já que eles logo iniciam a preempção e demoram para encerrar.

E esse é um dos principais problemas dos algoritmos mencionados, pois eles podem levar a uma situação de *starvation* para os outros veículos que estão em outras vias. Apesar de os experimentos não terem mais de um EV ao mesmo tempo, nos casos em que possam ocorrer EVs concorrentes no mesmo

¹https://nbviewer.org/github/rodrigogbranco/charts/blob/main/tese_2025-06-28-primeiro-experimento-app.ipynb

semáforo, alguns deles esperariam por um tempo desconhecido.

Portanto, é importante manter as propriedades de segurança presentes na TPN' pois tratam de alguns aspectos desejáveis mencionados na Seção 2.7. Revisitaremos tais aspectos em capítulos posteriores. Enquanto isso, observamos que o Algoritmo Filas obteve bons resultados em algumas situações e ruins em outras; por exemplo, na Figura 4.7 é possível observar que ele é inferior ao Algoritmo TPN' para algumas medições. No próximo Capítulo aprofundaremos em aspectos do Algoritmo Filas já mencionados na Subseção 3.2.3 e proporemos melhorias neste Algoritmo.

Segunda Solução - Onda de Choque

Neste capítulo, apresentamos a segunda solução desenvolvida neste trabalho, fundamentada no princípio de Onda de Choque, com o objetivo de superar as limitações identificadas nas abordagens baseadas em filas. Inicialmente, apresentamos a motivação para o desenvolvimento desta nova proposta (Seção 5.1). Em seguida, detalhamos o princípio teórico que norteia a solução (Seção 5.2) e descrevemos o algoritmo proposto (Seção 5.3). Posteriormente, discutimos os resultados experimentais obtidos nos cenários de uma interseção, Turim e Colônia (Seção 5.4). Realizamos, também, um estudo específico sobre as anomalias detectadas durante os experimentos (Seção 5.5). Por fim, apresentamos as conclusões do capítulo (Seção 5.6).

5.1 Motivação

Apesar do algoritmo TPN' ter apresentado um desempenho competitivo, o Algoritmo Filas (baseado no algoritmo proposto em Kapusta et al. (2017)) apresenta uma abordagem interessante para a dissipação da fila de veículos que se forma quando o semáforo está fechado. Contudo, conforme apresentado na Seção 3.2.3, esse algoritmo possui alguns pontos fracos.

Dentre os problemas encontrados no Algoritmo Filas (KAPUSTA et al., 2017), a utilização da ocupação média das vias foi um dos principais pontos de atenção, tendo em vista que entre o EV e o semáforo pode haver pelotões isolados que respondem por uma baixa ocupação, mas que, pontualmente, influenciam muito na dinâmica do trânsito naquele ponto. Outro fator importante é que, quanto maior a fila de veículos parada na interseção, mais tempo leva para que essa fila seja dissipada e, nesse algoritmo, a preempção era cancelada

sem levar esse fator em consideração. Por este motivo, apresentamos uma nova solução. E não menos importante, o algoritmo pode causar *starvation* já que a preempção não é interrompida enquanto o EV não cruza o semáforo, e, em caso de veículos de emergência concorrentes, um deles será prejudicado.

As linhas 4 e 11 do Algoritmo Filas 3.3 consideram o tempo de chegada do EV baseado em sua posição, mas, dependendo desta distância, a preempção não é iniciada, mesmo que a fila de veículos seja grande o suficiente. O contrário também é verdadeiro, ou seja, mesmo que haja pouco congestionamento, o algoritmo ainda é ativado, pois apenas os parâmetros T_α , T_β , s_α e s_β e o tempo do ciclo dos semáforos são usados para iniciar a preempção. Para fins de exemplificação, um EV trafegando a $70km/h$ irá ativar a preempção quando estiver a uma distância de $5,4km$ para os parâmetros $s_\alpha = 10m$ e $s_\beta = 15m$ utilizados em Kapusta et al. (2017), o que corresponderia a ter apenas de dois a quatro veículos na fila do semáforo. Por fim, uma vez que o algoritmo é ativado, ele fica alternando entre as fases verde e vermelha (Linhas 5 e 8), quando a luz verde já deveria ter sido fornecida para dissipar a fila de veículos à frente do EV. Dessa forma, o algoritmo manterá a preempção ativa por mais tempo. Cabe ressaltar que, como os autores não mencionam os fatores de aumento e redução dos ciclos (Linhas 5 e 8) e que as informações nem o código-fonte foram fornecidos para reprodução exata de seus resultados, uma redução de 10% (definida empiricamente) foi utilizada para os dois casos, conforme apresentado na Seção 3.2.3.

Podemos também observar que o algoritmo apenas considera o comprimento da fila após o início da preempção (Linhas 5 e 8), quando este ultrapassa os *thresholds* s_α e s_β . Neste capítulo, propomos um algoritmo que aborda tais problemas.

5.2 Princípio da Onda de Choque

Nesta Seção, apresentamos a definição deste princípio. Por meio de observações experimentais, Greenshields et al. (1935) constataram que a velocidade de um veículo $v(m/s)$, em média, possui uma relação linear com a densidade do tráfego $k(veh/m)$. A ideia é que os veículos podem trafegar na velocidade máxima permitida quando quase não há veículos na via, mas, quando a densidade do tráfego aumenta o suficiente, essa velocidade tende a zero. A Figura 5.1 apresenta esta relação, adaptada do trabalho original.

Lighthill e Whitham (1955), por sua vez, relacionam a densidade do tráfego k com o fluxo de veículos $q(veh/s)$, sendo a quantidade de veículos que cruzam um determinado ponto da via. Este relacionamento resulta na Figura 5.2. Quando k tende a zero, q também assume o valor zero. À medida

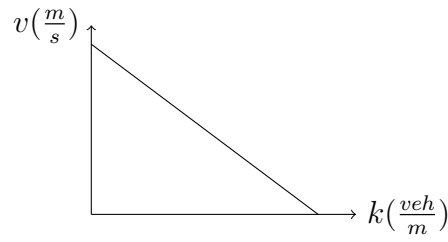


Figura 5.1: Relação linear entre velocidade e densidade do tráfego (GREENSHIELDS et al., 1935)

que k aumenta (mais veículos na via), q também aumenta, significando mais veículos passando por um determinado ponto da via. Neste momento, os veículos estão trafegando sem obstáculos, ou seja, em fluxo livre. A partir de um determinado ponto, os próprios veículos se tornam obstáculos, e o fluxo para de crescer na mesma velocidade de antes, até o ponto onde a capacidade da via é totalmente preenchida com veículos (k_m) e o fluxo não cresce mais deste ponto em diante. A partir daí, o fluxo começa a diminuir enquanto a densidade continua a aumentar, caracterizando a situação de congestionamento. Neste gráfico, ainda é possível encontrar a velocidade média dado um valor de densidade por meio da tangente de α .

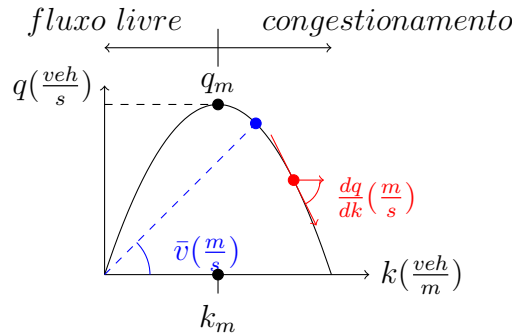


Figura 5.2: Curva de concentração de veículos vs fluxo (adaptada) de Lighthill e Whitham (1955).

Contudo, a novidade era a associação da dinâmica do trânsito com mecânica de fluidos. Lighthill e Whitham (1955) associaram a movimentação dos veículos com ondas, de forma que pequenas alterações no fluxo são propagadas com a velocidade da onda $w = \frac{dq}{dk}(m/s)$. Então, quando $w = 0$, ou $\frac{dq}{dk} = 0$, a via está com sua capacidade esgotada e não há movimentação da onda (quando $q = q_m$ e $k = k_m$). A consequência disso é que, quando uma fila de veículos está parada em um semáforo, no momento em que este fica verde, os veículos começam lentamente a se mover e, neste instante, um sistema de ondas é emitido. Cada uma dessas ondas em particular possui um valor de fluxo q , concentração k e possui um valor de velocidade de onda c , se propagando com velocidade uniforme. Uma das ondas permanece parada no ponto de origem, no local onde o veículo líder estava parado, com velocidade de onda

$c = 0$, e, como já visto, q atinge seu valor máximo (q_m). Assim, a definição formal de um onda de choque é a propagação de qualquer mudança brusca na concentração em determinado ponto da via (STEPHANOPOULOS; MICHALOPOULOS; STEPHANOPOULOS, 1979). O princípio da onda de choque é mencionado em Kapusta et al. (2017).

O princípio de onda de choque é útil para se calcular quanto tempo leva para que o último veículo da fila comece a se mover. Este tempo $t_{discharge}(s)$ é calculado pela Equação 5.1 (MARSHALL; BERG, 1997), onde Q é o comprimento da fila de veículos parada à frente do semáforo (m), k é a densidade máxima da via (veh/m) e s é o fluxo de saturação da via (veh/s). Para a via apresentada na Figura 5.2, por exemplo, $k = k_m$ e $s = q_m$. Quando os valores k e s não são conhecidos para as vias objeto de análise, Marshall e Berg (1997) sugerem utilizar os valores de 1.600 veh/h para s e 149,13 veh/km para k , originários de Transportation Research Board (1985).

$$t_{discharge} = \frac{Qk}{s} \quad (5.1)$$

Usando as definições e valores acima, descrevemos o algoritmo na seção a seguir.

5.3 Algoritmo Onda de Choque

Com o intuito de reduzir o tempo perdido pelo EV no trânsito, usando o Princípio da Onda de Choque, nesta Seção descrevemos um algoritmo que aborda uma proposta de aprimoramento do Algoritmo Filas. Entende-se que os limiares e parâmetros apresentados por Kapusta et al. (2017) não são adequados, pois o tempo de dissipação da fila de veículos independe desses fatores.

Como dito anteriormente, em (STEPHANOPOULOS; MICHALOPOULOS; STEPHANOPOULOS, 1979), o Princípio da Onda de Choque foi utilizado para modelar e analisar o comportamento de filas causadas por semáforos em interseções. A ideia básica é que as luzes do semáforo são vistas como choques de onda, levando a condições de congestionamento próximo ao semáforo durante o período de luz vermelha e de liberação da fila durante o período de luz verde. Com base em Kamalanathsharma e Hancock (2010), Kapusta et al. (2017) iniciam a preempção considerando o tempo de chegada do EV até o semáforo, considerando o tamanho da fila de veículos, mas o princípio de onda de choque não chega a ser realmente utilizado. O algoritmo proposto em Kapusta et al. (2017) utiliza alguns limiares como o tempo do ciclo do semáforo e parâmetros empíricos, mas não faz uso do tempo de dissipação da fila que não consta no algoritmo ou indica como esse tempo deve ser considerado. No algoritmo

Filas 5.1 essas premissas são mantidas.

Assim, o Algoritmo Filas 5.1 apresenta os passos do Algoritmo Onda de Choque. A ideia principal do algoritmo é utilizar o Princípio da Onda de Choque (STEPHANOPOULOS; MICHALOPOULOS; STEPHANOPOULOS, 1979; MARSHALL; BERG, 1997) para se determinar o tempo de dissipação de uma fila de veículos diante de um semáforo.

Algoritmo 5.1: Onda de Choque	
1	enquanto <i>EV</i> <i>está atendendo uma emergência</i> faça
2	$tempoChegada \leftarrow \frac{d}{v_{VE}};$
3	$Q_{flush} \leftarrow \frac{Qk}{r};$
4	se $Q \leq \frac{v^2}{2a}$ então
5	$Q_{flush} \leftarrow Q_{flush} + \sqrt{\frac{2Q}{a}};$
6	fim
7	senão
8	$Q_{flush} \leftarrow Q_{flush} + \frac{v}{a} + \frac{Q - \frac{v^2}{2a}}{v};$
9	fim
10	se $tempoChegada - (Q_{flush} + t_{flush}) \leq 0$ então
11	Altere para a fase verde;
12	fim
13	fim
14	Restaurar a fase padrão do semáforo;

Para se determinar quando a preempção deve iniciar, o tempo para o EV alcançar o semáforo é calculado na Linha 2. Porém, o algoritmo também considera o tempo necessário para esvaziar a fila de veículos à frente do EV (Q_{flush} , Linha 3) e o tempo para a alteração da fase atual do semáforo (t_{flush}), pois pode ser necessário fazer a transição para as luzes amarela e vermelha nas vias concorrentes à via do EV. Considere Q como o comprimento da fila de veículos na frente do semáforo (m). O último veículo da fila $Q(m)$ começará a se mover após $\frac{Qk}{r}(s)$, dado que k é a densidade do tráfego (veh/m) e r é a taxa do fluxo de saturação (veh/s). Este tempo é dado por Q_{flush} no Algoritmo 5.1.

Porém, é necessário também adicionar a Q_{flush} o tempo que esse último veículo levará para cruzar a interseção. Isso vai depender do tamanho da fila e da velocidade ($v(m/s)$) e da aceleração ($a(m/s^2)$) desse veículo. Para nossa implementação, consideramos que v é a velocidade máxima da via (m/s) e $a = 2,6m/s^2$, conforme determinado em estudos anteriores (LOPEZ et al., 2018). Se a fila for curta, o último veículo ainda estará acelerando quando cruzar a interseção (se a condição da Linha 4 for verdadeira) e levará $\sqrt{2Q/a}$ segundos. Caso contrário, se a fila for longa o suficiente para que o veículo atinja sua velocidade máxima e depois termine o restante do trajeto com velocidade constante, o veículo gastará $\frac{v}{a} + \frac{Q - \frac{v^2}{2a}}{v}$ segundos para cumprir esse

percurso. Se o tempo para esvaziar a fila de veículos (Q_{flush}) mais o tempo para alterar o semáforo para o sinal verde (t_{flush}) for maior ou igual ao tempo de chegada do EV até o cruzamento (Linha 10), a preempção deve ser iniciada imediatamente (Linha 11). O algoritmo mantém a preempção até que o EV cruze a interseção e depois restaura a programação normal do semáforo (Linha 14).

Se o tempo que o EV leva para alcançar o semáforo for inferior ao tempo necessário para esvaziar a fila de veículos à sua frente, o EV alcançará a fila ou o semáforo enquanto a fila ainda estiver sendo esvaziada, forçando-o a desacelerar. Para evitar essa situação, o Algoritmo 5.1 calcula periodicamente o tempo de chegada do EV e o tempo para esvaziar a fila de semáforos, iniciando a preempção se for necessário (Linha 10). Se a preempção ainda não foi iniciada, no próximo passo o EV estará mais próximo do semáforo e a fila de veículos pode estar maior ou menor. Nesse momento, uma nova verificação é feita, recalculando o tempo de chegada do EV e o tamanho da fila. Se a condição da linha 10 for atendida (quando o tempo de chegada do EV for inferior ao tempo de dissipação da fila de veículos mais o tempo da alteração da fase do semáforo), a preempção é iniciada. A preempção é mantida até que o EV cruze a interseção. Nesse momento, a preempção é finalizada e o estado do semáforo é restaurado. Note que o algoritmo proposto inicia a preempção uma única vez, evitando o *overhead* de alteração das fases do semáforo do Algoritmo 3.3.

5.3.1 Complexidade do Algoritmo

As mesmas considerações feitas para a complexidade do algoritmo da solução anterior na Subseção 4.4.2 são aplicáveis aqui. Considerando que os dados de telemetria do EV e do semáforo podem ser obtidos em $\mathcal{O}(1)$, o algoritmo dentro do laço entre as linhas 1 e 14 também seguirá essa complexidade. O detalhe é que o algoritmo proposto (KAPUSTA et al., 2017) foi projetado para apenas uma interseção. No fim das contas, portanto, dentro do laço indeterminado, o algoritmo precisa executar $\mathcal{O}(|TL|)$ vezes.

5.4 Resultados

Para estes experimentos, a versão do SUMO utilizada foi a versão 1.8. Os experimentos foram executados utilizando os cenários apresentados nas Subseções 3.4.2, 3.4.3.1 e 3.4.3.2, que são, respectivamente, os Cenários 'Uma interseção', 'Turim' e 'Colônia'. Para facilitar o entendimento, os resultados estão divididos pelos cenários citados, cada um com a sua Subseção respectiva, e são apresentados a seguir.

5.4.1 Cenário Uma interseção

Conforme apresentado na Subseção 3.4.2, pelo fato de Kapusta et al. (2017) usarem um cenário com apenas uma interseção na cidade de Zagreb, na Croácia, e como tal cenário não estava disponível para avaliação, um cenário com uma interseção, similar ao cenário do trabalho referenciado, foi gerado para fins comparativos. Portanto, na Figura 5.3, apresentamos o tempo perdido para o EV, para a versão sem preempção.

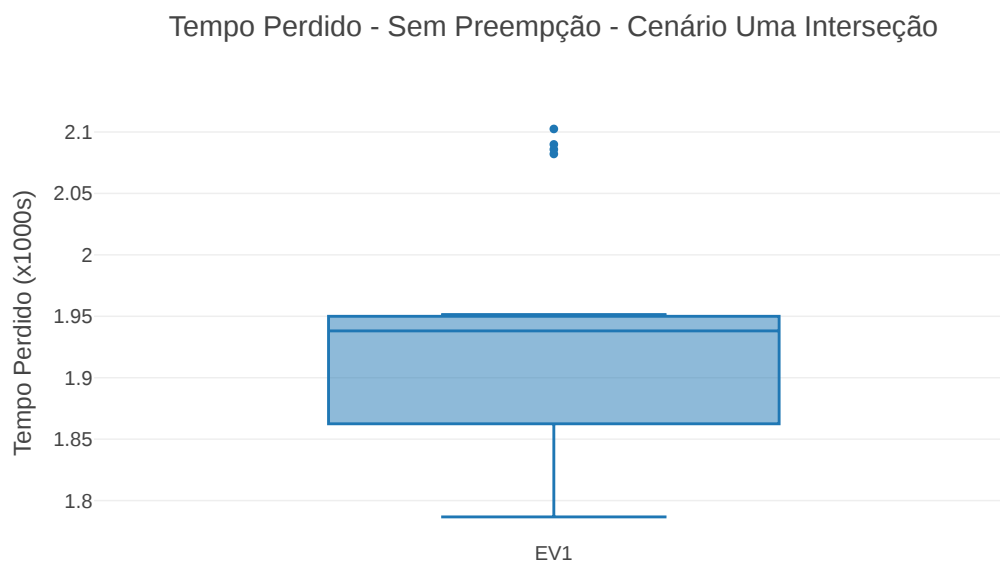


Figura 5.3: Tempo Perdido (s) - Versão Sem Preempção - Cenário Uma interseção.

A Tabela A.15 apresenta os dados correspondentes para análise (as colunas Min e Max só apresentam valores quando existem *outliers* na Figura).

A Figura 5.4 apresenta o percentual de tempo perdido em relação ao tempo total de viagem, acompanhada da Tabela A.16 com os dados relacionados.

Para os percentis Q1 e Q3, as Figuras mostram que quase todo o tempo do EV no trânsito é tempo perdido, ficando em torno de 92% de tempo perdido.

Quando analisamos a razão entre o tempo perdido antes da preempção e depois, aqui referida como Melhoria do Tempo perdido, o algoritmo proposto se comporta da mesma forma que o algoritmo Onda Verde para essa métrica, considerando a mediana dos experimentos. Tais algoritmos apresentaram resultados similares, com valor de aproximadamente 4 vezes (Figura 5.5).

Ainda considerando a mediana, os outros algoritmos não tiveram melhoria (valor de aproximadamente uma vez), enquanto o algoritmo Fuzzy, para o intervalo de confiança, obteve resultados negativos. Os resultados podem ser observados na Tabela A.17.

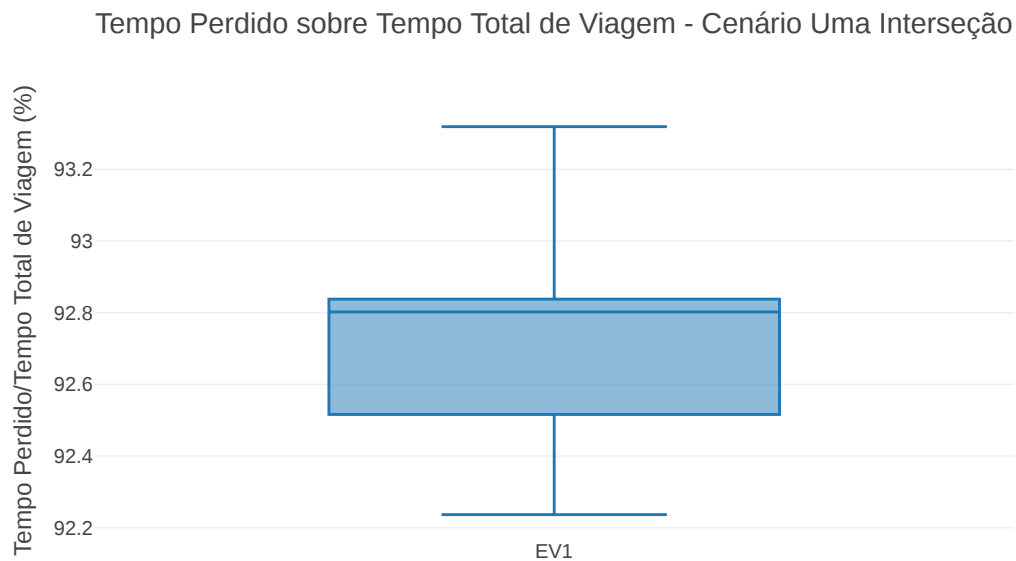


Figura 5.4: Tempo Perdido sobre o Tempo Total de Viagem (%) - Versão Sem Preempção - Cenário Uma interseção

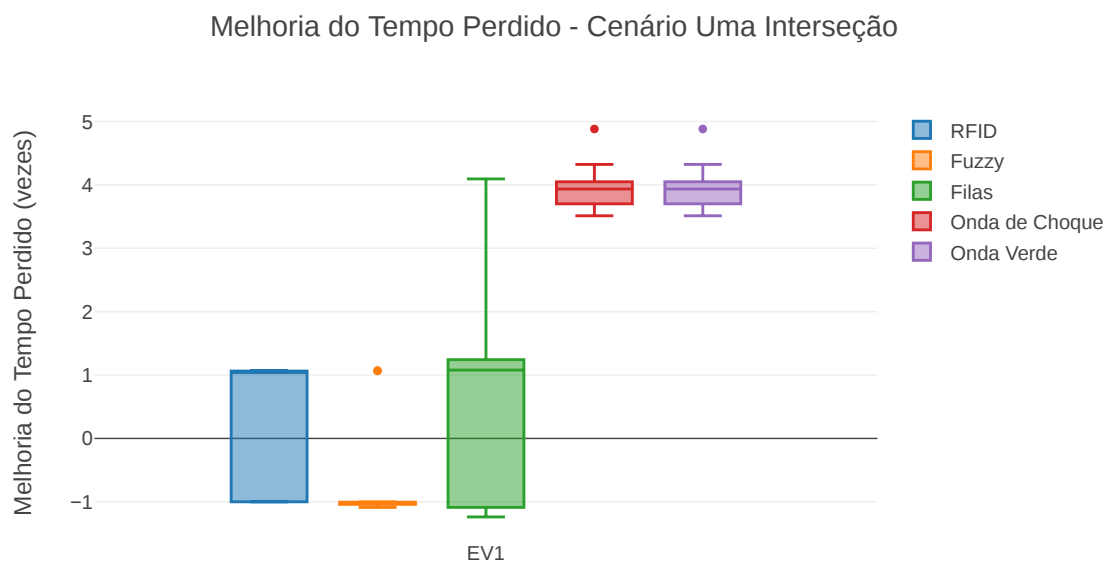


Figura 5.5: Melhoria do Tempo Perdido (vezes) - Cenário Uma Interseção

Para o *Cenário Uma Interseção*, o algoritmo proposto empata com o *Algoritmo Onda Verde* (possuindo ambos mediana de 74,5%, Figura 5.6) e é superior a todos os outros.

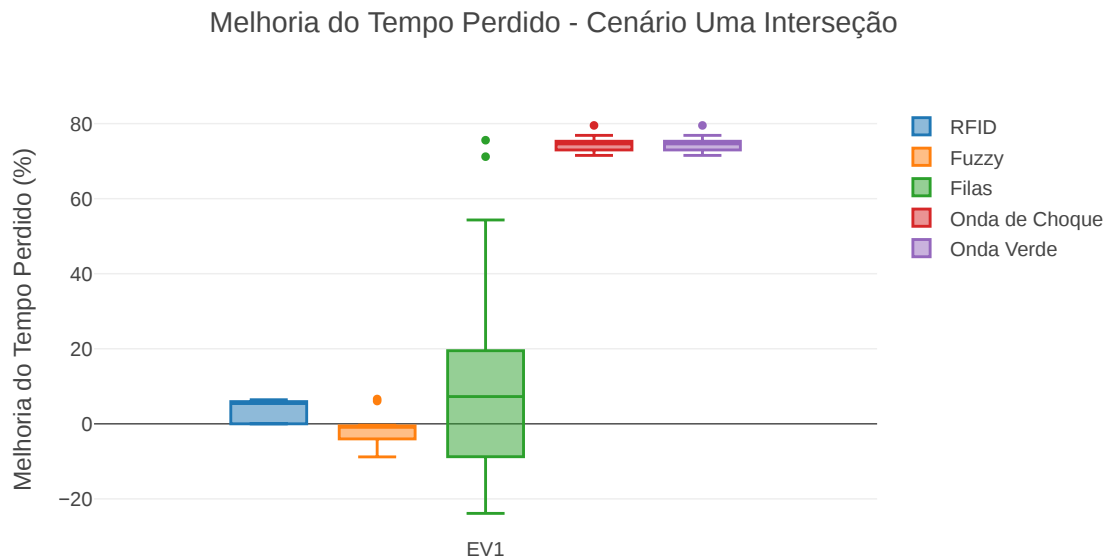


Figura 5.6: Melhoria do Tempo Perdido (%) - Cenário Uma Interseção

Neste cenário, quando o EV ingressa na simulação, a rede já está saturada e, portanto, o algoritmo proposto e o *Algoritmo Onda Verde* iniciam a preempção imediatamente. Com a rede saturada, os *Algoritmos RFID* e (*KAPUSTA et al., 2017*) não detectam o EV por um bom tempo, o que diminui a melhoria do tempo perdido, enquanto o *Algoritmo (DJAHEL et al., 2015)* causou atrasos maiores quando comparado à versão Sem Preempção. Isso pode ser observado na Figura 5.6. Os valores obtidos no experimento podem ser verificados na Tabela A.18.

A Figura 5.7 apresenta o tempo perdido (s) (quanto menor, melhor) dos algoritmos e a Tabela A.32 apresenta os dados correspondentes.

O algoritmo proposto apresenta o menor tempo perdido, juntamente com o algoritmo Onda Verde, o que condiz com o cenário de apenas uma interseção, já que, nesse caso, o algoritmo proposto se comporta tal qual o algoritmo Onda Verde. As medianas dos outros algoritmos ficam na faixa de 1.700 a 2.000 segundos, uma grande diferença quando comparada à mediana do algoritmo proposto, que é de aproximadamente 500 segundos.

Do ponto de vista de tempo de preempção médio, era esperado que os algoritmos que tiveram as maiores reduções do tempo de viagem do EV tivessem os valores mais altos para o tempo médio de preempção, pois eram acionados antes e permaneciam assim até que o EV cruzasse o semáforo. O tempo de preempção médio é equivalente para o Algoritmo proposto e o *Onda Verde*, já

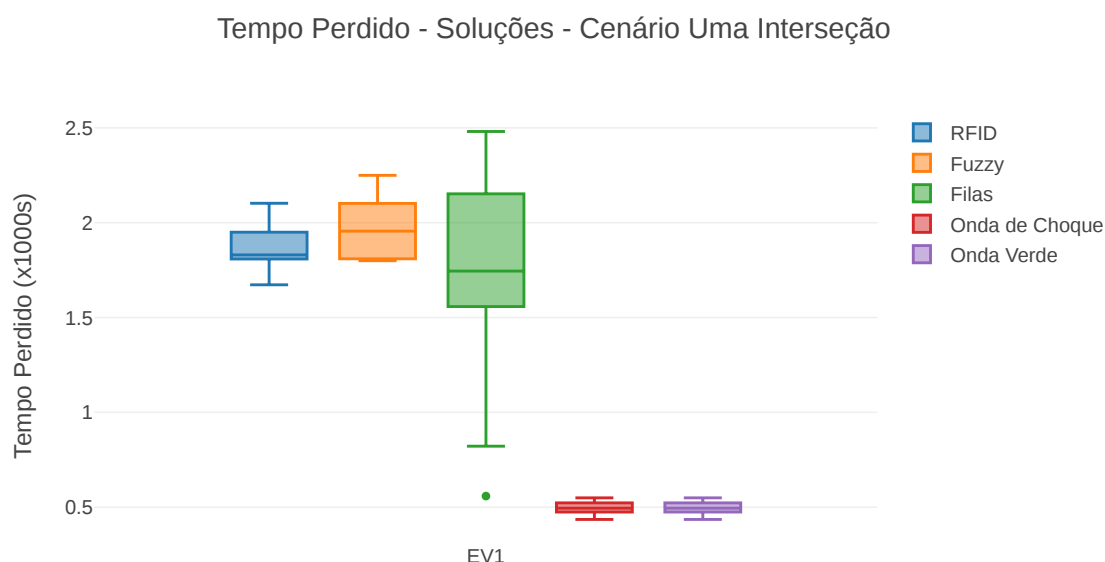


Figura 5.7: Tempo Perdido (s) - Algoritmos - Cenário Uma Interseção

que a preempção inicia e termina no mesmo momento, quando o EV cruzou o semáforo.

Contudo, o *Algoritmo* (KAPUSTA et al., 2017) ficou ativo por mais tempo do que o *Algoritmo* proposto e o *Algoritmo Onda Verde*, conforme apresentado pela Figura 5.8.

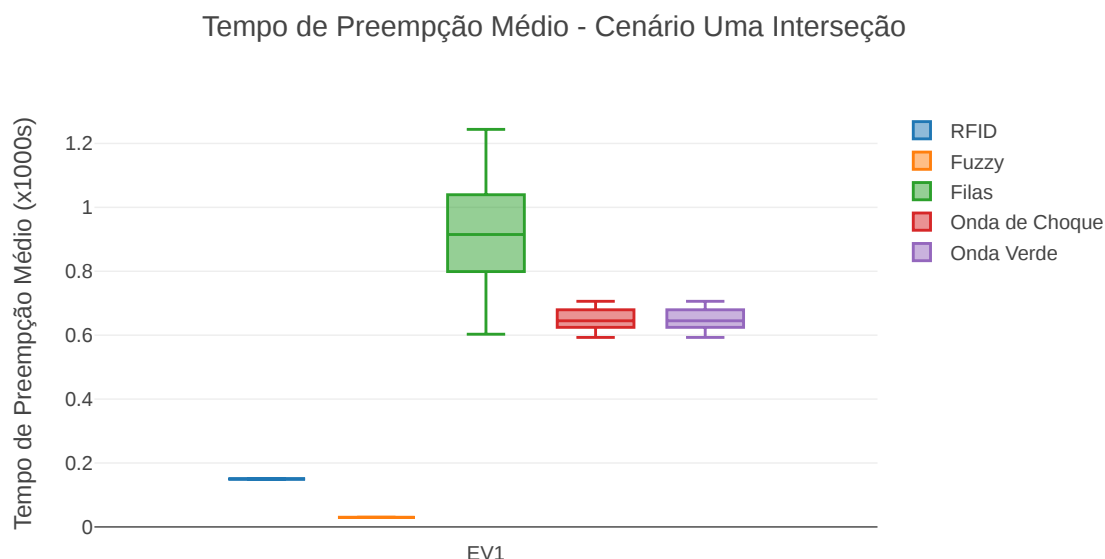


Figura 5.8: Tempo de Preempção Médio (s) - Cenário Uma Interseção

Isso é um indicativo da alternância das fases do semáforo na primeira parte do *Algoritmo* (KAPUSTA et al., 2017) (Linhas 3 a 10). Os valores obtidos no

experimento podem ser verificados na Tabela A.19.

Por fim, é interessante notar que o *Algoritmo Onda Verde* não obteve melhoria de 100% no tempo perdido pelo EV, demonstrando que há outros fatores que impactam essa métrica além dos atrasos provocados exclusivamente pelo semáforo.

5.4.2 Cenário Turim

Após a análise do Cenário com uma interseção, muito comum em outros trabalhos analisados no Capítulo 2, verificaremos agora situações onde o EV encontra situações variadas, com diferentes rotas, cada uma delas com um comprimento diferente, um número variável de semáforos, em mapas de cidades reais, situações estas mais próximas do cotidiano.

No caso do Cenário de Turim, apresentado na Subseção 3.4.3.1, é possível verificar as rotas dos EVs na Figura 3.6, repetida neste Capítulo para comodidade do leitor. Os detalhes de comprimento são reapresentados a seguir, na Tabela 3.3. É possível notar que o EV2 percorre uma rota mais retilínea e passa por um número menor de semáforos quando comparamos com as rotas do EV1 e EV3. Isso influencia no tempo perdido pelo EV, mesmo quando nenhum algoritmo de preempção é utilizado.

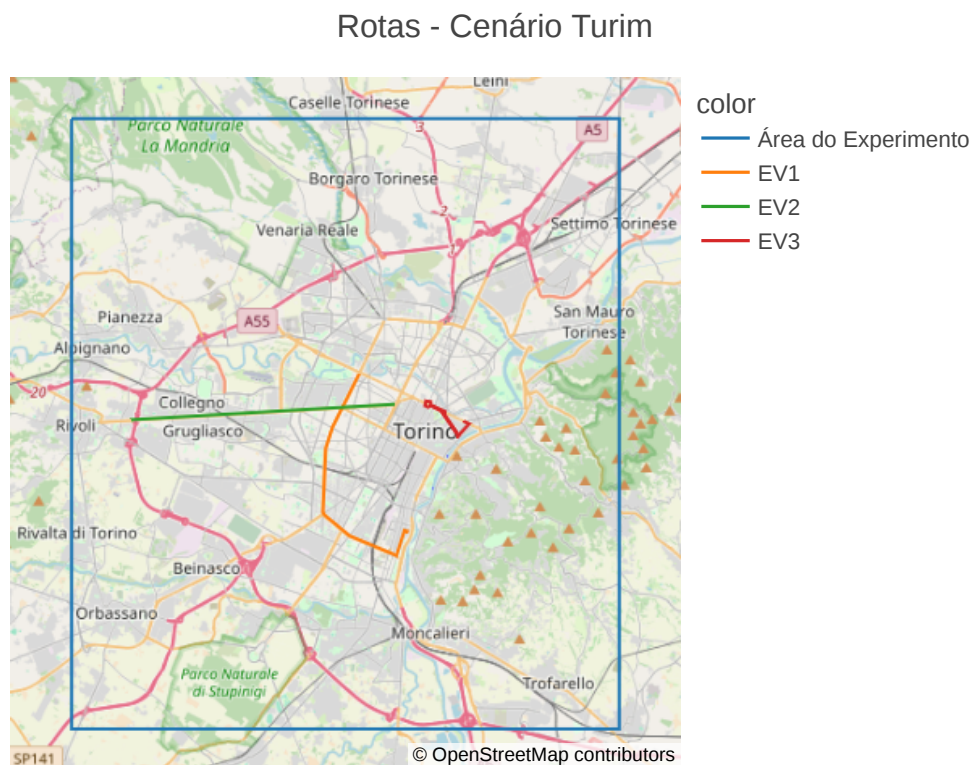


Figura 3.6: Rotas dos EVs dentro da área do experimento do Cenário Turim

Na avaliação utilizando o cenário com as demandas reais e sintéticas de Turim, a Figura 5.9, Tabela A.20, Figura 5.10 e Tabela A.21 apresentam o

Tabela 3.3: Parâmetros das rotas dos EVs - Cenário Turim

EV	Distância	Semáforos
EV1	9.961,68m	35
EV2	9.952,10m	19
EV3	5.300,60m	33

tempo perdido pelos EVs na versão sem preempção. É possível notar que não há correlação direta entre a distância percorrida e o número de semáforos cruzados com o tempo perdido pelos EVs, pois as condições do tráfego também influenciam esses valores.

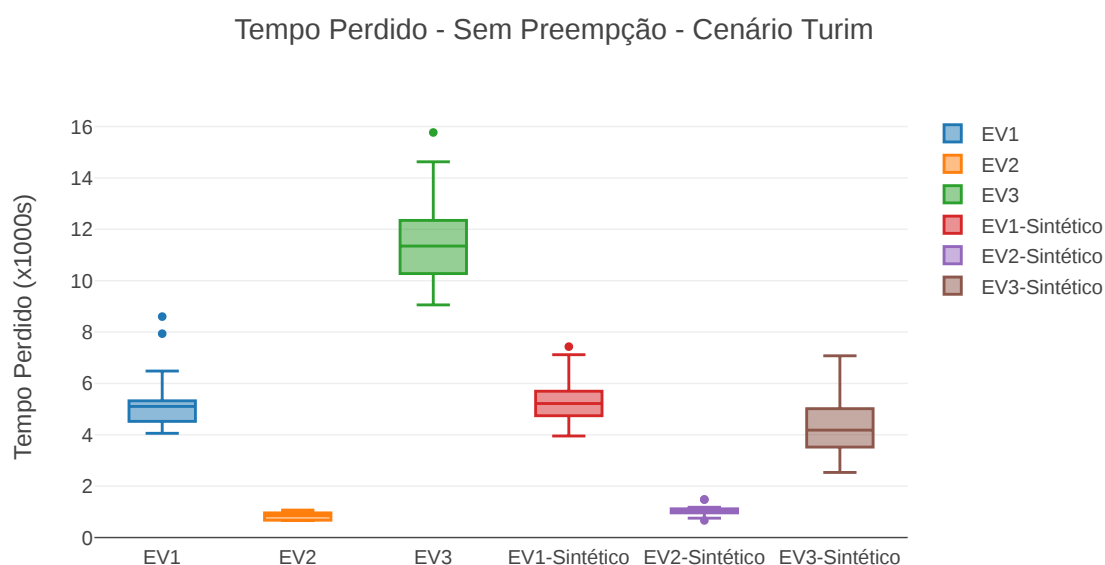


Figura 5.9: Tempo Perdido (s) - Versão Sem Preempção - Cenário Turim

O fato do tempo perdido pelo EV1 ser maior que o do EV2 nesse caso é esperado, mas a diferença entre eles não se explica apenas com esses valores. Além disso, o EV3 percorre uma distância menor, aproximadamente 5 mil e 300 metros, e cruza quase a mesma quantidade de semáforos que o EV1, no caso, 33. Ainda assim, seu tempo perdido é muito maior que o EV1 para a demanda real. É curioso notar que, para a demanda sintética, o do EV3 apresentou menor tempo perdido do que o EV1. E em relação aos EV1 e EV2, comparando as demandas reais e sintéticas, os resultados foram semelhantes.

Com exceção de EV1 e EV1-Sintético, o Algoritmo proposto é consideravelmente superior aos demais algoritmos, inclusive obtendo melhores resultados que o *Algoritmo Onda Verde* em alguns casos, com melhorias chegando até a 88,33% em sua mediana (Figura 5.11). Esse desempenho superior ao *Algoritmo Onda Verde* também explica o motivo do EV1 obter resultados não satisfatórios: é possível que a preempção esteja ativa, mas complicações no trânsito

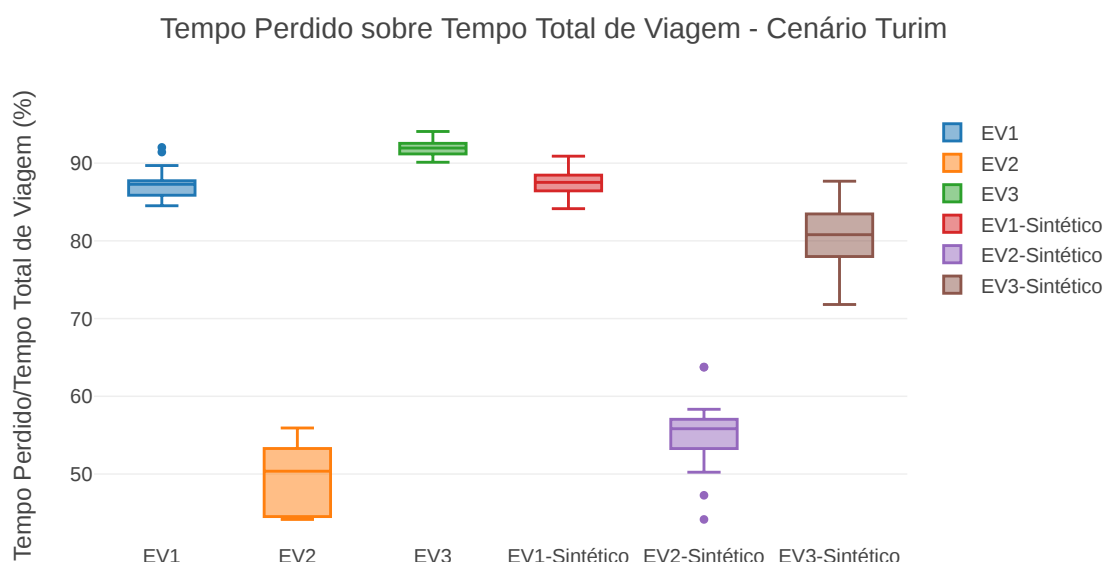


Figura 5.10: Tempo Perdido sobre o Tempo Total de Viagem (%) - Versão Sem Preempção - Cenário Turim

(congestionamento, acidentes etc.) possam impedir o bom andamento dos veículos.

Neste caso, em todos os algoritmos, manter a preempção pode piorar a situação. Para o tempo de preempção médio (Figura 5.12), o Algoritmo proposto apresenta valores mais baixos quando comparados ao Algoritmo (KAPUSTA et al., 2017). E, como esperado, como os semáforos ficam todos ativos e permanecem assim até o final, para o *Algoritmo Onda Verde* o *Tempo de Preempção Médio* se mostra proibitivo em alguns casos¹. Os valores do experimento podem ser vistos nas Tabelas A.22, A.23, A.24, A.25, A.26, A.33, A.34, A.35, A.36 e A.37.

A Figura 5.13 apresenta o tempo perdido (s) (quanto menor, melhor) e as Tabelas A.27, A.28, A.29, A.30 e A.31 são os dados correspondentes. Complementar à Figura 5.11, é possível notar que o EV1 possui sua mediana acima de um, mas seus valores de Q1 apresentam valores próximos de zero ou negativos. Para o restante dos EVs, os resultados foram positivos e alguns superiores ao algoritmo Onda Verde, com medianas apresentando resultados entre 6,53 e 8,57 vezes.

A Figura 5.14 apresenta o tempo perdido (s) (quanto menor, melhor) e as Tabelas A.38, A.39, A.40, A.41 e A.42 são os dados correspondentes. Analisado em conjunto com a Figura 5.11 e 5.13, novamente, é possível observar

¹Para navegar pelos gráficos deste trabalho e verificar outros valores que não as medianas, acesse https://nbviewer.org/github/rodrigogbranco/charts/blob/main/kapustaimp-paper.ipynb?flush_cache=true passando o mouse por cima dos *boxplots*

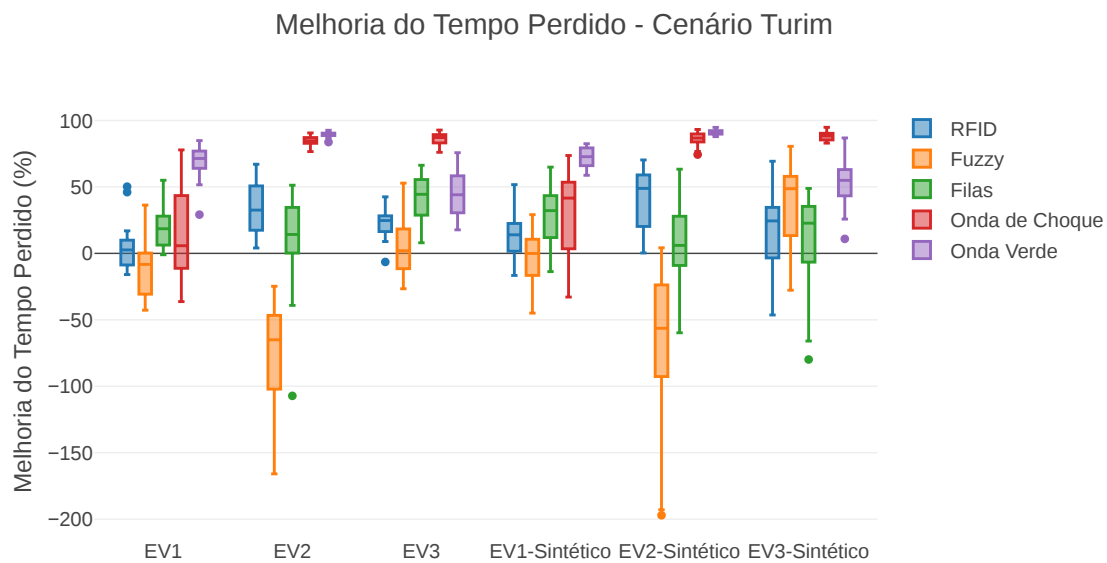


Figura 5.11: Melhoria do Tempo Perdido (%) - Cenário Turim

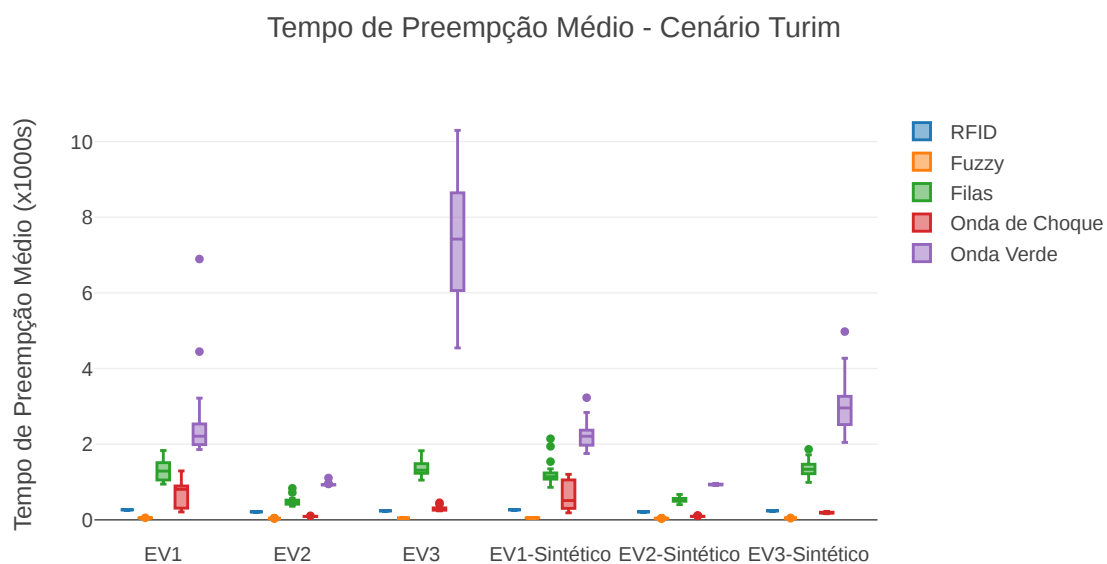


Figura 5.12: Tempo de Preempção Médio (s) - Cenário Turim

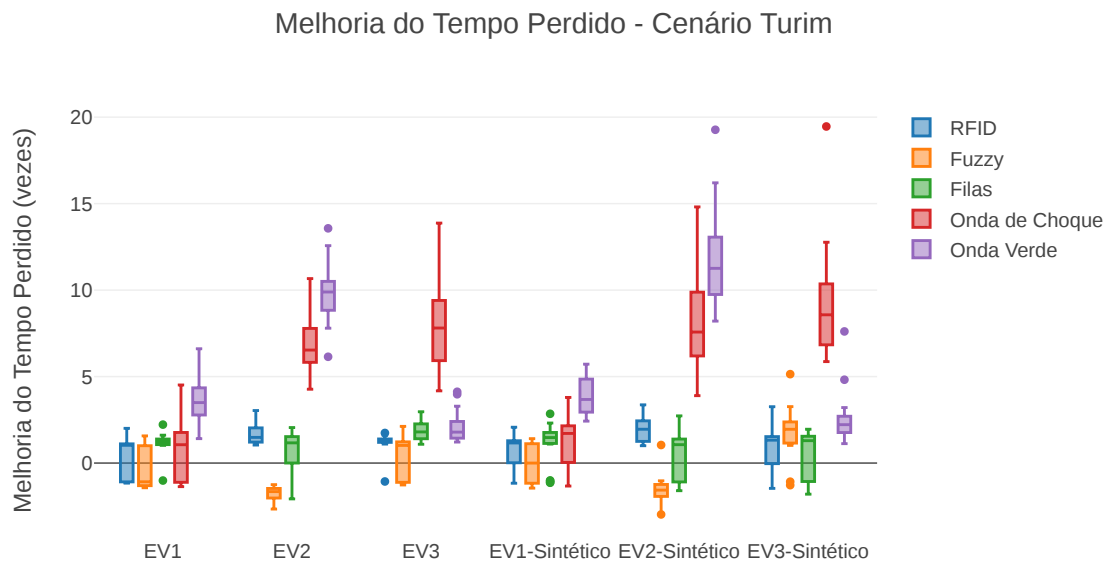


Figura 5.13: Melhoria do Tempo Perdido (vezes) - Cenário Turim

os maus resultados obtidos para o EV1. Em compensação, para o EV3, o tempo perdido obtido é inferior a todos os outros algoritmos, inclusive para o algoritmo Onda Verde. Ainda assim, é interessante observar como valores baixos são relativos ao contexto, já que baixo para o EV3 com dados reais é equivalente à sua mediana, com 1.556,25 segundos, o que resulta em aproximadamente 25 minutos de tempo perdido.

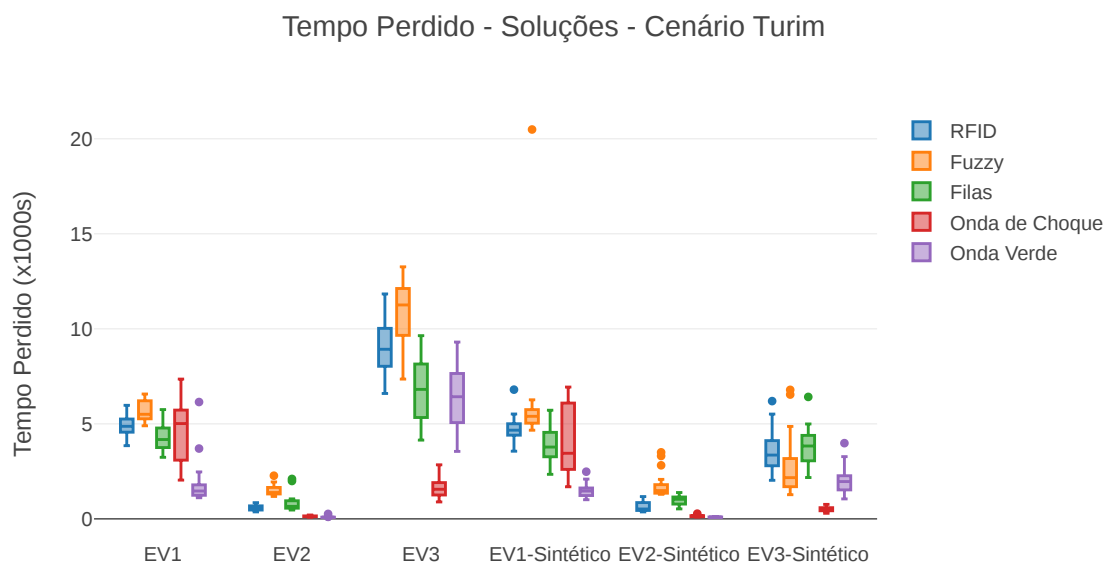


Figura 5.14: Tempo Perdido (s) - Algoritmos - Cenário Turim

Como foi possível observar, mesmo sem utilizar nenhum algoritmo de pre-

empção, o fato de termos rotas diferentes, com comprimentos variados, e condições de tráfego diferentes resulta em um tempo perdido pelo EV que não é diretamente comparável entre si. Tais valores devem ser usados como linha de base para os algoritmos de preempção, tendo assim uma forma objetiva de se verificar a efetividade dos algoritmos. Assim, foi possível observar que, na maioria dos casos, o Algoritmo Onda de Choque apresenta um desempenho superior aos outros algoritmos considerados para o Cenário de Turim.

5.4.3 Cenário Colônia

De forma análoga ao Cenário de Turim, o Cenário de Colônia apresenta rotas e condições de tráfego obtidos de medições realizadas *in loco* pelos criadores do cenário e, portanto, permite situações variadas para a avaliação dos algoritmos de preempção. A Figura 3.7 e a Tabela 3.4 foram apresentadas na Subseção 3.4.3, mas, para a comodidade do leitor, são novamente exibidas neste Capítulo. As mesmas considerações feitas para o Cenário de Turim são válidas para este Cenário, no que diz respeito ao comprimento de rota, número de semáforos cruzados, condições do tráfego, entre outras.

Rotas - Cenário Colônia

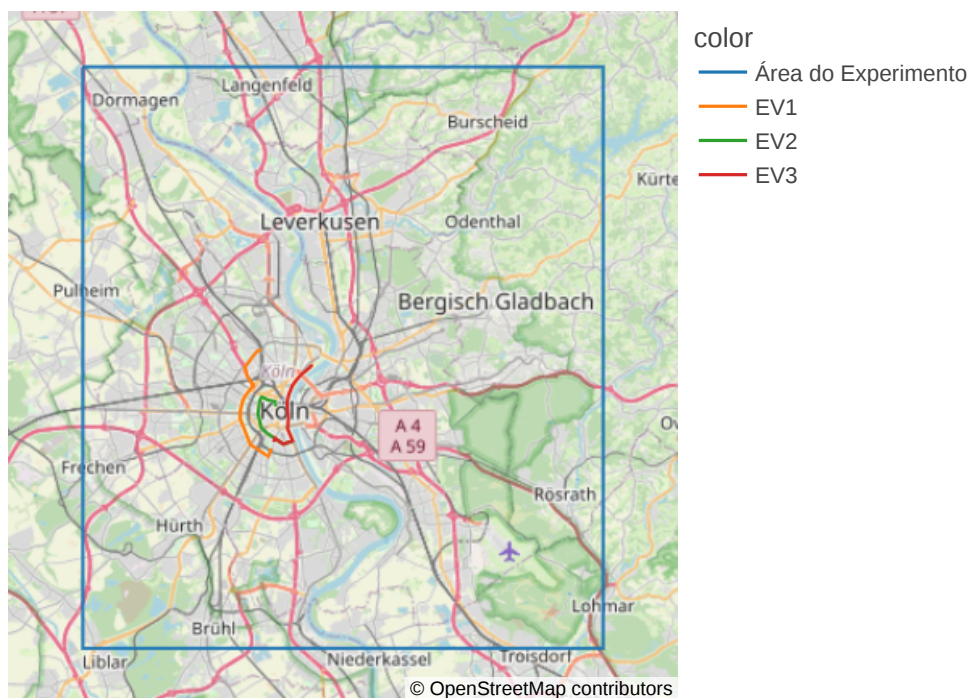


Figura 3.7: Rotas dos EVs dentro da área do experimento do Cenário Colônia

A Figura 5.15, Tabela A.43, Figura 5.16 e Tabela A.44 apresentam os dados de tempo perdido para a versão sem preempção. Os EV1, EV2 e EV3 cruzam quase a mesma quantidade de semáforos, mas percorrendo distâncias diferentes em suas rotas. Ainda assim, as medianas variaram bastante,

Tabela 3.4: Parâmetros das rotas dos EVs - Cenário Colônia

EV	Distância	Semáforos
EV1	8.351,42m	26
EV2	5.044,34m	27
EV3	6.344,31m	24

e as diferenças entre o EV1 e sua contraparte sintética chamam bastante a atenção, já que a única diferença para esses dois casos é a demanda. Neste caso, enquanto o EV1 com demandas reais obteve 10.369,03 segundos de tempo perdido, para o EV1 o resultado foi de 728,66 segundos. A consequência disso é que a proporção de tempo perdido, com relação ao tempo total de viagem para estes dois EVs, é de 95,12% e 57,83% para suas medianas.

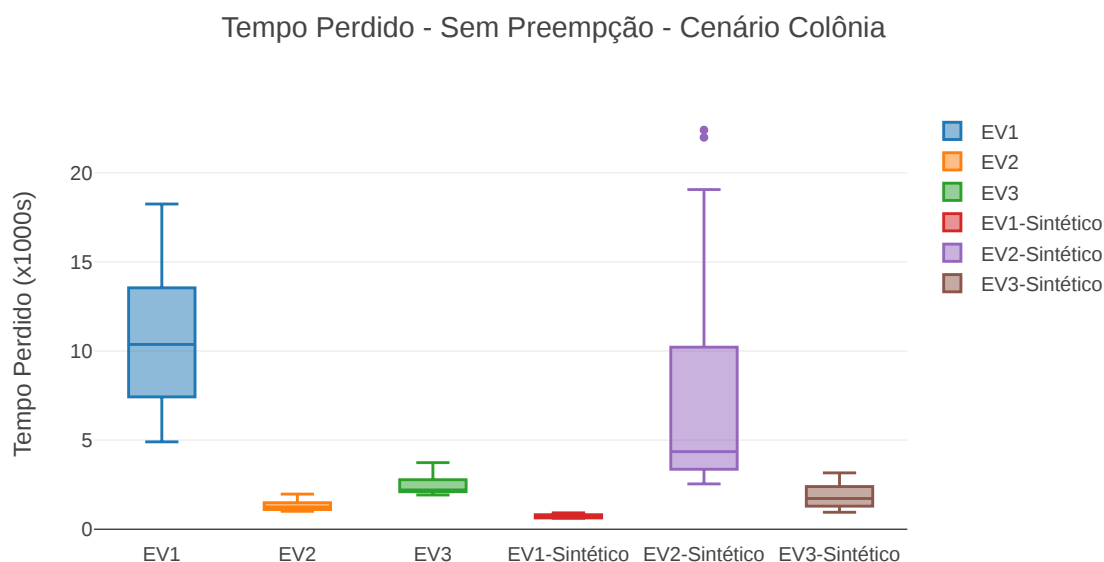


Figura 5.15: Tempo Perdido (s) - Versão Sem Preempção - Cenário Colônia

O Algoritmo Proposto também obteve bons resultados em comparação aos outros algoritmos, com melhorias atingindo 96,98% para a sua mediana (Figura 5.17). A exceção nesse cenário é para o EV3, cuja mediana é de 45,88% (o segundo maior valor), mas inferior ao Algoritmo RFID, cuja mediana apresenta melhoria de 58,17%. Contudo, neste cenário os Algoritmos 3.3 e Onda Verde obtiveram resultados negativos. Nem sempre, portanto, é útil manter a preempção ativa, sendo imperativo que os algoritmos detectem e evitem tais situações. Do ponto de vista do Tempo de Preempção Médio (Figura 5.18), o Algoritmo Proposto apresenta baixos valores, reforçando o custo-benefício em sua utilização, quando comparado com os Algoritmos concorrentes. Os valores obtidos neste experimento podem ser encontrados nas Tabelas A.45, A.46, A.47, A.48, A.49, A.50, A.51, A.52, A.53 e A.54.

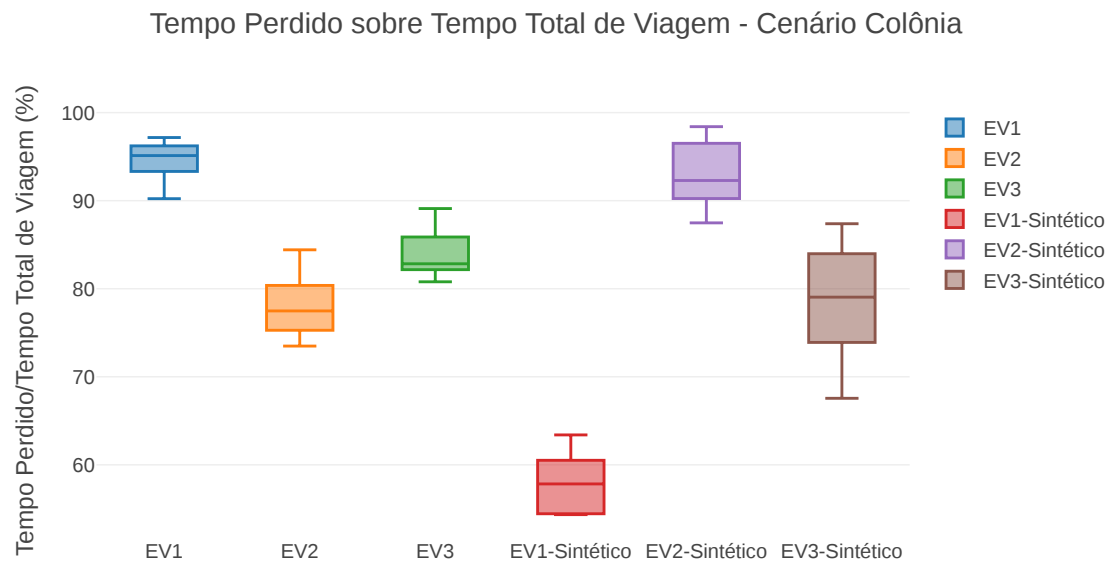


Figura 5.16: Tempo Perdido sobre o Tempo Total de Viagem (%) - Cenário Colônia

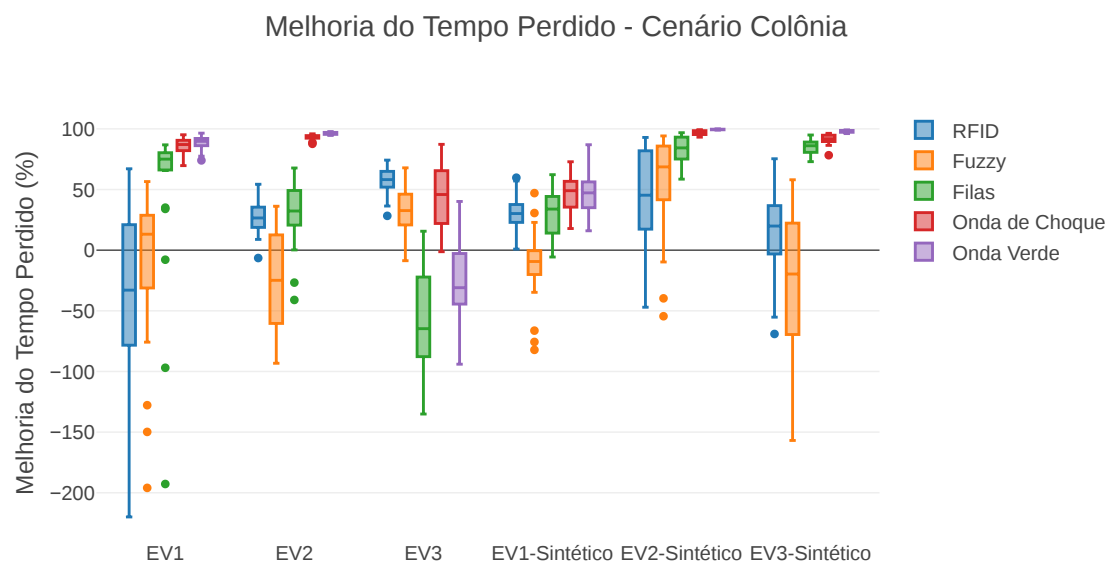


Figura 5.17: Melhoria do Tempo Perdido (%) - Cenário Colônia

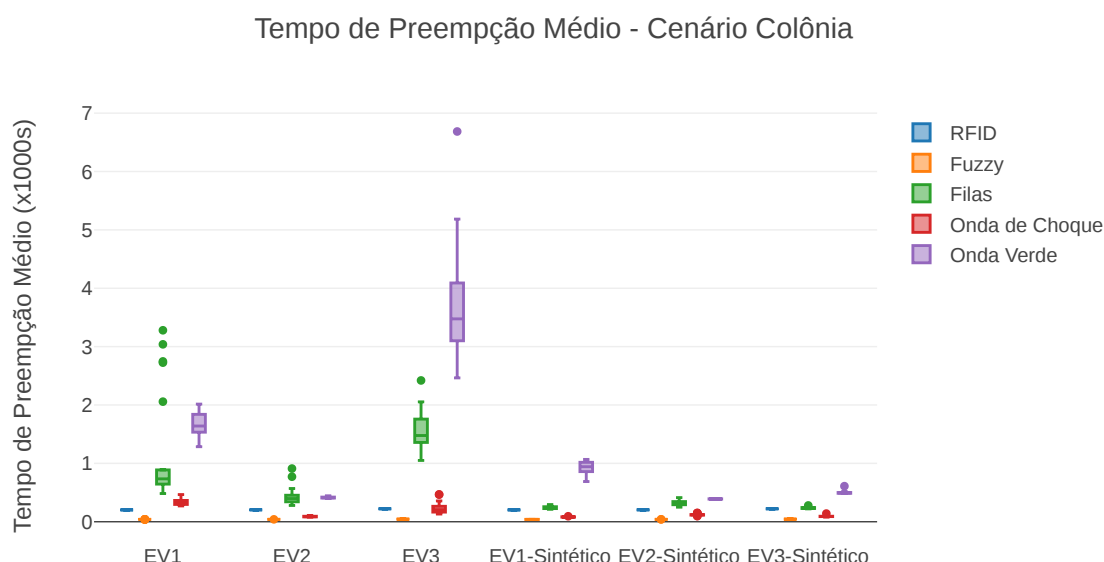


Figura 5.18: Tempo de Preempção Médio (s) - Cenário Colônia

A Figura 5.19 apresenta o tempo perdido (s) (quanto menor, melhor) e as Tabelas A.55, A.56, A.57, A.58 e A.59 são os dados correspondentes. Complementar à Figura 5.17, os dados de melhoria do algoritmo Onda Verde, para o EV2 sintético, dificultam a visualização dos outros cenários e algoritmos, pois ele destoa demais neste caso, obtendo melhoria de 212 vezes para a sua mediana. Ao se aproximar dos dados, deixando a escala entre 0 e 35 vezes, é possível notar que o algoritmo proposto obteve bons resultados, com medianas variando entre 7,75, 15,02, 1,84 e 1,96 vezes, para os EVs 1, 2, 3 (demandas reais) e EV1 (demanda sintética).

A Figura 5.20 apresenta o tempo perdido (s) (quanto menor, melhor) e as Tabelas A.60, A.61, A.62, A.63 e A.64 são os dados correspondentes. Assim como a Figura 5.19, os maus resultados dos algoritmos RFID e Fuzzy para o EV1 dificultam a visualização dos outros cenários, com esses algoritmos obtendo 14.600,71 e 10.230,94 segundos, respectivamente. Ao se alterar a escala da Figura, é possível notar que o algoritmo proposto, comparativamente aos outros algoritmos para as mesmas instâncias, obtém os menores valores de tempo perdido, com valores superiores apenas ao algoritmo Onda Verde, o que é esperado.

É possível navegar interativamente nos gráficos apresentados para verificação dos resultados². Os resultados deste experimento foram publicados em Branco, Ferreira e Cáceres (2022).

²https://nbviewer.org/github/rodrigobranco/charts/blob/main/tese_2025-06-28-segundo-experimento-app.ipynb

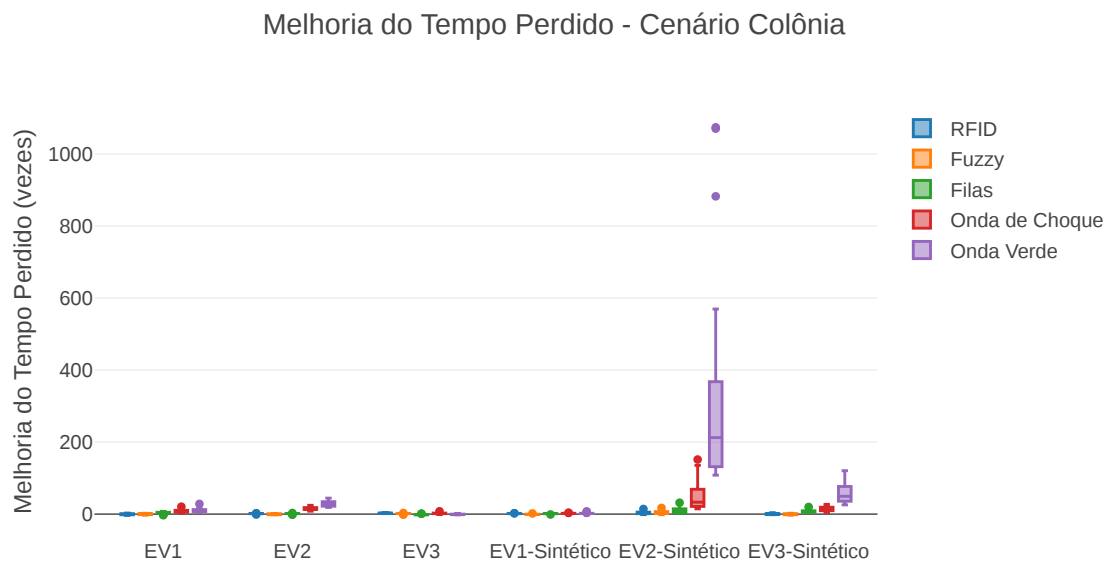


Figura 5.19: Melhoria do Tempo Perdido (vezes) - Cenário Colônia

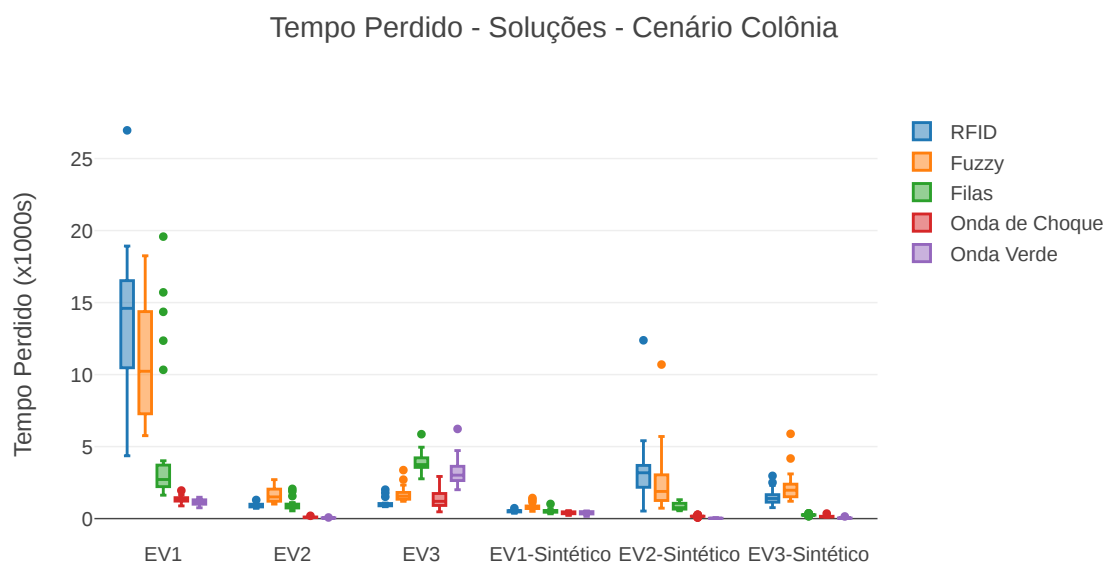


Figura 5.20: Tempo Perdido (s) - Algoritmos - Cenário Colônia.

5.5 Estudo das Anomalias

Embora a melhoria alcançada pelo Algoritmo Onda de Choque 5 seja significativa na maioria dos cenários e instâncias para a mediana, algumas instâncias apresentam valores negativos para o percentual de tempo perdido, nos casos de EV1 e EV1-Sintético do cenário de Turim e do EV3 do cenário de Colônia. O fato de os gráficos relacionados ao tempo de preempção médio apresentarem valores mais altos para essas instâncias colocou um alerta nesta situação.

Por este motivo, reexecutamos algumas dessas instâncias utilizando a interface gráfica do SUMO (SUMO-GUI) para averiguar a razão de tal comportamento. Para isso, um *Tracking* foi colocado no EV para verificar em quais situações essa instância estava pior que sua versão equivalente sem preempção.

Nesse tipo de comparação, foi possível notar que existe uma situação em que a execução da preempção, na verdade, atrapalha o tráfego do EV, ao invés do senso comum de que a abertura do semáforo ajudaria. E isso ocorre quando a preempção está ativa, mas o veículo líder do pelotão, que está na frente da fila de veículos, não pode se mover, mesmo que o semáforo esteja verde para ele.

Em uma simulação como a do SUMO, pode haver várias situações para isso ocorrer, como um *deadlock* inconciliável na rede, ou os ciclos do semáforo não estão bem projetados de modo que essa abertura do semáforo bloqueia este líder. Mas a situação mais comum, e é a que ocorre também na vida real, acontece quando a preempção está ativa, mas o líder não pode prosseguir porque a próxima aresta de sua rota está saturada. Sem ter para onde se mover, este veículo permanece onde está. Inclusive, esta é uma das situações em que nem sempre é possível que os veículos à frente do EV consigam ajudar saindo de sua frente, simplesmente por não haver espaço para tal manobra.

Para confirmar tal hipótese, criamos um cenário simples, com dois semáforos, mas apenas o primeiro é controlado pelos algoritmos de preempção, já que o EV encerra sua rota antes de alcançar o segundo semáforo. Neste caso então, o EV percorre quinhentos metros até o semáforo e mais quinhentos após cruzar a interseção, totalizando um quilômetro. Além disso, diminuímos a velocidade das vias de saída do semáforo controlado pelo algoritmo de preempção, para causar uma certa “resistência” na rede, que simularia a redução de velocidade presenciada pelos veículos quando o trânsito começa a ficar congestionado. Para finalizar, a interseção semafórica controlada pelo algoritmo possui fases como as da Figura 5.21. Isso significa que há apenas uma fase em que os veículos à frente do semáforo podem trafegar; caso essa

dias de todos os veículos que passaram pela simulação, dado que $atsb$ é a média das velocidades médias dos veículos no cenário sem preempção e $atsa$ é a média das velocidades médias dos veículos no cenário com preempção, o percentual de melhoria (ou piora) (%) da média das velocidades médias de todos os veículos é igual a $atsi$ e é dado pela Equação 5.2.

$$atsi = 100 * \left(\frac{atsa}{atsb} - 1 \right) \quad (5.2)$$

O valor desta métrica entregue pelo simulador em um cenário real com uma rede grande o suficiente não muda significativamente entre as versões sem e com preempção. Mas não porque o impacto não é significativo, mas sim porque a proporção de veículos que não participam da preempção é muito maior do que os que participam, e o simulador não faz distinção entre eles. Na realidade, é difícil delimitar quais veículos são realmente impactados pela preempção, já que não se pode considerar apenas os que estão nas arestas dos semáforos controlados pelos algoritmos. Qual o raio do impacto da preempção e quantas arestas partindo do semáforo controlado devem ser consideradas? As respostas para estas perguntas fogem do escopo do presente trabalho e podem ser consideradas em trabalhos futuros. De toda sorte, essa métrica é usada neste cenário controlado e não nos demais.

A segunda métrica é a média do tempo perdido médio de todos os veículos (*avg trip timeloss*). Considerando a média do tempo perdido de todos os veículos que passaram pela simulação, dado que $attb$ é a média do tempo perdido de todos os veículos no cenário sem preempção e $atta$ é a média do tempo perdido de todos os veículos no cenário com preempção, o percentual de melhoria (ou piora) (%) da média do tempo perdido de todos os veículos é igual a $atti$ e é dado pela Equação 5.3.

$$atti = 100 * \begin{cases} \left(1 - \frac{atta}{attb} \right), & \text{se } atta < attb \\ \left(\frac{atta}{attb} - 1 \right) * -1, & \text{caso contrário} \end{cases} \quad (5.3)$$

A terceira e última métrica se refere aos veículos teletransportados. Se o algoritmo de preempção está bloqueando o veículo líder do pelotão, e o mesmo algoritmo não vai alterar esse estado em decorrência dessa constatação, então a única forma da simulação prosseguir é se esse veículo bloqueante for removido da preempção via teletransporte. Assim como as outras métricas anteriores, dado tel_b o número de veículos teletransportados no cenário sem preempção e tel_a o número de veículos teletransportados no cenário com preempção, o percentual de melhoria (ou piora) (%) de veículos teletransportados

em virtude da utilização do algoritmo de preempção é igual a tel_{imp} e é dado pela Equação 5.4.

$$tel_{imp} = 100 * \begin{cases} (1 - \frac{tel_a}{tel_b}), \\ \text{se } tel_a < tel_b \\ (\frac{tel_a}{tel_b} - 1) * -1, \\ \text{caso contrário} \end{cases} \quad (5.4)$$

Tais métricas são usadas apenas neste experimento controlado. Fora deste ambiente, devido ao tamanho dos cenários e à dificuldade em se identificar quais veículos, que não o EV, são afetados pela preempção, a obtenção dos valores de tais métricas fica prejudicada.

Agora que as métricas apropriadas foram apresentadas, é possível verificar o impacto de um algoritmo de preempção que não é interrompido. A Figura 5.22 apresenta o quanto o Algoritmo Proposto piorou o tempo perdido do EV quando comparado ao cenário sem preempção. Esta é a métrica já usada nos cenários anteriores. Neste cenário, a mediana apresenta o valor de -473,31%, com o IQR de 85,41. A variação total dos dados neste cenário varia entre -400,15% e -608,74%, como pode ser vista na Tabela A.65.

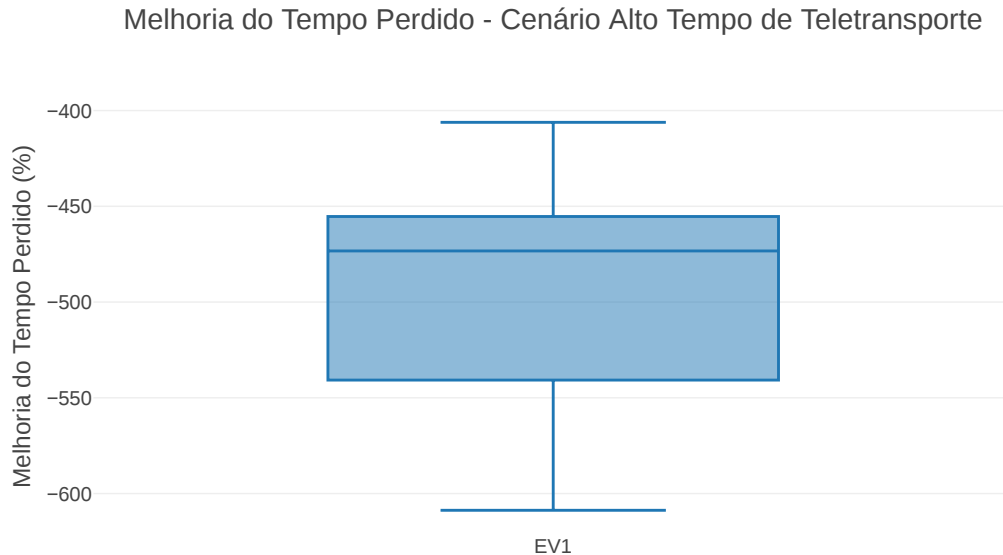


Figura 5.22: Melhoria do Tempo Perdido (%) - Cenário Alto Tempo de Teletransporte

Mas os resultados ruins não param no tempo perdido apenas pelo EV. Para todos os outros veículos, o tempo perdido médio (Equação 5.3) é apresentado na Figura 5.23. A mediana apresenta valor de -302,42%, com o IQR de 23,27. Os valores para este cenário variam entre -281,92% e -344,50%, como pode

ser visto na Tabela A.66.

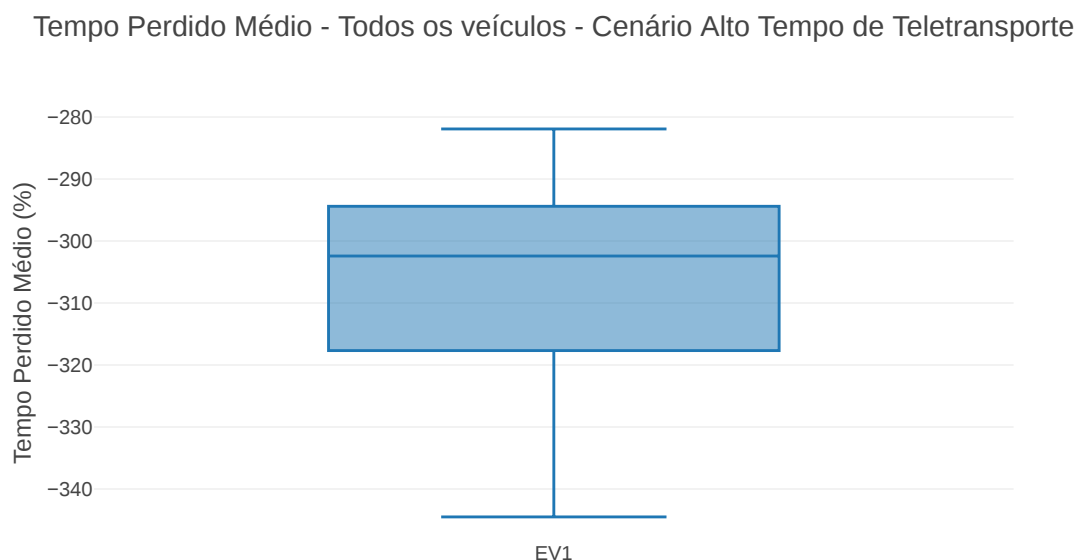


Figura 5.23: Tempo Perdido Médio (%) - Todos os Veículos - Cenário Alto Tempo de Teletransporte

Em complemento à Figura 5.23, a Figura 5.24 apresenta o percentual de melhoria da média da velocidade média de todos os veículos (Equação 5.3). A mediana apresenta o valor de -54,28%, com IQR de 1,42. Os valores permaneceram na faixa entre -54,28% e -51,42%, conforme apresentados na Tabela A.67.

Por fim, a Figura 5.25 apresenta o percentual de melhoria (ou piora) do número de veículos teletransportados quando os cenários sem e com preempção são comparados, conforme Equação 5.4. Com mediana de -1832% e IQR de 193,25 e percentuais variando entre -1.656% e -2.132% (conforme Tabela A.68), sustentar um algoritmo de preempção quando veículos estão bloqueados por ele não é uma situação desejável. Aqui cabem algumas considerações. Primeiro, o percentual apresentado não tem valor definido para o tempo de teletransporte dos veículos, mas sim para o número de veículos afetados pelo bloqueio. Contudo, esse valor, obviamente, reflete no tempo perdido pelo EV e pelos outros veículos.

A métrica da Equação 5.4 é pouco útil para outros cenários, ou até mesmo na vida real, pois essa situação de teletransporte não ocorre. Até entre outros simuladores, o comportamento desse procedimento de teletransporte pode variar. Por exemplo, o simulador VISSIM possui um parâmetro chamado *Waiting time before diffusion (s)*, que é a quantidade de tempo que um veículo pode esperar quando este faz uma parada de emergência para poder trocar de faixa. Se este *threshold* for alcançado, o veículo é removido da simulação, o que pode

Velocidade Média das Viagens - Todos os veículos - Cenário Alto Tempo de Teletransporte

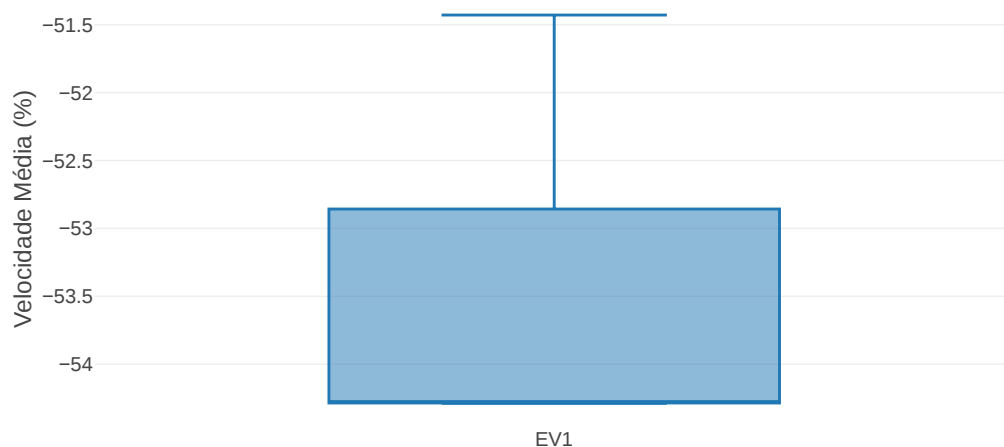


Figura 5.24: Velocidade Média das Viagens (%) - Todos os Veículos - Cenário Alto Tempo de Teletransporte

Veículos Teletransportados - Alto Tempo de Teletransporte

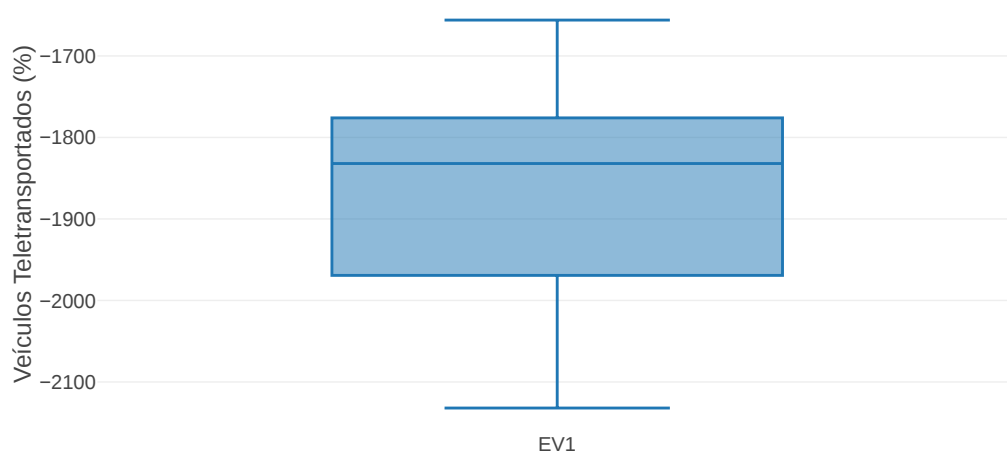


Figura 5.25: Veículos Teletransportados (%) - Cenário Alto Tempo de Teletransporte

ser interpretado como o teletransporte do SUMO³. Contudo, os simuladores tentarão encontrar formas de remover veículos problemáticos para que a simulação possa continuar.

É possível navegar interativamente nos gráficos apresentados para verificação dos resultados⁴.

5.6 Conclusão

Independentemente das diferenças entre os simuladores, é importante notar que o algoritmo de preempção pode bloquear os veículos e, até onde se pode dizer, não foram encontrados trabalhos na literatura que indiquem como esse problema pode ser detectado e solucionado.

Apesar da melhoria significativa quando comparado com a implementação do Algoritmo Kapusta et al. (2017), o Algoritmo Onda de Choque, apresentado neste capítulo, contém falhas que precisam ser corrigidas.

O Algoritmo Onda de Choque não incorpora mecanismos de segurança. Por exemplo, podemos citar o comportamento indefinido que ocorre quando a preempção é disparada erroneamente duas vezes no algoritmo, ou, pior ainda, quando o congestionamento é tão grande que pode causar *starvation* em veículos concorrentes, e quando um desses é um veículo de emergência, a situação pode ser agravada.

Portanto, esta é a oportunidade de incorporar as propriedades matemáticas da Rede Petri temporizada da primeira solução, para que mecanismos de segurança sejam assim considerados e deixem a solução final mais robusta. Consideraremos tais aspectos no próximo capítulo.

³Parâmetros de calibragem do VISSIM: <<https://wisconsin.gov/dtsdManuals/traffic-ops/manuals-and-standards/teops/16-20att6.3.pdf>>

⁴<https://nbviewer.org/github/rodrigogbranco/charts/blob/main/tese_2025-06-28-highteleporttime-app.ipynb>

Terceira Solução - Onda de Choque + TPN*

Neste capítulo, apresentamos a terceira solução desenvolvida neste trabalho, denominada Onda de Choque + TPN*, que combina a eficiência da dissipação de filas baseada no princípio da onda de choque com as garantias de segurança de uma Rede de Petri Temporizada aprimorada. Inicialmente, apresentamos uma visão geral da solução e sua respectiva motivação (Seção 6.1), e detalhamos o modelo de controle e a estrutura da TPN* proposta (Seção 6.2). Em seguida, demonstramos as propriedades formais da solução, assegurando sua corretude por meio das análises de *boundedness* e alcançabilidade (Seção 6.3). Posteriormente, descrevemos o algoritmo desenvolvido e sua complexidade (Seção 6.4), seguido de suas especificidades de implementação (Seção 6.5). Apresentamos, então, os resultados obtidos nos experimentos realizados com dados reais das cidades de Turim, Colônia e do Centro Expandido de São Paulo (Seção 6.6). Por fim, apresentamos as conclusões deste capítulo (Seção 6.7).

6.1 Visão Geral da Solução

Considerando as considerações do Capítulo 5, neste capítulo, a terceira solução é apresentada, partindo do Algoritmo Onda de Choque, mas voltando a adotar uma Rede Petri Temporizada modificada e algoritmos de Controle que asseguram tanto a segurança do trânsito de forma geral como a eficiência frente a outras soluções encontradas na Literatura. Como anteriormente, neste sistema de controle, o *Middleware* de cidades inteligentes recebe infor-

mação sobre a localização, velocidade e aceleração do EV, volume do tráfego das vias, estado e localização dos semáforos ao longo da rota do EV. O Algoritmo de Preempção altera a temporização das luzes do semáforo para garantir passagem ao EV quando este atinge as interseções temporizadas de sua rota.

A Rede Petri Temporizada foi definida formalmente na Seção 4.1, mantendo as propriedades mínimas de segurança e evitando situações conflitantes para os semáforos. O sistema precisa reagir quando eventos inesperados acontecessem e cancelar preempções vigentes e agendadas, e isso é alcançado utilizando a TPN que é construída neste capítulo. Assim como a solução apresentada no Capítulo 4, diferente de outros trabalhos (KAPUSTA et al., 2017), um dos *insights* desta solução é utilizar uma TPN principal para controlar toda a rota do EV, enquanto trabalhos anteriores modelam apenas uma interseção (HUANG; WENG; ZHOU, 2015).

O modelo da TPN' apresentado no Capítulo 4 foi aprimorado, contemplando simplificações e melhorias significativas, e esta nova TPN é agora definida como TPN*, diferenciando-a da anterior. Conforme apresentado pela Figura 4.2, a TPN' anterior fazia com que o *token* percorresse todos os blocos de controle sequencialmente. A TPN* proposta neste capítulo transfere o *token* para as posições em paralelo.

Assim, a rota é controlada pela Rede Petri (onde parte dela é apresentada na Figura 6.1b), e cada um dos semáforos possui uma rede Petri própria para controlar sua interseção, que será apresentada posteriormente. O bloco de controle é apresentado com mais detalhes na Seção 6.2. Uma preocupação crítica em qualquer sistema de controle de tráfego é a segurança de todos os veículos nas proximidades, e portanto isso deve ser considerado na solução proposta.

6.2 Modelo de Controle

A seguir são apresentadas as diferenças da TPN* descrita neste Capítulo, quando comparada à TPN' apresentada no Capítulo 4.3, além de fornecer explicações da lógica do fluxo de *tokens* dentro desse novo modelo.

Como vimos anteriormente, uma TPN' possui transições que são disparadas por eventos externos, e no caso deste modelo, uma TPN*, as transições que possuem esse comportamento são as transições t_{Cancel} e t_3 , que se referem a eventos externos de cancelamento e cruzamento do EV pelo semáforo, respectivamente.

A Figura 6.1b apresenta a visão global do Bloco de Controle do modelo proposto. Cada Bloco de Controle TL_i é modelado com uma TPN* apresentada na Figura 6.2. A transição t_{Start} na Figura 6.1b coloca um *token* na posição

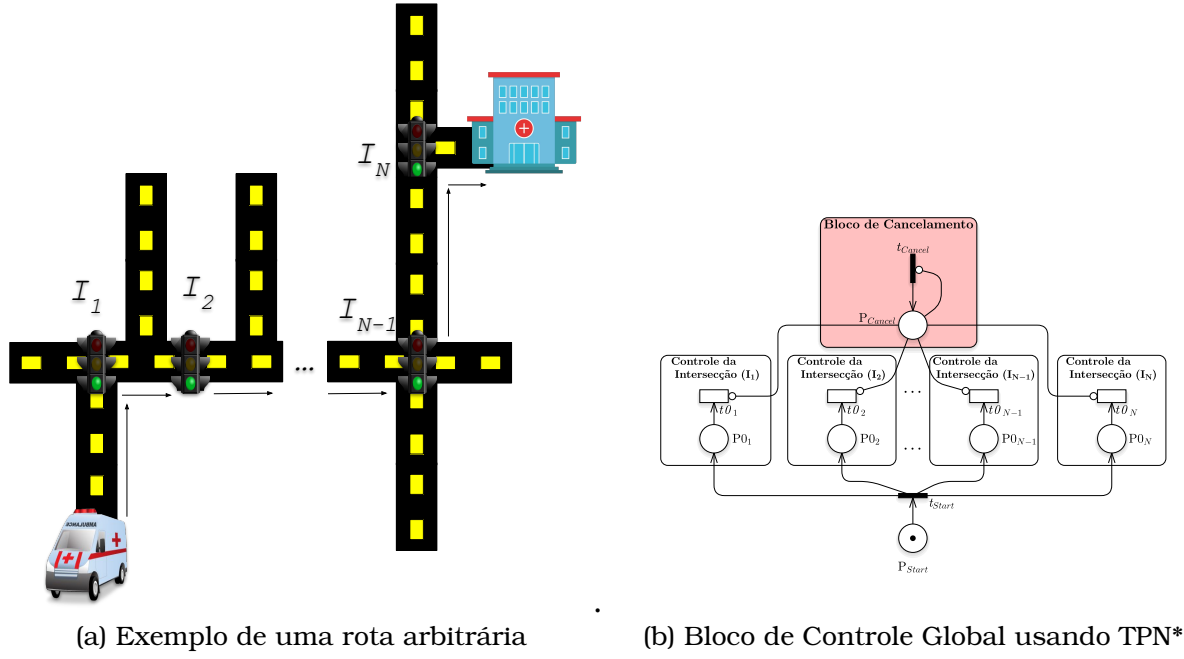


Figura 6.1: Exemplo de uma rota e o respectivo Bloco de Controle Global. O número de semáforos ao longo da rota do EV equivale ao número de Controles de interseção no Bloco de Controle Global.

P_0 de todos os Blocos de Controle TL_i ($1 \leq i \leq N$) no início da emergência. Por uma questão de simplicidade, TL_i é omitido nas posições e transições na Figura 6.2. Isto significa que cada interseção possui suas próprias posições e transições, com exceção da transição t_{Cancel} e da posição P_{Cancel} , que são globais e únicas para todo o modelo.

Uma posição pode ser associada a uma ação no mundo real, por exemplo, a posição P_1 dispara o início da preempção para a transição TL_i (isto é, garante que a luz esteja verde quando o EV se aproxima de TL_i) enquanto a posição P_4 restaura TL_i para sua operação normal (isto é, desabilita a preempção, resincronizando o semáforo), enquanto outras posições são usadas para controlar *tokens* e manter a TPN* em um estado seguro e estável. Por exemplo, um *token* na posição P_2 não dispara ação alguma, mas significa que a preempção foi iniciada e esta deve aguardar até que o EV cruze a interseção ou que o comando de cancelamento seja disparado.

Todas as transições da referida TPN* são imediatas (isto é, podem disparar a qualquer momento desde que esta esteja habilitada). A exceção é a transição t_0 , que é uma transição temporizada e já possui um tempo preciso de disparo no início da preempção. O objetivo dessa transição é limpar a fila de veículos em TL_i e para que o EV alcance a interseção sem perder tempo. A maneira como este tempo é calculado será apresentada posteriormente.

Quando a TPN* é construída, todos os tempos das transições t_0 de todos os TL_i são computados e definidos nas respectivas transições. A transição t_{Start}

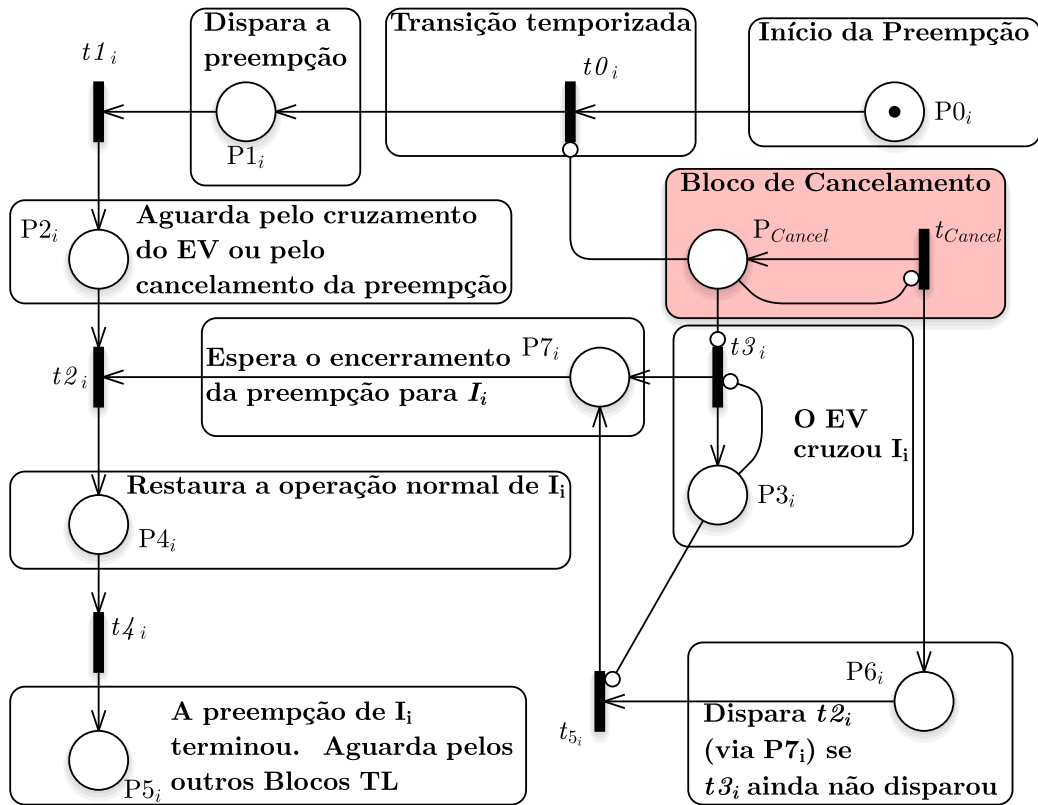


Figura 6.2: O modelo TPN* proposto. A região vermelha (Bloco de Cancelamento) é compartilhada por todos os blocos de controle de todos os TL_i , o restante é replicado para cada TL_i ($1 \leq i \leq N$).

da Figura 6.1b dispara, e um *token* é colocado na posição P_0 . É importante notar que, até este momento, nada de preempção é mencionado. Isso porque, neste modelo proposto, antes da preempção iniciar, há um estágio chamado “Preempção Antecipada”, que verifica se a fila de veículos de TL_i é longa o suficiente antes da preempção real iniciar. Se este for o caso, uma preempção curta temporizada é feita para suavizar a fila de veículos, preparando a situação para a preempção principal. Detalhes dessa operação serão fornecidos posteriormente.

Enquanto a transição t_0 não é disparada, seu tempo é continuamente atualizado, dependendo dos novos parâmetros do trânsito e do tamanho das filas dos semáforos. Quando a transição t_0 atinge seu tempo de disparo, esta transição é disparada e o *token* é movido para a posição P_1 , e a preempção principal é iniciada (Figura 6.3). A luz verde de TL_i é fornecida, a transição t_1 dispara e o *token* vai para a posição P_2 , enquanto este aguarda um próximo evento para poder prosseguir seu fluxo (Figura 6.4). Considerando uma operação normal, o evento esperado é o cruzamento do EV por TL_i , o que dispara a transição t_3 , que coloca um *token* na posição P_3 para prevenir futuros disparos incorretos da transição t_3 e coloca um *token* na posição P_7 (Figura 6.5).

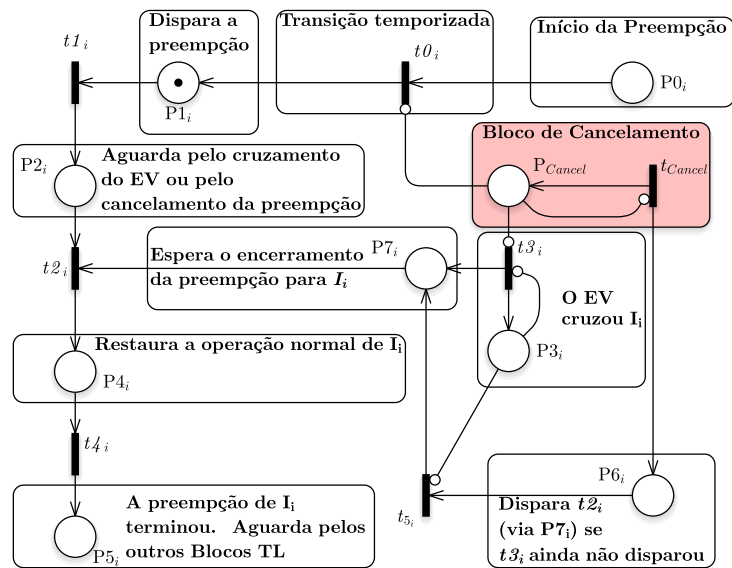


Figura 6.3: O disparo da transição t_0 remove o *token* da posição P_0 e o coloca na posição P_1 , iniciando a preempção para este semáforo.

Agora, os *tokens* nas posições P_2 e P_7 habilitam a transição t_2 . Esta transição é disparada, removendo os *tokens* das posições citadas e colocando um *token* na posição P_4 (Figura 6.6), significando que a preempção para este semáforo acabou e o mesmo é resincronizado com a sua temporização normal. Disparar a transição t_3 mesmo que o EV ainda não tenha cruzado o semáforo

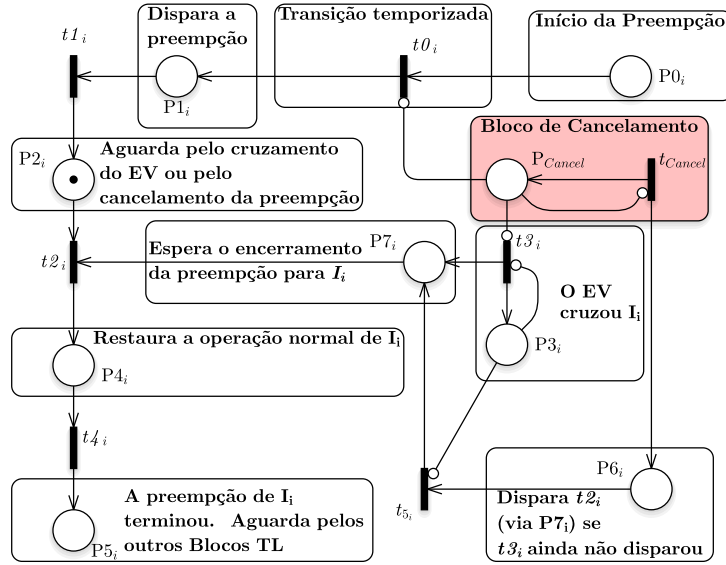


Figura 6.4: O disparo da transição t_1 remove o *token* da posição P_1 e o coloca na posição P_2 . Neste ponto, para que o fluxo da TPN* continue, é necessário ou que o EV cruze o semáforo ou que a preempção seja cancelada.

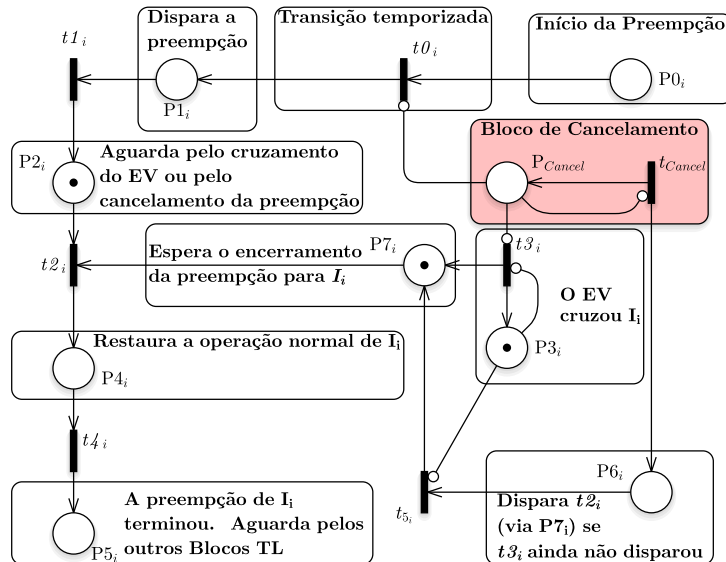


Figura 6.5: Quando o EV cruza o semáforo, a transição t_3 é disparada, colocando tokens nas posições P_3 e P_7 . Esse evento indica que o semáforo será ressincronizado, mas isso acontecerá quando o *token* estiver na posição P_4 .

força que este comportamento da TPN* descrito até o momento seja executado, tendo como resultado o cancelamento e a ressincronização deste semáforo em específico, mas mantendo a preempção dos demais semáforos. Depois disso, a transição t_4 dispara, e o *token* vai para a posição P_5 . A preempção para TL_i é, portanto, considerada completa, e basta aguardar que os outros Blocos de Controle dos demais semáforos finalizem sua lógica (Figura 6.7).

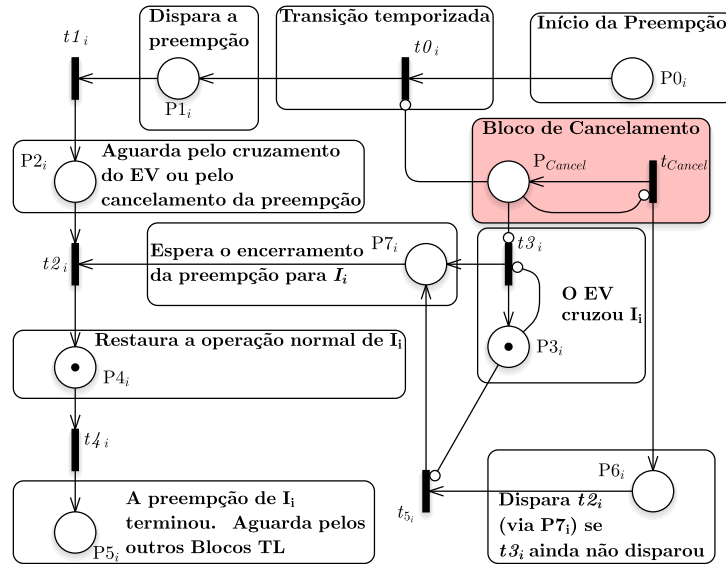


Figura 6.6: Com *tokens* nas posições P_2 e P_7 , a transição t_2 é habilitada. A transição t_2 é então disparada, o que remove os *tokens* das posições P_2 e P_7 e coloca um *token* na posição P_4 . Neste momento, o semáforo é ressincronizado.

Como observamos anteriormente, é possível que o sistema precise cancelar a preempção a qualquer momento porque o EV está muito lento ou até mesmo parado em virtude de um alto congestionamento. Manter a preempção nesta situação pode criar caos nas ruas vizinhas com longas filas de veículos parados no sinal vermelho. Quando o sistema determina que é necessário cancelar a preempção (por meio do Algoritmo descrito na Seção 6.4), ele dispara a transição t_{Cancel} , que coloca um *token* nas posições P_{Cancel} e P_6 (Figura 6.8). O estado da rede e o momento em que esta transição é disparada determinam como o sistema se comportará. Se a transição t_0 ainda não disparou, a aresta inibidora $P_{Cancel} \rightarrow t_0$ previne a transição t_0 de disparar. Isto significa que a preempção nunca inicia para TL_i neste caso. Embora a posição P_0 contenha um *token*, o Estágio de Preempção Antecipada é também finalizado nesta situação. O *token* da posição P_6 segue o caminho $P_6 \rightarrow t_5 \rightarrow P_7$, mas sem *token* na posição P_2 , nada mais é feito.

Se a transição t_0 já foi disparada quando a transição t_{Cancel} dispara, é necessário encerrar a preempção e restaurar o estado de TL_i imediatamente. Neste

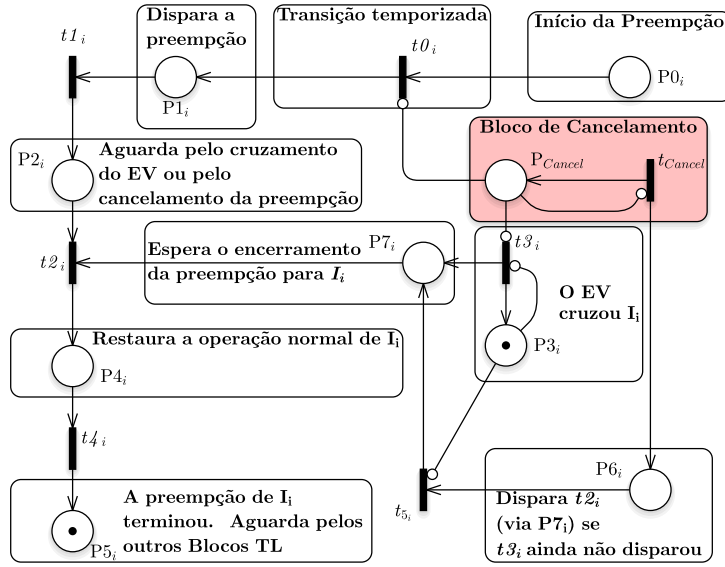


Figura 6.7: O disparo da transição t_4 remove o *token* da posição P_4 e o coloca na posição P_5 , finalizando a preempção para este semáforo.

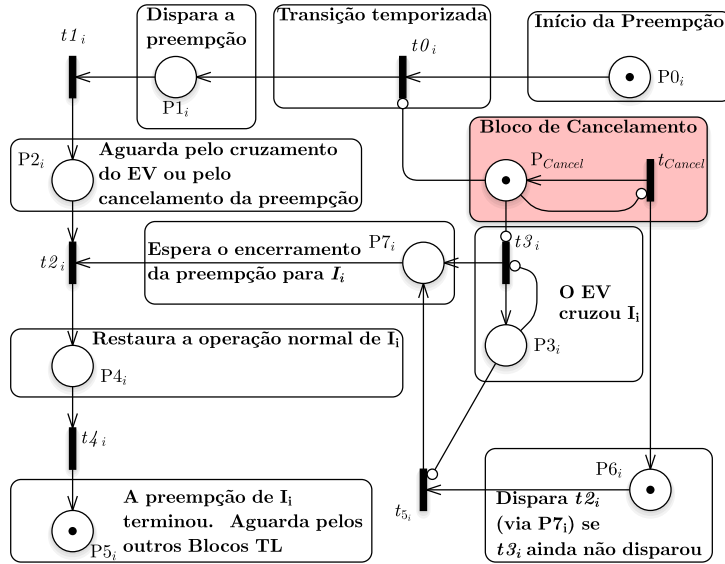


Figura 6.8: Quando a preempção é cancelada, a transição global t_{Cancel} é disparada, colocando um *token* nas posições P_{Cancel} e P_6 . Nesta Figura, há um *token* na posição P_0 , mas a preempção pode ser cancelada a qualquer momento, significando que o *token* pode estar nas posições P_0 , P_1 , P_2 , P_4 ou P_5 quando isso acontecer. Para o caso desta Figura, a posição P_{Cancel} inibe a transição t_0 , significando que a preempção não iniciará para este semáforo.

caso, o *token* está aguardando na posição P_2 (Figura 6.9). Como a situação considerada neste momento é a situação em que o EV ainda não cruzou TL_i , o caminho anterior $P_6 \rightarrow t_5 \rightarrow P_7$ usará o *token* da posição P_2 para disparar a transição t_2 (Figura 6.10). O restante do procedimento é igual, restaurando assim o semáforo ao seu funcionamento normal.

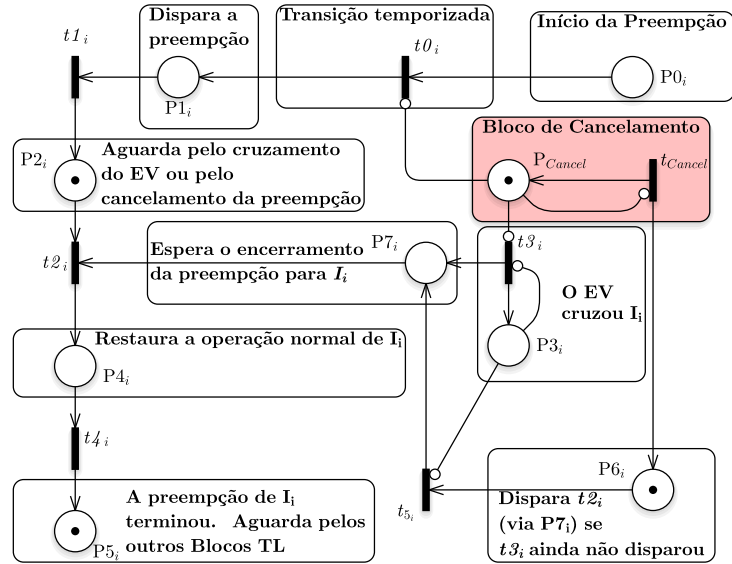


Figura 6.9: A preempção foi cancelada, então a transição global t_{Cancel} é disparada, colocando um *token* nas posições P_{Cancel} e P_6 . Contudo, a preempção já havia sido iniciada, então a transição t_0 já havia disparado, e o *token* passou pela posição P_1 e está na posição P_2 .

Neste modelo, não importa qual *token* é colocado na posição P_7 , seja pelo cruzamento do EV ou pelo comando de cancelamento. Na verdade, um inibe o outro: se a transição t_{Cancel} dispara primeiro, a transição t_3 não pode disparar devido à aresta inibidora $P_{Cancel} \rightarrow t_3$. Um *token* é colocado nas posições P_6 e P_7 logo em seguida. No caso em que a transição t_3 dispara primeiro, um *token* é colocado na posição P_3 e diretamente na posição P_7 . A transição t_{Cancel} ainda pode disparar (afinal de contas, essa é uma transição global e o sistema ainda está em funcionamento), mas o *token* na posição P_6 ficará preso lá, por causa da aresta inibidora $P_3 \rightarrow t_5$ (Figura 6.11). Apesar disso, tudo o que é necessário é o *token* na posição P_7 para que, se a preempção já iniciou, esse token habilite a transição t_2 para restaurar a operação normal de TL_i (Figura 6.12).

As operações aqui apresentadas mitigam as limitações de segurança e corrigem a impossibilidade de cancelamento da preempção do Algoritmo Onda de Choque. A próxima Seção apresenta o motivo de tais operações assegurarem as propriedades de segurança da TPN*.

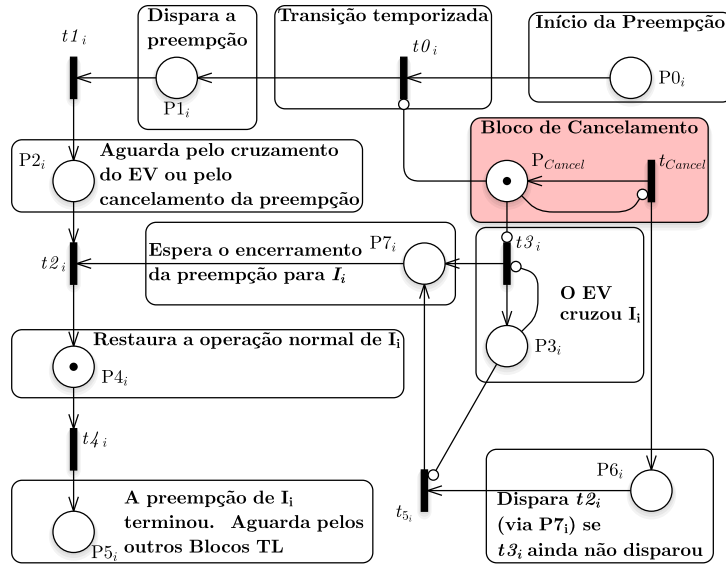


Figura 6.10: O *token* que estava na posição P_6 é colocado na posição P_7 pelo disparo da transição t_5 . O *token* nas posições P_7 e P_2 fazem disparar a transição t_2 , que os retira das respectivas posições e coloca o *token* na posição P_4 .

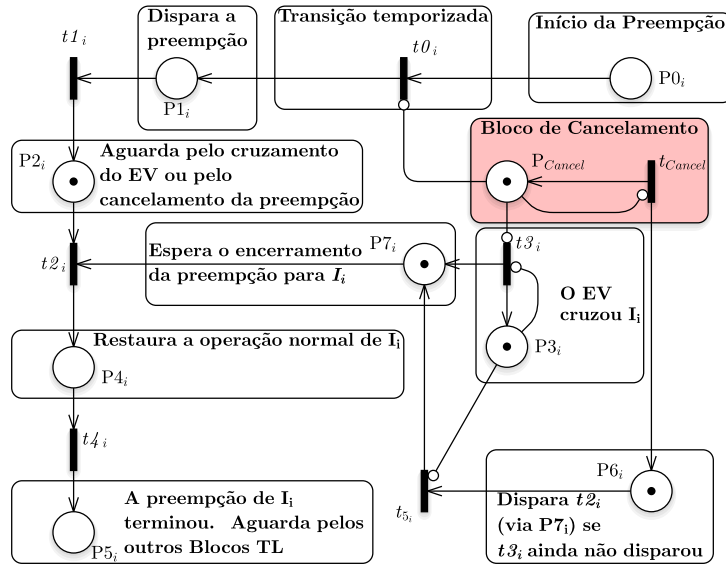


Figura 6.11: A preempção iniciou (disparo da transição t_0) e o EV cruzou o semáforo (disparo da transição t_3 , colocando *tokens* nas posições P_3 e P_7). Neste momento, a preempção é cancelada (disparo da transição t_{Cancel} , colocando *tokens* nas posições P_{Cancel} e P_6). Neste caso, a posição P_3 inibe a transição t_5 , represando o *token* na posição P_6 . Contudo, já há um *token* na posição P_7 , que é usado para concluir a preempção.

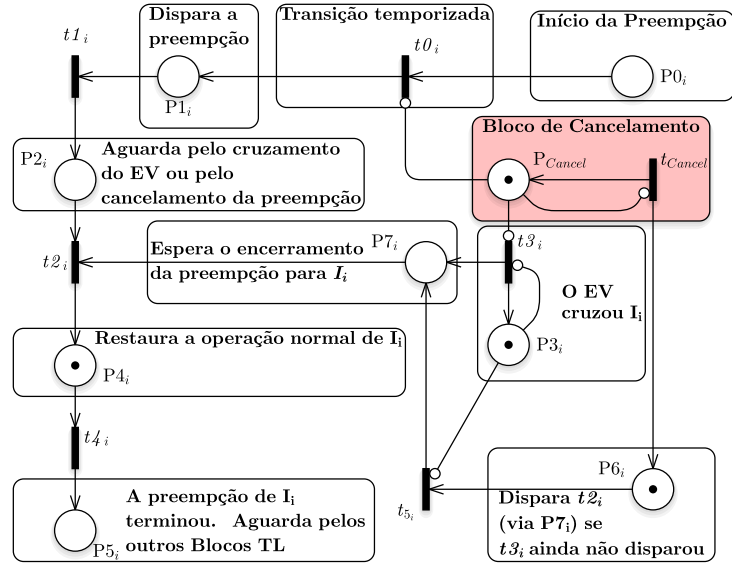


Figura 6.12: A preempção foi finalizada com a informação de cruzamento do EV (disparo da transição t_3), mesmo que o cancelamento tenha ocorrido durante este processo (utilização dos *tokens* nas posições P_2 e P_7 para habilitar a transição t_2 e colocar o *token* na posição P_4).

6.3 Propriedades da TPN* Proposta

Para mostrar que a solução proposta não executa ações concorrentes conflitantes, são definidas propriedades que cada bloco de controle precisa satisfazer: (i) executar no máximo uma ação de preempção; (ii) não restaurar o estado de qualquer TL_i antes de sua preempção; e (iii) executar no máximo uma ação de restauração.

Embora o modelo utilize uma transição temporizada, para a análise, devem ser considerados os casos em que a preempção é urgente (porque o EV está muito próximo de TL_i , por exemplo), e, portanto, o tempo de disparo da transição temporizada $t_0 = 0$. Também há o caso em que a preempção é cancelada (ou seja, quando a transição t_{Cancel} é disparada externamente) logo após a criação da TPN*. Finalmente, a transição t_3 pode disparar logo no início da execução, devido a um erro na leitura do sensor de cruzamento do EV por TL_i . É possível notar que, na verdade, as transições t_0 , t_{Cancel} e t_3 podem disparar a qualquer momento. Portanto, na análise, a transição t_0 também é considerada uma transição imediata, e todas as transições no modelo possuem a mesma probabilidade de disparo por causa das situações descritas.

6.3.1 Boundedness

Nessa subseção mostramos que a TPN* proposta tem, a qualquer tempo, no máximo um *token* em qualquer uma de suas posições, mostrando que a referida rede é *segura*, ou seja, é $[0 - 1]$ -*bounded*. Isso será importante para as demais demonstrações das propriedades de segurança desta rede.

A TPN* foi modelada utilizando o pacote GreatSPN (CHIOLA et al., 1995), bem como suas ferramentas de análise, como o *solver* MC_4CSL^{TA} (GILBERTO; DONATELLI, 2010). Começando com *Boundedness*, é importante saber quantos *tokens* podem estar em qualquer uma das posições da rede, a qualquer momento. Isso porque, se houver mais de um, significa que a ação de pre-empção, por exemplo, pode ter sido executada mais de uma vez para o mesmo TL_i , o que é um comportamento indesejado.

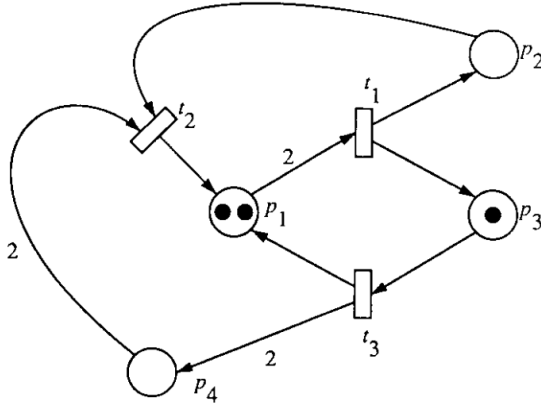
As observações a seguir valem tanto para uma TPN quanto para uma TPN*. Para manter a coerência com as definições encontradas na literatura, a seguir nos referimos à TPN, mas entende-se que essas observações são válidas para uma TPN* também.

Por definição, uma TPN é *k-bounded* limitada se o número de *tokens* em cada posição não ultrapassa um número finito k para qualquer conjunto de marcas alcançável a partir de M_0 , ou seja, $M(P_x) \leq k$ para cada posição $P_x \in P$ (MURATA, 1989). Uma TPN é dita *segura* se ela for $[0 - 1]$ -*bounded*, ou seja, $0 \leq k \leq 1$. A seguir, apresentaremos como é possível fazer essa verificação.

Uma matriz de incidência de uma TPN é uma matriz que relaciona as posições (p) e transições (t), tendo as posições nas linhas e transições nas colunas. Os pares ordenados (p, t) iguais a 1 indicam que a transição t coloca um *token* na posição p . Já quando (p, t) é igual a -1, isso indica que a posição p é uma das condições para ativar a transição t , quando há pelo menos um *token* em p , e esse *token* é removido de p . Quando (p, t) é igual a 0, significa dizer que a posição p e a transição t não estão relacionadas. A Figura 6.13a apresenta uma rede Petri exemplificativa, e a Figura 6.13b apresenta a matriz de incidência relacionada.

Quando as transições disparam, é possível definir f como o vetor de contagem de disparos. É um vetor de comprimento $|t|$, ou seja, cada posição $k, k \leq |t|$ do vetor indica quantas vezes determinada transição disparou no momento da análise, e deve ser um inteiro não negativo. Tendo, portanto, uma marca M_i , a matriz de incidência C e o vetor de contagem de disparos f , podemos encontrar a próxima marca M_j usando a Equação 6.1. Note que usamos a versão transposta de f e M_i , para que possamos multiplicar e somar as matrizes e vetores de forma condizente.

$$M_j^T = M_i^T + C \cdot f^T \quad (6.1)$$



(a) Exemplo de uma rede Petri (MURATA, 1989)

$$\begin{pmatrix} -2 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & -2 & 2 \end{pmatrix}$$

(b) Matriz de incidência da rede Petri ao lado (MURATA, 1989)

Figura 6.13: Exemplo de uma Rede Petri e a respectiva matriz de incidência (MURATA, 1989). As linhas representam as posições e as colunas, as transições.

Para fins exemplificativos, usando ainda o estado e valores das Figuras 6.13a e 6.13b, suponha que a marca da Figura 6.13a seja M_i . Então, $M_i = (2, 0, 1, 0)^T$, indicando que a posição P_1 possui dois *tokens* e a posição P_3 possui um *token*. Suponha agora que a transição t_3 dessa mesma rede Petri tenha disparado. Ou seja, $f = (0, 0, 1)^T$. Considerando a matriz de incidência C da Figura 6.13b, qual é o próximo estado M_j ? Aplicando a Equação 6.1, chegamos ao sistema 6.2.

$$M_j = \begin{pmatrix} 2 \\ 0 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} -2 & 1 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & -2 & 2 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad (6.2)$$

O resultado é a marca $M_j = (3, 0, 0, 2)^T$, significando que t_3 retirou o *token* de P_3 , ficando assim sem nenhum *token*, mas colocou (mais) um *token* em P_1 , ficando assim com 3 *tokens*, e também colocou dois *tokens* de uma vez em P_4 , devido à aresta com peso 2 entre t_3 e P_4 na Figura 6.13a.

Portanto, partindo da premissa de que a marca inicial M_0 seja $[0 - 1]$ -bounded, considerando a matriz de incidência C , deve-se utilizar a Equação 6.1 e verificar se toda marca M_j também é $[0 - 1]$ -bounded, para todo conjunto f válido. Valores válidos de f são vetores com valores não negativos que, aplicados na referida Equação, não deixam nenhuma posição negativa em M_j , significando que a transição retirou mais tokens do que a posição possuía. Considerando que na nossa tpn^* , a marca M_0 é $[0 - 1]$ -bounded por definição, e fazendo $M_i = M_0$, é possível recursivamente validar os valores de M_j e de f , validando assim se a rede é $[0 - 1]$ -bounded para todo $M_j \in M$.

Contudo, isso só é possível se existir a matriz de incidência C . Sobre Ma-

triz de Incidência e Equações de Estado, Murata (1989), na página 551, diz: “Contudo, a resolatividade desse tipo de equações é de alguma forma limitada, parcialmente por causa da natureza não determinística inerente de modelos de rede Petri e por causa das restrições de que as soluções precisam ser inteiros não negativos. Onde quer que matrizes de incidência sejam discutidas neste trabalho, é assumido que a rede Petri é pura ou é tornada pura pela adição de um par de uma transição e posição fictícia como discutido na Seção II (Fig. 3)” (em tradução livre). O ponto é que nossa TPN* não é pura e não pode ser transformada utilizando a técnica descrita por (MURATA, 1989), pois o problema aqui é que a nossa rede possui arestas inibidoras.

Matrizes de incidência de redes Petri não podem ser representadas nessa condição, afinal, a matriz de incidência relaciona uma posição e uma transição, mas quando a rede possui arestas inibidoras, *tokens* em uma outra posição influenciam diretamente o disparo da transição analisada.

Devido a essa limitação na representação do modelo, analisaremos a TPN* proposta considerando as definições anteriores, com o objetivo de verificar se todas as posições contêm entre zero e um *token*, no máximo, a qualquer tempo. Devido ao fato de a TPN* proposta possuir poucas posições e transições, e também desta rede não possuir laços, é viável analisar o estado da rede e verificar se há alguma posição que pode ter mais de um *token* em algum momento.

Para as posições P_{Cancel} e P_3 , é claro que suas arestas inibidoras desabilitam as transições t_{Cancel} e t_3 . Não há arestas de saída de tais posições, portanto, se um *token* está nessas posições, ele permanece nelas, não importa o que aconteça.

Se há um *token* na posição P_6 , há um *token* também na posição P_{Cancel} , que desabilita a transição t_{Cancel} . A única transição que pode colocar *tokens* na posição P_6 é a transição t_{Cancel} , e considerando o que foi previamente dito, é possível afirmar que apenas um *token* pode chegar na posição P_6 .

Para a posição P_7 , *tokens* podem chegar oriundos das transições t_3 e t_5 . As transições t_{Cancel} e t_3 podem disparar apenas uma vez. Então, o que acontece com a posição P_7 depende da ordem de disparo das transições. Se a transição t_{Cancel} dispara primeiro, ela desabilita a transição t_3 (e a si própria). A transição t_5 dispara usando o *token* na posição P_6 , colocando o *token* na posição P_7 . Nenhum *token* mais pode chegar na posição P_7 neste caso. Por outro lado, se a transição t_3 dispara primeiro, um *token* é colocado na posição P_7 . Não importa o que aconteça depois, um *token* já está presente na posição P_3 , que desabilita as transições t_3 e t_5 ao mesmo tempo. Nenhum outro *token* pode chegar até a posição P_7 .

Ao solicitar os valores de *boundedness* para a ferramenta GreatSPN (CHI-

OLA et al., 1995), a mesma apresenta os limites das posições e todas elas são $[0 - 1]$ -bounded, exceto pelas posições P_{Cancel} , P_3 , P_6 e P_7 , em virtude das arestas inibidoras ditas anteriormente. Para essas posições, considerando a análise feita sobre as possibilidades das situações apontadas, é possível afirmar que tais posições também são $[0 - 1]$ -bounded. A Figura 6.14 resume essa análise e apresenta os limites de todas as posições, e todas são $[0 - 1]$ -bounded.

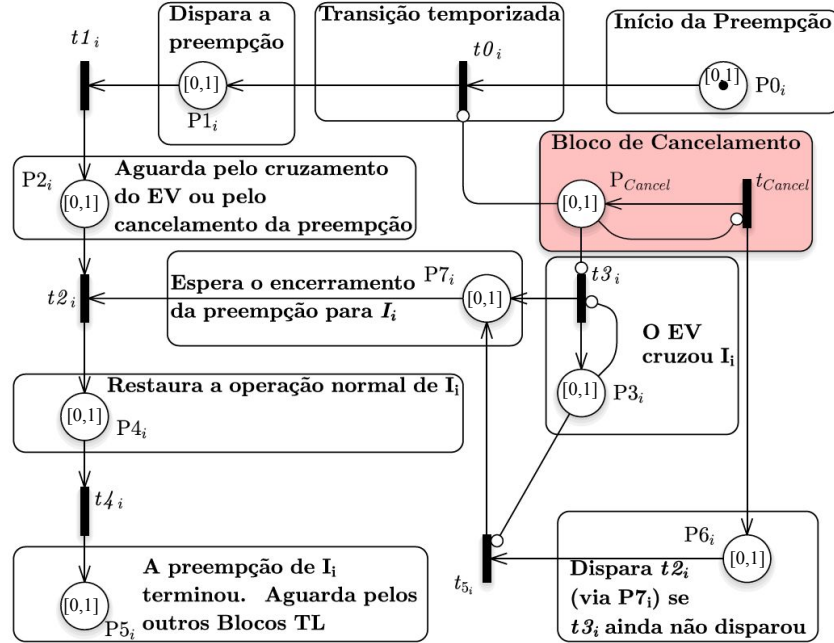


Figura 6.14: Limite de *tokens* nas posições na TPN* proposta. É importante notar que todas as posições são $[0, 1]$ -bounded.

Assumindo que M_0 é $[0 - 1]$ -bounded por definição, considere por absurdo que mais de um *token* passe ou esteja em qualquer posição da rede a qualquer tempo. Se isso fosse verdade, o disparo das transições empurraria tais *tokens* para as últimas posições da rede. Contudo, a rede não é cíclica, e ao chegar em determinadas posições que não possuem transições de saída, haveria um acúmulo de *tokens* em tais posições (em nossa TPN*, as posições “terminais” são P_3 e P_5). Mas isso é impossível, já que a TPN* é $[0 - 1]$ -bounded.

Se isso é verdade, apenas um *token* passa por P_1 , iniciando a preempção, e apenas um *token* passa por P_4 , restaurando o semáforo. Isso garante diretamente as propriedades (i) e (iii), como queríamos demonstrar.

Contudo, isso não garante que a propriedade (ii) seja atendida, pois isso depende da ordem dos eventos e da ordem do disparo das transições. A próxima subseção apresenta como essa propriedade é garantida na TPN* proposta.

6.3.2 Grafo de Alcançabilidade

Alcançabilidade é outro conceito importante na análise de propriedades dinâmicas das TPN. Uma marca M_n é dita **alcançável** se existir uma *sequên-*

cia de disparos de transições que, partindo da marca M_0 , resultem na marca M_n . Essa sequência de disparos é denotada por $\sigma = M_0 t_1 M_1 t_2 M_2 \dots t_n M_n$ (ou a versão simplificada apresentando apenas a sequência de transições $\sigma = t_1 t_2 t_n$). Se existe uma sequência de disparos que leve da marca M_0 para a marca M_n , temos que $M_0[\sigma > M_n$.

Apesar de decidível, a enumeração dos estados para a geração do grafo de alcançabilidade em Redes Petri é exponencial em tempo e espaço (MURATA, 1989; PETERSON, 1981). Para uma rede pequena como a deste trabalho, é possível listar todas as marcas da rede sem grandes problemas, se cada transição habilitada for disparada em um algoritmo recursivo.

E por que é importante saber quais marcas são alcançáveis partindo da marca M_0 ? Ao construir um **Grafo de Alcançabilidade**, é possível verificar o estado do sistema e se as operações desejadas são sempre efetuadas, e mais importante que isso, a ordem em que os eventos ocorrem.

A Figura 6.15 apresenta o Grafo de Alcançabilidade gerado pelo *Solver MC₄CSL^{TA}* (GILBERTO; DONATELLI, 2010). A operação normal segue o seguinte disparo de transições: t_0, t_1, t_3, t_2 e t_4 (As marcas $V1, V3, V6, V10, V14$ e $V17$), terminando com um *token* na posição P_3 por causa do cruzamento do EV e do *token* na posição P_5 , o que significa que TL_i finalizou e está aguardando pelos outros Blocos de Controle. Essa situação poderia se encerrar dessa forma, ou um outro Bloco de Controle pode encontrar algum problema, o que dispararia a transição t_{Cancel} . Neste caso, a marca final da rede (momento em que o estado da rede não é mais alterado, e os *tokens* permanecem nas posições em que estão) seria a marca $M4$.

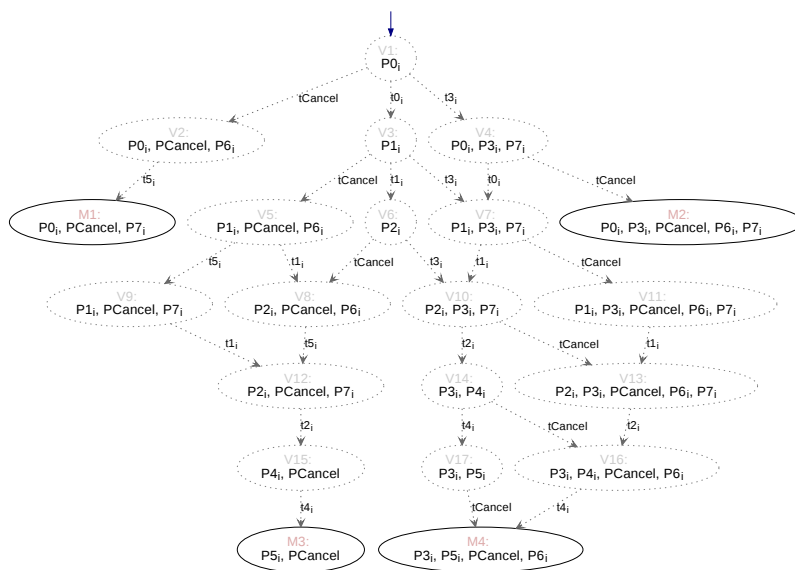


Figura 6.15: Grafo de Alcançabilidade da TPN proposta na Figura 6.2.

Para a propriedade (ii) falhar, é necessário encontrar uma marca (estado

da rede em um determinado momento, apresentando quais posições possuem *tokens*) onde a posição P_1 aparece depois da posição P_4 na Figura 6.15. Como somente as marcas V_{14} , V_{15} e V_{16} possuem a posição P_4 , e seus sucessores não a possuem, portanto, a propriedade (ii) é satisfeita.

Essa consequência é importante, pois mesmo que um operador humano envie dois ou mais comandos de cancelamento, ou o sensor de cruzamento do EV envie mais dados do que deveria, a TPN aqui proposta, por meio dessas propriedades de segurança, garante que as condições no tráfego permaneçam em um estado correto e previsível.

6.4 Algoritmo Onda de Choque + TPN*

O Modelo de Controle descrito acima provê garantias formais para o sistema. Contudo, ele isoladamente não fornece redução do tempo perdido pelo EV no trânsito. A melhoria de desempenho vem da preempção dos semáforos efetuada no momento certo para permitir que o EV e os veículos à sua frente cruzem o semáforo sem gastar muito tempo aguardando na interseção. Por este motivo, os mecanismos de cálculo de dissipação da fila de veículos utilizando o princípio da onda de choque do Algoritmo 5.1 são combinados com as características existentes da TPN* para fornecer uma solução eficiente e robusta. Para isso, o sistema precisa computar o tempo da transição temporizada t_0 para cada bloco de controle na rota do EV quando ele inicia o atendimento à emergência. O sistema atualiza estes tempos para disparo da transição constantemente, considerando as mesmas considerações e Equações da Seção 5.3, e para esta solução, é definido pelas Equações 6.3, Equação 6.4 e Equação 6.5.

A seguir descrevemos o funcionamento do algoritmo. O Algoritmo 6.1 descreve o controle da solução proposta. A TPN* das Figuras 6.1b é criada (Linha 10), dispara transições imediatas quando possível (Linha 12) e executa as ações associadas com as posições quando os *tokens* as alcançam (Linha 13). Por exemplo, quando o *token* alcança a posição P_1 de TL_i , o sistema precisa iniciar a preempção para TL_i . Quando o EV cruza TL_i , o sistema precisa disparar a transição t_3 (Linha 15). O último passo completa a operação normal habilitando a transição t_2 e, logo em seguida, a transição t_4 . A ação associada com a posição P_4 desabilita a preempção do TL_i e o restaura para sua programação normal. A linha 35 é responsável por atualizar os tempos de abertura da transição t_0 .

$$chegada_{EV} \leftarrow \frac{d}{v_{VE}} \quad (6.3)$$

$$Q_{flush} = \frac{Qk}{r} + \begin{cases} \sqrt{\frac{2Q}{a}}, & \text{se } Q \leq \frac{v^2}{2a} \\ \frac{v}{a} + \frac{Q - \frac{v^2}{2a}}{v} & \text{caso contrário} \end{cases} \quad (6.4)$$

$$t0_{open} = data/hora\ atual + chegada_{EV} - (Q_{flush} + t_{flush}) \quad (6.5)$$

Algoritmo 6.1: Onda de Choque + TPN*

```

1  TPN ← NULO;
2  tempo_parado ← 0;
3  enquanto EV está atendendo uma emergência faça
4    TLi ← próximo TL;
5    se temporizador expirou então
6      TPN ← NULO;
7    fim
8    se TPN == NULO então
9      limpe o temporizador;
10     TPN ← construa a TPN* usando a Equação 6.5;
11   fim
12   Dispare as transições habilitadas;
13   Execute as ações das posições com tokens;
14   se EV cruzou TLi então
15     Dispare a transição t3 de TLi, se habilitado;
16   fim
17   senão se TLi está em preempção E EV está parado então
18     se líder está parado então
19       limiar ← limiar_bloqueio;
20     fim
21     senão
22       limiar ← limiar_normal;
23     fim
24     se tempo_parado ≥ limiar então
25       Dispare a transição tCancel, se habilitado;
26       defina o temporizador;
27     fim
28     senão
29       tempo_parado ← tempo_parado + 1;
30     fim
31   fim
32   senão
33     tempo_parado ← 0;
34   fim
35   Atualize os tempos de t0open;
36 fim

```

As Linhas de 17 até 34 são responsáveis por tratar o problema que ocorre quando a preempção está ativa, mas o EV não consegue avançar. A Linha 17 testa justamente essa condição. Se isso não for verdade, a Linha 33 rei-

inicializa o contador usado para verificar quanto tempo o EV fica parado sem se mover. Em caso positivo, a contagem é iniciada (Linha 29). Mas antes, é necessário verificar qual é a tolerância da espera, que ocorre quando o veículo líder do pelotão está parado, mesmo com o semáforo aberto. Se isso é verdade (Linha 18), o *limiar* usado é diferente, mais restritivo, do que o *limiar_normal* de parada do EV. Independentemente do *limiar* escolhido, caso o tempo de parada do EV alcance esse limiar, a preempção é cancelada (Linha 25) e um tempo de *cooldown* é definido para que a preempção seja novamente ativada em um futuro próximo (Linha 26).

O algoritmo de preempção pretende dissipar uma fila de tamanho Q , mas, depois de t_{flush} , a fila se torna $Q + \Delta Q$. Portanto, o problema é descobrir quanto tempo leva para dissipar uma fila de tamanho $Q + \Delta Q$. Neste caso, há uma fila de veículos de tamanho Q metros, parada, e o semáforo começou a transicionar para a luz verde, e esse processo levará $t_{flush}(s)$. Portanto, neste cenário, não há fluxo de partida de veículos, apenas de chegada, e esse fluxo de chegada é conhecido e estimado em $c(veh/s)$. Sabendo que cada veículo tem um comprimento médio de $l(m/veh)$, a nova fila que irá se formar é dada pela Equação 6.6.

$$\Delta Q = clt_{flush} \quad (6.6)$$

O tempo de dissipação da fila é, portanto, alterado, conforme a Equação 6.7.

$$(Q + \Delta Q)_{flush} = \frac{(Q + \Delta Q)k}{r} + \begin{cases} \sqrt{\frac{2(Q + \Delta Q)_i}{a}}, & \text{se } Q + \Delta Q \leq \frac{v_{max}^2}{2a} \\ \frac{v}{a} + \frac{Q + \Delta Q - \frac{v^2}{2a}}{v} & \text{caso contrário} \end{cases} \quad (6.7)$$

Portanto, um valor mais preciso de abertura pode ser obtido se os valores c , l , k e s forem previamente conhecidos. Como já previamente apresentado na Seção 5.3, k e s dependem de características e geometria da via em questão, e c depende ainda de outras características, como hora do dia, se é um dia útil ou não, horários de pico, entre outros. É possível utilizar valores históricos para tais parâmetros, mas a utilização da solução, dependendo dessa base de dados, a torna inviável. Para efeitos de simulação, a implementação da solução calcula o tempo futuro para abertura do semáforo considerando apenas o valor conhecido Q , mas posteriormente um redutor percentual ϵ é aplicado à Equação 6.5 para “representar” o ΔQ desconhecido, com bons resultados empíricos obtidos.

Diante do exposto, é possível notar que o Algoritmo 6.1, além de utilizar o Princípio da Onda de Choque devido às questões de desempenho até aqui observadas, ainda incorpora as propriedades de segurança fornecidas pela

TPN*, resolvendo assim os problemas encontrados nas soluções propostas nos Capítulos 4 e 5.

6.4.1 Complexidade do Algoritmo

Novamente, as mesmas considerações feitas para a complexidade do algoritmo da solução anterior na Subseção 4.4.2 são aplicáveis aqui. As duas primeiras linhas possuem complexidade $\mathcal{O}(1)$, e o tempo de execução entre as linhas 3 e 36 depende do cruzamento do EV por todos os semáforos, e embora uma parte desse tempo seja de fato responsabilidade do algoritmo, há uma boa parte do tempo que o EV gasta independentemente do algoritmo, pois ele leva tempo para percorrer uma certa distância física em sua rota.

Entre as linhas 4 e 11, as operações são $\mathcal{O}(1)$, com exceção da linha 10. A linha 10 constrói a TPN', e como visto anteriormente, isso depende do número de semáforos $|TL|$, do número de posições $|P|$, do número de transições $|t|$ e do número de arestas de relacionamento entre esses componentes $|\rightarrow|$, portanto, a complexidade também é $\mathcal{O}(|TL| \cdot (|P| + |t| + |\rightarrow|))$. Contudo, uma das vantagens da TPN* em relação à TPN' é a simplicidade. Para a TPN*, temos 8 posições, 5 transições e 17 arestas de relacionamento em cada bloco de controle de cada TL , mais 1 posição, 1 transição e 2 arestas de relacionamento do bloco de cancelamento global. Percebe-se, portanto, que o menor número de componentes da TPN* em relação à TPN' colabora na diminuição do número de iterações para a construção da rede.

As linhas 12 e 13 são equivalentes às do algoritmo TPN'. Se o algoritmo for reativo, apenas as transições habilitadas são disparadas, e apenas as ações das posições habilitadas são executadas. Consideraremos que o algoritmo precisa percorrer a lista de $|t|$ e $|P|$, para cada $|TL|$. Portanto $\mathcal{O}(|TL| \cdot (|P| + |t|))$. O restante das linhas, de 14 a 35, possui complexidade $\mathcal{O}(1)$.

Note que foi possível simplificar a rede e o algoritmo; agora, há apenas um lugar onde a TPN* é construída (e reconstruída, se necessário), e isso acontece na linha 10. Também não há um procedimento adicional para o cancelamento, pois isso está incorporado no algoritmo apresentado. Isso deixa o algoritmo mais elegante e simples de entender.

6.5 Implementação

Os algoritmos, incluindo os apresentados no Capítulo 3, foram implementados em *Python* 3, e a implementação da TPN* do Algoritmo 6.1 foi feita utilizando a biblioteca *SNAKES* (POMMEREAU, 2015). Foi necessário fazer alterações nas classes *Transition* e *PetriNet* da biblioteca *SNAKES*, para permitir que as transições temporizadas tivessem seus tempos atualizados

quando isso fosse necessário. Essa alteração está presente no arquivo `tpn.py` e é apresentada no Apêndice A.

Os valores das constantes usados no Algoritmo 6.1 foram definidos empiricamente, sendo $\text{limiar_bloqueio} = t_{flush}$, $\text{limiar_normal} = \max(\frac{Q^k}{s}, 15)$ e $\text{temporizador} = \text{tempo_do_ciclo}(TL)$. Para os limiares e temporizadores, como isso se trata de operações de cancelamento, é importante considerar neste caso tempos de ciclo completos do semáforo; afinal, uma vez que a luz verde é fornecida pelo menos uma vez dentro do ciclo para todas as vias desse semáforo, é razoável supor que nesse intervalo de tempo o problema possa ter sido resolvido. Já o redutor percentual ϵ utilizado na Equação 6.5 foi de 50%, e este valor está relacionado com a liberação da fila $Q + \Delta Q$. Valores próximos de 0% consideram apenas a fila Q , enquanto valores próximos de 100% liberam o semáforo quase instantaneamente, algo parecido com o que faz o Algoritmo Onda Verde, o que também não é desejável. Várias execuções experimentais foram utilizadas variando esses valores, e 50% foi um valor considerado adequado e que apresenta bons resultados.

Nossa implementação do Algoritmo Fuzzy (DJAHEL et al., 2015) também utiliza a biblioteca *pyfuzzy*, tal qual o trabalho original.

Ao final, por estarem implementados utilizando o *framework* proposto neste trabalho na Subseção 3.5.1, apresentamos na Tabela 6.1 a quantidade de *Linhas de Código* (LOC) de nossa implementação dos algoritmos. O *framework* é usado apenas para uniformizar as chamadas de preempção e tomada de tempo e não tem relação com o desempenho dos algoritmos.

Tabela 6.1: Linhas de código da implementação dos Algoritmos

Algoritmo	Técnica	LOC
Algoritmo 6.1	TPN*	447
Algoritmo 3.1	RFID	54
Algoritmo 3.2	Fuzzy	247
Algoritmo 3.3	Filas	106
Algoritmo 5.1 Onda de Choque		110
Algoritmo 3.4	Onda verde	25

6.6 Resultados

Para estes experimentos, a versão do SUMO utilizada foi a versão 1.16. Nas próximas subseções, são apresentados os resultados obtidos neste trabalho, destacando as considerações e achados importantes sobre cada situação específica. Os resultados obtidos para estes cenários foram consistentes com os capítulos anteriores, já que foram feitas várias simulações, com vários cenários, e os resultados são semelhantes.

6.6.1 Cenário Turim

Nesta Subseção são apresentados os resultados para o cenário de Turim. Para fins de comodidade, repetimos os dados das rotas dos EVs na Figura 3.6 e na Tabela 3.3.

Rotas - Cenário Turim

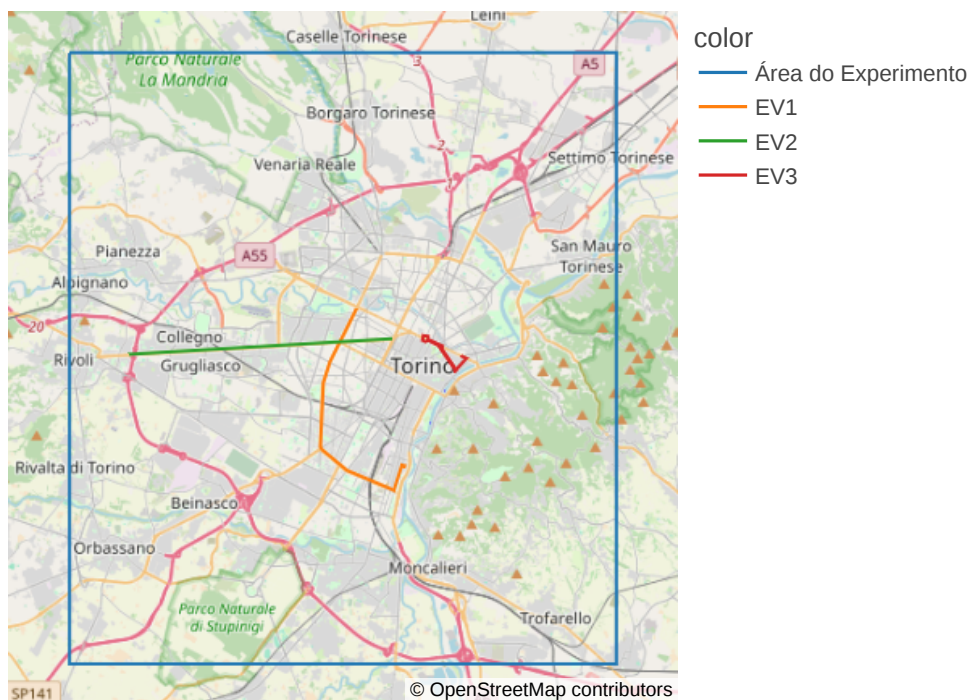


Figura 3.6: Rotas dos EVs dentro da área do experimento do Cenário Turim

Tabela 3.3: Parâmetros das rotas dos EVs - Cenário Turim

EV	Distância	Semáforos
EV1	9.961,68 m	35
EV2	9.952,10 m	19
EV3	5.300,60 m	33

A Figura 6.16 apresenta o tempo perdido para os EVs 1, 2 e 3, tanto para a demanda real quanto para a demanda artificial (quanto menor, melhor), para a versão sem preempção. A Tabela A.69 apresenta os dados correspondentes para análise (as colunas Min e Max só apresentam valores quando existem outliers na Figura). A Figura 6.17 apresenta o percentual de tempo perdido em relação ao tempo total de viagem, acompanhada da Tabela A.70 com os dados relacionados. Com exceção do EV2, que apresentou 30% de tempo perdido quando comparado com o tempo total de viagem, percebe-se que, nos demais casos, esses valores superam os 60% em suas medianas.

A Figura 6.18 apresenta quantas vezes os algoritmos são melhores que a versão Sem Preempção (portanto, quanto maior, melhor). As Tabelas A.71,

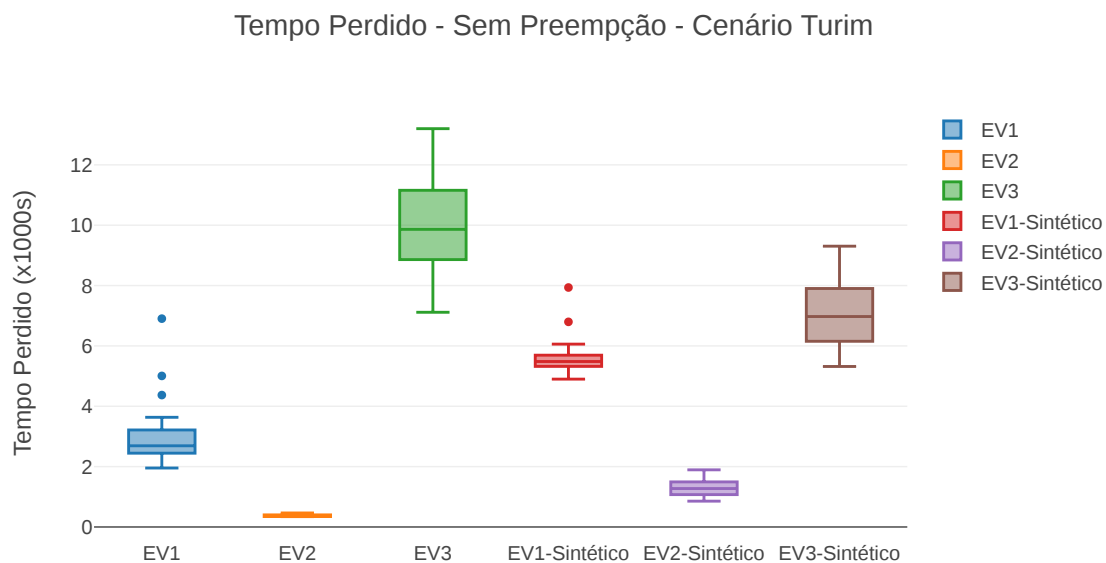


Figura 6.16: Tempo Perdido (s) - Versão Sem Preempção - Cenário Turim

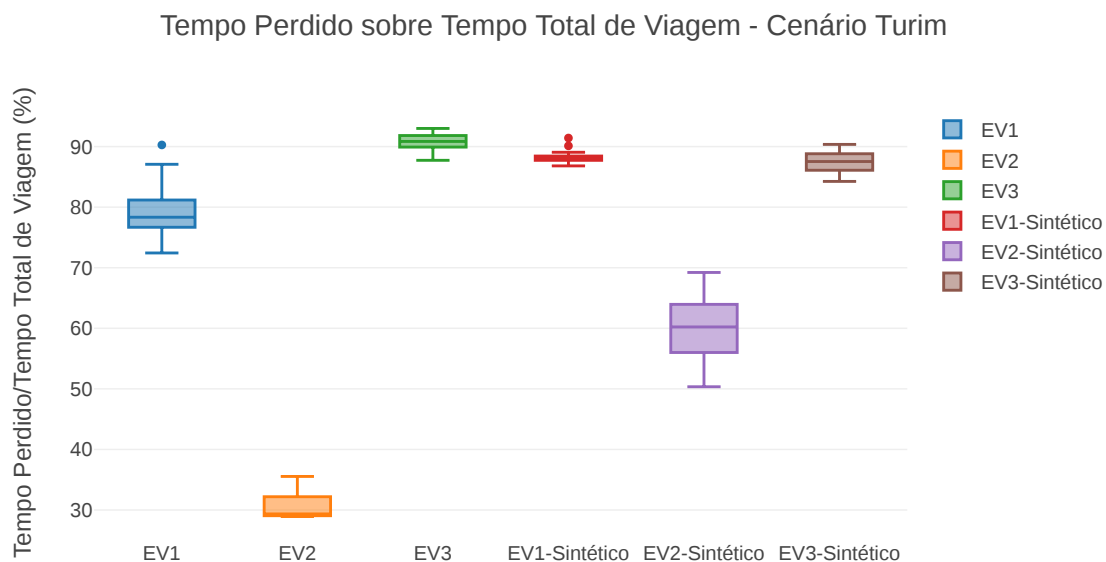


Figura 6.17: Tempo Perdido sobre o Tempo Total de Viagem (%) - Versão Sem Preempção - Cenário Turim

A.72, A.73, A.74, A.75 e A.76 são os dados correspondentes.

Como a melhoria proposta foi feita sobre o Algoritmo Onda de Choque (Algoritmo 5.1), considerando ainda as garantias fornecidas pelas TPN* e considerando o ΔQ adicional das filas, esperamos que a introdução dessas características deixe os tempos competitivos e traga segurança ao mesmo tempo que mantém o desempenho. A Figura 6.18 mostra que os resultados são competitivos, e os melhores resultados foram obtidos para o EV1 no cenário real, com valores de melhoria variando entre 20,48 e 30,99 entre os percentis 25º e 75º. Resultados inclusive melhores que o Algoritmo Onda Verde na maioria dos casos (que possui vantagens e defeitos similares ao Algoritmo Onda de Choque), mesmo considerando o baixo valor de teletransporte de 150 segundos, com exceção da Rota do EV3. Cabe salientar que o Algoritmo Onda Verde é inferior ao Algoritmo Onda de Choque + TPN* (Algoritmo 6.1) para este caso específico, mas que esses dois são inferiores ao Algoritmo Onda de Choque. Esse comportamento é analisado a seguir.

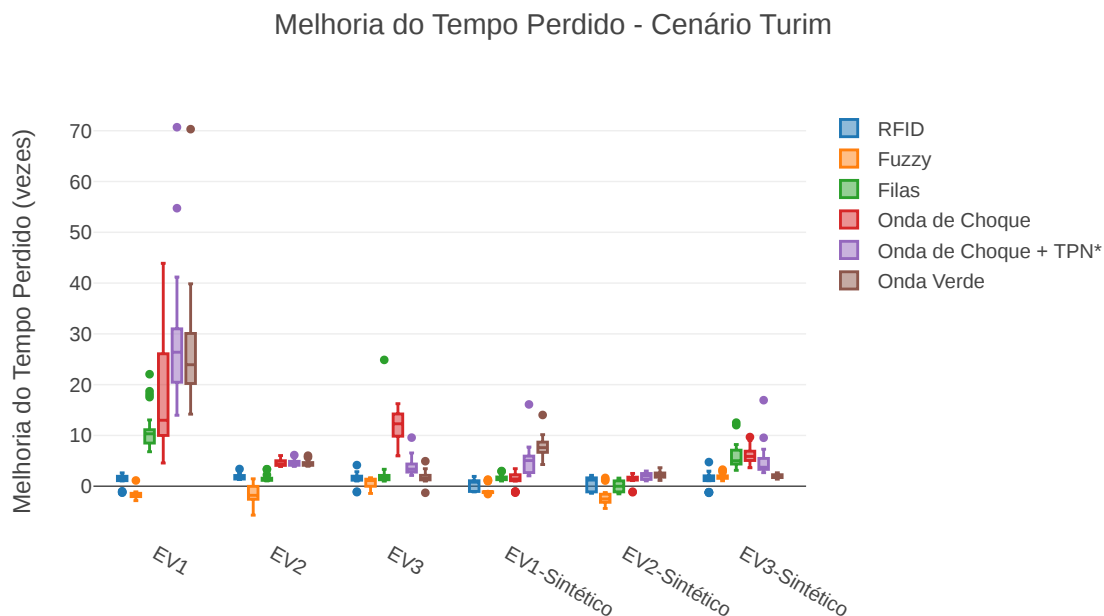


Figura 6.18: Melhoria do Tempo Perdido (vezes) - Cenário Turim

Notamos que o Algoritmo Onda de Choque + TPN* (Algoritmo 6.1) obteve bons resultados, já que para o EV3, ele é o segundo melhor algoritmo, com melhorias atingindo valores entre 2,78 e 4,37 vezes em relação à situação sem preempção, entre os percentis 25º e 75º. Contudo, ele ainda é inferior ao Algoritmo Onda de Choque (Algoritmo 5.1), cuja melhoria atinge valores entre 9,85 e 14,23 vezes para os mesmos percentis. Por este motivo, investigamos as situações e razões desse comportamento.

Neste cenário, observamos que um dos problemas ocorre na interseção da Figura 6.19. Essa interseção é constituída por um conjunto de semáforos não

relacionados e, no caso particular, a rota do EV3 está destacada em azul.

O fato é que o cenário apresenta semáforos bloqueantes não controlados pelo algoritmo. E quando um líder da rota está bloqueado e a preempção é cancelada, rapidamente outros veículos de outras vias preenchem o espaço esperado e são novamente bloqueados. Não foi encontrado um padrão válido para alterar o Algoritmo Onda de Choque + TPN* (Algoritmo 6.1), ou mesmo alguns de seus parâmetros empíricos previamente definidos.

A defasagem entre os algoritmos Onda de Choque (Algoritmo 5.1) e o Algoritmo Onda Verde também é um indicativo, já que esses dois algoritmos têm o mesmo comportamento, mas em momentos diferentes, e, depois que a preempção é ativada, eles se mantêm assim até que o EV cruze o semáforo. As Figuras 6.20a e 6.20b mostram uma foto do Google Maps da interseção real.

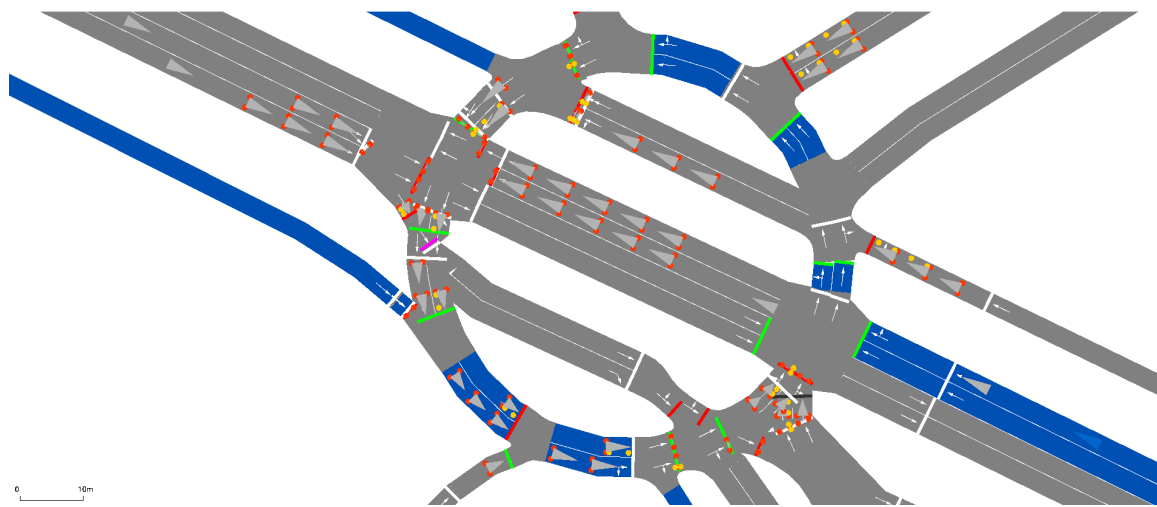
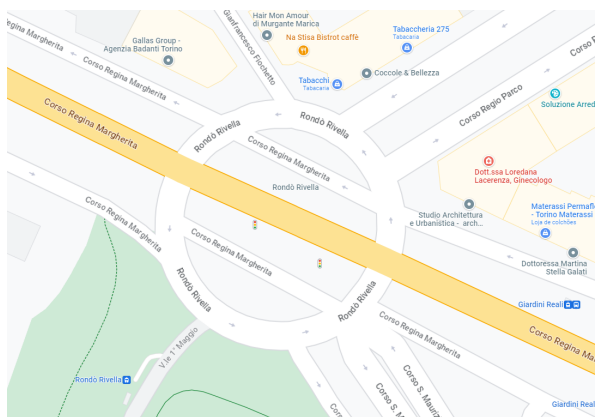


Figura 6.19: interseção Complexa para o Cenário de Turim - Visão SUMO



(a) interseção Complexa para o Cenário de Turim - Visão Google Maps



(b) interseção Complexa para o Cenário de Turim - Visão Google Earth

Figura 6.20: Interseção Complexa para o Cenário de Turim - Visões de *Layers* do Google

A Figura 6.21 apresenta um comparativo baseado em percentuais (quanto maior, melhor), de acordo com as Tabelas A.77, A.78, A.79, A.80, A.81 e A.82.

No intervalo interquartil, o Algoritmo Onda de Choque + TPN* (Algoritmo 6.1) varia de 24,04% até 96,76%, melhoria esta não encontrada em nenhum outro trabalho relacionado (mesmo para o Algoritmo Onda de Choque (Algoritmo 5.1), proposto neste trabalho, que alcançou melhoria máxima para o percentil de 75% no valor de 96,17% para o EV1). O gráfico de percentis é útil para apresentar com clareza como alguns dos outros algoritmos se comportam em relação ao tempo perdido do EV, especialmente na implementação que fizemos do Algoritmo Fuzzy (DJAHEL et al., 2015).

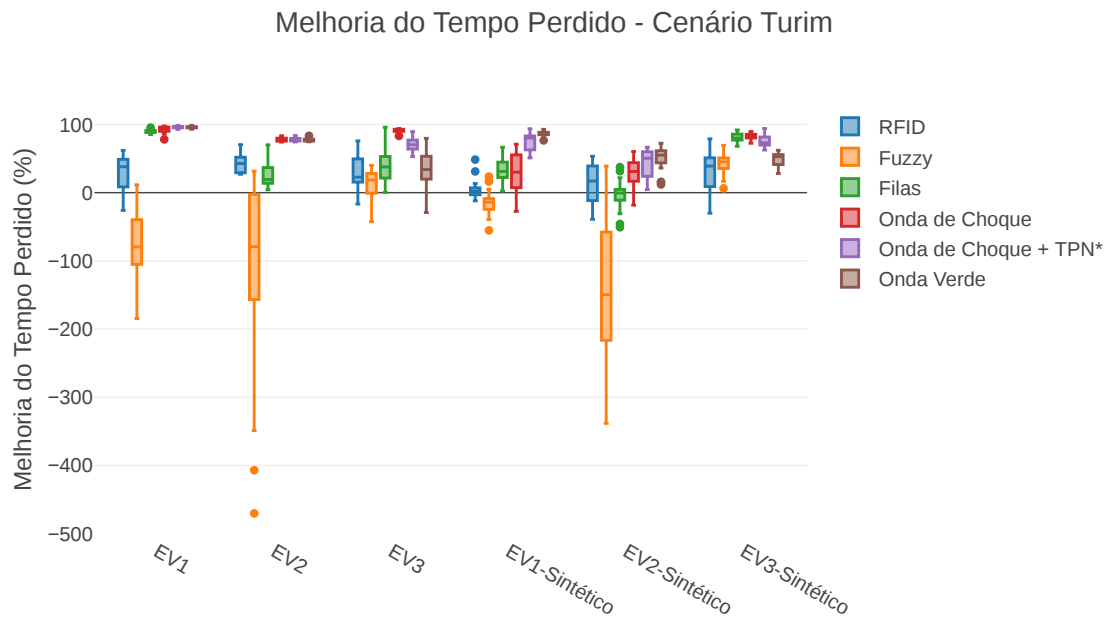


Figura 6.21: Melhoria do Tempo Perdido (%) - Algoritmos - Cenário Turim

A Figura 6.22 apresenta o tempo perdido (s) (quanto menor, melhor) e as Tabelas A.83, A.84, A.85, A.86, A.87 e A.88 são os dados correspondentes. É possível notar que a irregularidade dos tempos perdidos depende da rota do EV, mas o Algoritmo Onda de Choque+ TPN* (Algoritmo 6.1) apresenta os tempos perdidos mais baixos, com medianas ficando abaixo dos 2000 segundos, com exceção do EV3 para o cenário real, visto que sua mediana nesse caso tem valor de 2.956,70 segundos. Ainda assim, é o segundo melhor valor nesse cenário, cujo melhor resultado é o Algoritmo Onda de Choque (Algoritmo 5.1), com mediana de 1.275,73 segundos.

A Figura 6.23 apresenta o tempo de preempção médio, ou seja, quanto tempo os semáforos ficaram com a preempção ativa, em média, enquanto o EV estava na simulação. Portanto, quanto menor, melhor. Os dados relacionados a essa Figura constam nas tabelas A.89, A.90, A.91, A.92, A.93 e A.94. Percebe-se que o Algoritmo Onda de Choque + TPN (Algoritmo 6.1) é competitivo em relação aos outros algoritmos, com baixo tempo de preempção médio, significando que o trânsito é pouco perturbado em decorrência da alte-

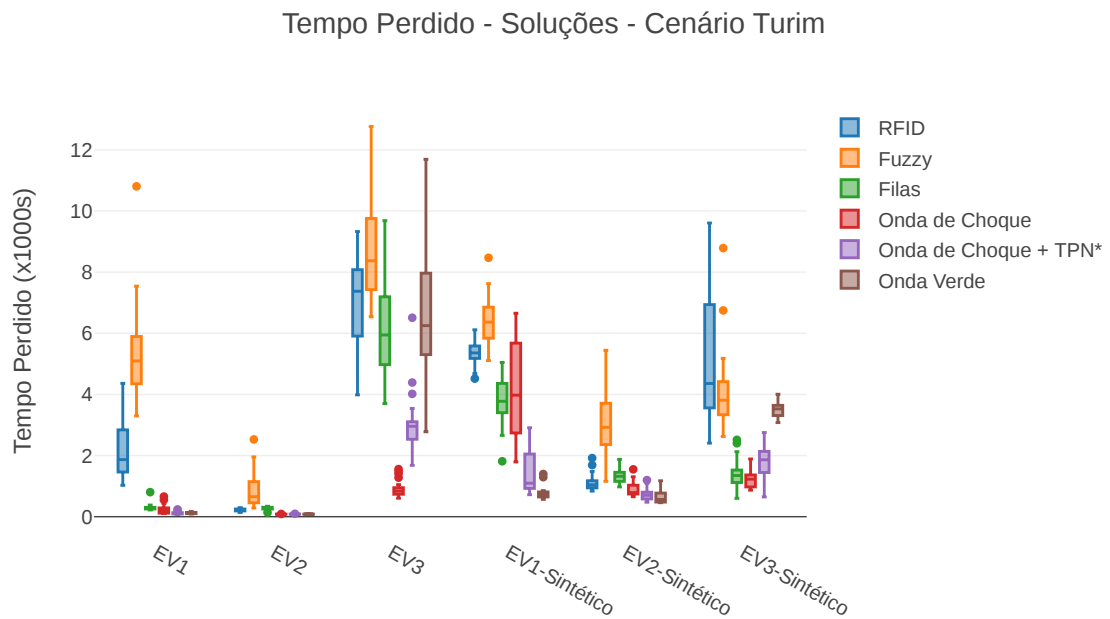


Figura 6.22: Tempo Perdido (s) - Algoritmos - Cenário Turim

ração do ciclo dos semáforos. O maior valor encontrado dentro do intervalo de confiança ocorre para o cenário EV1 sintético, para o percentil de 75%, com 1.017,46 segundos.

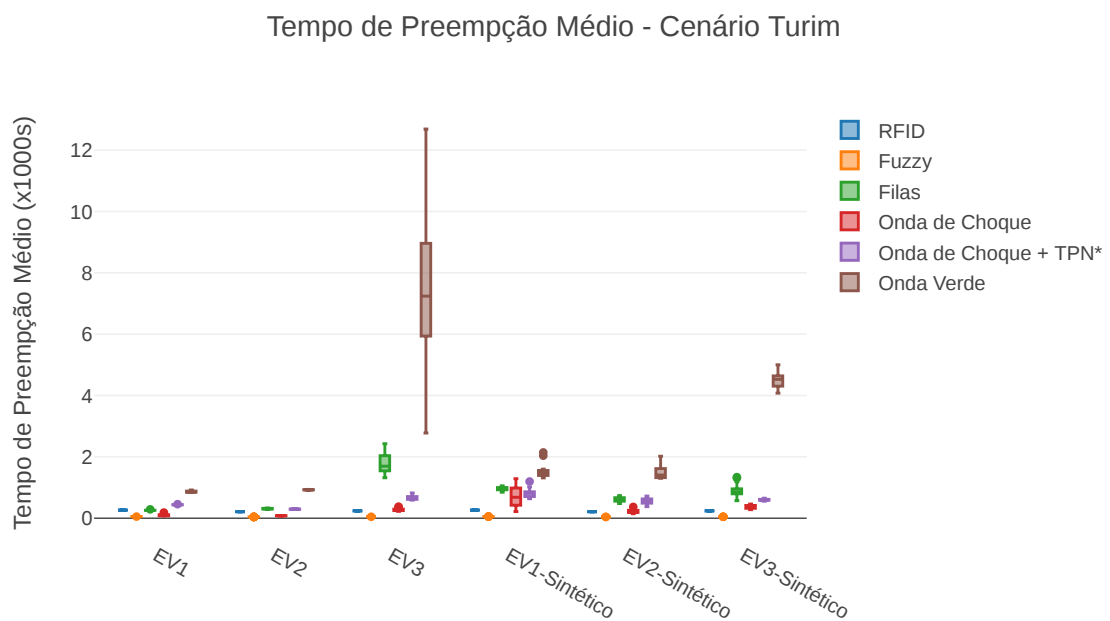


Figura 6.23: Tempo de Preempção Médio (s) - Algoritmos - Cenário Turim

6.6.2 Cenário Colônia

Da mesma forma que o cenário de Turim, por motivos de comodidade, repetimos os dados das rotas dos EVs na Figura 3.7 e na Tabela 3.4 para o cenário de Colônia.

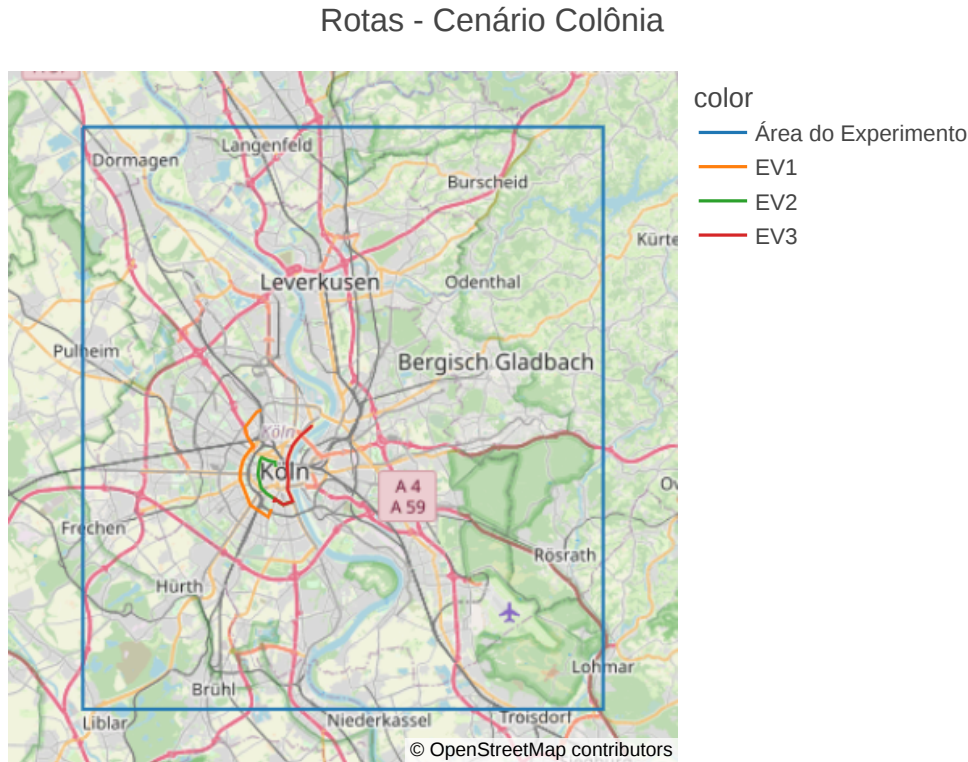


Figura 3.7: Rotas dos EVs dentro da área do experimento do Cenário Colônia

Tabela 3.4: Parâmetros das rotas dos EVs - Cenário Colônia

EV	Distância	Semáforos
EV1	8.351,42 m	26
EV2	5.044,34 m	27
EV3	6.344,31 m	24

Para este cenário, a proposta aqui apresentada também obteve resultados competitivos. Começando com o tempo perdido da Versão Sem Preempção, a Figura 6.24 mostra que o tempo perdido pelos EVs no cenário sem preempção, para a maioria das rotas, fica abaixo de 3.200 segundos para o 75% percentil, com exceção do EV1 e EV3, com 13.151,37 e 9.833,88 segundos para o mesmo percentil, respectivamente. A demanda do trânsito mudou bastante os valores de tempo perdido, já que para as mesmas rotas, mas com dados sintéticos, os mesmos EVs obtiveram valores bem inferiores (A Tabela A.95 apresenta os dados completos para análise). Considerando a informação da Tabela 3.4, não foi possível relacionar a distância percorrida e semáforos cruzados com o tempo perdido resultante, deixando, portanto, uma questão em

aberto se parâmetros como demanda, volume de tráfego ou as conexões da rede poderiam diretamente estar correlacionados a isto. A Figura 6.25 apresenta a proporção do tempo perdido considerando o tempo total de viagem, e os dados relacionados estão presentes na Tabela A.96. É possível observar, para estes EVs e para estas rotas, que a maior parte do tempo do EV no trânsito foi tempo perdido, com medianas acima de 85%, com exceção dos EV1 e EV3 sintéticos, que ainda assim possuem valores altos, com medianas acima de 67%.

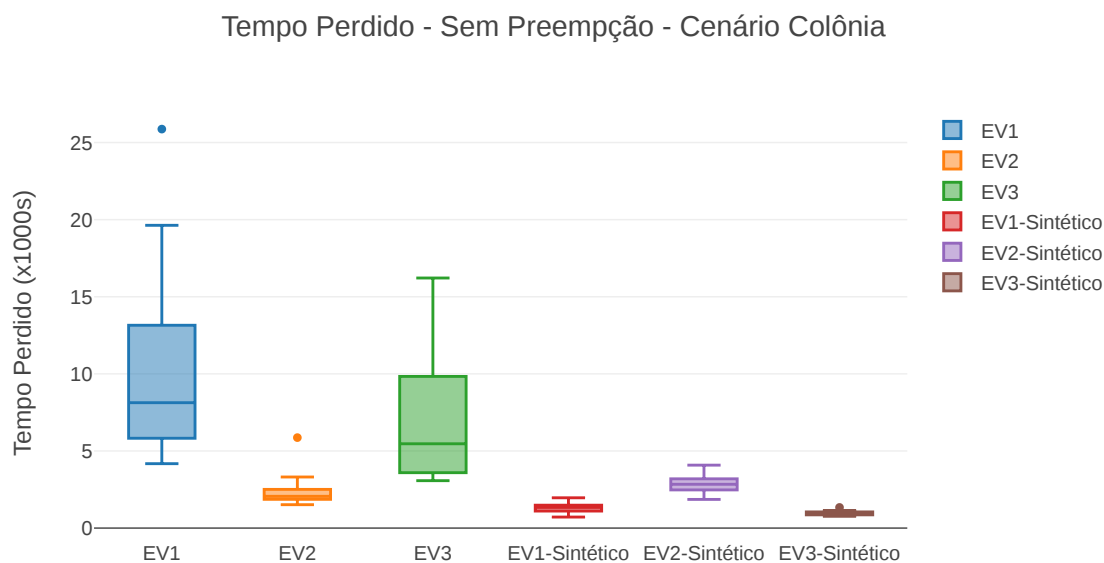


Figura 6.24: Tempo Perdido (s) - Versão Sem Preempção - Cenário Colônia

A Figura 6.26 apresenta a melhoria de tempo perdido dos Algoritmos em relação à Versão Sem Preempção, e os dados respectivos estão nas Tabelas A.97, A.98, A.99, A.100, A.101 e A.102. Novamente, observa-se que a Solução proposta neste capítulo é competitiva, tendo valores compatíveis com os do Algoritmo Onda de Choque (Algoritmo 5.1), isso quando ele não se sobressai mais, como no caso do EV2 sintético, com mediana de 67,83 vezes e percentil de 75% com valor de 96,42 vezes melhor que a versão sem preempção. Contudo, não é possível desconsiderar o resultado do EV3, já que, embora quase todos os algoritmos, com exceção da implementação baseada em Kapusta et al. (2017) ficarem positivos para o intervalo de confiança, o pior valor de todos os experimentos ficou negativo, significando então que a preempção em todos os casos piorou o tempo perdido pelo EV3, em relação ao cenário sem preempção.

A Figura 6.27 coloca os resultados anteriores na mesma escala, e as Tabelas A.103, A.104, A.105, A.106, A.107 e A.108 apresentam os dados relacionados. Nesta Figura, fica claro o que foi dito anteriormente sobre o Algoritmo Onda de Choque + TPN* (Algoritmo 6.1), com ótimos resultados (chegando a

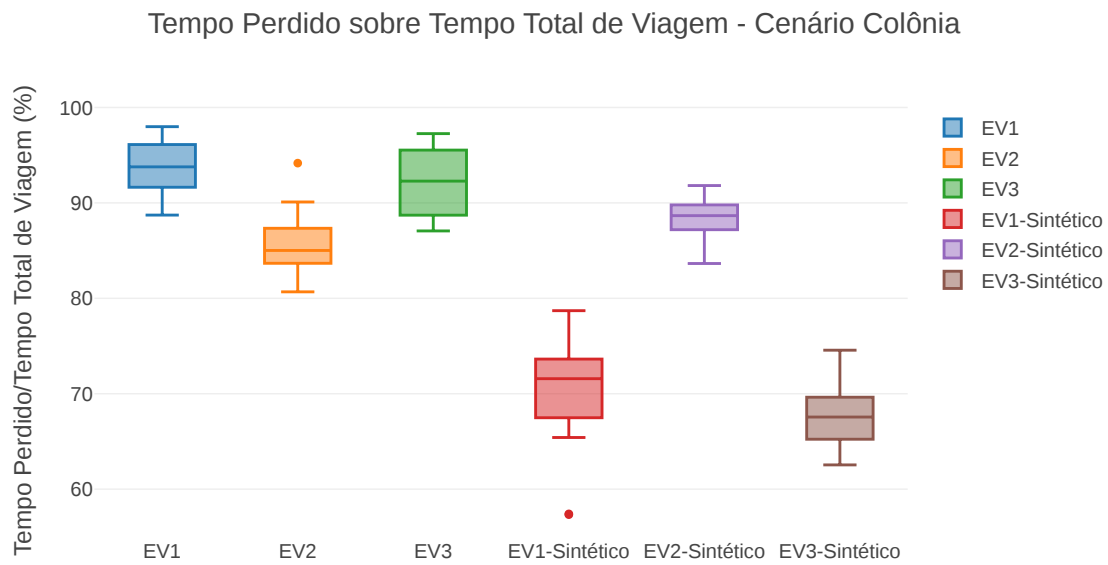


Figura 6.25: Tempo Perdido sobre o Tempo Total de Viagem (%) - Cenário Colônia

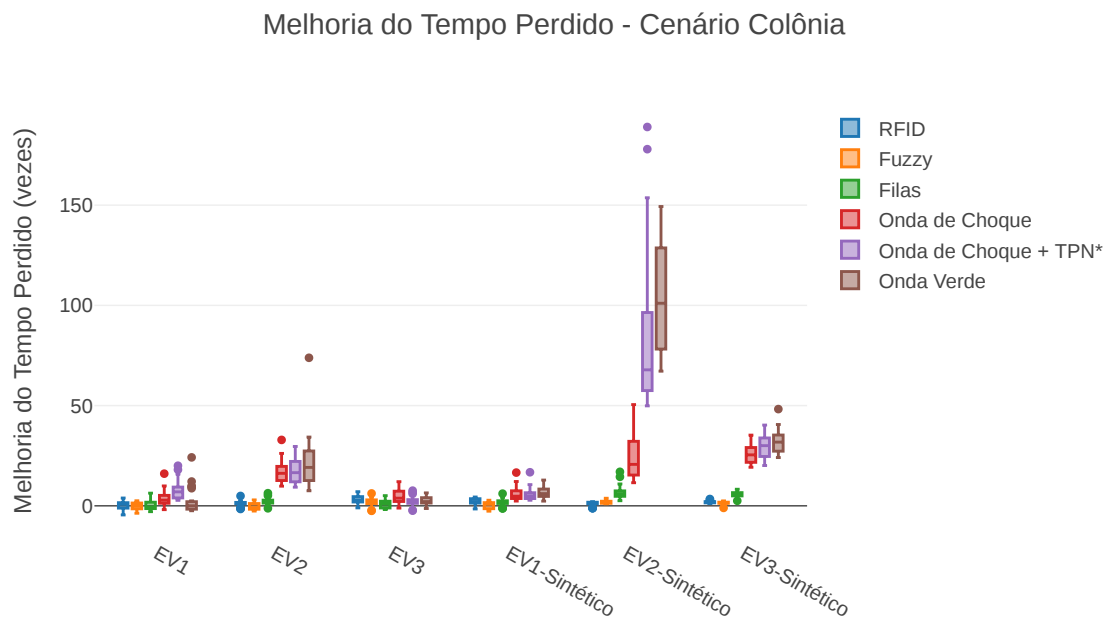


Figura 6.26: Tempo Perdido (s) - Algoritmos - Cenário Colônia

98,96% para o percentil 75% no cenário EV2 sintético); os resultados obtidos abaixo do percentil de 25% do EV3 chamam a atenção. Não que os outros algoritmos tenham se saído muito melhor neste cenário, pois em todos eles há algum outlier que obteve algum resultado ruim. De qualquer forma, é necessário verificar o que houve nesse caso em particular.

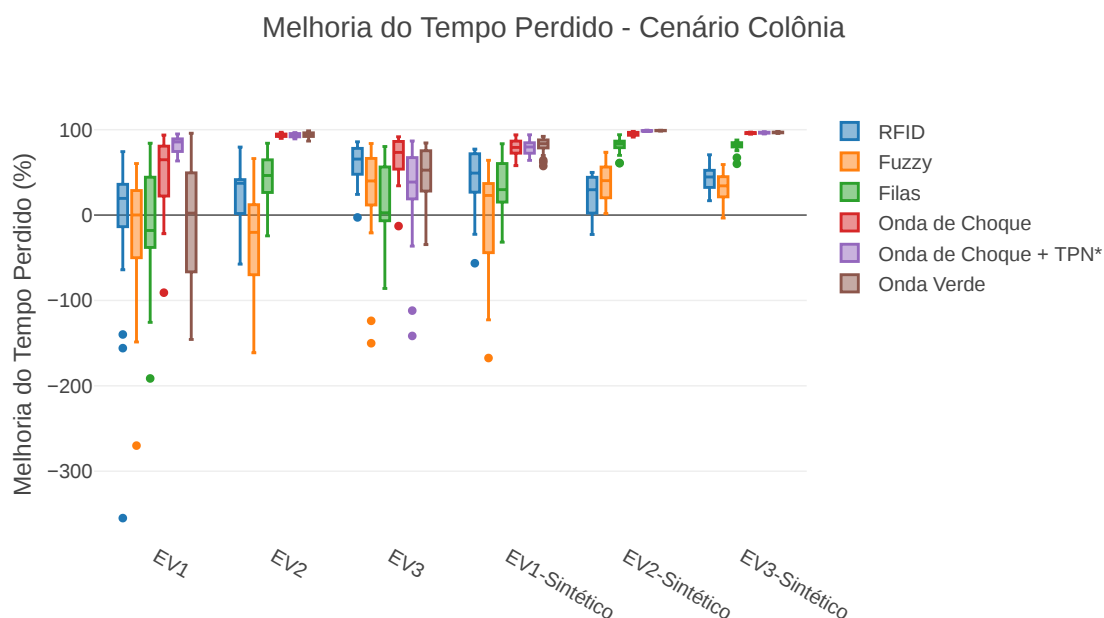


Figura 6.27: Melhoria do Tempo Perdido (%) - Algoritmos - Cenário Colônia

Vamos analisar os *outliers* que fizeram com que o Algoritmo Onda de Choque + TPN* (Algoritmo 6.1) tivesse um desempenho inferior ao da versão sem preempção para o EV3. Assim como o cenário de Turim, foram analisados alguns trechos de gargalos para o cenário cujo resultado foi ruim. A Figura 6.28 apresenta um dos trechos problemáticos, onde se observa o EV3 para a rota que o SUMO traçou (destacada em azul), onde ele deve fazer uma leve curva à direita, e depois, seguindo a seta de ligação das arestas da rede, ele deverá seguir sentido sul.

Contudo, esse trecho não é controlado por semáforos. Depois que o EV3 chega nessa leve curva, ele prossegue até onde a seta preta leva e fica esperando, mesmo que não haja veículos bloqueando. Comparando com os mapas reais desse trecho (Figuras 6.29a e 6.29b), percebe-se que, nesse trecho, há dois semáforos. Ou seja, a rota e o mapa não condizem com a situação real.

Além disso, problemas desse tipo não são relatados no trabalho do cenário original em Varschen e Wagner (2006). Isso porque, no trabalho original, os autores usaram uma simulação mesoscópica, que não apresenta os problemas de interação dos elementos encontrados em nosso trabalho, em virtude do modelo necessário neste caso ser a utilização de simulação microscópica.

Problemas deste tipo atrapalham a avaliação geral, já que os algoritmos não têm condições de resolver problemas de possíveis falhas do simulador.

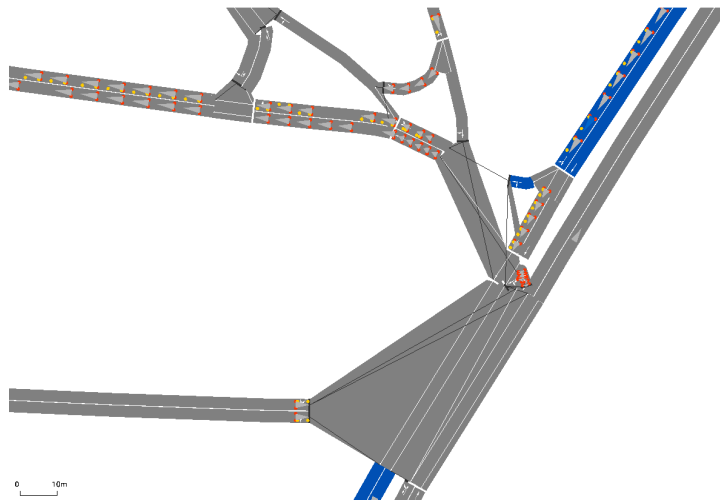
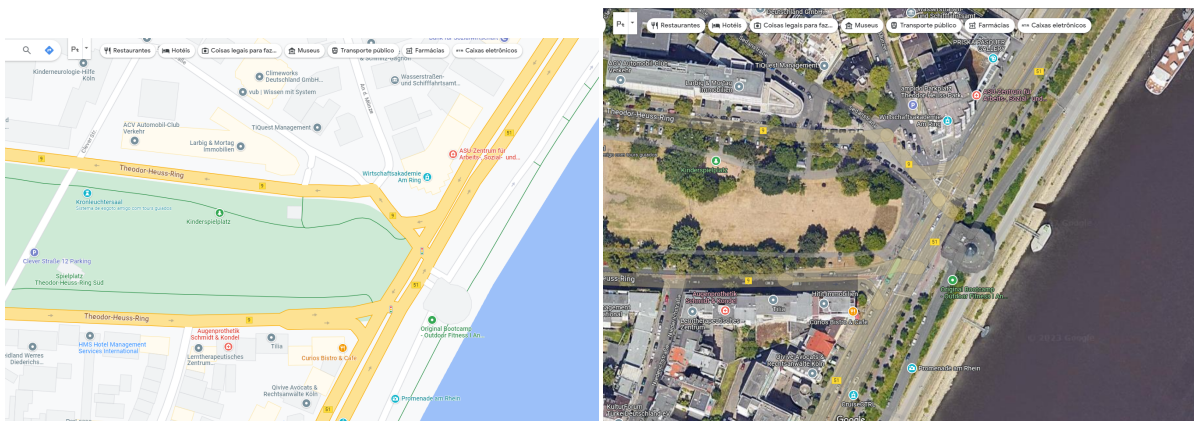


Figura 6.28: Uma das interseções Complexas para o Cenário de Colônia - Visão SUMO



(a) Uma das interseções Complexas para o Cenário Colônia - Visão Google Maps

(b) Uma das interseções Complexas para o Cenário Colônia - Visão Google Earth

Figura 6.29: Uma das interseções Complexas para o Cenário Colônia - Visões de *Layers* do Google

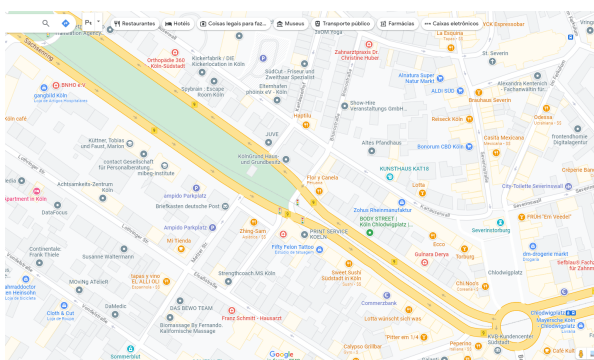
Há ainda mais dificuldades mais à frente do trajeto. A Figura 6.30 apresenta a próxima interseção, com o EV3 no começo do trecho no canto inferior direito, onde ele deve continuar pela rotatória sentido noroeste. Quase na parte superior esquerda da Figura, a rota do EV3 deixa de ser uma única via e vira uma avenida com duas vias. Neste momento, há um semáforo que controla a entrada de veículos na perpendicular. O primeiro ponto de atenção é que, novamente, o trecho apresentado no SUMO não condiz com o trecho visto nos mapas reais nas Figuras 6.31a e 6.31b. Não parece haver uma forma, pelo menos nesta parte, da avenida que tem direção contrária, possuir um retorno para a avenida da rota do EV3.

Esta situação apresenta uma limitação do Algoritmo Onda de Choque + TPN* (Algoritmo 6.1). Todas as vias da rota do EV3 estão completamente saturadas, mas não só elas, as vias de entrada concorrentes também. Assim, quando a preempção é interrompida, será fornecida luz verde para as vias concorrentes, que logo preencherão a rota do EV3 com mais veículos.

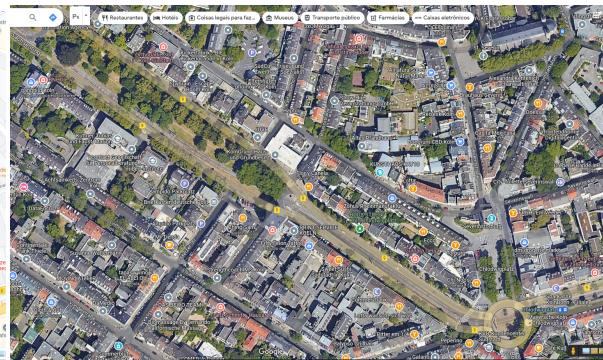
O Algoritmo Onda de Choque (Algoritmo 5.1) apresentou um bom desempenho nesse caso, pois há um valor de teletransporte de 150 segundos, que é irreal para situações práticas. Com altos valores de teletransporte (tendendo ao infinito), os resultados, como já vistos anteriormente, não apresentariam boas perspectivas.



Figura 6.30: Mais uma interseção complexa para o Cenário de Colônia - Visão SUMO



(a) Mais uma interseção complexa para o Cenário Colônia - Visão Google Maps



(b) Mais uma interseção complexa para o Cenário Colônia - Visão Google Earth

Figura 6.31: Mais uma interseção complexa para o Cenário Colônia - Visões de *Layers* do Google

A Figura 6.32 apresenta o tempo perdido dos Algoritmos para o Cenário Colônia (e os dados completos estão apresentados nas Tabelas A.109, A.110, A.111, A.112, A.113 e A.114). A complicação verificada para o EV3, explicada anteriormente, não surpreende ao se observar essa Figura (com um outlier

que chama a atenção de 22.871,76 segundos, que é o equivalente ao outlier de -141,56% da Figura 6.27). Mas ignorando essa anomalia, é possível notar a constância da solução, com valores de IQR variando menos que os das outras soluções.

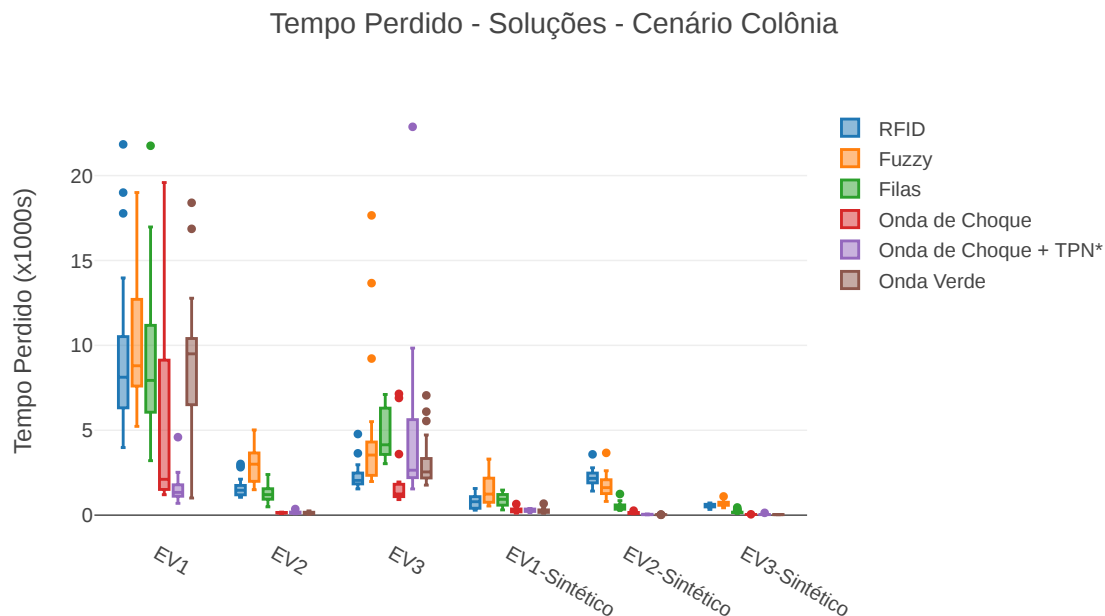


Figura 6.32: Tempo Perdido (s) - Algoritmos - Cenário Colônia

Por fim, a Figura 6.33 apresenta os tempos de preempção médios das soluções, acompanhados pelos dados nas Tabelas A.115, A.116, A.117, A.118, A.119 e A.120. Nota-se que o tempo que os semáforos ficaram com a preempção ativa, em média, para o Algoritmo Onda de Choque + TPN* (Algoritmo 6.1) é similar aos tempos dos outros algoritmos, o que confirma o baixo impacto nos outros veículos (percentil de 75% de 528,17 segundos para o EV1). A exceção é, novamente, o EV3, cujo valor atingido pelo experimento no percentil 75% é de 992,35 segundos, mas ainda assim, perdendo apenas para o Algoritmo Onda de Choque (Algoritmo 5.1), também apresentado neste trabalho.

6.6.3 Cenário do Centro Expandido de São Paulo - Metrô OD SP 2017

Esta Subseção apresenta os resultados obtidos para o cenário criado neste trabalho, o Cenário de São Paulo - Centro Expandido - Metrô OD SP 2017. Para fins de comodidade, como fizemos com os cenários de Turim e Colônia, repetimos os detalhes das rotas dos EVs na Figura 3.9 e Tabela 3.5.

É importante reafirmar o que foi dito na Subseção 3.4.5. Foi verificado que o esforço necessário para a criação de cenários sintéticos, devido ao tamanho

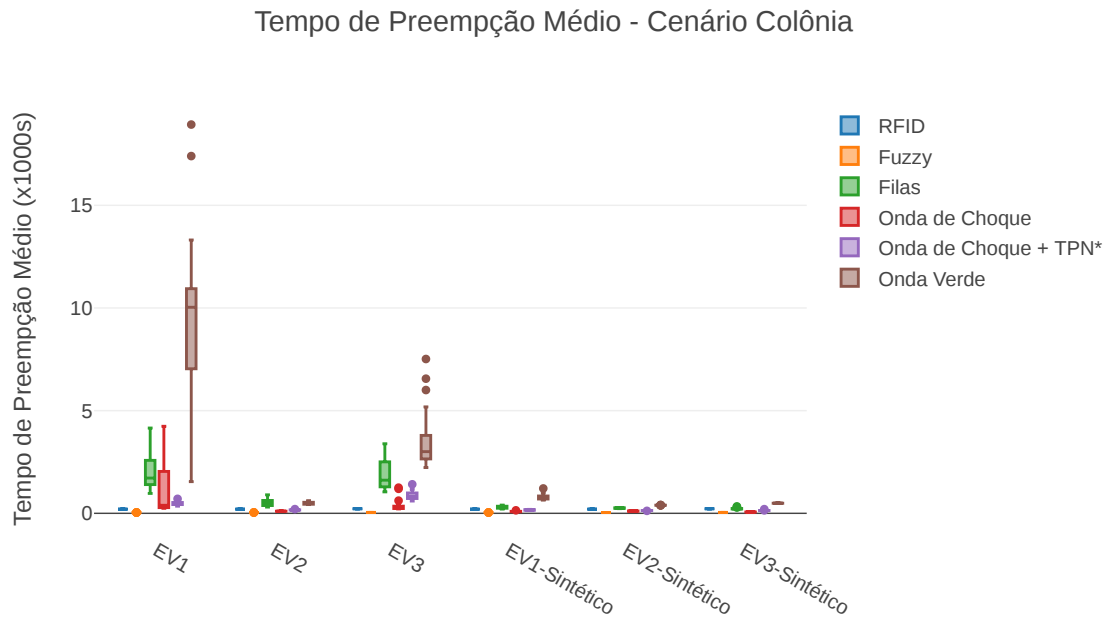


Figura 6.33: Tempo de Preempção Médio (s) - Algoritmos - Cenário Colônia

Rotas - Cenário Centro Expandido - Metrô OD SP 2017

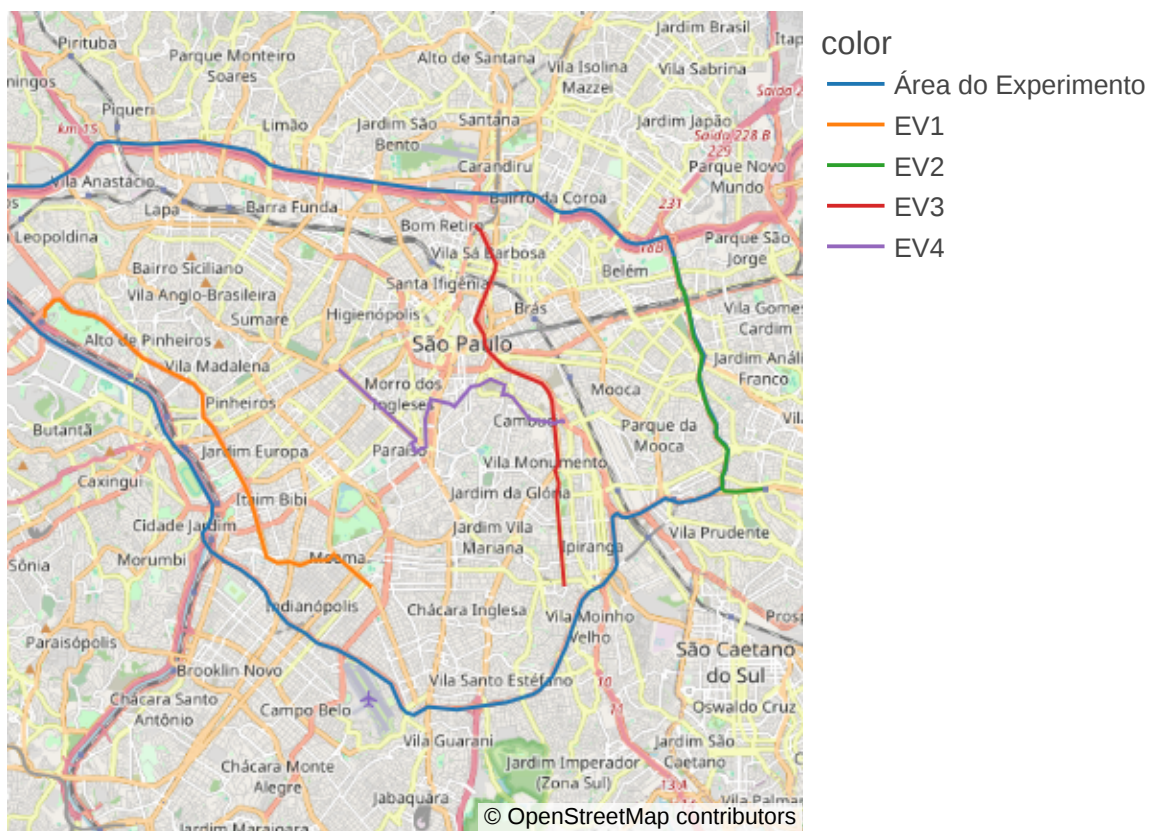


Figura 3.9: Rotas dos EVs dentro da área do experimento do cenário Centro Expandido de São Paulo (Companhia Metropolitana de São Paulo (METRÔ), 2017).

Tabela 3.5: Parâmetros das rotas dos EVs - cenário Centro Expandido de São Paulo

EV	Distância	Semáforos
EV1	12.140,19 m	55
EV2	6.914,45 m	15
EV3	9.922,56 m	23
EV4	8.602,20 m	43

desse cenário e ao tempo de execução necessário para executá-los, não agregaria tanto quanto adicionar um EV, a partir de uma rota já existente escolhida aleatoriamente. Por esse motivo, esse cenário não possui dados sintéticos, mas há um EV adicional com uma rota diferente da dos outros EVs.

Isto posto, avaliaremos os resultados para este cenário. A Figura 6.37 apresenta o tempo perdido da Versão Sem Preempção para o Cenário Centro Expandido - Metrô OD SP 2017, enquanto a Tabela A.121 apresenta os dados correspondentes. É compreensível que o EV1 possua um tempo perdido maior, quando comparado aos outros EVs, já que ele possui a maior rota e cruza o maior número de semáforos, conforme apresentado na Tabela 3.5. Contudo, é importante ressaltar que há divergências detectadas no mapa gerado em relação ao efetivamente apurado na realidade. Note na Figura 6.34 que há duas interseções sem semáforos. Por conta disso, os carros, na simulação, se resolvem de acordo com a prioridade da via, que não necessariamente condiz com o que deveria ser. Observe como o mesmo trecho é retratado no Google Maps nas Figuras 6.35 e 6.36. Além disso, é importante notar que, no simulador, o EV1 cruza por 55 semáforos. Contudo, esse EV1 cruza um total de 73 interseções, significando que 18 interseções os algoritmos de preempção não têm gerência. O tempo perdido pelo EV, portanto, é sensível às prioridades dessas vias, e a dinâmica do trânsito pode impactar a variação dos resultados para as diferentes execuções.

Pelo mesmo raciocínio, EV2 apresenta um dos valores mais baixos de tempo perdido, e este EV possui a menor rota e cruza o menor número de semáforos. Contudo, a discrepância entre os EV3 e EV4 não é explicada apenas por esses parâmetros, já que a diferença da rota dos dois EVs é de apenas 1.320,36 m, mesmo que o EV4 cruze vinte semáforos a mais que o EV3. Apenas com parâmetros de comprimento de rota e números de semáforos cruzados não se explicam tais diferenças, e características da rede e volume do trânsito nas proximidades podem contribuir para essa situação. As considerações feitas sobre as diferenças entre o mapa gerado pelo simulador e os aferidos na realidade também são aplicáveis a todos os EVs.

Já com relação ao percentual de tempo perdido, dados apresentados pela Figura 6.38 e Tabela A.122, quando comparado ao tempo total de viagem,

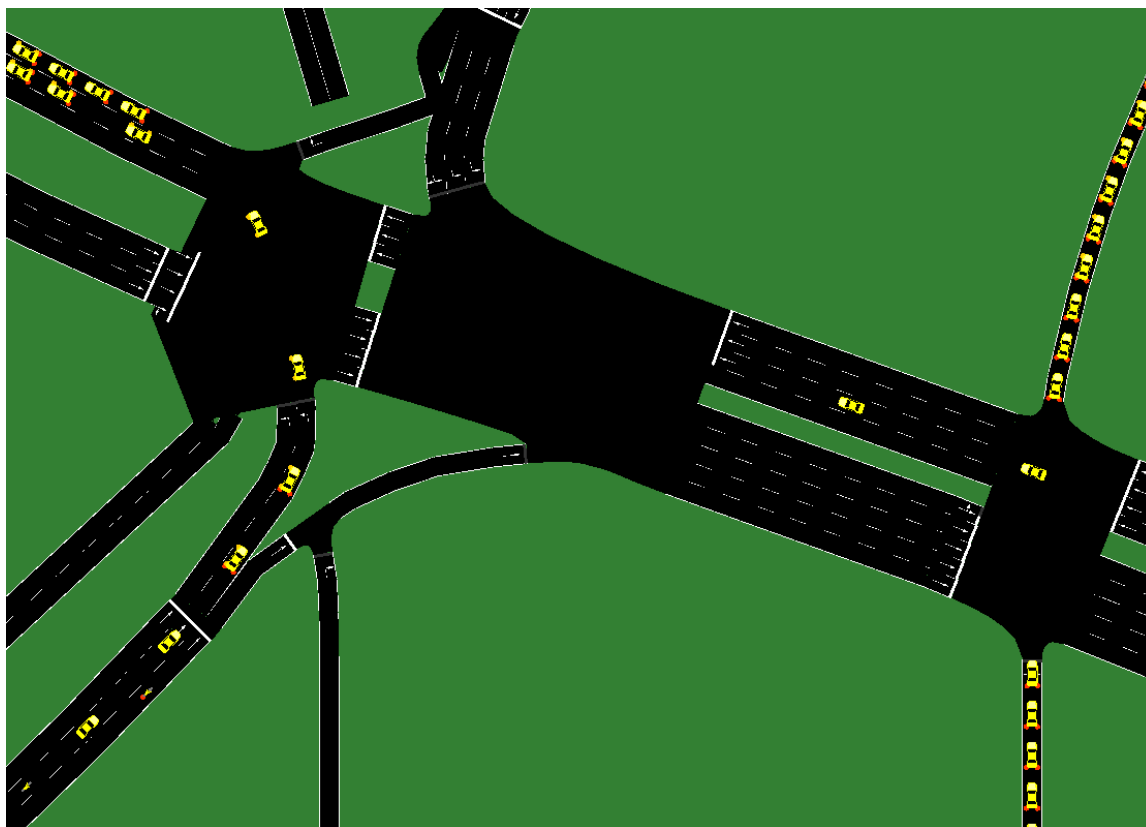


Figura 6.34: Trecho da rota do EV1 no simulador SUMO para o Cenário do Centro Expandido de São Paulo. Cruzamento da Av. Pedroso de Moraes e da R. Ferreira de Araújo.

mesmo para o EV que menos presenciou atrasos (EV3), os valores das medianas foram superiores a 70%, chegando até a 97,01% para o EV1. Tais dados demonstram o impacto do congestionamento nas viagens para este cenário. O EV3 enfrentou menos resistência quando comparado aos outros EVs, mas de forma alguma tais valores podem ser considerados baixos.

A Figura 6.39, com os dados relacionados apresentados nas Tabelas A.123, A.124, A.125, A.126, A.127 e A.128, apresenta os bons resultados do Algoritmo Onda de Choque + TPN*, em especial pelos resultados de EV2, EV3 e EV4. Há um *outlier* para o EV4, mas os *outliers* do Algoritmo Onda de Choque e os valores entre Q1 e a mediana do Algoritmo Onda Verde indicam uma complicação que os algoritmos de preempção não conseguem resolver, nestes casos específicos. Essa é a mesma situação apresentada anteriormente para o cenário de Turim, na Figura 6.19. Já para o EV1, o teletransporte auxiliou os Algoritmos Onda de Choque e Onda Verde, mas, como dito anteriormente, essa situação é impraticável em uma situação real. Ainda assim, esse cenário sugere que há alguns casos em que, combinando fatores da rede, volume de tráfego e complexidade do entorno, pode ser produtivo desativar completamente a preempção por um determinado trecho, para manter assim a linha de base mais próxima possível da versão sem preempção.

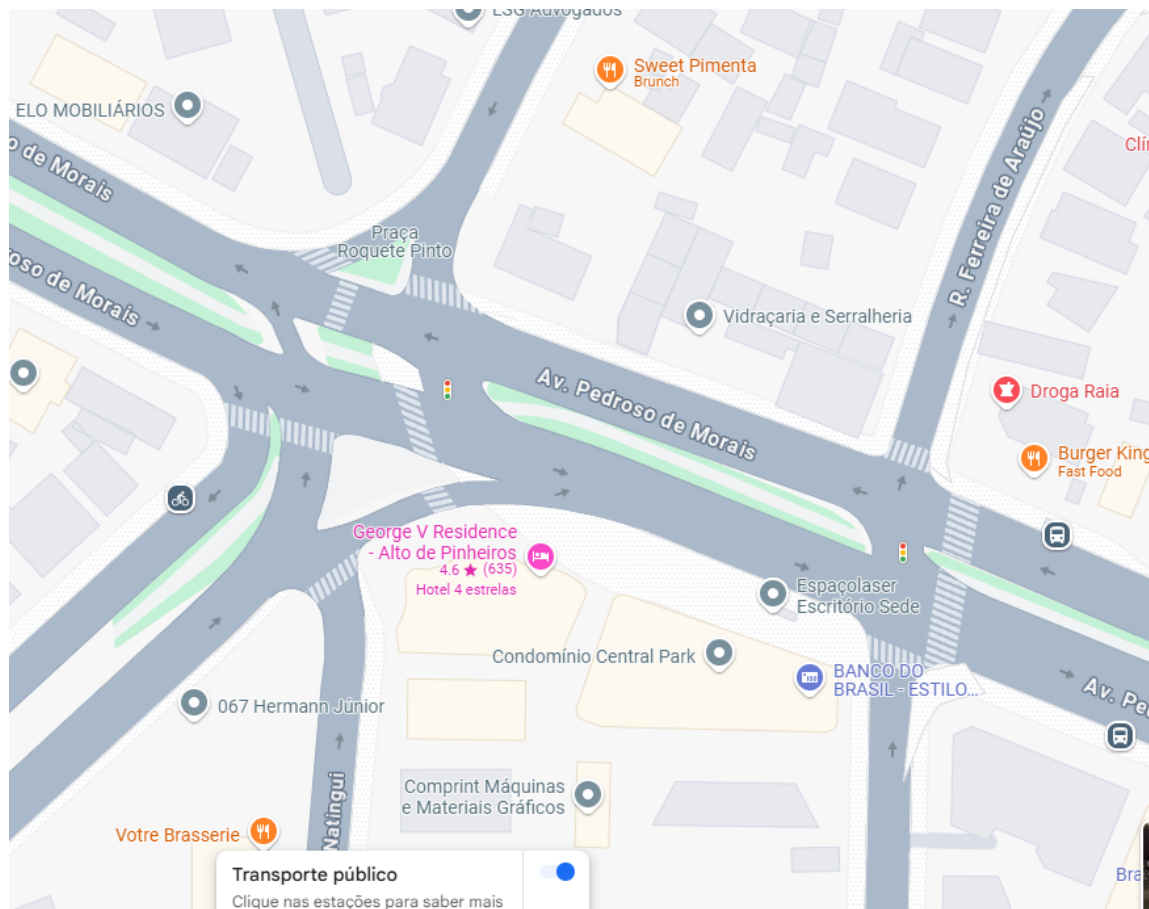


Figura 6.35: Trecho da rota do EV1 no Google Maps para o Cenário do Centro Expandido de São Paulo. Cruzamento da Av. Pedroso de Moraes e da R. Ferreira de Araújo.

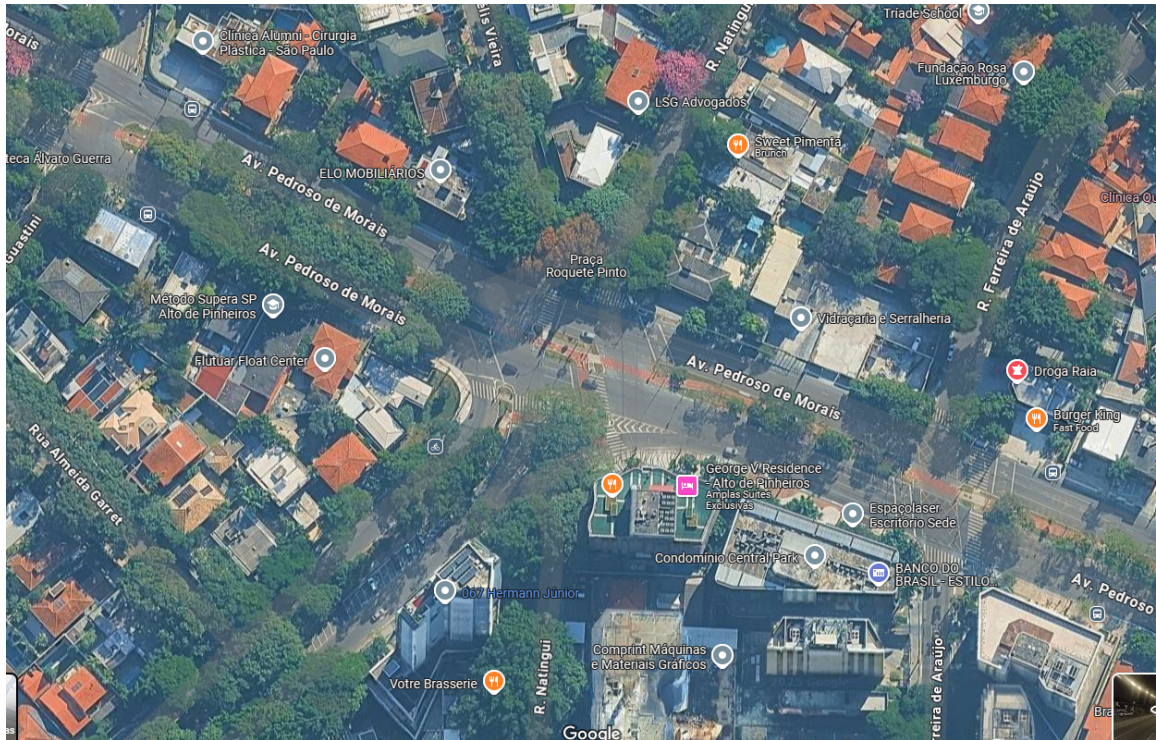


Figura 6.36: Trecho da rota do EV1 no Google Earth para o Cenário do Centro Expandido de São Paulo. Cruzamento da Av. Pedroso de Moraes e da R. Ferreira de Araújo.

Tempo Perdido - Sem Preempção - Cenário Centro Expandido - Metrô OD SP 2017

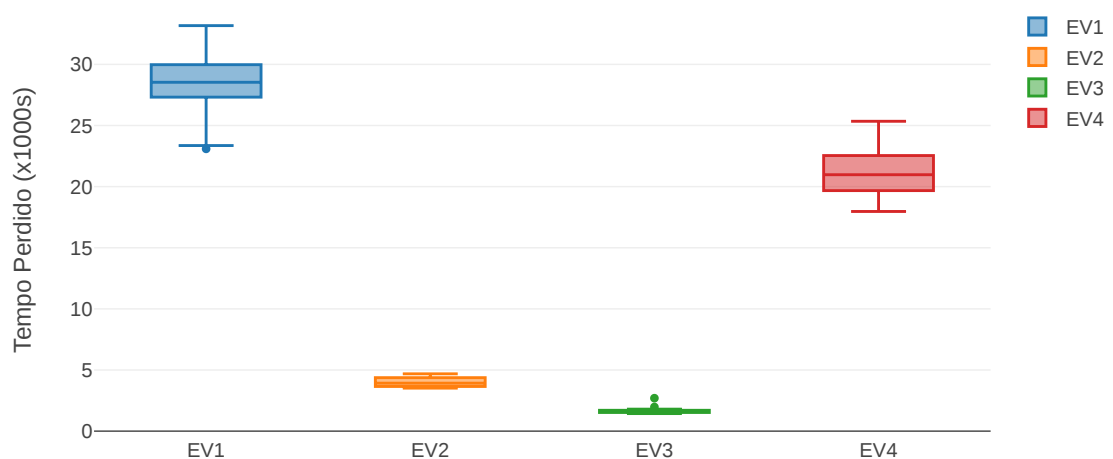


Figura 6.37: Tempo Perdido (s) - Versão Sem Preempção - Cenário Centro Expandido - Metrô OD SP 2017.

mpo Perdido sobre Tempo Total de Viagem - Cenário Centro Expandido - Metrô OD SP 20

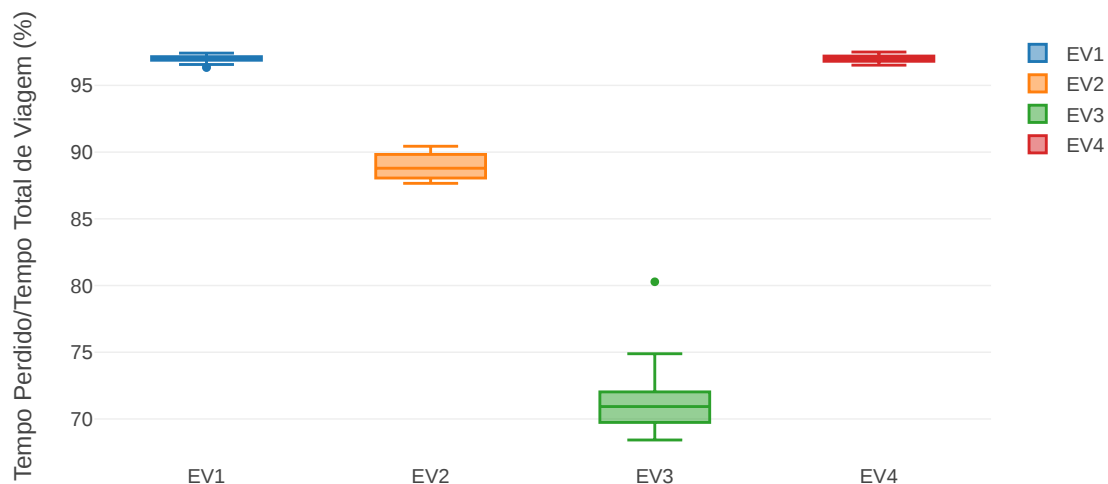


Figura 6.38: Tempo Perdido sobre o Tempo Total de Viagem (%) - Cenário Centro Expandido - Metrô OD SP 2017.

Melhoria do Tempo Perdido - Cenário Centro Expandido - Metrô OD SP 2017

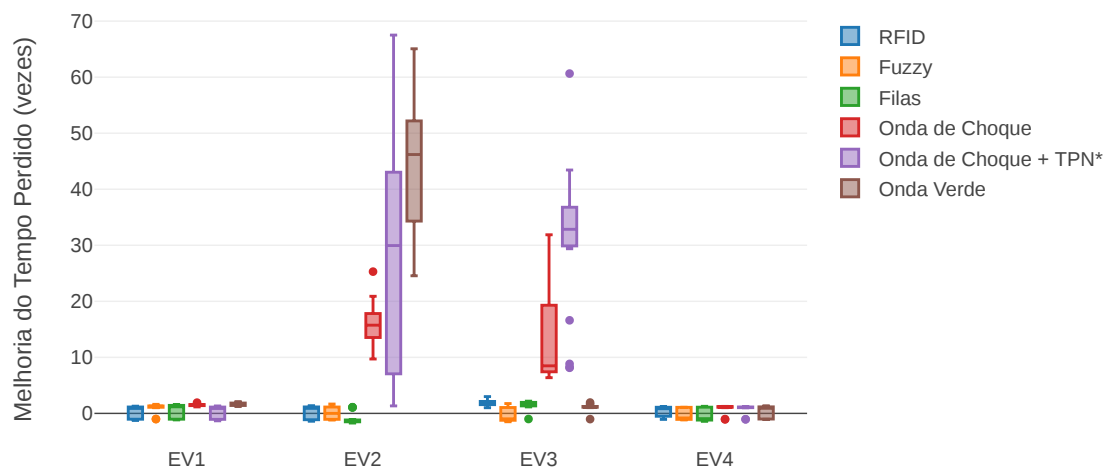


Figura 6.39: Melhoria de Tempo Perdido (vezes) - Algoritmos - Cenário Centro Expandido - Metrô OD SP 2017.

A Figura 6.40 e as Tabelas A.129, A.130, A.131, A.132, A.133 e A.134 reforçam os dados apresentados anteriormente. As medianas apresentaram resultados positivos em todos os casos para o Algoritmo Onda de Choque + TPN*, com o pior resultado apresentado de 4,56% no caso do EV1, mas atingindo valores de 97,28% para o Q3 do EV3. Para o intervalo de confiança, como já apresentado, o pior resultado foi obtido para o EV1, com o Q1 igual a -6,07%. Ainda assim, 6% de piora em relação à versão sem preempção, considerando todos os complicadores deste cenário, não é um resultado ruim.

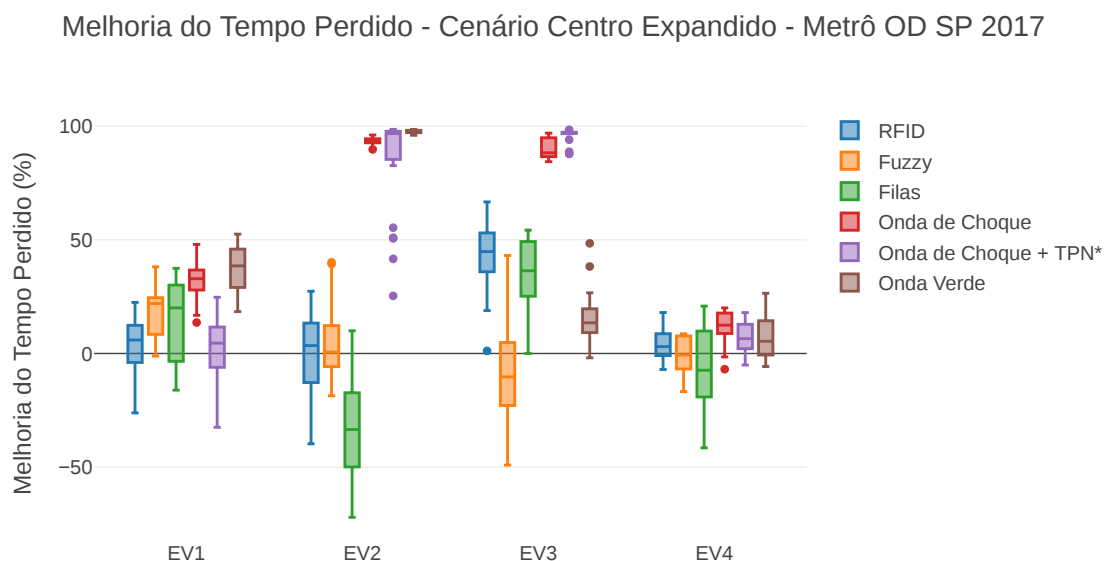


Figura 6.40: Tempo Perdido (%) - Algoritmos - Cenário Centro Expandido - Metrô OD SP 2017.

A Figura 6.41 e as Tabelas A.135, A.136, A.137, A.138, A.139 e A.140 apresentam o tempo perdido dos Algoritmos. O tempo perdido da solução apresentada fica no patamar do Algoritmo Onda de Choque e Onda Verde, para os EV2, EV3 e EV4. A exceção é, novamente, o EV1. Para este EV, o tempo perdido ficou dentro dos quartis Q1 e Q3, entre 26.374,01 e 29.569,31 segundos. Para fins comparativos de escala, a versão sem preempção apresenta, para estes mesmos quartis, 27.320,67 e 29.975,78 segundos, resultados bem próximos, o que é um bom indicador para um cenário tão complicado.

Por fim, a Figura 6.42 apresenta o tempo de preempção médio, ou seja, quanto tempo os semáforos ficaram com a preempção ativa enquanto o EV estava na simulação, em média. Os dados relacionados a essa Figura constam nas tabelas A.141, A.142, A.143, A.144, A.145 e A.146. Percebe-se que o tempo médio de preempção para o Algoritmo Onda de Choque + TPN* está entre os menores, indicando pouca perturbação para os outros veículos. Mesmo tendo um desempenho inferior ao dos outros algoritmos para o EV1, nesta

Tempo Perdido - Soluções - Cenário Centro Expandido - Metrô OD SP 2017

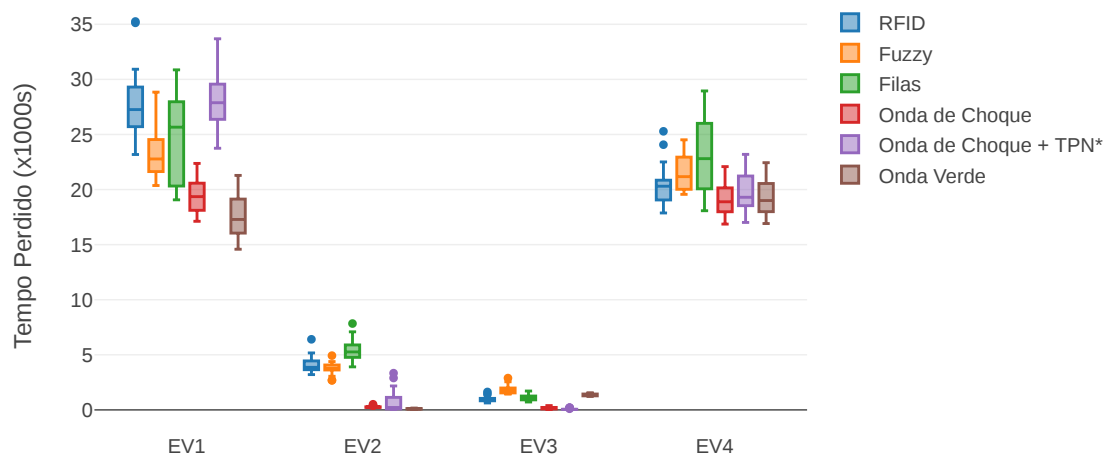


Figura 6.41: Tempo Perdido (s) - Algoritmos - Cenário Centro Expandido - Metrô OD SP 2017.

instância o Algoritmo percebe que não adianta permanecer com a preempção ativa, já que o trânsito não está fluindo de acordo, e, portanto, dando oportunidade para os outros veículos. Um alto valor neste caso, para o Algoritmo Onda Verde, indica que foi necessário manter a preempção ativa por muito tempo, confirmando mais uma vez que algoritmos sem cancelamento de preempção podem levar a situações de *starvation*.

É possível navegar interativamente nos gráficos apresentados para verificação dos resultados¹.

6.7 Conclusão

O Algoritmo Onda de Choque + TPN* (Algoritmo 6.1) apresentou bons resultados e se mostrou competitivo frente aos outros algoritmos para os cenários avaliados. Levando em consideração os algoritmos avaliados, o Algoritmo Onda de Choque + TPN* é o algoritmo que pode ser utilizado em situações práticas e que obteve os melhores resultados, além de apresentar as garantias apresentadas na Seção 6.3.

Nas situações em que o Algoritmo Onda de Choque + TPN* apresentava resultados inferiores, verificamos que o simulador e problemas no cenário contribuíram para esta situação. Ainda assim, constatamos uma limitação em uma situação particular, quando as vias estão saturadas, mas o cancela-

¹https://nbviewer.org/github/rodrigogbranco/charts/blob/main/tese_2025-06-28-terceiroexperimento-app.ipynb

Tempo de Preempção Médio - Cenário Centro Expandido - Metrô OD SP 2017

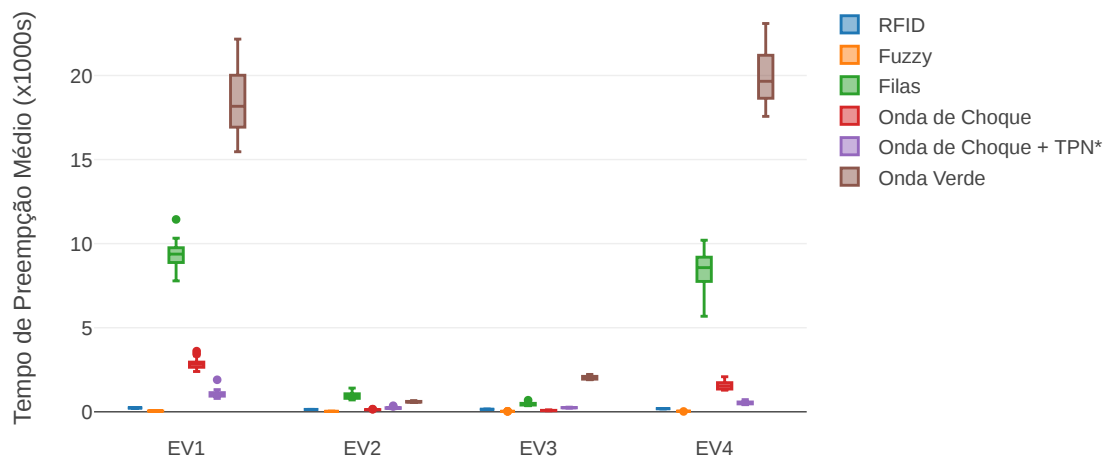


Figura 6.42: Tempo de Preempção Médio (s) - Algoritmos - Centro Expandido - Metrô OD SP 2017.

mento da preempção implica em outros veículos concorrentes ocupando os lugares vagos, o que resulta em um resultado inferior como consequência desta situação. Há algumas abordagens que podem ser usadas, como “desligar” a preempção até que o trecho problemático fosse ultrapassado, por exemplo. Restaria descobrir a extensão do trecho problemático e com quanto tempo de antecipação se deve detectar essa situação. Pode ser necessário trabalhar com um algoritmo com múltiplas políticas ou trabalhar segmento a segmento individualmente. É importante notar que em nossas experiências, quando tentávamos resolver situações particulares no algoritmo, muitas vezes os casos gerais eram penalizados no processo. No momento, a interrupção da preempção garante que os veículos não entrem em *starvation*, que serão objetos de trabalhos futuros.

Conclusão

Nesta tese, investigamos e propusemos soluções para o problema da preempção semafórica para veículos de emergência em malhas urbanas congestionadas, focando na atuação em rota completa. A pesquisa resultou no desenvolvimento de três abordagens distintas: a TPN' (focada em segurança), o Algoritmo Onda de Choque (focado em eficiência de limpeza de filas) e a solução híbrida Onda de Choque + TPN*.

Os resultados obtidos por meio de simulações microscópicas extensivas, utilizando dados reais de Turim, Colônia e do Centro Expandido de São Paulo, demonstram a eficácia da abordagem proposta:

- **Eficiência Operacional:** A solução híbrida (Onda de Choque + TPN*) apresentou os melhores resultados globais, alcançando uma redução de até 98,96% no tempo perdido pelo EV em comparação com o cenário base. Em cenários críticos, o desempenho da solução superou a estratégia teórica de Onda Verde, provando ser mais robusta contra anomalias de tráfego e bloqueios de interseções.
- **Segurança e Formalismo:** Diferente de abordagens puramente heurísticas, a modelagem baseada em Redes de Petri Temporizadas permitiu garantir formalmente propriedades de segurança, assegurando a ausência de estados inválidos ou *deadlocks* no controlador semafórico, um requisito essencial para sistemas críticos em Cidades Inteligentes.
- **Robustez:** O modelo mostrou-se capaz de lidar com a inserção dinâmica de veículos e tempos de transição semafórica (amarelo/vermelho), mantendo a integridade da malha viária mesmo sob alta demanda.

Na próxima seção, são apresentadas as principais contribuições deste trabalho.

7.1 Contribuições

Os algoritmos propostos neste trabalho, que utilizam duas Redes Petri Temporizadas modificadas, a TPN' e TPN*, garantem que ocorram condições de segurança e estabilidade para que a preempção de semáforos seja feita de forma segura. Dos trabalhos analisados, este é o primeiro que provê garantias formais de que ações concorrentes não ocorram em qualquer semáforo da rota do EV.

Em relação ao desempenho, os nossos algoritmos se mostram competitivos comparados aos algoritmos da literatura que avaliamos, sendo que os Algoritmos Onda de Choque e Onda de Choque + TPN* apresentam resultados melhores dentro do intervalo de confiança em quase todos os cenários, atingindo desempenho de até 98,54% para o primeiro caso e de 98,96% para o segundo. O Algoritmo Onda de Choque + TPN* apresenta alguns resultados melhores até mesmo que o Algoritmo Onda Verde, que em teoria deveria ser o melhor dentre todos os analisados, sendo que em alguns casos o desempenho foi inferior por limitações do simulador. As razões para essa anomalia foram explicadas ao longo deste trabalho.

Em nosso modelo, consideramos a rota completa do EV, situação não considerada na maioria dos trabalhos aqui analisados. Assim, os resultados atestam que, com exceção dos *outliers* já analisados e descritos, o atraso mensurado pelo EV é de fato reduzido, em comparação com o Algoritmo linha de base Sem Preempção.

Dentro do que pudemos verificar, em relação aos trabalhos analisados, este é o único que incorpora em seu modelo o ingresso de veículos após o cálculo do momento futuro da preempção, apresentando uma análise de como esse comportamento afeta as soluções. Além disso, os algoritmos propostos permitem explicitamente que um ser humano consiga cancelar a preempção, comportamento não observado em outros trabalhos.

Nos trabalhos relacionados, há uma situação importante não considerada. No momento em que a preempção for iniciada, o semáforo leva alguns segundos para transitar entre as luzes amarela e vermelha para chegar, enfim, à luz verde desejada pelo EV. Essa transição leva um tempo e varia de acordo com cada semáforo, mas valores comuns variam entre 5 e 10 segundos. Os algoritmos avaliados não consideram esse tempo de transição, mas isso foi incorporado em nossos algoritmos. Dependendo do fluxo de entrada da via, com esse tempo em que o semáforo permanece fechado até sua total abertura, vá-

rios veículos podem chegar, e agora é necessário liberar uma fila de tamanho $Q + \Delta Q$. Qualquer algoritmo de preempção de semáforos terá esse comportamento, mas não encontramos trabalhos que abordem esse problema. Essa situação foi tratada no Algoritmo Onda de Choque + TPN* e é uma contribuição importante apresentada neste trabalho.

Para que os algoritmos pudessem ser avaliados em condições comparáveis, este trabalho provê um *framework* de simulação genérico para fazer a mediação entre o algoritmo e o simulador, que uniformiza o envio e recebimento de dados entre os dois e simplifica a implementação destes e de novos algoritmos de controle. O objetivo inicial é fornecer uma interface comum para o simulador SUMO, mas este pode ser usado para o envio de comandos para um *Middleware* Smartcity já implantado, como por exemplo, o *InterSCity* (BATISTA et al., 2016).

Por fim, em nosso trabalho, utilizamos cenários sintéticos criados com a ferramenta *OsmWebWizard* do SUMO para partes das cidades de Nova Iorque e São Paulo, e dois cenários reais presentes na literatura, das cidades de Turim - Itália e Colônia - Alemanha, respectivamente (RAPELLI; CASETTI; GAGLIARDI, 2022; VARSCHEN; WAGNER, 2006). Além disso, um novo cenário foi criado, utilizando os dados reais de viagens de origem e destino da Pesquisa do Metrô da Cidade de São Paulo (Companhia Metropolitana de São Paulo (METRÔ), 2017), sendo o maior cenário disponível para simulação no SUMO, tanto para o tamanho da rede quanto para o volume do tráfego, com aproximadamente 13 milhões de veículos em 24 horas. Deste cenário foi derivado um cenário um pouco menor, apenas com o Centro Expandido de São Paulo, com aproximadamente 300 mil veículos para uma janela de trinta minutos.

De forma resumida, portanto, esta pesquisa apresentou e avaliou soluções inovadoras para o problema de preempção semaforica para veículos de emergência, com destaque para a abordagem híbrida que combina o algoritmo de Onda de Choque com Redes de Petri Temporizadas estendidas (TPN*). Os resultados experimentais, obtidos através de simulações microscópicas extensivas em cenários reais (Turim, Colônia, Centro Expandido de São Paulo) e sintéticos (São Paulo, Nova Iorque), demonstraram que a solução proposta é altamente competitiva.

Em comparações com abordagens da literatura e com o cenário de linha de base (sem preempção), a solução Onda de Choque + TPN* atingiu reduções no tempo perdido pelo veículo de emergência de até 98,96%. Além disso, a solução mostrou-se capaz de superar, em diversos casos, a estratégia de Onda Verde, oferecendo um equilíbrio superior entre a rapidez do atendimento e a segurança do tráfego. Um diferencial importante validado pelos resultados

foi a capacidade do modelo de lidar com a inserção dinâmica de veículos na rede e considerar o tempo de transição das fases semafóricas, garantindo a corretude e a segurança das operações de preempção ao longo de toda a rota.

Na próxima Seção, serão apresentadas as Limitações encontradas durante o desenvolvimento deste trabalho.

7.2 Limitações e Trabalhos Futuros

Apesar dos avanços alcançados, o modelo proposto e a metodologia de avaliação apresentam limitações que delineiam oportunidades claras para a continuidade da pesquisa.

7.2.1 Representatividade do Modelo de Car-Following e Comportamento Humano

As validações foram realizadas em ambiente de simulação microscópica (SUMO), regido por modelos de *Car-Following* (KRAUSS, 1998; TREIBER; HENNECKE; HELBING, 2000; WIEDEMANN; VERKEHRSWESSEN; KÖHLER, 1974). Embora robustos para o fluxo geral, tais modelos são normativos e não capturam com fidelidade a imprevisibilidade da reação humana diante de sirenes (o “efeito mar vermelho”, hesitações ou manobras irregulares).

Para trabalhos futuros, a calibração dos parâmetros do modelo de *Car-Following* com dados de telemetria reais de interações entre motoristas e EVs é um passo natural. Adicionalmente, o uso de modelos comportamentais baseados em agentes mais complexos poderia simular a “cooperação imperfeita” dos motoristas, aumentando o realismo dos cenários de teste.

7.2.2 Inteligência Artificial e Aprendizado de Máquina

Soluções baseadas exclusivamente em Inteligência Artificial, embora eficientes em média, carecem de determinismo e garantias formais de segurança — atributos essenciais para sistemas críticos. O modelo atual prioriza essa segurança por meio de Redes de Petri, mas pode se beneficiar da capacidade de adaptação dinâmica de abordagens baseadas em dados.

Neste contexto, a integração de abordagens de Inteligência Artificial (como *Safe Reinforcement Learning*) sob a supervisão do modelo formal de Redes de Petri pode ser considerada (CHENG et al., 2025; XIAO et al., 2022; WEI et al., 2018; YAN et al., 2020). Tal integração é viável e recomendada em dois cenários específicos:

- **Otimização do Tempo de Preempção:** Uma vez assegurada a integridade da rede pelo modelo formal, a IA atuaria para refinar o cálculo

do tempo estimado, antecipando a abertura do sinal com base em padrões históricos, desde que restrições sejam implementadas para impedir a convergência para soluções triviais (como uma “Onda Verde” cega).

- **Identificação de Bloqueios Latentes:** Após o cálculo inicial de segurança, a IA pode identificar padrões latentes nos dados que indiquem a ineficácia da ação devido a tráfego parado à frente (que o modelo físico pode não ter previsto). Nesse caso, o sistema abortaria a preempção preventivamente para evitar o agravamento do congestionamento.

7.2.3 Escalonamento e Concorrência de Múltiplos EVs

A solução foi otimizada para o atendimento prioritário de um único EV por vez na rota. O sistema atual utiliza uma política de “última ordem recebida” para lidar com múltiplos chamados, carecendo de uma lógica de priorização baseada na gravidade da ocorrência.

Portanto, o desenvolvimento de um orquestrador centralizado capaz de gerenciar conflitos de rotas entre múltiplos EVs, aplicando algoritmos de negociação ou priorização baseados em parâmetros da ocorrência, deve ser considerado.

7.2.4 Granularidade da Preempção e Impacto no Entorno

A reserva de “rota completa” garante fluidez máxima ao EV, mas pode gerar ociosidade em interseções distantes se ativada muito precocemente.

Por isso, a implementação de uma estratégia de “Janela Deslizante” (*Sliding Window*) poderia permitir ao sistema controlar dinamicamente apenas um subconjunto de semáforos à frente do EV (ex: raio de 1 km), liberando as interseções já cruzadas e reservando as futuras conforme o deslocamento real.

7.2.5 Validação em Infraestrutura Real (V2I/V2V/VTL)

O modelo assume uma infraestrutura de comunicação ideal. No mundo real, a latência de rede e as falhas de sensores são variáveis críticas não completamente capturadas na simulação.

Como trabalho futuro, a realização de Provas de Conceito (PoC) integrando o algoritmo a *middlewares* reais de Cidades Inteligentes (como o *InterSCity* (BATTISTA et al., 2016)) é essencial para validar a robustez da solução frente a falhas de hardware e latência de comunicação. E o desenvolvimento natural do trabalho é estender essa implementação para situações de utilização de comunicação V2V/V2I/VTL.

7.2.6 Fidelidade dos Cenários e Atualização da Base de Dados

Os cenários foram criados utilizando dados reais da Pesquisa Origem e Destino do Metrô de 2017 (Companhia Metropolitana de São Paulo (METRÔ), 2017). Contudo, uma nova versão da pesquisa, cujos dados foram disponibilizados em 2025 (Companhia Metropolitana de São Paulo (METRÔ), 2023), aponta uma redução geral nos deslocamentos pós-pandemia de Covid-19, realidade não refletida nos cenários atuais. Além disso, características da rede como ciclos e clusterização semaforica foram inferidas via ferramentas do SUMO, e as viagens compartilhadas basearam-se apenas em zonas OD, sem a granularidade do transporte público real.

O próximo passo é, portanto, adaptar o trabalho para utilizar a base de dados atualizada (2023/2025) e adequar os cenários inferidos aos equivalentes da vida real (ciclos semaforicos reais e dados de transporte público). Como contribuição final, pretendemos que os cenários construídos e calibrados a partir da Pesquisa Origem e Destino constem oficialmente na página de cenários do SUMO¹, servindo de *benchmark* para a comunidade científica.

7.3 Produção Bibliográfica

Os conceitos, modelagens e resultados parciais desenvolvidos ao longo desta tese foram submetidos à avaliação da comunidade científica e publicados em eventos relevantes da área de Pesquisa Operacional e Sistemas Inteligentes. Destaca-se a publicação do trabalho completo intitulado 'Controle de Semáforos Para Veículos de Emergência' nos anais do LIV Simpósio Brasileiro de Pesquisa Operacional (BRANCO; FERREIRA; CÁCERES, 2022).

Esta publicação validou a modelagem inicial do problema e a proposta de utilização de Redes de Petri para a coordenação segura dos semáforos, servindo de base para o refinamento dos algoritmos apresentados nos capítulos subsequentes desta tese.

Além disso, os resultados parciais foram disseminados no Workshop do INCT em 2020².

Adicionalmente, os resultados finais e a abordagem unificada proposta nesta tese foram consolidados no artigo científico intitulado "*Traffic Light Control for Emergency Vehicles*", que se encontra em submissão para publicação.

¹<https://sumo.dlr.de/docs/Data/Scenarios.html>

²<https://intercity.org/events/seminarios-intercity-sao-realizados-digitalmente-durante-pandemia/>

Referências Bibliográficas

- AGBELIE, B. R.; ROSHANDEH, A. M. Impacts of Signal-Related Characteristics on Crash Frequency at Urban Signalized Intersections. *Journal of Transportation Safety & Security*, Taylor & Francis, v. 7, n. 3, p. 199–207, 2015. Disponível em: <<https://doi.org/10.1080/19439962.2014.943867>>. Citado na página 5.
- ALBAGUL, A.; HRAIRI, M.; HIDAYATHULLAH, M. F. Design and Development of Sensor Based Traffic Light System. *American Journal of Applied Sciences*, Science Publications, v. 3, p. 1745–1749, Mar 2006. Disponível em: <<https://thescipub.com/abstract/ajassp.2006.1745.1749>>. Citado nas páginas 27 e 30.
- ALMURAYKHI, K. M.; AKHLAQ, M. STLS: Smart Traffic Lights System for Emergency Response Vehicles. In: *2019 International Conference on Computer and Information Sciences (ICCIS)*. Sakaka, Arábia Saudita: IEEE, 2019. p. 1–6. Citado nas páginas 28 e 30.
- ALZAMZAMI, O. et al. Passable: An Intelligent Traffic Light System with Integrated Incident Detection and Vehicle Alerting. *Sensors*, v. 25, n. 18, 2025. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/25/18/5760>>. Citado nas páginas 28 e 30.
- AXHAUSEN, K. W.; HORNI, A.; NAGEL, K. *The multi-agent transport simulation MATSim*. Londres, Reino Unido: Ubiquity Press, 2016. Citado na página 42.
- BAI, W. et al. Equilibrium Traffic Guidance Strategy Based on Queuing Theory for Emergency Vehicles. In: *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. Praga, República Checa: IEEE, 2022. p. 462–467. Citado nas páginas 28 e 30.
- BANIARDALANI, S.; SOLEIMANI, S. Modeling and Analysis of Preemption System for Emergency Vehicles using Timed Petri Nets. *Journal of Modeling in Engineering*, Semnan University Press, v. 23, n. 81, p. 139–156, 2025. ISSN 2008-4854. Disponível em: <https://modelling.semnan.ac.ir/article_9625.html>. Citado nas páginas 28 e 30.
- BARCELÓ, J. *Fundamentals of traffic simulation*. Nova York, NY, EUA: Springer, 2010. v. 145. Citado na página 42.
- BATISTA, D. M. et al. InterSCity: Addressing Future Internet research challenges for Smart Cities. In: *2016 7th International Conference on the*

Network of the Future (NOF). Buzios, Brasil: IEEE, 2016. p. 1–6. Citado nas páginas 9, 25, 26, 155, e 157.

BORJA, R.; GAMA, K. Middleware para Cidades Inteligentes baseado em um Barramento de Serviços. In: *Anais do X Simpósio Brasileiro de Sistemas de Informação*. Porto Alegre, RS: SBC, 2014. p. 584–590. ISSN 0000-0000. Disponível em: <<https://sol.sbc.org.br/index.php/sbsi/article/view/6147>>. Citado na página 24.

BRANCO, R. G. d.; FERREIRA, R. A.; CÁCERES, E. N. Controle de Semáforos Para Veículos de Emergência. In: *Anais do LIV Simpósio Brasileiro de Pesquisa Operacional (SBPO)*. Juiz de Fora, MG: Galoá, 2022. Disponível em: <<https://proceedings.science/sbpo/sbpo-2022/trabalhos/controle-de-semaforos-para-veiculos-de-emergencia?lang=pt-br>>. Acesso em: 21 de dezembro de 2025. Citado nas páginas 99 e 158.

BRASIL. Código de trânsito brasileiro - lei nº 9.503/97. *Diário Oficial da União*, Brasília, DF, 1997. ISSN 1677-7042. Disponível em: <http://www.planalto.gov.br/ccivil_03/leis/l9503compilado.htm>. Citado na página 2.

BRASIL. *Leilão do 5G confirma expectativas e arrecada R\$ 47,2 bilhões*. 2021. Disponível em: <<https://www.gov.br/pt-br/noticias/transito-e-transportes/2021/11/leilao-do-5g-confirma-expectativas-e-arrecada-r-47-2-bilhoes>>. Acesso em: 16 de abril de 2025. Citado na página 11.

BUENO, E. *Evento de Smart Cities no MS apresenta plano de Cidade Inteligente de Campo Grande*. 2021. Disponível em: <<https://portal.connectedsmartcities.com.br/2021/03/10/evento-de-smart-cities-no-ms-apresenta-plano-de-cidade-inteligente-de-campo-grande/>>. Acesso em: 25 de abril de 2025. Citado na página 25.

BURSZTYN, M.; ARAÚJO, C. *Da utopia à exclusão: vivendo nas ruas em Brasília*. Garamond, 1997. Disponível em: <<https://books.google.com.br/books?id=2N00hCB3EOgC>>. Citado na página 1.

CHAKRABORTY, P. S.; TIWARI, A.; SINHA, P. R. Adaptive and Optimized Emergency Vehicle Dispatching Algorithm for Intelligent Traffic Management System. *Procedia Computer Science*, v. 57, p. 1384–1393, 2015. ISSN 1877-0509. 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015). Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1877050915019833>>. Citado nas páginas 27 e 30.

CHENG, M. et al. Role-aware Multi-agent Reinforcement Learning for Coordinated Emergency Traffic Control. In: *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. NeurIPS Foundation, 2025. Disponível em: <<https://openreview.net/forum?id=R3xbcRlzUd>>. Citado nas páginas 28, 30, e 156.

CHIOLA, G. et al. GreatSPN 1.7: Graphical editor and analyzer for timed and stochastic Petri nets. *Performance Evaluation*, v. 24, n. 1, p. 47–68, 1995. ISSN 0166-5316. Performance Modeling Tools. Disponível em: <<https://www.sciencedirect.com/science/article/pii/016653169500008L>>. Citado nas páginas 120 e 123.

CHOI, E.-H. *Crash factors in intersection-related crashes: an on-scene perspective*. 1200 New Jersey Avenue SE., Washington DC, EUA, 2010. 37 p. Citado na página 5.

CHOU, L.-D.; TSENG, J.-H.; YANG, J.-Y. Adaptive virtual traffic light based on vanets for mitigating congestion in smart city. In: THE SOCIETY OF DIGITAL INFORMATION AND WIRELESS COMMUNICATION. *The Third International Conference on Digital Information and Communication Technology and its Applications (DICTAP2013)*. Bangucoque, Tailândia, 2013. p. 40–44. Citado nas páginas 24, 27, e 30.

COHEN, B. *IEEE 1609-family of standards for wireless access in vehicular environments (wave)*. Atlanta, GA, EUA: January, 2006. Citado na página 22.

COLLURA, J. et al. Traffic Signal Preemption and Priority: Technologies, Past Deployments, and System Requirements. In: *ITS America 11th Annual Meeting and Exposition, ITS: Connecting the Americas*. Miami Beach, FL, EUA: ITS, 2001. Citado na página 21.

Companhia Metropolitana de São Paulo (METRÔ). *Pesquisa Origem e Destino 2017*. São Paulo: Metrô, 2017. Disponível em: <https://transparencia.metrosp.com.br/dataset/pesquisa-origem-e-destino>. Acesso em: 16 de abril de 2025. Citado nas páginas xviii, xxii, 50, 52, 143, 155, 158, e 176.

Companhia Metropolitana de São Paulo (METRÔ). *Pesquisa Origem e Destino 2023*. São Paulo: Metrô, 2023. Disponível em: <https://www.metro.sp.gov.br/pesquisa-od/>. Acesso em: 28 de junho de 2025. Citado na página 158.

Connectivity Standards Alliance (CSA). *Zigbee | Complete IOT Solution*. 2025. Disponível em: <https://csa-iot.org/all-solutions/zigbee/>. Acesso em: 25 de abril de 2025. Citado na página 19.

Conselho Nacional do Trânsito (CONTRAN). Resolução nº 970/22. *Diário Oficial da União*, Brasília, DF, 2022. ISSN 1677-7042. Disponível em: <https://www.gov.br/transportes/pt-br/assuntos/transito/conteudo-contran/resolucoes/resolucao9702022.pdf>. Citado na página 2.

Digi. *Digi XBee RF Modules*. 2025. Disponível em: <https://www.digi.com/products/embedded-systems/digi-xbee/rf-modules>. Acesso em: 16 de abril de 2025. Citado na página 19.

Digital Dubai. *Digital Dubai Home Page*. 2025. Disponível em: <https://www.digitaldubai.ae/>. Acesso em: 25 de abril de 2025. Citado na página 25.

DISBRO, J. E.; FRAME, M. *Traffic flow theory and chaotic behavior*. Nova York, NY, EUA, 1989. Citado nas páginas 6 e 15.

DJAHEL, S. et al. Reducing emergency services response time in smart cities: An advanced adaptive and fuzzy approach. In: *2015 IEEE First International Smart Cities Conference (ISC2)*. Guadalajara, México: IEEE, 2015. p. 1–8. Citado nas páginas 22, 30, 35, 37, 43, 89, 129, 134, e 171.

ELTAYEB, A. S.; ALMUBARAK, H. O.; ATTIA, T. A. A GPS based traffic light pre-emption control system for emergency vehicles. In: *2013 International Conference on Computing, Electrical and Electronic Engineering (ICCEEE)*. Cartum, Sudão: IEEE, 2013. p. 724–729. Citado nas páginas 28 e 30.

Escola São Paulo de Ciência Avançada. *São Paulo School of Advanced Science on Smart Cities*. 2017. Disponível em: <https://espca.fapesp.br/escola/sao_paulo_school_of_advanced_science_on_smart_cities/62/>. Acesso em: 25 de abril de 2025. Citado na página 25.

ESPOSTE, A. M. D. et al. InterSCity: A Scalable Microservice-based Open Source Platform for Smart Cities. In: INSTICC. *Proceedings of the 6th International Conference on Smart Cities and Green ICT Systems - Volume 1: SMARTGREENS*,. Porto, Portugal: SciTePress, 2017. p. 35–46. ISBN 978-989-758-241-7. Citado na página 26.

ETSI. *Intelligent Transport Systems (ITS); Testing; Conformance test specifications for GeoNetworking ITS-G5; Part 3: Abstract Test Suite (ATS) and Protocol Implementation eXtra Information for Testing (PIXIT)*. 650 Route des Lucioles, F-06921 Sophia Antipolis Cedex - França, 2017. 31 p. Citado na página 22.

EXAME. *Frota de automóveis no Brasil é a mais velha dos últimos 25 anos*. 2021. Disponível em: <<https://exame.com/negocios/frota-de-automoveis-no-brasil-e-a-mais-velha-dos-ultimos-25-anos/>>. Acesso em: 16 de abril de 2025. Citado na página 11.

FAYOLLAS, C. et al. Fault-Tolerant Interactive Cockpits for Critical Applications: Overall Approach. In: AVGERIOU, P. (Ed.). *Software Engineering for Resilient Systems*. Berlim, Alemanha: Springer Berlin Heidelberg, 2012. p. 32–46. ISBN 978-3-642-33176-3. Citado na página 65.

FERREIRA, B. *Análise formal de redes veiculares com verificação de modelos probabilística*. Tese (Doutorado) — Universidade Federal de Minas Gerais, setembro 2016. Citado na página 24.

FERREIRA, M. et al. Self-organized traffic control. In: *Proceedings of the Seventh ACM International Workshop on VehiculAr InterNETworking*. Nova York, NY, EUA: Association for Computing Machinery, 2010. (VANET '10), p. 85–90. ISBN 9781450301459. Disponível em: <<https://doi.org/10.1145/1860058.1860077>>. Citado na página 23.

G1 Amazonas. *Ambulância capota após colisão com carro entre avenidas da Zona Sul de Manaus*. 2017. Disponível em: <<https://g1.globo.com/am/amazonas/transito/noticia/ambulancia-capota-apos-colisao-com-carro-entre-avenidas-da-zona-sul-de-manaus.ghtml>>. Acesso em: 16 de abril de 2025. Citado na página 3.

G1 DF. *Engarrafamento faz morador do DF passar 19% mais tempo no trânsito*. 2016. Disponível em: <<http://g1.globo.com/distrito-federal/noticia/2016/03/engarrafamento-faz-morador-do-df-passar-19-mais-tempo-no-transito.html>>. Acesso em: 25 de abril de 2025. Citado na página 1.

G1 Presidente Prudente. *Ambulância se envolve em acidente a caminho de atendimento a grávida*. 2017. Disponível em: <<http://g1.globo.com/sp/presidente-prudente-regiao/noticia/2017/03/ambulancia-se-envolve-em-acidente-caminho-de-atendimento-gravida.html>>. Acesso em: 16 de abril de 2025. Citado na página 3.

G1 São Paulo. *SP bate recorde histórico com 344 km de vias congestionadas, diz CET*. 2014. Disponível em: <<http://g1.globo.com/sao-paulo/noticia/2014/05/sp-bate-recorde-historico-com-344-km-de-vias-congestionadas-diz-cet.html>>. Acesso em: 25 de abril de 2025. Citado nas páginas xvii e 2.

GEDAWY, H. K.; DIAS, M.; HARRAS, K. Dynamic path planning and traffic light coordination for emergency vehicle routing. *Comp. Sci. Dept., Carnegie Mellon Univ., Pittsburgh, Pennsylvania, USA*, v. 189, 2008. Citado nas páginas 28 e 30.

GERLA, M. et al. Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds. In: *2014 IEEE World Forum on Internet of Things (WF-IoT)*. Seul, Coréia do Sul: IEEE, 2014. p. 241–246. Citado na página 22.

GILBERTO, E. A.; DONATELLI, S. MC4CSLTA: An Efficient Model Checking Tool for CSLTA. In: *2010 Seventh International Conference on the Quantitative Evaluation of Systems*. Williamsburg, VA, EUA: IEEE, 2010. p. 153–154. Citado nas páginas 120 e 124.

GRAHAM, L. A. et al. *Analysis of Ground Ambulance Crash Data From 2012 to 2018*. Washington, DC, EUA, 2023. Citado na página 4.

GREENSHIELDS, B. D. et al. A study of traffic capacity. In: *Highway research board proceedings*. Washington, DC: Highway Research Board, 1935. v. 14, n. 1, p. 448–477. Citado nas páginas xviii, 82, e 83.

HAGHANI, M.; KLUGER, R. Developing a Shockwave-Based Queue Estimation Model to Quantify the Disruption Created by Emergency Vehicle Preemption on Arterial Cross Streets. *Journal of Transportation Engineering, Part A: Systems*, v. 151, n. 11, p. 04025092, 2025. Disponível em: <<https://ascelibrary.org/doi/abs/10.1061/JTEPBS.TEENG-9220>>. Citado nas páginas 28 e 30.

HAJIEBRAHIMI, S.; IRANMANESH, S. An Adaptive Control Method of Traffic Signal-Timing under Emergency Situations for Smart Cities. In: *2018 3rd IEEE International Conference on Intelligent Transportation Engineering (ICITE)*. Singapura: IEEE, 2018. p. 225–230. Citado nas páginas 28 e 30.

HASHIM, N. et al. Traffic light control system for emergency vehicles using radio frequency. *IOSR Journal of Engineering (IOSRJEN)*, v. 3, n. 7, p. 43–52, 2013. Citado nas páginas 27, 30, e 36.

HEGDE, R.; SALI, R. R.; INDIRA, M. RFID and GPS based automatic lane clearance system for ambulance. *International Journal of Advanced Electrical and Electronics Engineering*, v. 2, n. 3, p. 102–107, 2013. Citado na página 36.

HUANG, Y.-S.; WENG, Y.-S.; ZHOU, M. Design of Traffic Safety Control Systems for Emergency Vehicle Preemption Using Timed Petri Nets. *IEEE Transactions on Intelligent Transportation Systems*, v. 16, n. 4, p. 2113–2120, 2015. Citado nas páginas 28, 30, 66, e 110.

HUMAGAIN, S.; SINHA, R. Dynamic Prioritization of Emergency Vehicles For Self-Organizing Traffic using VTL+EV. In: *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*. Singapura: IEEE, 2020. p. 789–794. Citado nas páginas 24, 27, e 30.

HUSSIN, R. et al. Wireless Traffic Light Controller for Emergency Vehicle through XBee and Basic Stamp Microcontroller. *Procedia Engineering*, v. 41, p. 636–642, 2012. ISSN 1877-7058. International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012). Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877705812026239>. Citado nas páginas 27 e 30.

IEEE Standard for Information Technology-Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks-Specific Requirements. *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. Nova York, NY, EUA: IEEE, 2020. (IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016)). Citado na página 22.

IMPINJ. *Types of RFID Systems*. 2025. Disponível em: <https://www.impinj.com/products/technology/how-can-rfid-systems-be-categorized>. Acesso em: 16 de abril de 2025. Citado nas páginas 20 e 55.

Instituto Brasileiro de Geografia e Estatística. *IBGE | Cidades@ | Distrito Federal | Brasília | Panorama*. 2025. Disponível em: <https://cidades.ibge.gov.br/brasil/df/brasil/panorama>. Acesso em: 16 de abril de 2025. Citado na página 1.

KAMALANATHSHARMA, R. K.; HANCOCK, K. L. *Congestion-based emergency vehicle preemption*. Pennsylvania, PA, EUA, 2010. Disponível em: <https://rosap.ntl.bts.gov/view/dot/25651>. Citado na página 84.

KAPUSTA, B. et al. Preemptive traffic light control based on vehicle tracking and queue lengths. In: *2017 International Symposium ELMAR*. Zadar, Croácia: IEEE, 2017. p. 11–16. Citado nas páginas xviii, 30, 35, 39, 40, 47, 66, 81, 82, 84, 86, 87, 89, 90, 93, 107, 110, e 137.

KHAN, A. et al. EVP-STC: Emergency Vehicle Priority and Self-Organising Traffic Control at Intersections Using Internet-of-Things Platform. *IEEE Access*, v. 6, p. 68242–68254, 2018. Citado nas páginas 28 e 30.

KNUTH, E. *The Art of Computer Programming, Volume 3*. Boston, MA, EUA: Addison-Wesley, 1973. 123–129 p. Citado na página 73.

KRAUSS, S. *Microscopic modeling of traffic flow: investigation of collision free vehicle dynamics*. 1998. Citado na página 156.

- LI, D. et al. Adaptive Traffic Signal Control Model on Intersections Based on Deep Reinforcement Learning. *Journal of Advanced Transportation*, v. 2020, n. 1, p. 6505893, 2020. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1155/2020/6505893>>. Citado na página 10.
- LIGHTHILL, M. J.; WHITHAM, G. B. On kinematic waves II. A theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, v. 229, n. 1178, p. 317–345, 05 1955. ISSN 0080-4630. Disponível em: <<https://doi.org/10.1098/rspa.1955.0089>>. Citado nas páginas xviii, 82, e 83.
- LOPEZ, P. A. et al. Microscopic Traffic Simulation using SUMO. In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. Maui, HI, EUA: IEEE, 2018. p. 2575–2582. Citado nas páginas 43 e 85.
- MARSHALL, P. S.; BERG, W. D. Design guidelines for railroad preemption at signalized intersections. *Institute of Transportation Engineers. ITE Journal*, v. 67, n. 2, p. 20, 02 1997. Disponível em: <<https://www.proquest.com/scholarly-journals/design-guidelines-railroad-preemption-at/docview/224878600/se-2>>. Acesso em: 16 de abril de 2025. Citado nas páginas 84 e 85.
- MERLIN, P.; FARBER, D. Recoverability of Communication Protocols - Implications of a Theoretical Study. *IEEE Transactions on Communications*, v. 24, n. 9, p. 1036–1043, 1976. Citado na página 64.
- METROCOUNT. *Traffic data collection with induction loops*. 2020. Disponível em: <<https://www.metrocount.com/traffic-data-collection-induction-loops>>. Acesso em: 25 de abril de 2025. Citado nas páginas xvii e 20.
- Miovision Technologies Incorporated. *Miovision Opticom™ Emergency Vehicle Preemption*. 2025. Disponível em: <<https://miovision.com/opticom/emergency-vehicle-preemption/>>. Acesso em: 16 de abril de 2025. Citado na página 19.
- MISBAHUDDIN, S. et al. IoT based dynamic road traffic management for smart cities. In: *2015 12th International Conference on High-capacity Optical Networks and Enabling/Emerging Technologies (HONET)*. Islamabad, Paquistão: IEEE, 2015. p. 1–5. Citado nas páginas 28 e 30.
- MITTAL, A. K.; BHANDARI, D. A novel approach to implement green wave system and detection of stolen vehicles. In: *2013 3rd IEEE International Advance Computing Conference (IACC)*. Gaziabade, Índia: IEEE, 2013. p. 1055–1059. Citado na página 36.
- MIYAWAKI, M.; YAMASHIRO, Z.; YOSHIDA, T. Fast emergency pre-emption systems (FAST). In: *Proceedings 199 IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems (Cat. No.99TH8383)*. Tóquio, Japão: IEEE, 1999. p. 993–997. Citado nas páginas 27 e 30.
- MOHAMED, B. et al. Dynamic Traffic Light Control For Intelligent Mobility In Smart Cities. *Journal of Theoretical & Applied Information Technology*, v. 73, n. 2, 2015. Citado na página 22.

- MOROI, Y.; TAKAMI, K. A method of securing priority-use routes for emergency vehicles using inter-vehicle and vehicle-road communication. In: *2015 7th International Conference on New Technologies, Mobility and Security (NTMS)*. Paris, França: IEEE, 2015. p. 1–5. Disponível em: <https://ieeexplore.ieee.org/abstract/document/7266466>. Citado nas páginas 27 e 30.
- MU, H.; SONG, Y.; LIU, L. Route-Based Signal Preemption Control of Emergency Vehicle. *Journal of Control Science and Engineering*, v. 2018, n. 1, p. 1024382, 2018. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1155/2018/1024382>. Citado nas páginas 28 e 30.
- MURATA, T. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, v. 77, n. 4, p. 541–580, 1989. Citado nas páginas xxi, 63, 120, 121, 122, e 124.
- National Safety Council. *Emergency vehicles - injury facts*. 2023. Disponível em: <https://injuryfacts.nsc.org/motor-vehicle/road-users/emergency-vehicles/>. Acesso em: 16 de abril de 2025. Citado na página 3.
- NOGRADY, B. *Smart city: using technology to tackle traffic and social isolation in Melbourne*. 2017. Disponível em: <https://www.theguardian.com/sustainable-business/2017/jun/19/smart-city-traffic-social-isolation-melbourne-technology>. Acesso em: 16 de abril de 2025. Citado na página 25.
- OLAVERRI-MONREAL, C. et al. In-Vehicle Virtual Traffic Lights: A graphical user interface. In: *7th Iberian Conference on Information Systems and Technologies (CISTI 2012)*. Madri, Espanha: IEEE, 2012. p. 1–6. Citado nas páginas xvii e 24.
- OpenStreetMap contributors. *Planet dump retrieved from https://planet.osm.org*. 2025. Disponível em: <https://www.openstreetmap.org>. Acesso em: 16 de abril de 2025. Citado na página 44.
- Organização Mundial da Saúde (OMS). *Global status report on road safety 2018*. Genebra, Suíça: Organização Mundial da Saúde, 2018. 403 p. ISBN 978-92-4-156568-4. Citado na página 5.
- PANIATI, J. F.; AMONI, M. *Traffic signal preemption for emergency vehicles: a cross-cutting study*. Washington, DC, EUA: U.S. Department of Transportation, 2006. Citado nas páginas 4 e 5.
- PARUCHURI, V. Adaptive Preemption of Traffic for Emergency Vehicles. In: *2017 UKSim-AMSS 19th International Conference on Computer Modelling & Simulation (UKSim)*. Cambridge, Reino Unido: IEEE, 2017. p. 45–49. Citado nas páginas 28 e 30.
- PEI, Y. et al. Modeling and Analyzing of Emergency Vehicle Preemption in a Fourphase Intersection via TPN. *Journal of Traffic and Logistics Engineering* Vol, v. 8, n. 2, 2020. Citado nas páginas 28 e 30.
- PETERSON, J. L. *Petri net theory and the modeling of systems*. Armonk, NY, EUA: Prentice Hall PTR, 1981. Citado na página 124.

PICONE, M.; AMORETTI, M.; ZANICHELLI, F. Simulating smart cities with DEUS. In: *Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques*. Bruxelas, Bélgica: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2012. (SIMUTOOLS '12), p. 172–177. ISBN 9781450315104. Citado na página 42.

PISHUE, B. *2022 INRIX Global Traffic Scorecard*. INRIX, 2023. Disponível em: <<https://inrix.com/scorecard/>>. Citado na página 1.

POMMEREAU, F. Snakes: A flexible high-level petri nets library (tool paper). In: DEVILLERS, R.; VALMARI, A. (Ed.). *Application and Theory of Petri Nets and Concurrency*. Bruxelas, Bélgica: Springer International Publishing, 2015. p. 254–265. ISBN 978-3-319-19488-2. Citado na página 128.

Portal Unibus. *Em ritmo lento: Veja um diagnóstico de como está o trânsito de Natal*. 2013. Disponível em: <<https://portalunibus.com.br/2013/02/24/em-ritmo-lento-veja-um-diagnostico-de/>>. Acesso em: 16 de abril de 2025. Citado nas páginas xvii e 3.

QIN, X.; KHAN, A. M. Control strategies of traffic signal timing transition for emergency vehicle preemption. *Transportation Research Part C: Emerging Technologies*, v. 25, p. 1–17, 2012. ISSN 0968-090X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0968090X1200054X>>. Citado na página 30.

RAPELLI, M.; CASETTI, C.; GAGLIARDI, G. Vehicular Traffic Simulation in the City of Turin From Raw Data. *IEEE Transactions on Mobile Computing*, IEEE, v. 21, n. 12, p. 4656–4666, 2022. Citado nas páginas 48, 52, e 155.

RGH Pacific Inc. *Sonem 2000 in Action*. 2025. Disponível em: <<https://www.rghpacific.com/rghpacific-sonem2000.html>>. Acesso em: 16 de abril de 2025. Citado na página 18.

ROW, T.-C.; LEE, S.-C.; CHENG, J.-F. Designing Traffic Light Signals to Ease Congestion Caused by Traffic Accidents Using Timed Petri Net. *Applied Sciences*, v. 14, n. 23, 2024. ISSN 2076-3417. Disponível em: <<https://www.mdpi.com/2076-3417/14/23/11260>>. Citado nas páginas 28 e 30.

SAHANA, N. et al. Dynamic traffic light control for intelligent mobility of emergency vehicles. *Bonfring International Journal of Software Engineering and Soft Computing*, Bonfring, v. 6, n. Special Issue Special Issue on Advances in Computer Science and Engineering and Workshop on Big Data Analytics Editors: Dr. SB Kulkarni, Dr. UP Kulkarni, Dr. SM Joshi and JV Vadavi, p. 167–170, 2016. Disponível em: <<http://www.journal.bonfring.org/abstract.php?id=7&archiveid=724>>. Citado nas páginas 22, 28, e 30.

SANTANA, E. F. Z. et al. Interscsimulator: Large-scale traffic simulation in smart cities using erlang. In: DIMURO, G. P.; ANTUNES, L. (Ed.). *Multi-Agent Based Simulation XVIII*. São Paulo, Brasil: Springer International Publishing, 2018. p. 211–227. ISBN 978-3-319-91587-6. Citado na página 42.

SHANKAR, V.; GAUTHAM, R.; VEDAPRAKASHVARMA. Automated traffic signal for hassle free movement of ambulance. In: *2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*. Coimbatore, Índia: IEEE, 2015. p. 1–5. Citado na página 30.

SHARMA, S. et al. Traffic light priority control for emergency vehicle using RFID. *Int. J. Innov. Eng. Technol*, Citeseer, v. 2, n. 2, p. 363–366, 2013. Citado nas páginas 24, 27, 30, e 36.

SIDDIQI, M. H. et al. Development of a Smart Signalization for Emergency Vehicles. *Sensors*, v. 23, n. 10, 2023. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/23/10/4703>. Citado nas páginas 28 e 30.

SMITH, M. R.; DAVIDSON, P. J.; PFISTER, H. L. *Emergency vehicle warning and traffic control system*. [S.l.]: Google Patents, 1988. US Patent 4,775,865. Citado na página 30.

Society of Automotive Engineers. *SAE J2735 - Dedicated short range communications (DSRC) message set dictionary*. Warrendale, PA, EUA, 2009. Disponível em: <https://www.standards.its.dot.gov/Factsheets/Factsheet/71>. Citado na página 22.

STEPHANOPOULOS, G.; MICHALOPOULOS, P. G.; STEPHANOPOULOS, G. Modelling and analysis of traffic queue dynamics at signalized intersections. *Transportation Research Part A: General*, v. 13, n. 5, p. 295–307, 1979. ISSN 0191-2607. Disponível em: <https://www.sciencedirect.com/science/article/pii/0191260779900281>. Citado nas páginas 84 e 85.

SUNDAR, R.; HEBBAR, S.; GOLLA, V. Implementing Intelligent Traffic Control System for Congestion Control, Ambulance Clearance, and Stolen Vehicle Detection. *IEEE Sensors Journal*, v. 15, n. 2, p. 1109–1113, 2015. Citado nas páginas 27 e 30.

THOMPSON, W. R. On Confidence Ranges for the Median and Other Expectation Distributions for Populations of Unknown Distribution Form. *The Annals of Mathematical Statistics*, Institute of Mathematical Statistics, v. 7, n. 3, p. 122–128, 1936. ISSN 00034851, 21688990. Disponível em: <http://www.jstor.org/stable/2957563>. Citado na página 60.

TOH, C. *Ad hoc mobile wireless networks: protocols and systems*. Pearson Education, 2001. ISBN 9780132442046. Disponível em: <https://books.google.com.br/books?id=9fUchlqTXDQC>. Citado na página 22.

TONGUZ, O. Biologically inspired solutions to fundamental transportation problems. *IEEE Communications Magazine*, v. 11, n. 49, p. 106–115, 2011. Citado na página 24.

Transportation Research Board. Special Report 209: Highway Capacity Manual. *TRB, Washington, DC*, Washington, DC, EUA, 1985. Disponível em: <https://onlinepubs.trb.org/Onlinepubs/sr/sr209/209.pdf>. Acesso em: 10 de setembro de 2023. Citado na página 84.

TREIBER, M.; HENNECKE, A.; HELBING, D. Congested traffic states in empirical observations and microscopic simulations. *Phys. Rev. E*, American Physical Society, v. 62, p. 1805–1824, Aug 2000. Disponível em: <https://link.aps.org/doi/10.1103/PhysRevE.62.1805>. Citado na página 156.

UNIBASO, G. et al. A novel CAM-based traffic light preemption algorithm for efficient guidance of emergency vehicles. In: *13th International IEEE Conference on Intelligent Transportation Systems*. Funchal, Portugal: IEEE, 2010. p. 74–79. Citado nas páginas 22, 27, e 30.

VARSCHEN, C.; WAGNER, P. Mikroskopische Modellierung der Personenverkehrsnachfrage auf Basis von Zeitverwendungstagebüchern. In: BECKMANN, K. J. (Ed.). *Integrierte Mikro-Simulation von Raum- und Verkehrsentwicklung. Theorie, Konzepte, Modelle, Praxis*. Institut für Stadtbauwesen und Stadtverkehr, RWTH Aachen, 2006. (Stadt Region Land, v. 81), p. 63–69. ISBN 3-88354-149-4. Disponível em: <https://elib.dlr.de/45058/>. Citado nas páginas 49, 52, 139, e 155.

VUJIĆ, M. *Dynamic priority systems for public transport in urban automatic traffic control*. Tese (Doutorado) — Universidade de Zagreb, 2013. Citado na página 47.

WATANABE, B. L. et al. Is Use of Warning Lights and Sirens Associated With Increased Risk of Ambulance Crashes? A Contemporary Analysis Using National EMS Information System (NEMSIS) Data. *Annals of Emergency Medicine*, v. 74, n. 1, p. 101–109, 2019. ISSN 0196-0644. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0196064418313258>. Citado na página 3.

WEI, H. et al. IntelliLight: A Reinforcement Learning Approach for Intelligent Traffic Light Control. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Nova York, NY, EUA: Association for Computing Machinery, 2018. (KDD '18), p. 2496–2505. ISBN 9781450355520. Disponível em: <https://doi.org/10.1145/3219819.3220096>. Citado na página 156.

WIEDEMANN, R.; VERKEHRSWESEN, U. K. I. für; KÖHLER, U. *Simulation des Strassenverkehrsflusses*. Institut für Verkehrswesen der Universität Karlsruhe, 1974. (Schriftenreihe, v. 8-10). Disponível em: <https://books.google.com.br/books?id=lhhyxgEACAAJ>. Citado na página 156.

XIAO, N. et al. A cold-start-free reinforcement learning approach for traffic signal control. *Journal of Intelligent Transportation Systems*, Taylor & Francis, v. 26, n. 4, p. 476–485, 2022. Disponível em: <https://doi.org/10.1080/15472450.2021.1934679>. Citado nas páginas 10 e 156.

YAN, L. et al. MobiRescue: Reinforcement Learning based Rescue Team Dispatching in a Flooding Disaster. In: *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*. Singapura, Singapura: IEEE, 2020. p. 111–121. Citado na página 156.

YAN, R.; DUNNETT, S.; ANDREWS, J. A Petri net model-based resilience analysis of nuclear power plants under the threat of natural hazards. *Reliability Engineering & System Safety*, v. 230, p. 108979, 2023. ISSN 0951-8320. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0951832022005944>>. Citado na página 65.

YOUNES, M. B.; BOUKERCHE, A. An efficient dynamic traffic light scheduling algorithm considering emergency vehicles for intelligent transportation systems. *Wireless Networks*, Springer Nature America, Inc, v. 24, n. 7, p. 2451–2463, mar 2017. Citado na página 24.

ZADEH, L. Fuzzy sets. *Information and Control*, v. 8, n. 3, p. 338–353, 1965. ISSN 0019-9958. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S001999586590241X>>. Citado na página 37.

ZHAO, L. et al. Emergency Vehicle Preemption Control of T-shaped Intersection with Time Petri Nets. In: *2018 Chinese Automation Congress (CAC)*. Xian, China: IEEE, 2018. p. 2503–2508. Citado nas páginas 28 e 30.

ZLATKOVIC, M. et al. Concepts of Signal Control Preemption for Emergency Vehicles in Connected Vehicle Environments. *Journal of Road and Traffic Engineering*, v. 69, n. 2, p. 1–7, Jun. 2023. Disponível em: <<https://www.putisaobracaj.rs/index.php/PiS/article/view/279>>. Citado nas páginas 27 e 30.

Apêndice

Implementação Fuzzy

Este é o conteúdo do arquivo `djahel.fcl`, criado neste trabalho seguindo as orientações em Djahel et al. (2015), para ser usado com a biblioteca Python *pyfuzzy*.

```
FUNCTION_BLOCK djahel
  VAR_INPUT
    avs: REAL;
    ol: REAL;
  END_VAR

  VAR_OUTPUT
    cl: REAL;
  END_VAR

  FUZZIFY avs
    TERM low := (0, 1) (20, 1) (35, 0);
    TERM medium := (20, 0) (35, 1) (50, 0);
    TERM high := (35, 0) (50, 1);
  END_FUZZIFY

  FUZZIFY ol
    TERM min := (0, 1) (0.08, 1) (0.15, 0);
    TERM low := (0.08, 0) (0.15, 1) (0.25, 1) (0.35, 0);
    TERM medium := (0.25, 0) (0.35, 1) (0.5, 1) (0.6, 0);
    TERM high := (0.5, 0) (0.6, 1) (0.7, 1) (0.78, 0);
    TERM critical := (0.7, 0) (0.78, 1);
  END_FUZZIFY

  DEFUZZIFY cl
```

```

TERM negligible := (0, 0) (0.5, 1) (1, 0);
TERM low := (1, 0) (1.5, 1) (2, 0);
TERM medium := (2, 0) (2.5, 1) (3, 0);
TERM high := (3, 0) (3.5, 1) (4, 0);
TERM critical := (4, 0) (4.5, 1) (5, 0);
ACCU:MAX;
METHOD: COG;
END_DEFUZZIFY

RULEBLOCK first
  RULE 0: IF (ol IS min) AND (avs IS high) THEN (cl IS negligible);
  RULE 1: IF (ol IS min) AND (avs IS medium) THEN (cl IS negligible);
  RULE 2: IF (ol IS min) AND (avs IS low) THEN (cl IS low);

  RULE 3: IF (ol IS low) AND (avs IS high) THEN (cl IS negligible);
  RULE 4: IF (ol IS low) AND (avs IS medium) THEN (cl IS low);
  RULE 5: IF (ol IS low) AND (avs IS low) THEN (cl IS low);

  RULE 6: IF (ol IS medium) AND (avs IS high) THEN (cl IS low);
  RULE 7: IF (ol IS medium) AND (avs IS medium) THEN (cl IS medium);
  RULE 8: IF (ol IS medium) AND (avs IS low) THEN (cl IS medium);

  RULE 9: IF (ol IS high) AND (avs IS high) THEN (cl IS medium);
  RULE 10: IF (ol IS high) AND (avs IS medium) THEN (cl IS high);
  RULE 11: IF (ol IS high) AND (avs IS low) THEN (cl IS high);

  RULE 12: IF (ol IS critical) AND (avs IS high) THEN (cl IS high);
  RULE 13: IF (ol IS critical) AND (avs IS medium) THEN (cl IS critical);
  RULE 14: IF (ol IS critical) AND (avs IS low) THEN (cl IS critical);
END_RULEBLOCK

END_FUNCTION_BLOCK

```

Arquivo tpn.py

A seguir são apresentadas as classes adaptadas Transition e PetriNet para permitir atualizações das transições temporizadas depois que a TPN* é construída.

```

import snakes.plugins

from snakes import ConstraintError

@snakes.plugins.plugin("snakes.nets")
def extend (module) :
    class Transition (module.Transition) :
        def __init__ (self, name, guard=None, **args) :
            self.time = None
            self.min_time = args.pop("min_time", None)
            module.Transition.__init__(self, name, guard, **args)

```

```

def enabled (self, binding, **args) :
    if self.min_time is None:
        return module.Transition.enabled(self, binding)
    elif self.time is None:
        return False
    else:
        return (self.min_time <= self.time) and \
            module.Transition.enabled(self, binding)
def update_min_time(self,new_min_time):
    self.min_time = new_min_time

class PetriNet (module.PetriNet) :
    def time (self, step=None) :
        transactions = [trans for trans in self.transition() \
            if trans.min_time is not None]
        for trans in transactions:
            trans.time = step
return Transition, PetriNet

```

Geração dos Cenários - Metrô OD 2017 e Centro Expandido de São Paulo

Este cenário foi completamente gerado neste trabalho. O banco de dados da Pesquisa Origem e Destino 2017 está acessível neste link¹, e entre esses arquivos está um que delimita as zonas de pesquisa e um que descreve as viagens, identificando a zona de origem, a zona de destino, a modalidade de transporte e a hora de partida. Da maneira como estão, os arquivos não podem ser usados e precisam passar por um processamento para serem passíveis de utilização no SUMO.

Primeiro, é necessário obter o mapa da Região Metropolitana de São Paulo (Figura A.1). A região é muito grande para ser obtida de uma vez, portanto, devem ser obtidos *tiles* menores que devem ser mesclados posteriormente. Os *tiles* foram obtidos usando o script *osmGet.py* fornecido pelo pacote do SUMO².

Para a construção da rede, o comando *netconvert* foi usado com os parâmetros informados abaixo:

```

$ netconvert --type-files \
$SUMO_HOME/data/typemap/osmNetconvert.typ.xml --osm-files \
tile_1.osm.xml,tile_2.osm.xml,...,tile_n.osm.xml \
--output-file $SCENARIO_PATH/osm.net.xml \
--output.street-names --output.original-names --proj.utm \
--geometry.remove --tls.discard-simple --tls.join \
--tls.guess-signals --tls.allred.time 3 \
--tls.minor-left.max-speed 10000 --tls.left-green.time 0 \

```

¹<https://transparencia.metrosp.com.br/sites/default/files/OD-2017.zip>

²<https://sumo.dlr.de/docs/Networks/Import/OpenStreetMapDownload.html>



Figura A.1: Área Metropolitana de São Paulo obtido diretamente do OpenStreetMap em <https://www.openstreetmap.org/relation/2661855#map=10/-23.5756/-46.1604>

```
--ramps.guess --keep-edges.by-vclass \
passenger,bus,motorcycle,emergency --no-turnarounds \
--junctions.join -v --ignore-errors
```

A maioria dos parâmetros está presente como recomendação para construção de redes com menos problemas no script *osmWebWizard.py* (que invoca o script *osmBuild.py*, definindo a variável *DEFAULT_NETCONVERT_OPTS*³), com exceção dos parâmetros abaixo:

- **-tls.allred.time 3**: para cada uma das fases de sinal verde, foi adicionado uma fase apenas com sinal amarelo e vermelho seguido por uma fase apenas com sinal vermelho (durando 3 segundos), para dar tempo de esvaziar a interseção (compondo assim o *flush time*);
- **-tls.minor-left.max-speed 10000**: quando a velocidade ao limite estabelecido, conversões à esquerda são permitidas ao mesmo tempo que o tráfego conflitante em linha reta, mesmo que o veículo da conversão ceda;
- **-tls.left-green.time 0**: especifica o tempo da fase de conversão à esquerda, quando disponível; e
- **-keep-edges.by-vclass passenger,bus,motorcycle,emergency**: descarta as demais arestas, mantendo apenas as que são trafegáveis pelos veículos.

³https://sumo.dlr.de/docs/Networks/Import/OpenStreetMap.html#recommended_netconvert_options

culos utilizados na simulação.

Todos os parâmetros relacionados a semáforos são importantes para que a rede construída apresente o menor número de erros/problemas durante a execução. Isso porque os semáforos são “adivinhados” e suas fases e respectivos tempos de duração são definidos pela ferramenta *netconvert* seguindo a estratégia de Melhor Esforço. Nada, contudo, garante que defeitos na rede não surjam, e os problemas relacionados às interseções, em geral, só são verificados quando a simulação está em execução.

Com o mapa em mãos, é necessário verificar quais arestas/vias pertencem a quais zonas. Entre os arquivos disponibilizados pela Pesquisa, um deles fornece a delimitação das zonas no formato *Esri Shapefile*. Esse arquivo foi importado na ferramenta JOSM⁴ e depois exportado para o formato *OpenStreetMap* (OSM), com o nome *zones.osm.xml*. Uma conversão adicional é necessária para transformar os elementos OSM para os polígonos que são interpretados pelo SUMO, feito pelo comando abaixo:

```
$ cd $SCENARIO_PATH && polyconvert -n osm.net.xml \
--type-file osmPolyconvert.typ.xml --osm-files \
zones.osm.xml -o zones.osm.poly.xml --verbose
```

O conteúdo do arquivo *osmPolyconvert.typ.xml* é descrito no trecho eXtensible Markup Language (XML) abaixo, usado para selecionar apenas os elementos de interesse, que, nesse caso particular, são os elementos *boundary*:

```
<polygonTypes>
  <polygonType id="boundary" name="boundary" color="1.0,.33,.33" layer="0"/>
</polygonTypes>
```

De posse da rede e das zonas com os polígonos corretos, é necessário encontrar os distritos, ou seja, quais arestas pertencem a quais zonas. Para isso, foi executado o comando abaixo:

```
$ cd $SUMO_PATH/tools && python edgesInDistricts.py -n \
$SCENARIO_PATH/osm.net.xml -t \
$SCENARIO_PATH/zones.osm.poly.xml -o \
$SCENARIO_PATH/districts.taz.osm.xml
```

O próximo passo é tratar o arquivo de demanda. O arquivo do Banco de Dados OD_2017_v1.dbf contém a amostra das viagens, e foi exportado como *Comma Separated Value* (CSV) usando o programa LibreOffice Calc. O arquivo contém uma amostra de 183.092 viagens que, excluindo as viagens não motorizadas (por exemplo, metrô, trem, bicicleta etc.), resultou em uma amostra filtrada de 91.014 viagens.

⁴<https://josm.openstreetmap.de/>

Por fim, para que o arquivo de demanda seja usado, ele precisa estar em um dos formatos reconhecíveis pelo SUMO. Entre os disponíveis, foi escolhido o formato Amitran⁵, que permite utilizar as modalidades de transporte. Para isto, este trabalho fornece o script *parse_od_csv.py*⁶, construindo assim o arquivo demand.xml a partir do CSV previamente apresentado. Neste momento, para cada registro do arquivo de demanda original, é possível especificar a quantidade de registros gerados de arquivos de demanda final (campo *amount*), para que assim as 91.014 viagens se transformem na representação aproximada do traço real de 23.670.360 viagens motorizadas diárias. Usando como base a representação da amostra apresentada na Figura A.2, o campo *amount* foi calculado da seguinte forma:

VIAGENS DIÁRIAS MOTORIZADAS POR MODO PRINCIPAL
Região Metropolitana de São Paulo – 2007 e 2017

MODO	2007		2017	
	(x 1.000)	%	(x 1.000)	%
Metrô	2.223	8,8	3.400	12,0
Trem	815	3,2	1.245	4,4
Ônibus	9.034	35,9	8.304	29,4
Fretado	514	2,0	252	0,9
Escolar	1.327	5,3	2.094	7,4
Auto	10.381	41,2	11.341	40,1
Táxi	91	0,4	468	1,7
Moto	721	2,9	1.064	3,8
Outros	61	0,2	112	0,4
TOTAL	25.167	100,0	28.280	100,0

Fonte: Metrô-Pesquisas OD 2007 e 2017

Figura A.2: Representação da Amostra da Pesquisa OD Metrô 2017 (Companhia Metropolitana de São Paulo (METRÔ), 2017).

- **carros:** 41.114 registros no arquivo dBase Database File (DBF), representando 11,341 milhões de viagens de automóveis, 468 mil viagens de táxis e 112 mil viagens por outros meios motorizados, totalizando 11,921 milhões de viagens nesta categoria. Portanto, $amount = \frac{11,921M}{41,114k} \approx 290$;
- **motocicletas:** 3.075 registros no arquivo DBF, representando 1,064 milhão de viagens. Portanto, $amount = \frac{1,064M}{3,075k} \approx 346$; e
- **ônibus:** 29.962 registros no arquivo DBF, representando 8,304 milhões de viagens de ônibus, 252 mil viagens de fretado e 2,094 viagens de

⁵<<https://sumo.dlr.de/xsd/amitran/linkdata.xsd>>

⁶<https://gitlab.com/rodrigo.g.branco/docker-sumo-intercity-spres-ev/blob/master/tools/parse_od_csv.py>

O arquivo *districts.taz.osm.xml* já informa quais arestas pertencem a quais zonas, portanto, basta eliminar as arestas e elementos relacionados das zonas fora da área delimitada. Para isto, este trabalho disponibiliza o script *trim_map.py*⁹.

O último passo é ajustar as viagens. Cada uma das viagens no arquivo *osm.trips.xml* foi roteada na rede (pois elas possuem apenas arestas de origem e destino definidos). Com a rota definida, a rota é varrida do início ao fim para localizar a primeira aresta que faça parte da área delimitada, servindo agora como a nova origem do veículo. O mesmo procedimento é feito para a aresta de destino, mas percorrendo a rota em sua ordem inversa. Devido ao volume de viagens, o script *restrict_trips_to_edges.py*, que divide o arquivo original em chunks de 5000 viagens, para que o roteador do SUMO (*duarouter*) seja invocado em paralelo e o processo termine em um tempo razoável. Ao fim do processo, 5.896.245 viagens no período de 24 horas restaram neste cenário reduzido¹⁰.

Para o futuro, há maneiras de se aproximar os dois cenários, o completo e o reduzido, de uma situação mais próxima do real. Duas maneiras diretas seriam fixar as rotas dos ônibus públicos/escolares e ajustar os ciclos dos semáforos para seus equivalentes físicos.

Detalhamento dos Resultados

Tabela A.1: Tempo Perdido (s) - Primeira Solução - Cenário São Paulo (Sintético) - Sem Preempção

#Veículos	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
7.016	-	996,54	1.014,03	1.037,14	1.071,77	1.140,43	1.335,05
12.842	-	1.017,14	1.282,96	1.367,56	1.586,70	1.920,14	-
17.500	-	1.292,65	1.375,36	1.621,57	1.764,48	2.218,17	2.557,69
22.154	-	1.290,19	1.637,84	1.915,12	2.189,84	2.855,21	3.924,55
25.882	-	1.323,43	1.703,59	1.923,27	2.735,74	3.885,87	5.652,50

Tabela A.2: Tempo Perdido (s) - Primeira Solução - Cenário Nova Iorque (Sintético) - Sem Preempção

#Veículos	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
4.265	671,44	693,91	814,89	863,42	900,81	1.000,71	1.035,60
7.789	-	794,57	873,09	906,54	1.001,60	1.177,60	1.562,27
10.799	-	828,95	952,44	1.158,92	1.549,50	2.185,16	2.550,12
13.622	-	841,81	1.056,38	1.309,79	1.805,92	2.433,73	3.251,73
16.454	-	947,64	1.434,82	1.830,27	2.502,18	3.972,50	-

⁹https://gitlab.com/rodrigo.g.branco/docker-sumo-intercity-spres-ev/blob/master/tools/trim_map.py

¹⁰Cenário disponível em <https://gitlab.com/rodrigo.g.branco/augmented-downtown>

Tabela A.3: Tempo Perdido sobre o Tempo Total de Viagem (%) - Primeira Solução - Cenário São Paulo (Sintético) - Sem Preempção

#Veículos	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
7.016	-	55,22	57,74	58,95	60,98	65,63	66,43
12.842	-	57,87	63,49	65,38	68,28	73,32	-
17.500	-	62,67	66,18	68,39	71,63	79,56	-
22.154	-	64,82	68,95	72,15	76,24	84,86	-
25.882	-	66,00	69,94	73,36	78,66	88,99	-

Tabela A.4: Tempo Perdido sobre o Tempo Total de Viagem (%) - Primeira Solução - Cenário Nova Iorque (Sintético) - Sem Preempção

#Veículos	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
4.265	-	65,03	67,57	68,97	70,50	73,60	75,70
7.789	-	65,65	68,99	70,68	73,63	79,29	81,92
10.799	-	68,54	71,36	74,52	80,64	88,03	-
13.622	-	68,06	72,19	77,04	83,21	90,55	-
16.454	-	71,90	80,64	83,76	87,25	91,57	-

Tabela A.5: Melhoria do Tempo Perdido (%) - Primeira Solução - Cenário São Paulo (Sintético) - RFID

#Veículos	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
7.016	43,53	52,58	60,56	66,86	70,33	84,33	-
12.842	-	-2,54	36,60	56,82	64,38	74,35	-
17.500	-52,88	-28,92	13,77	36,24	52,45	74,18	-
22.154	-	-125,88	-47,45	6,34	39,55	71,92	-
25.882	-332,09	-148,14	-50,10	-6,94	17,56	50,56	-

Tabela A.6: Melhoria do Tempo Perdido (%) - Primeira Solução - Cenário São Paulo (Sintético) - Fuzzy

#Veículos	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
7.016	-	5,84	24,73	38,75	50,47	65,22	-
12.842	-127,74	-58,54	-14,11	11,67	22,64	55,25	-
17.500	-233,18	-149,59	-59,77	-10,29	12,21	56,45	-
22.154	-	-179,07	-82,80	-32,42	3,35	46,49	-
25.882	-214,05	-187,13	-92,35	-47,11	-23,94	19,47	-

Tabela A.7: Melhoria do Tempo Perdido (%) - Primeira Solução - Cenário São Paulo (Sintético) - Filas

#Veículos	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
7.016	-	79,97	84,82	87,98	89,58	92,03	-
12.842	40,43	63,40	77,41	83,84	87,79	90,73	-
17.500	-	3,24	28,51	41,09	52,18	72,67	-
22.154	-	-53,66	4,08	32,61	50,85	75,41	-
25.882	-110,69	-79,50	-17,63	7,67	24,05	54,96	-

Tabela A.8: Melhoria do Tempo Perdido (%) - Primeira Solução - Cenário São Paulo (Sintético) - TPN

#Veículos	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
7.016	81,15	85,23	88,16	89,47	90,16	92,41	-
12.842	43,11	80,32	84,63	88,45	89,69	91,70	-
17.500	6,51	13,77	43,52	54,65	65,22	88,89	-
22.154	-	-99,93	-2,69	51,93	77,28	91,21	-
25.882	-169,51	-95,47	-27,67	24,46	41,03	86,65	-

Tabela A.9: Melhoria do Tempo Perdido (%) - Primeira Solução - Cenário São Paulo (Sintético) - Onda Verde

#Veículos	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
7.016	-	86,80	88,56	89,58	90,21	92,47	-
12.842	83,02	86,46	87,85	88,87	89,77	91,16	-
17.500	58,73	67,10	78,30	81,99	87,32	90,02	-
22.154	-51,16	70,58	78,46	86,59	88,21	91,33	-
25.882	-38,90	3,46	42,71	64,13	74,38	89,27	-

Tabela A.10: Melhoria do Tempo Perdido (%) - Primeira Solução - Cenário Nova Iorque (Sintético) - RFID

#Veículos	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
7.016	-	41,99	55,44	60,62	66,12	72,98	-
12.842	-0,37	36,43	52,29	59,58	65,92	79,18	-
17.500	-115,32	-74,90	-1,09	50,21	57,76	69,99	-
22.154	-136,33	-85,53	-26,06	31,00	47,42	64,84	-
25.882	-	-258,21	-107,90	-57,80	-6,45	54,66	-

Tabela A.11: Melhoria do Tempo Perdido (%) - Primeira Solução - Cenário Nova Iorque (Sintético) - Fuzzy

#Veículos	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
7.016	-	13,08	33,12	45,17	53,45	67,02	-
12.842	-19,29	6,83	30,89	45,31	56,67	65,95	-
17.500	-	-97,34	-35,21	29,79	42,71	59,75	-
22.154	-230,24	-121,13	-30,62	10,97	36,29	54,16	-
25.882	-	-287,22	-114,51	-45,25	0,76	32,69	-

Tabela A.12: Melhoria do Tempo Perdido (%) - Primeira Solução - Cenário Nova Iorque (Sintético) - Filas

#Veículos	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
7.016	64,19	71,70	80,62	83,60	88,63	95,54	-
12.842	-27,51	60,06	77,63	84,62	89,78	95,82	-
17.500	-8,60	3,56	51,24	75,50	90,80	95,56	-
22.154	-	-9,18	42,07	71,48	88,78	95,02	-
25.882	-200,24	-101,24	-8,03	41,94	65,36	92,63	-

Tabela A.13: Melhoria do Tempo Perdido (%) - Primeira Solução - Cenário Nova Iorque (Sintético) - TPN'

#Veículos	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
7.016	64,62	71,21	81,94	88,44	89,78	95,40	-
12.842	38,02	74,04	83,18	87,65	89,75	95,88	-
17.500	-79,66	-26,23	38,50	64,96	83,59	91,14	-
22.154	-	-93,76	1,81	61,35	79,99	95,54	-
25.882	-	-209,75	-105,58	-41,31	32,96	88,87	-

Tabela A.14: Melhoria do Tempo Perdido (%) - Primeira Solução - Cenário Nova Iorque (Sintético) - Onda Verde

#Veículos	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
7.016	73,48	83,21	89,09	90,59	93,84	95,70	-
12.842	29,09	92,22	93,47	94,35	95,09	95,94	-
17.500	20,05	23,71	64,92	82,67	93,60	95,82	-
22.154	-	8,56	54,17	75,42	92,09	95,65	-
25.882	-79,86	-30,66	22,88	50,69	72,33	95,29	-

Tabela A.15: Tempo Perdido (s) - Segunda Solução - Cenário Uma Interseção - Sem Preempção

Algoritmo	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	1.786,72	1.862,50	1.938,11	1.949,99	1.951,43	2.102,48

Tabela A.16: Tempo Perdido sobre o Tempo Total de Viagem (%) - Segunda Solução - Cenário Uma Interseção - Sem Preempção

Algoritmo	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	92,24	92,52	92,80	92,84	93,32	-

Tabela A.17: Melhoria do Tempo Perdido (vezes) - Segunda Solução - Cenário Uma Interseção

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
RFID	-	-1,00	-1,00	1,06	1,06	1,07	-
Fuzzy	-	-1,09	-1,04	-1,01	-1,01	-1,00	1,07
Filas	-	-1,24	-1,09	1,08	1,24	4,09	-
Onda de Choque	-	3,51	3,70	3,94	4,05	4,32	4,88
Onda Verde	-	3,51	3,70	3,94	4,05	4,32	4,88

Tabela A.18: Melhoria do Tempo Perdido (%) - Segunda Solução - Cenário Uma Interseção

Algoritmo	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
RFID	-	-	0,00	5,40	5,92	6,41	-
Fuzzy	-	-8,81	-4,02	-0,87	-0,55	-0,37	6,61
Filas	-23,87	-8,77	7,26	19,49	54,32	75,57	
Onda de Choque	-	71,53	72,98	74,59	75,29	76,87	79,52
Onda Verde	-	71,53	72,98	74,59	75,29	76,87	79,52

Tabela A.19: Tempo de Preempção Médio (s) - Segunda Solução - Cenário Uma Interseção

Algoritmo	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
RFID	-	-	-	150,00	-	-	-
Fuzzy	-	-	-	30,00	-	-	-
Filas	-	603,00	799,00	915,00	1.309,50	1.244,00	-
Onda de Choque	-	593,00	624,50	645,00	679,50	706,00	-
Onda Verde	-	593,00	624,50	645,00	679,50	706,00	-

Tabela A.20: Tempo Perdido (s) - Segunda Solução - Cenário Turim - Sem Preempção

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	4.057,63	4.543,19	5.104,79	5.322,43	6.481,92	8.600,99
EV2	-	665,43	675,43	853,90	960,42	1.069,27	-
EV3	-	9.056,56	10.275,39	11.347,12	12.346,31	14.627,42	15.768,42
EV1-Sintético	-	3.952,96	4.742,09	5.216,97	5.696,32	7.117,92	7.431,69
EV2-Sintético	665,45	754,32	960,29	1.064,65	1.118,70	1.178,23	1.480,83
EV3-Sintético	-	2.534,26	3.521,38	4.181,59	5.016,68	7.075,39	-

Tabela A.21: Tempo Perdido sobre o Tempo Total de Viagem (%) - Segunda Solução - Cenário Turim - Sem Preempção

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	84,52	85,92	87,28	87,73	89,69	92,03
EV2	-	44,16	44,52	50,36	53,28	55,92	-
EV3	-	90,12	91,18	91,94	92,55	94,07	-
EV1-Sintético	-	84,14	86,43	87,53	88,46	90,90	-
EV2-Sintético	44,16	50,23	53,28	55,83	57,04	58,33	63,77
EV3-Sintético	-	71,81	77,98	80,79	83,46	87,68	-

Tabela A.22: Melhoria do Tempo Perdido (%) - Segunda Solução - Cenário Turim - RFID

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	15,96	-8,71	2,59	9,93	16,95	50,20
EV2	-	4,02	17,41	32,57	50,76	67,07	-
EV3	-6,47	8,89	16,42	24,70	28,23	42,54	-
EV1-Sintético	-	-16,62	1,61	13,98	22,49	51,73	-
EV2-Sintético	-	0,30	20,26	48,87	59,03	70,29	-
EV3-Sintético	-	-46,36	-3,42	24,43	34,60	69,31	-

Tabela A.23: Melhoria do Tempo Perdido (%) - Segunda Solução - Cenário Turim - Fuzzy

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-42,76	-30,68	-8,29	0,24	36,34	-
EV2	-	-165,94	-102,14	-65,03	-46,66	-24,74	-
EV3	-	-26,63	-11,50	1,99	18,37	52,87	-
EV1-Sintético	-	-44,94	-16,57	-0,04	10,62	29,14	-
EV2-Sintético	-197,07	-192,90	-92,69	-56,36	-23,73	4,21	-
EV3-Sintético	-	27,73	13,40	48,68	57,94	80,55	-

Tabela A.24: Melhoria do Tempo Perdido (%) - Segunda Solução - Cenário Turim - Filas

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-1,05	6,26	18,53	28,00	55,02	-
EV2	-107,22	-39,16	0,23	14,22	34,57	51,29	-
EV3	-	7,96	28,76	44,44	55,56	66,29	-
EV1-Sintético	-	-13,72	11,87	32,17	43,44	64,90	-
EV2-Sintético	-	-59,76	-9,07	5,99	27,92	63,34	-
EV3-Sintético	-79,86	-65,99	-6,58	22,72	35,35	48,87	-

Tabela A.25: Melhoria do Tempo Perdido (%) - Segunda Solução - Cenário Turim - Onda de Choque

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-36,29	-11,23	5,74	43,48	77,85	-
EV2	-	76,58	82,83	84,68	87,15	90,62	-
EV3	-	76,04	83,10	87,16	89,36	92,79	-
EV1-Sintético	-	-32,87	3,45	41,60	53,55	73,65	-
EV2-Sintético	74,33	76,69	83,85	86,78	89,87	93,25	-
EV3-Sintético	-	82,95	85,36	88,33	90,34	94,86	-

Tabela A.26: Melhoria do Tempo Perdido (%) - Segunda Solução - Cenário Turim - Onda Verde

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	29,16	51,62	63,99	71,39	76,99	84,87	-
EV2	83,73	87,17	88,68	89,89	90,48	92,63	-
EV3	-	17,69	30,51	44,09	58,36	75,74	-
EV1-Sintético	-	58,75	66,00	72,75	79,37	82,51	-
EV2-Sintético	-	87,82	89,73	91,12	92,34	94,81	-
EV3-Sintético	10,90	25,77	43,31	54,94	63,06	86,86	-

Tabela A.27: Melhoria do Tempo Perdido (vezes) - Segunda Solução - Cenário Turim - RFID

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-1,16	-1,07	1,03	1,10	2,01	-
EV2	-	1,04	1,21	1,48	2,03	3,04	-
EV3	-1,06	1,10	1,20	1,33	1,39	1,60	1,74
EV1-Sintético	-	-1,17	0,02	1,16	1,29	2,07	-
EV2-Sintético	-	1,00	1,25	1,96	2,44	3,37	-
EV3-Sintético	-	-1,46	-0,03	1,32	1,53	3,26	-

Tabela A.28: Melhoria do Tempo Perdido (vezes) - Segunda Solução - Cenário Turim - Fuzzy

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-1,43	-1,30	-1,08	1,00	1,57	-
EV2	-	-2,66	-2,02	-1,65	-1,47	-1,25	-
EV3	-	-1,27	-1,12	1,02	1,23	2,12	-
EV1-Sintético	-	-1,45	-1,17	-0,00	1,12	1,41	-
EV2-Sintético	-2,97	-2,93	-1,93	-1,56	-1,24	-1,03	1,04
EV3-Sintético	-1,28	1,02	1,16	1,95	2,38	3,26	5,14

Tabela A.29: Melhoria do Tempo Perdido (vezes) - Segunda Solução - Cenário Turim - Filas

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-1,01	1,03	1,07	1,23	1,37	1,61	2,22
EV2	-	-2,07	0,00	1,17	1,53	2,05	-
EV3	-	1,09	1,41	1,80	2,26	2,97	-
EV1-Sintético	-1,14	1,10	1,13	1,47	1,77	2,31	2,85
EV2-Sintético	-	-1,60	-1,09	1,06	1,39	2,73	-
EV3-Sintético	-	-1,80	-1,07	1,29	1,55	1,96	-

Tabela A.30: Melhoria do Tempo Perdido (vezes) - Segunda Solução - Cenário Turim - Onda de Choque

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-1,36	-1,09	1,06	1,76	4,51	-
EV2	-	4,27	5,82	6,53	7,78	10,67	-
EV3	-	4,17	5,92	7,81	9,40	13,87	-
EV1-Sintético	-	-1,33	0,04	1,71	2,15	3,79	-
EV2-Sintético	-	3,90	6,19	7,57	9,88	14,81	-
EV3-Sintético	-	5,87	6,84	8,57	10,36	12,77	19,46

Tabela A.31: Melhoria do Tempo Perdido (vezes) - Segunda Solução - Cenário Turim - Onda Verde

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	1,41	2,83	3,50	4,28	4,84	6,61
EV2	6,15	7,80	8,83	9,89	10,50	12,57	13,57
EV3	-	1,21	1,44	1,79	2,41	3,29	4,12
EV1-Sintético	-	2,42	2,94	3,68	4,85	5,72	-
EV2-Sintético	-	8,21	9,75	11,26	13,06	16,20	19,27
EV3-Sintético	-	1,12	1,76	2,22	2,71	3,21	7,61

Tabela A.32: Tempo Perdido (s) - Segunda Solução - Cenário Uma Interseção

Algoritmo	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
RFID	-	1.672,59	1.808,74	1.830,24	1.949,99	2.102,48	-
Fuzzy	-	1.799,59	1.809,66	1.955,57	2.101,72	2.249,75	-
Filas	558,38	821,55	1.557,78	1.744,92	2.152,32	2.481,08	-
Onda de Choque	-	435,16	474,29	494,58	522,57	549,05	-
Onda Verde	-	435,16	474,29	494,58	522,57	549,05	-

Tabela A.33: Tempo de Preempção Médio (s) - Segunda Solução - Cenário Turim - RFID

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-	-	265,17	-	-	-
EV2	-	-	-	211,32	-	-	-
EV3	-	-	-	238,33	-	-	-
EV1-Sintético	-	-	-	265,17	-	-	-
EV2-Sintético	-	-	-	211,32	-	-	-
EV3-Sintético	-	-	-	238,33	-	-	-

Tabela A.34: Tempo de Preempção Médio (s) - Segunda Solução - Cenário Turim - Fuzzy

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	52,21	52,53	52,66	53,07	53,67	54,06
EV2	-	-	-	41,79	41,83	42,33	-
EV3	-	-	-	47,30	47,47	-	-
EV1-Sintético	-	52,21	-	52,66	53,10	53,67	-
EV2-Sintético	40,61	41,12	41,46	41,79	-	41,83	42,33
EV3-Sintético	-	-	-	47,30	-	-	47,47

Tabela A.35: Tempo de Preempção Médio (s) - Segunda Solução - Cenário Turim - Filas

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	944,86	1.052,89	1.288,73	1.511,06	1.833,14	-
EV2	-	359,05	414,16	456,89	521,71	565,89	834,10
EV3	-	1.050,69	1.230,89	1.312,67	1.484,98	1.827,39	-
EV1-Sintético	-	860,00	1.076,50	1.141,06	1.240,19	1.348,91	2.144,89
EV2-Sintético	-	401,16	491,21	515,63	567,18	671,95	-
EV3-Sintético	-	990,90	1.219,79	1.337,12	1.469,90	1.719,15	1.865,69

Tabela A.36: Tempo de Preempção Médio (s) - Segunda Solução - Cenário Turim - Onda de Choque

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	208,51	309,11	806,97	895,26	1.293,51	-
EV2	-	84,42	88,42	90,68	94,13	101,68	107,26
EV3	-	238,24	255,68	291,45	316,04	368,03	452,51
EV1-Sintético	-	185,89	303,30	510,13	1.055,70	1.203,26	-
EV2-Sintético	-	87,21	88,84	92,95	94,50	102,79	114,68
EV3-Sintético	-	162,09	178,83	187,06	201,71	222,90	-

Tabela A.37: Tempo de Preempção Médio (s) - Segunda Solução - Cenário Turim - Onda Verde

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	1.858,69	1.988,84	2.211,80	2.535,54	3.218,02	6.895,62
EV2	-	916,74	922,24	928,29	932,68	947,42	1.107,32
EV3	-	4.545,39	6.062,29	7.422,26	8.645,83	10.297,73	-
EV1-Sintético	-	1.753,89	1.973,73	2.210,50	2.371,04	2.839,57	3.229,29
EV2-Sintético	-	917,74	924,13	929,42	944,92	963,89	-
EV3-Sintético	-	2.046,39	2.519,10	2.958,40	3.265,21	4.273,00	4.977,73

Tabela A.38: Tempo Perdido (s) - Segunda Solução - Cenário Turim - RFID

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	3.849,25	4.559,63	4.868,94	5.251,41	5.973,59	-
EV2	-	365,90	463,96	549,78	683,42	852,10	-
EV3	-	6.595,60	8.027,97	8.922,06	10.022,98	11.833,18	-
EV1-Sintético	-	3.559,39	4.402,39	4.665,24	5.003,29	5.516,35	6.797,75
EV2-Sintético	-	365,35	441,51	518,39	855,77	1.172,31	-
EV3-Sintético	-	2.027,29	2.793,01	3.355,33	4.115,70	5.512,63	6.194,38

Tabela A.39: Tempo Perdido (s) - Segunda Solução - Cenário Turim - Fuzzy

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	4.895,89	5.263,14	5.502,03	6.211,70	6.568,23	-
EV2	-	1.178,73	1.309,16	1.379,84	1.652,38	1.941,52	2.270,03
EV3	-	7.352,42	9.654,73	11.253,77	12.120,26	13.260,06	-
EV1-Sintético	-	4.662,98	5.034,77	5.395,70	5.746,54	6.263,52	20.484,97
EV2-Sintético	-	1.290,69	1.350,16	1.486,76	1.798,74	2.077,42	3.500,13
EV3-Sintético	-	1.270,50	1.691,58	2.167,80	3.173,09	4.863,87	6.787,49

Tabela A.40: Tempo Perdido (s) - Segunda Solução - Cenário Turim - Filas

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	3.236,97	3.750,86	4.173,89	4.779,45	5.748,84	-
EV2	-	464,35	561,60	657,60	950,54	1.050,77	2.100,53
EV3	-	4.142,54	5.331,98	6.812,08	8.145,67	9.638,25	-
EV1-Sintético	-	2.342,21	3.266,41	3.777,74	4.550,71	5.715,07	-
EV2-Sintético	-	523,29	777,78	1.029,76	1.149,40	1.384,56	-
EV3-Sintético	-	2.173,65	3.053,94	3.832,74	4.391,16	4.992,73	6.415,44

Tabela A.41: Tempo Perdido (s) - Segunda Solução - Cenário Turim - Onda de Choque

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	2.037,65	3.088,85	5.012,70	5.721,69	7.353,06	-
EV2	-	85,89	113,25	121,77	155,14	202,71	-
EV3	-	889,93	1.255,60	1.556,26	1.906,36	2.843,42	-
EV1-Sintético	-	1.687,10	2.602,44	3.450,42	6.091,98	6.933,74	-
EV2-Sintético	-	94,11	111,76	139,38	166,78	248,86	273,36
EV3-Sintético	-	292,05	425,60	468,63	587,48	762,05	-

Tabela A.42: Tempo Perdido (s) - Segunda Solução - Cenário Turim - Onda Verde

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	1.109,86	1.240,38	1.464,47	1.788,59	2.469,80	6.146,89
EV2	-	73,17	78,45	85,06	88,24	102,50	261,03
EV3	-	3.548,30	5.064,01	6.424,72	7.648,44	9.301,21	-
EV1-Sintético	-	1.006,29	1.226,78	1.463,19	1.624,07	2.092,60	2.480,94
EV2-Sintético	-	73,29	80,84	86,34	97,80	117,72	-
EV3-Sintético	-	1.049,18	1.523,38	1.961,37	2.268,78	3.274,36	3.982,17

Tabela A.43: Tempo Perdido (s) - Segunda Solução - Cenário Colônia - Sem Preempção

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	4.903,73	7.425,10	10.369,03	13.543,97	18.246,62	-
EV2	-	1.008,23	1.106,25	1.252,65	1.488,93	1.972,03	-
EV3	-	1.922,71	2.106,28	2.206,64	2.779,35	3.737,92	-
EV1-Sintético	-	632,54	634,78	728,66	814,80	920,51	-
EV2-Sintético	-	2.543,87	3.365,53	4.354,31	10.214,70	19.057,53	22.397,78
EV3-Sintético	-	952,52	1.296,09	1.725,91	2.395,36	3.163,18	-

Tabela A.44: Tempo Perdido sobre o Tempo Total de Viagem (%) - Segunda Solução - Cenário Colônia - Sem Preempção

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	90,23	93,32	95,12	96,22	97,17	-
EV2	-	73,49	75,28	77,48	80,37	84,42	-
EV3	-	80,79	82,16	82,83	85,88	89,10	-
EV1-Sintético	-	54,34	54,44	57,83	60,51	63,40	-
EV2-Sintético	-	87,48	90,24	92,29	96,51	98,40	-
EV3-Sintético	-	67,55	73,90	79,04	83,97	87,38	-

Tabela A.45: Melhoria do Tempo Perdido (%) - Segunda Solução - Cenário Colônia - RFID

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	219,94	-78,33	-32,95	21,00	67,09	-
EV2	-6,48	8,87	18,77	26,55	35,46	54,34	-
EV3	28,29	36,38	51,96	58,17	64,96	74,25	-
EV1-Sintético	-	0,86	22,91	30,19	37,64	57,06	59,75
EV2-Sintético	-	-47,04	17,32	45,30	81,93	92,94	-
EV3-Sintético	-69,08	-55,23	-3,14	19,88	36,73	75,36	-

Tabela A.46: Melhoria do Tempo Perdido (%) - Segunda Solução - Cenário Colônia - Fuzzy

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-195,98	-75,84	-31,17	13,16	28,76	56,56	-
EV2	-	-93,24	-60,39	-24,85	12,58	36,21	-
EV3	-	-8,69	20,81	32,63	46,15	67,88	-
EV1-Sintético	-82,19	-34,76	-20,07	-9,34	-0,43	22,89	47,04
EV2-Sintético	-54,47	-9,72	41,57	68,66	85,84	94,19	-
EV3-Sintético	-	156,88	-69,54	-19,67	22,29	58,04	-

Tabela A.47: Melhoria do Tempo Perdido (%) - Segunda Solução - Cenário Colônia - Filas

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-192,78	65,71	66,10	75,06	80,34	86,82	-
EV2	-41,04	0,19	20,66	32,29	49,19	67,74	-
EV3	-	135,08	-87,79	-64,72	-22,16	15,61	-
EV1-Sintético	-	-5,64	14,07	33,87	44,28	62,24	-
EV2-Sintético	-	58,55	75,13	84,31	93,18	96,84	-
EV3-Sintético	-	72,96	80,64	86,07	89,14	94,94	-

Tabela A.48: Melhoria do Tempo Perdido (%) - Segunda Solução - Cenário Colônia - Onda de Choque

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	69,69	82,06	87,09	90,57	95,14	-
EV2	87,66	89,68	92,33	93,34	94,46	95,92	-
EV3	-	-1,27	22,00	45,88	65,52	87,24	-
EV1-Sintético	-	17,84	35,48	49,03	56,75	72,89	-
EV2-Sintético	-	93,16	95,33	96,98	98,54	99,34	-
EV3-Sintético	78,28	86,39	89,55	91,84	94,78	96,31	-

Tabela A.49: Melhoria do Tempo Perdido (%) - Segunda Solução - Cenário Colônia - Onda Verde

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	73,99	77,49	86,14	89,73	92,09	96,47	-
EV2	-	94,59	95,45	96,52	97,10	97,74	-
EV3	-	-94,01	-44,45	-30,93	-2,76	40,16	-
EV1-Sintético	-	15,98	35,02	47,26	56,29	86,93	-
EV2-Sintético	-	99,08	99,24	99,53	99,73	99,91	-
EV3-Sintético	-	96,10	97,18	97,98	98,69	99,17	-

Tabela A.50: Tempo de Preempção Médio (s) - Segunda Solução - Cenário Colônia - RFID

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-	-	202,69	-	-	-
EV2	-	-	-	203,70	-	-	-
EV3	-	-	-	221,46	-	-	-
EV1-Sintético	-	-	-	202,69	-	-	-
EV2-Sintético	-	-	-	203,70	-	-	-
EV3-Sintético	-	-	-	221,46	-	-	-

Tabela A.51: Tempo de Preempção Médio (s) - Segunda Solução - Cenário Colônia - Fuzzy

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	39,04	-	-	40,54	-	-	40,56
EV2	40,69	-	-	40,74	-	-	-
EV3	-	-	-	44,29	-	-	-
EV1-Sintético	-	-	39,04	40,54	-	40,56	-
EV2-Sintético	40,69	-	-	40,74	-	-	-
EV3-Sintético	-	-	-	44,29	-	-	-

Tabela A.52: Tempo de Preempção Médio (s) - Segunda Solução - Cenário Colônia - Filas

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	482,96	644,62	734,56	887,56	888,88	3.279,89
EV2	-	278,89	341,42	397,18	453,64	568,92	910,96
EV3	1.050,08	1.358,50	1.476,94	1.758,14	2.053,42	2.420,42	-
EV1-Sintético	-	213,27	225,25	238,83	257,02	294,00	-
EV2-Sintético	-	249,26	291,35	320,85	345,07	412,93	-
EV3-Sintético	-	220,20	226,86	233,39	243,20	263,58	278,70

Tabela A.53: Tempo de Preempção Médio (s) - Segunda Solução - Cenário Colônia - Onda de Choque

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	270,58	296,23	330,12	364,75	465,08	-
EV2	-	81,56	83,11	87,15	93,78	105,74	-
EV3	-	131,83	165,17	207,42	264,25	356,00	468,29
EV1-Sintético	-	72,50	78,58	80,96	84,17	90,96	93,80
EV2-Sintético	96,41	99,81	113,28	119,76	122,56	123,67	151,92
EV3-Sintético	-	78,96	87,89	90,31	94,17	102,17	140,00

Tabela A.54: Tempo de Preempção Médio (s) - Segunda Solução - Cenário Colônia - Onda Verde

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	1.285,42	1.534,19	1.639,69	1.839,54	2.014,96	-
EV2	-	399,59	406,09	411,59	421,96	444,22	-
EV3	-	2.464,04	3.100,29	3.475,46	4.090,35	5.184,54	6.685,67
EV1-Sintético	-	688,42	858,58	942,75	1.018,06	1.067,42	-
EV2-Sintético	-	386,59	387,59	388,59	392,59	398,59	-
EV3-Sintético	-	484,33	486,83	496,83	500,83	518,33	609,33

Tabela A.55: Melhoria do Tempo Perdido (vezes) - Segunda Solução - Cenário Colônia - RFID

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-3,20	-1,78	-1,33	1,27	3,04	-
EV2	-1,06	1,10	1,23	1,36	1,55	1,89	2,19
EV3	-	1,39	2,08	2,39	2,89	3,88	-
EV1-Sintético	-	1,01	1,30	1,43	1,60	1,86	2,48
EV2-Sintético	-	-1,47	1,22	1,84	5,56	11,14	14,18
EV3-Sintético	-	-1,69	-1,03	1,25	1,58	4,06	-

Tabela A.56: Melhoria do Tempo Perdido (vezes) - Segunda Solução - Cenário Colônia - Fuzzy

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-2,96	-1,31	1,15	1,41	2,30	-
EV2	-	-1,93	-1,58	-1,25	1,13	1,57	-
EV3	-1,09	1,02	1,26	1,49	1,86	2,53	3,11
EV1-Sintético	-	-1,82	-1,20	-1,09	-0,00	1,44	1,89
EV2-Sintético	-	-1,54	1,71	3,26	7,08	12,68	17,21
EV3-Sintético	-	-2,57	-1,70	-1,20	1,29	2,38	-

Tabela A.57: Melhoria do Tempo Perdido (vezes) - Segunda Solução - Cenário Colônia - Filas

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-2,93	1,52	2,95	4,01	5,09	7,59	-
EV2	-1,41	1,00	1,28	1,48	1,96	2,44	3,10
EV3	-	-2,35	-1,88	-1,65	-1,22	-1,02	1,18
EV1-Sintético	-1,06	1,06	1,16	1,51	1,79	2,65	-
EV2-Sintético	-	2,41	4,02	6,38	14,68	29,20	31,60
EV3-Sintético	-	3,70	5,17	7,21	9,22	11,53	19,79

Tabela A.58: Melhoria do Tempo Perdido (vezes) - Segunda Solução - Cenário Colônia - Onda de Choque

EV		Min L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	3,30	5,58	7,75	10,62	14,34	20,57
EV2	-	8,11	13,05	15,03	18,07	24,50	-
EV3	-	-1,01	1,28	1,85	2,90	4,89	7,84
EV1-Sintético	-	1,22	1,55	1,96	2,31	2,92	3,69
EV2-Sintético	-	14,62	21,40	33,12	68,99	135,45	151,59
EV3-Sintético	-	4,60	9,57	12,41	19,18	27,14	-

Tabela A.59: Melhoria do Tempo Perdido (vezes) - Segunda Solução - Cenário Colônia - Onda Verde

EV		Min L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	3,85	7,28	9,74	12,66	15,97	28,31
EV2	-	18,49	21,99	28,75	34,52	44,40	-
EV3	-	-1,94	-1,44	-1,31	-0,03	1,67	-
EV1-Sintético	-	1,19	1,54	1,90	2,29	2,90	7,65
EV2-Sintético	-	108,19	131,72	212,51	367,87	569,70	1.074,43
EV3-Sintético	-	25,65	35,80	49,53	76,80	120,62	-

Tabela A.60: Tempo Perdido (s) - Segunda Solução - Cenário Colônia - RFID

EV		Min L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	4.361,08	10.473,74	14.600,70	16.522,61	18.924,48	26.959,58
EV2	-	714,23	810,44	908,63	997,95	1.263,08	1.293,70
EV3	-	825,07	887,90	965,97	1.047,70	1.079,25	2.009,96
EV1-Sintético	-	370,50	463,02	523,62	554,33	684,57	722,28
EV2-Sintético	-	524,45	2.174,39	3.182,56	3.694,06	5.408,67	12.381,82
EV3-Sintético	-	762,07	1.139,97	1.397,74	1.662,62	2.299,99	2.967,90

Tabela A.61: Tempo Perdido (s) - Segunda Solução - Cenário Colônia - Fuzzy

EV		Min L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	5.758,65	7.277,61	10.230,94	14.372,69	18.248,92	-
EV2	-	1.009,07	1.207,53	1.506,21	2.023,88	2.706,47	-
EV3	-	1.200,65	1.346,58	1.563,94	1.818,30	2.327,36	3.366,99
EV1-Sintético	-	505,58	698,55	820,98	876,11	1.068,69	1.430,43
EV2-Sintético	-	722,81	1.256,30	1.886,94	3.038,74	5.701,39	10.697,44
EV3-Sintético	-	1.204,59	1.516,36	1.971,12	2.391,18	3.103,03	5.887,46

Tabela A.62: Tempo Perdido (s) - Segunda Solução - Cenário Colônia - Filas

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	1.626,21	2.218,37	2.715,18	3.715,15	4.016,57	19.579,81
EV2	-	538,63	740,86	832,80	1.003,28	1.115,20	2.065,74
EV3	-	2.762,86	3.556,41	3.752,08	4.218,55	4.956,94	5.861,19
EV1-Sintético	-	334,16	443,04	504,37	564,44	682,11	1.018,74
EV2-Sintético	-	545,16	658,49	745,25	1.061,79	1.311,84	-
EV3-Sintético	162,98	231,35	237,49	250,49	266,35	282,64	366,97

Tabela A.63: Tempo Perdido (s) - Segunda Solução - Cenário Colônia - Onda de Choque

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	882,36	1.215,89	1.306,42	1.453,29	1.660,69	1.950,41
EV2	-	61,29	71,96	83,75	112,37	124,48	194,83
EV3	-	471,33	912,77	1.199,86	1.743,08	2.925,95	-
EV1-Sintético	-	223,39	338,21	413,99	470,80	553,63	-
EV2-Sintético	75,62	150,79	156,90	161,47	167,41	180,73	273,01
EV3-Sintético	-	113,08	123,32	127,52	161,07	200,00	350,13

Tabela A.64: Tempo Perdido (s) - Segunda Solução - Cenário Colônia - Onda Verde

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	750,63	999,98	1.105,71	1.304,33	1.481,98	-
EV2	-	33,24	39,85	45,52	55,16	69,67	78,15
EV3	-	2.005,83	2.640,84	3.017,04	3.631,57	4.723,39	6.226,48
EV1-Sintético	-	153,88	323,02	407,74	484,57	532,60	-
EV2-Sintético	-	20,37	20,93	22,49	26,45	32,02	-
EV3-Sintético	-	23,66	26,34	36,32	40,45	57,67	148,73

Tabela A.65: Tempo Perdido sobre o Tempo Total de Viagem (%) - Cenário Alto Tempo de Teletransporte - Onda de Choque

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-608,74	-540,74	-473,31	-455,33	-406,15	-

Tabela A.66: Tempo Perdido Médio (%) - Todos os Veículos - Cenário Alto Tempo de Teletransporte - Onda de Choque

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-344,50	-317,67	-302,42	-294,40	-281,92	-

Tabela A.67: Velocidade Média das Viagens (%) - Todos os Veículos - Cenário Alto Tempo de Teletransporte - Onda de Choque

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-	-	-54,29	-52,86	-51,42	-

Tabela A.68: Veículos Teletransportados (%) - Cenário Alto Tempo de Teletransporte - Onda de Choque

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-2.132,00	-1.969,25	-1.832,00	-1.776,00	-1.656,00	-

Tabela A.69: Tempo Perdido (s) - Terceira Solução - Cenário Turim - Sem Preempção

EV		Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	1.953,57	2.445,69	2.690,07	3.215,12	3.634,60	6.903,89	
EV2	-	342,68	344,93	350,04	400,94	464,14	-	
EV3	-	7.115,05	8.860,67	9.862,92	11.157,36	13.199,45	-	
EV1-Sintético	-	4.899,60	5.321,66	5.483,41	5.693,07	6.059,70	7.935,61	
EV2-Sintético	-	855,40	1.072,16	1.273,76	1.493,92	1.893,74	-	
EV3-Sintético	-	5.318,58	6.153,23	6.973,82	7.902,80	9.305,26	-	

Tabela A.70: Tempo Perdido sobre o Tempo Total de Viagem (%) - Terceira Solução - Cenário Turim - Sem Preempção

EV		Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	72,43	76,68	78,34	81,18	87,08	90,27	
EV2	-	28,94	29,06	29,35	32,19	35,54	-	
EV3	-	87,73	89,91	90,85	91,82	92,99	-	
EV1-Sintético	-	86,81	87,73	88,05	88,44	89,06	91,41	
EV2-Sintético	-	50,35	56,00	60,23	63,95	69,22	-	
EV3-Sintético	-	84,25	86,09	87,53	88,82	90,36	-	

Tabela A.71: Melhoria do Tempo Perdido (vezes) - Terceira Solução - Cenário Turim - RFID

EV		Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-1,26	1,05	1,09	1,61	1,96	2,63	-	
EV2	-	1,37	1,41	1,75	2,11	2,63	3,39	
EV3	-1,17	1,02	1,18	1,29	1,98	2,86	4,16	
EV1-Sintético	-	-1,12	-1,03	1,02	1,08	1,94	-	
EV2-Sintético	-	-1,39	-1,12	1,21	1,65	2,15	-	
EV3-Sintético	-1,30	1,09	1,10	1,64	2,04	2,97	4,75	

Tabela A.72: Melhoria do Tempo Perdido (vezes) - Terceira Solução - Cenário Turim - Fuzzy

EV		Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-2,85	-2,05	-1,80	-1,40	-1,11	1,13	
EV2	-	-5,70	-2,57	-1,79	-0,02	1,46	-	
EV3	-	-1,43	-0,01	1,23	1,39	1,67	-	
EV1-Sintético	-1,55	-1,39	-1,22	-1,14	-1,09	-1,04	1,31	
EV2-Sintético	-	-4,38	-3,17	-2,50	-1,58	-1,24	1,64	
EV3-Sintético	-	1,07	1,55	1,82	2,04	2,40	3,27	

Tabela A.73: Melhoria do Tempo Perdido (vezes) - Terceira Solução - Cenário Turim - Filas

EV		Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	6,79	8,50	10,33	11,13	13,05	22,05	
EV2	-	1,04	1,16	1,24	1,58	2,17	3,35	
EV3	-	1,00	1,27	1,60	2,13	3,34	24,87	
EV1-Sintético	-	1,02	1,29	1,45	1,82	2,21	2,99	
EV2-Sintético	-	-1,51	-1,11	-0,01	1,05	1,60	-	
EV3-Sintético	-	3,13	4,37	5,02	7,10	8,21	12,53	

Tabela A.74: Melhoria do Tempo Perdido (vezes) - Terceira Solução - Cenário Turim - Onda de Choque

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	4,57	10,00	12,99	26,08	43,90	-
EV2	-	3,88	4,10	4,28	5,02	6,04	-
EV3	-	5,97	9,85	12,29	14,23	16,25	-
EV1-Sintético	-1,27	1,03	1,08	1,43	2,25	3,46	-
EV2-Sintético	-1,18	1,11	1,23	1,45	1,78	2,52	-
EV3-Sintético	-	3,63	5,06	5,86	6,90	9,05	9,68

Tabela A.75: Melhoria do Tempo Perdido (vezes) - Terceira Solução - Cenário Turim - Onda de Choque + TPN*

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	13,97	20,48	26,39	30,99	41,20	70,69
EV2	-	3,90	4,12	4,30	4,90	5,61	6,13
EV3	-	2,12	2,78	3,37	4,37	6,55	9,57
EV1-Sintético	-	2,04	2,68	5,05	5,93	7,72	16,11
EV2-Sintético	-	1,05	1,34	2,02	2,50	2,99	-
EV3-Sintético	-	2,67	3,32	3,74	5,48	7,29	16,94

Tabela A.76: Melhoria do Tempo Perdido (vezes) - Terceira Solução - Cenário Turim - Onda Verde

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	14,19	20,23	23,92	30,08	39,86	70,31
EV2	-	3,93	4,11	4,44	4,66	5,33	6,02
EV3	-1,29	1,02	1,26	1,51	2,12	3,29	4,92
EV1-Sintético	-	4,27	6,67	7,60	8,68	10,15	14,03
EV2-Sintético	-	1,14	1,78	2,22	2,60	3,64	-
EV3-Sintético	-	1,39	1,70	2,11	2,26	2,65	-

Tabela A.77: Melhoria do Tempo Perdido (%) - Terceira Solução - Cenário Turim - RFID

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-26,03	8,48	37,95	48,87	62,03	-
EV2	-	26,90	29,20	42,76	51,96	70,48	-
EV3	-	-16,75	15,51	22,74	49,50	75,95	-
EV1-Sintético	-	-12,08	-3,02	2,27	7,09	13,42	48,45
EV2-Sintético	-	-39,09	-11,63	17,04	39,19	53,42	-
EV3-Sintético	-	-30,19	8,94	38,98	51,05	78,96	-

Tabela A.78: Melhoria do Tempo Perdido (%) - Terceira Solução - Cenário Turim - Fuzzy

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-184,61	-105,11	-79,54	-39,62	11,52	-
EV2	-470,31	-348,92	-156,73	-79,23	-2,40	31,57	-
EV3	-	-42,55	-0,87	18,54	28,15	40,10	-
EV1-Sintético	-55,32	-39,40	-21,83	-13,96	-8,79	5,10	23,62
EV2-Sintético	-	-338,28	-216,57	-149,61	-57,82	38,93	-
EV3-Sintético	6,37	16,65	35,49	45,16	50,95	69,38	-

Tabela A.79: Melhoria do Tempo Perdido (%) - Terceira Solução - Cenário Turim - Filas

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	85,26	88,23	90,32	91,01	94,65	95,47
EV2	-	4,17	13,76	19,23	36,78	70,16	-
EV3	-	0,30	21,41	37,67	53,03	95,98	-
EV1-Sintético	-	2,38	22,29	31,18	44,81	66,59	-
EV2-Sintético	-50,75	-31,00	-10,98	-0,59	4,78	21,96	37,49
EV3-Sintético	-	68,03	77,09	80,08	85,86	92,02	-

Tabela A.80: Melhoria do Tempo Perdido (%) - Terceira Solução - Cenário Turim - Onda de Choque

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	78,10	86,06	90,00	92,30	96,17	97,72	-
EV2	-	74,21	75,60	76,65	79,94	83,45	-
EV3	83,25	85,22	89,81	91,86	92,97	93,85	-
EV1-Sintético	-	-27,46	7,36	30,07	55,37	71,06	-
EV2-Sintético	-	-18,34	18,79	31,15	43,73	60,37	-
EV3-Sintético	-	72,46	80,24	82,93	85,49	89,67	-

Tabela A.81: Melhoria do Tempo Perdido (%) - Terceira Solução - Cenário Turim - Onda de Choque + TPN*

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	92,84	95,11	96,21	96,76	98,59	-
EV2	-	74,34	75,75	76,73	79,56	83,70	-
EV3	-	52,79	63,94	70,28	77,12	89,55	-
EV1-Sintético	-	50,91	62,74	80,18	83,11	93,79	-
EV2-Sintético	-	4,54	24,04	50,53	60,00	66,59	-
EV3-Sintético	-	62,51	69,83	73,20	81,61	94,10	-

Tabela A.82: Melhoria do Tempo Perdido (%) - Terceira Solução - Cenário Turim - Onda Verde

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	92,95	95,06	95,82	96,68	98,58	-
EV2	-	74,56	75,68	77,47	78,52	82,48	83,38
EV3	-29,19	1,57	20,78	33,93	52,88	79,69	-
EV1-Sintético	76,56	83,97	85,01	86,84	88,48	92,87	-
EV2-Sintético	11,98	36,08	43,73	54,99	61,54	72,56	-
EV3-Sintético	-	28,09	41,27	52,65	55,82	62,22	-

Tabela A.83: Tempo Perdido (s) - Terceira Solução - Cenário Turim - RFID

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	1.025,42	1.461,94	1.867,37	2.840,27	4.363,23	-
EV2	-	137,03	188,49	246,80	252,00	297,96	-
EV3	-	3.985,38	5.908,55	7.375,76	8.080,38	9.328,88	-
EV1-Sintético	4516,46	4.681,48	5.178,93	5.357,36	5.584,66	6.114,66	-
EV2-Sintético	-	836,12	942,36	1.042,66	1.175,51	1.480,31	1.917,77
EV3-Sintético	-	2.407,14	3.558,61	4.357,12	6.938,19	9.607,94	-

Tabela A.84: Tempo Perdido (s) - Terceira Solução - Cenário Turim - Fuzzy

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	3.298,64	4.348,89	5.097,03	5.895,50	7.542,04	10.807,18
EV2	-	283,37	455,00	653,87	1.147,65	1.957,69	2528,30
EV3	-	6.540,43	7.429,50	8.375,15	9.755,28	12.765,35	-
EV1-Sintético	-	5.107,45	5.885,49	6.365,42	6.768,78	7.624,89	8.469,80
EV2-Sintético	-	1.156,54	2.360,28	2.922,42	3.710,72	5.445,24	-
EV3-Sintético	-	2.623,99	3.335,32	3.812,14	4.418,77	5.180,77	8.787,34

Tabela A.85: Tempo Perdido (s) - Terceira Solução - Cenário Turim - Filas

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	219,45	252,18	285,38	305,68	381,73	802,51
EV2	134,85	203,08	250,04	295,30	308,75	341,80	-
EV3	-	3.698,58	4.973,42	5.944,79	7.193,69	9.684,94	-
EV1-Sintético	1813,50	2.657,76	3.401,84	3.775,23	4.360,96	5.047,08	-
EV2-Sintético	-	973,58	1.155,87	1.321,90	1.453,07	1.875,93	-
EV3-Sintético	-	599,77	1.117,97	1.346,57	1.524,74	2.127,15	2.515,10

Tabela A.86: Tempo Perdido (s) - Terceira Solução - Cenário Turim - Onda de Choque

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	100,26	116,78	203,39	287,58	436,63	659,30
EV2	-	76,02	80,83	81,98	84,32	88,76	90,02
EV3	-	605,06	734,67	839,53	948,22	1.050,14	1.557,34
EV1-Sintético	-	1.790,82	2.738,73	3.976,33	5.677,53	6.656,13	-
EV2-Sintético	-	652,16	745,46	788,59	1.026,63	1.306,79	1.549,42
EV3-Sintético	-	866,46	973,84	1.228,36	1.365,67	1.889,15	-

Tabela A.87: Tempo Perdido (s) - Terceira Solução - Cenário Turim - Onda de Choque + TPN*

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	81,53	90,75	98,47	132,79	157,38	237,67
EV2	-	73,55	80,84	83,47	86,69	90,05	95,74
EV3	-	1.678,41	2.533,34	2.956,70	3.105,11	3.542,08	6.508,57
EV1-Sintético	-	719,16	925,56	1.096,35	2.050,09	2.910,01	-
EV2-Sintético	-	473,26	579,98	699,36	800,99	1.099,56	1.197,66
EV3-Sintético	-	646,06	1.445,82	1.862,37	2.134,59	2.753,29	-

Tabela A.88: Tempo Perdido (s) - Terceira Solução - Cenário Turim - Onda Verde

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	88,71	98,86	113,72	138,55	174,30	-
EV2	-	75,55	78,65	83,03	86,12	94,54	-
EV3	-	2.777,52	5.362,51	6.250,58	7.964,45	11.692,61	-
EV1-Sintético	-	565,68	646,06	727,50	809,74	853,22	1.396,70
EV2-Sintético	-	457,35	478,39	564,45	773,01	1.175,38	-
EV3-Sintético	-	3.077,83	3.309,57	3.527,82	3.642,71	4.006,78	-

Tabela A.89: Tempo de Preempção Médio (s) - Terceira Solução - Cenário Turim - RFID

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	265,17	265,17	265,17	265,17	265,17	-
EV2	-	211,32	211,32	211,32	211,32	211,32	-
EV3	-	238,33	238,33	238,33	238,33	238,33	-
EV1-Sintético	-	265,17	265,17	265,17	265,17	265,17	-
EV2-Sintético	-	211,32	211,32	211,32	211,32	211,32	-
EV3-Sintético	-	238,33	238,33	238,33	238,33	238,33	-

Tabela A.90: Tempo de Preempção Médio (s) - Terceira Solução - Cenário Turim - Fuzzy

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	52,35	52,54	52,66	52,70	52,88	53,00
EV2	40,61	41,12	41,45	41,79	41,79	41,79	42,33
EV3	-	47,30	47,30	47,30	47,30	47,30	47,59
EV1-Sintético	52,21	52,35	52,53	52,66	52,73	52,88	53,90
EV2-Sintético	-	41,79	41,79	41,79	41,81	41,83	42,33
EV3-Sintético	-	47,30	47,30	47,30	47,30	47,30	47,47

Tabela A.91: Tempo de Preempção Médio (s) - Terceira Solução - Cenário Turim - Filas

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	252,37	255,13	258,37	260,43	263,91	288,80
EV2	-	283,42	297,16	310,37	321,87	333,89	-
EV3	-	1.317,67	1.544,97	1.696,58	2.039,03	2.429,94	-
EV1-Sintético	-	847,26	926,56	963,97	1.001,19	1.056,69	-
EV2-Sintético	-	477,58	555,61	603,39	669,05	738,16	-
EV3-Sintético	-	569,76	795,52	855,00	956,94	1.189,85	1.339,79

Tabela A.92: Tempo de Preempção Médio (s) - Terceira Solução - Cenário Turim - Onda de Choque

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	74,86	78,36	96,67	113,81	165,40	185,06
EV2	-	77,05	79,63	81,39	82,32	83,47	-
EV3	-	223,79	248,39	259,65	295,74	332,79	371,67
EV1-Sintético	-	215,17	424,04	682,81	984,64	1.288,60	-
EV2-Sintético	-	145,16	189,24	205,55	257,82	295,00	367,89
EV3-Sintético	-	278,64	322,53	366,48	416,33	462,70	-

Tabela A.93: Tempo de Preempção Médio (s) - Terceira Solução - Cenário Turim - Onda de Choque + TPN*

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	433,17	435,49	437,67	440,76	445,46	457,43
EV2	-	286,00	289,58	294,21	299,05	309,05	-
EV3	-	588,64	614,05	667,73	705,77	823,58	-
EV1-Sintético	-	635,00	706,97	808,91	861,69	1.017,46	1.196,14
EV2-Sintético	-	376,11	489,71	530,97	628,37	722,11	-
EV3-Sintético	-	554,70	586,39	599,68	615,82	653,45	-

Tabela A.94: Tempo de Preempção Médio (s) - Terceira Solução - Cenário Turim - Onda Verde

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	836,40	846,76	861,99	885,33	921,20	-
EV2	-	918,74	922,24	927,29	930,05	939,47	-
EV3	-	2.776,00	5.937,27	7.240,61	8.961,50	12.689,03	-
EV1-Sintético	-	1.313,77	1.394,01	1.474,20	1.557,17	1.601,40	2.144,00
EV2-Sintético	-	1.301,68	1.322,97	1.408,47	1.617,68	2.020,58	-
EV3-Sintético	-	4.076,24	4.305,50	4.525,33	4.639,80	5.002,03	-

Tabela A.95: Tempo Perdido (s) - Terceira Solução - Cenário Colônia - Sem Preempção

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	4.176,11	5.824,07	8.129,62	13.151,37	19.632,63	25.871,33
EV2	-	1.516,73	1.866,22	2.060,76	2.496,83	3.311,94	5.865,62
EV3	-	3.073,22	3.592,01	5.466,25	9.833,88	16.215,85	-
EV1-Sintético	-	714,34	1.103,69	1.339,18	1.484,42	1.963,67	-
EV2-Sintético	-	1.858,62	2.496,09	2.840,77	3.083,73	3.382,42	4.080,30
EV3-Sintético	-	763,64	858,01	951,90	1.048,01	1.144,23	1.339,15

Tabela A.96: Tempo Perdido sobre o Tempo Total de Viagem (%) - Terceira Solução - Cenário Colônia - Sem Preempção

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	88,72	91,64	93,77	96,11	97,99	-
EV2	-	80,68	83,69	85,01	87,29	90,10	94,17
EV3	-	87,06	88,71	92,28	95,53	97,26	-
EV1-Sintético	57,33	65,41	67,48	71,58	73,63	78,70	-
EV2-Sintético	83,65	84,83	87,29	88,66	89,45	91,82	-
EV3-Sintético	-	62,54	65,23	67,56	69,64	74,56	-

Tabela A.97: Melhoria do Tempo Perdido (vezes) - Terceira Solução - Cenário Colônia - RFID

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-4,55	-1,14	1,24	1,56	3,89	-
EV2	-1,57	1,01	1,03	1,59	1,70	2,46	4,92
EV3	-	-1,03	1,92	2,91	4,58	7,05	-
EV1-Sintético	-	-1,56	1,37	1,97	3,55	4,40	-
EV2-Sintético	-1,23	1,00	1,05	1,43	1,76	2,00	-
EV3-Sintético	-	1,20	1,48	1,81	2,10	2,55	3,40

Tabela A.98: Melhoria do Tempo Perdido (vezes) - Terceira Solução - Cenário Colônia - Fuzzy

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-3,70	-1,50	0,00	1,40	2,52	-
EV2	-	-2,61	-1,69	-1,20	1,11	2,96	-
EV3	-2,50	-1,21	1,14	1,67	2,97	4,46	6,18
EV1-Sintético	-	-2,67	-1,44	1,30	1,59	2,79	-
EV2-Sintético	-	1,02	1,33	1,68	2,24	3,60	3,78
EV3-Sintético	-1,03	1,00	1,27	1,52	1,82	2,45	-

Tabela A.99: Melhoria do Tempo Perdido (vezes) - Terceira Solução - Cenário Colônia - Filas

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-2,91	-1,38	-1,18	1,80	6,33	-
EV2	-1,24	1,10	1,41	1,90	2,71	3,94	6,30
EV3	-	-1,86	-1,06	1,03	2,28	5,08	-
EV1-Sintético	-1,32	1,01	1,18	1,42	2,52	4,20	6,08
EV2-Sintético	-	2,55	4,87	5,96	7,38	10,92	16,97
EV3-Sintético	2,50	3,05	4,95	5,64	6,58	8,27	-

Tabela A.100: Melhoria do Tempo Perdido (vezes) - Terceira Solução - Cenário Colônia - Onda de Choque

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-1,91	1,29	2,84	5,21	9,92	16,05
EV2	-	9,81	12,77	16,21	19,67	26,15	32,88
EV3	-	-1,13	2,17	3,82	7,30	12,06	-
EV1-Sintético	-	2,37	3,64	4,81	7,49	12,14	16,58
EV2-Sintético	-	11,52	16,55	20,69	29,91	46,64	50,50
EV3-Sintético	-	19,23	21,70	25,42	29,08	35,28	-

Tabela A.101: Melhoria do Tempo Perdido (vezes) - Terceira Solução - Cenário Colônia - Onda de Choque + TPN*

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	2,73	3,94	6,97	9,35	15,85	20,01
EV2	-	9,30	12,02	16,55	22,03	29,63	-
EV3	-2,42	-1,36	1,24	1,63	3,08	5,24	7,58
EV1-Sintético	-	2,78	3,63	4,94	6,54	10,58	16,75
EV2-Sintético	-	49,86	57,67	67,84	96,25	153,68	188,99
EV3-Sintético	-	20,09	24,62	30,02	33,86	40,29	-

Tabela A.102: Melhoria do Tempo Perdido (vezes) - Terceira Solução - Cenário Colônia - Onda Verde

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-2,46	-1,67	0,03	2,06	2,46	24,16
EV2	-	7,51	12,70	19,16	27,10	34,31	73,85
EV3	-	-1,35	1,39	2,11	4,06	6,46	-
EV1-Sintético	-	2,35	4,70	6,10	8,36	12,86	-
EV2-Sintético	-	67,17	79,30	101,06	122,06	149,31	-
EV3-Sintético	-	24,07	27,26	31,78	35,32	40,58	48,26

Tabela A.103: Melhoria do Tempo Perdido (%) - Terceira Solução - Cenário Colônia - RFID

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-354,93	-64,10	-13,61	19,55	36,03	74,30	-
EV2	-57,49	-21,50	3,17	37,24	41,29	79,67	-
EV3	-2,76	24,19	47,82	65,60	78,14	85,83	-
EV1-Sintético	-56,42	-22,55	26,82	49,16	71,80	77,27	-
EV2-Sintético	-	-22,76	4,54	29,67	43,31	50,01	-
EV3-Sintético	-	16,82	32,38	44,62	52,41	70,63	-

Tabela A.104: Melhoria do Tempo Perdido (%) - Terceira Solução - Cenário Colônia - Fuzzy

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-270,02	-148,66	-49,96	0,10	28,73	60,35	-
EV2	-	-161,30	-69,45	-20,26	9,79	66,19	-
EV3	-150,12	-20,89	11,84	39,98	66,37	83,82	-
EV1-Sintético	-167,43	-122,71	-44,04	23,00	36,97	64,16	-
EV2-Sintético	-	1,56	24,42	40,31	55,40	73,57	-
EV3-Sintético	-	-3,48	21,33	34,23	44,97	59,21	-

Tabela A.105: Melhoria do Tempo Perdido (%) - Terceira Solução - Cenário Colônia - Filas

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-191,37	-125,67	-37,90	-18,07	44,34	84,20	-
EV2	-24,36	8,79	28,94	46,36	63,02	84,13	-
EV3	-	-85,98	-6,19	2,68	56,05	80,32	-
EV1-Sintético	-	-31,77	15,23	29,80	60,34	83,56	-
EV2-Sintético	60,76	69,95	79,44	83,18	86,44	94,11	-
EV3-Sintético	60,02	75,42	79,79	82,27	84,80	87,91	-

Tabela A.106: Melhoria do Tempo Perdido (%) - Terceira Solução - Cenário Colônia - Onda de Choque

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-90,90	-21,83	22,31	64,80	80,73	93,77	-
EV2	-	89,81	92,17	93,83	94,92	96,96	-
EV3	-12,90	34,27	53,81	73,40	86,24	91,70	-
EV1-Sintético	-	57,87	72,54	79,23	86,64	93,97	-
EV2-Sintético	-	91,32	93,93	95,15	96,64	98,02	-
EV3-Sintético	-	94,80	95,39	96,07	96,56	97,17	-

Tabela A.107: Melhoria do Tempo Perdido (%) - Terceira Solução - Cenário Colônia - Onda de Choque + TPN*

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	63,37	74,48	85,65	89,27	95,00	-
EV2	-	89,24	91,68	93,96	95,46	96,63	-
EV3	-141,56	-36,40	18,96	38,67	67,30	86,80	-
EV1-Sintético	-	64,06	72,45	79,77	84,70	94,03	-
EV2-Sintético	-	97,99	98,27	98,53	98,96	99,47	-
EV3-Sintético	-	95,02	95,92	96,67	97,05	97,52	-

Tabela A.108: Melhoria do Tempo Perdido (%) - Terceira Solução - Cenário Colônia - Onda Verde

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-145,68	-66,50	2,27	49,45	95,86	-
EV2	-	86,68	92,12	94,78	96,31	98,65	-
EV3	-	-34,60	28,12	52,61	75,34	84,51	-
EV1-Sintético	57,45	69,82	78,64	83,60	88,04	92,22	-
EV2-Sintético	-	98,51	98,74	99,01	99,18	99,33	-
EV3-Sintético	-	95,84	96,33	96,85	97,17	97,93	-

Tabela A.109: Tempo Perdido (s) - Terceira Solução - Cenário Colônia - RFID

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	3.979,02	6.320,78	8.122,68	10.514,94	13.972,77	21.838,29
EV2	-	1.050,58	1.196,39	1.466,07	1.749,94	2.119,54	3.016,83
EV3	-	1.542,97	1.840,31	2.047,39	2.479,88	2.969,57	4.776,92
EV1-Sintético	-	291,25	405,76	790,46	1.088,95	1.570,65	-
EV2-Sintético	-	1.416,69	1.900,15	2.166,72	2.479,06	2.789,16	3.580,63
EV3-Sintético	-	336,09	485,44	572,34	638,12	717,57	-

Tabela A.110: Tempo Perdido (s) - Terceira Solução - Cenário Colônia - Fuzzy

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	5.229,01	7.603,37	8.798,07	12.709,57	19.002,16	-
EV2	-	1.496,09	1.992,17	3.001,83	3.661,73	5.021,52	-
EV3	-	1.979,77	2.344,19	3.539,23	4.307,98	5.506,95	17.655,09
EV1-Sintético	-	537,54	757,75	1.239,78	2.177,59	3.304,43	-
EV2-Sintético	-	807,58	1.269,60	1.620,37	2.084,38	2.616,33	3.667,99
EV3-Sintético	-	430,63	580,57	627,21	759,87	1.013,15	1.102,05

Tabela A.111: Tempo Perdido (s) - Terceira Solução - Cenário Colônia - Filas

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	3.208,85	6.062,44	7.937,09	11.178,97	16.973,76	21.756,56
EV2	-	488,76	946,20	1.210,27	1.546,55	2.394,43	-
EV3	-	3.029,42	3.621,86	4.143,82	6.293,95	7.108,97	-
EV1-Sintético	-	307,38	588,68	932,38	1.216,33	1.476,79	-
EV2-Sintético	-	274,18	347,52	472,23	601,49	854,46	1.245,28
EV3-Sintético	-	139,06	154,06	166,45	174,99	176,19	453,51

Tabela A.112: Tempo Perdido (s) - Terceira Solução - Cenário Colônia - Onda de Choque

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	1.201,90	1.513,37	2.110,97	9.128,51	19.589,88	-
EV2	-	100,55	128,32	139,54	160,13	178,61	-
EV3	-	914,82	1.089,83	1.247,00	1.817,60	1.953,13	7.144,70
EV1-Sintético	-	116,93	193,48	250,88	362,95	465,01	655,14
EV2-Sintético	-	75,59	91,34	148,94	155,63	179,91	265,08
EV3-Sintético	-	32,20	36,19	39,17	40,98	46,69	48,86

Tabela A.113: Tempo Perdido (s) - Terceira Solução - Cenário Colônia - Onda de Choque + TPN*

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	694,78	1.113,99	1.340,21	1.786,78	2.524,88	4.592,42
EV2	-	80,98	119,56	141,70	180,95	248,33	359,92
EV3	-	1.543,24	2.207,88	2.647,16	5.624,53	9.840,54	22.871,76
EV1-Sintético	-	140,32	218,32	266,26	359,80	382,88	-
EV2-Sintético	-	22,30	29,24	32,11	51,30	64,59	-
EV3-Sintético	-	28,24	30,54	32,78	36,69	45,03	136,75

Tabela A.114: Tempo Perdido (s) - Terceira Solução - Cenário Colônia - Onda Verde

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	1.007,58	6.504,85	9.500,85	10.403,56	12.777,32	18.397,83
EV2	-	60,66	76,67	116,97	171,09	251,69	-
EV3	-	1.767,11	2.189,90	2.544,57	3.336,02	4.722,64	7.054,44
EV1-Sintético	-	108,73	161,71	234,48	308,84	463,21	677,81
EV2-Sintético	21,26	25,81	26,96	28,59	30,38	35,34	36,17
EV3-Sintético	-	27,75	29,27	30,68	32,36	36,43	-

Tabela A.115: Tempo de Preempção Médio (s) - Terceira Solução - Cenário Colônia - RFID

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	202,69	202,69	202,69	202,69	202,69	-
EV2	-	203,70	203,70	203,70	203,70	203,70	-
EV3	-	221,46	221,46	221,46	221,46	221,46	-
EV1-Sintético	-	202,69	202,69	202,69	202,69	202,69	-
EV2-Sintético	-	203,70	203,70	203,70	203,70	203,70	-
EV3-Sintético	-	221,46	221,46	221,46	221,46	221,46	-

Tabela A.116: Tempo de Preempção Médio (s) - Terceira Solução - Cenário Colônia - Fuzzy

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	39,04	40,54	40,54	40,54	40,55	40,56	-
EV2	40,69	40,74	40,74	40,74	40,74	40,74	-
EV3	-	44,29	44,29	44,29	44,29	44,29	-
EV1-Sintético	39,04	40,54	40,54	40,54	40,54	40,54	40,56
EV2-Sintético	-	40,74	40,74	40,74	40,74	40,74	-
EV3-Sintético	-	44,29	44,29	44,29	44,29	44,29	-

Tabela A.117: Tempo de Preempção Médio (s) - Terceira Solução - Cenário Colônia - Filas

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	969,19	1.398,94	1.716,92	2.579,35	4.153,88	-
EV2	-	303,96	379,30	511,04	626,83	900,44	-
EV3	-	1.044,62	1.290,31	1.613,98	2.506,15	3.388,38	-
EV1-Sintético	-	207,08	247,40	307,60	345,58	400,69	-
EV2-Sintético	-	228,15	243,72	256,80	266,31	282,41	-
EV3-Sintético	-	200,00	209,15	218,31	235,81	246,67	336,58

Tabela A.118: Tempo de Preempção Médio (s) - Terceira Solução - Cenário Colônia - Onda de Choque

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	246,38	277,58	391,25	2.040,27	4.239,27	-
EV2	-	95,37	98,19	101,07	103,89	109,26	-
EV3	-	213,50	233,92	273,04	348,96	389,21	1.250,00
EV1-Sintético	-	70,65	76,65	81,27	96,33	103,65	145,65
EV2-Sintético	-	88,26	97,70	112,26	115,43	140,81	-
EV3-Sintético	-	61,88	63,38	64,25	64,79	66,88	-

Tabela A.119: Tempo de Preempção Médio (s) - Terceira Solução - Cenário Colônia - Onda de Choque + TPN*

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	347,58	426,00	465,58	528,17	580,58	704,96
EV2	-	136,19	156,50	163,69	170,89	185,56	205,00
EV3	-	597,33	720,79	826,08	992,35	1.165,46	1.413,04
EV1-Sintético	-	134,23	150,13	165,46	177,48	192,00	-
EV2-Sintético	120,33	126,30	132,56	134,89	137,69	144,11	-
EV3-Sintético	-	136,00	136,73	137,44	138,88	141,38	203,38

Tabela A.120: Tempo de Preempção Médio (s) - Terceira Solução - Cenário Colônia - Onda Verde

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	1.542,42	7.039,58	10.034,56	10.938,81	13.310,96	18.932,42
EV2	-	426,22	442,22	483,28	537,31	617,22	-
EV3	-	2.227,96	2.649,19	3.004,19	3.794,85	5.183,54	7.515,33
EV1-Sintético	-	642,77	695,92	769,42	843,48	995,96	1.210,96
EV2-Sintético	387,59	391,59	393,59	394,59	396,59	399,59	402,59
EV3-Sintético	-	488,33	489,83	491,33	492,83	497,33	-

Tabela A.121: Tempo Perdido (s) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Sem Preempção

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	23090,32	23.360,41	27.320,67	28.535,17	29.975,78	33.170,31	-
EV2	-	3.517,76	3.652,60	3.923,32	4.372,78	4.686,57	-
EV3	-	1.434,76	1.525,31	1.615,01	1.704,20	1.797,47	2.694,25
EV4	-	17.967,55	19.788,05	20.974,66	22.504,36	25.346,19	-

Tabela A.122: Tempo Perdido sobre o Tempo Total de Viagem (%) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017)

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	96,33	96,57	96,89	97,01	97,15	97,42	-
EV2	-	87,66	88,06	88,79	89,83	90,44	-
EV3	-	68,42	69,74	70,93	72,03	74,89	80,28
EV4	-	96,52	96,82	96,99	97,19	97,50	-

Tabela A.123: Melhoria de Tempo Perdido (vezes) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - RFID

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-1,26	-1,04	1,06	1,14	1,29	-
EV2	-	-1,40	-1,13	1,04	1,15	1,38	-
EV3	-	1,01	1,56	1,81	2,14	3,00	-
EV4	-	-1,07	-0,01	1,03	1,09	1,22	-

Tabela A.124: Melhoria de Tempo Perdido (vezes) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Fuzzy

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-1,01	1,02	1,09	1,28	1,32	1,62	-
EV2	-	-1,19	-1,06	0,01	1,14	1,67	-
EV3	-	-1,49	-1,23	-1,10	1,05	1,76	-
EV4	-	-1,17	-1,05	-1,01	1,08	1,10	-

Tabela A.125: Melhoria de Tempo Perdido (vezes) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Filas

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-1,16	-1,02	1,25	1,43	1,60	-
EV2	-	-1,72	-1,50	-1,33	-1,17	-1,08	1,11
EV3	-1,00	1,16	1,34	1,57	1,97	2,19	-
EV4	-	-1,41	-1,19	-1,07	1,10	1,26	-

Tabela A.126: Melhoria de Tempo Perdido (vezes) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Onda de Choque

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	1,16	1,39	1,49	1,58	1,81	1,92
EV2	-	9,70	13,54	15,74	17,82	20,90	25,30
EV3	-	6,37	7,42	8,49	19,30	31,88	-
EV4	-1,07	1,02	1,10	1,14	1,22	1,25	-

Tabela A.127: Melhoria de Tempo Perdido (vezes) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Onda de Choque + TPN*

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-1,32	-1,06	1,05	1,13	1,33	-
EV2	-	1,34	7,06	29,95	43,04	67,51	-
EV3	8,14	29,38	29,88	32,85	36,79	43,43	60,64
EV4	-1,05	1,00	1,02	1,07	1,14	1,22	-

Tabela A.128: Melhoria de Tempo Perdido (vezes) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Onda Verde

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	1,23	1,41	1,63	1,85	2,10	-
EV2	-	24,56	34,32	46,19	52,19	65,07	-
EV3	-1,02	1,02	1,10	1,16	1,24	1,36	1,94
EV4	-	-1,06	-1,00	1,06	1,17	1,36	-

Tabela A.129: Melhoria de Tempo Perdido (%) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - RFID

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-26,14	-3,94	5,94	12,37	22,46	-
EV2	-	-39,71	-12,78	3,48	13,36	27,37	-
EV3	1,13	18,85	35,92	44,80	52,99	66,64	-
EV4	-	-7,03	-0,54	3,02	8,35	18,02	-

Tabela A.130: Melhoria de Tempo Perdido (%) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Fuzzy

	EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-1,14	8,41	21,98	24,52	38,10	-	
EV2	-	-18,59	-5,77	0,64	12,26	39,06	40,10	
EV3	-	-49,06	-22,86	-10,28	4,82	43,08	-	
EV4	-	-16,75	-4,82	-0,70	7,59	8,68	-	

Tabela A.131: Melhoria de Tempo Perdido (%) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Filas

	EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-16,12	-2,46	20,06	29,96	37,47	-	
EV2	-	-72,02	-49,87	-33,42	-17,27	9,98	-	
EV3	-	-0,00	25,17	36,35	49,21	54,24	-	
EV4	-	-41,49	-18,50	-7,34	8,69	20,84	-	

Tabela A.132: Melhoria de Tempo Perdido (%) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Onda de Choque

	EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	13,62	16,78	27,90	32,86	36,70	47,97	-	
EV2	89,70	90,11	92,62	93,64	94,39	96,05	-	
EV3	-	84,31	86,53	88,21	94,81	96,86	-	
EV4	-6,90	-1,51	9,44	12,45	17,74	20,04	-	

Tabela A.133: Melhoria de Tempo Perdido (%) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Onda de Choque + TPN*

	EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	-32,46	-6,07	4,57	11,60	24,73	-	
EV2	25,33	82,57	85,32	96,66	97,68	98,52	-	
EV3	87,72	96,60	96,65	96,96	97,28	97,70	98,35	
EV4	-	-5,07	2,21	6,51	12,56	17,98	-	

Tabela A.134: Melhoria de Tempo Perdido (%) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Onda Verde

	EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	18,38	29,04	38,51	45,88	52,47	-	
EV2	-	95,93	97,08	97,83	98,08	98,46	-	
EV3	-	-1,90	9,17	13,50	19,64	26,66	48,42	
EV4	-	-5,73	-0,47	5,38	14,30	26,48	-	

Tabela A.135: Tempo Perdido (s) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - RFID

	EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	23.180,47	25.707,04	27.259,23	29.303,74	30.921,15	35.255,66	
EV2	-	3.202,49	3.648,53	3.863,14	4.445,57	5.169,26	6.405,65	
EV3	-	629,41	825,10	898,66	1.043,68	1.360,51	1.628,82	
EV4	-	17.875,31	19.054,99	20.302,58	20.849,37	22.507,05	25.285,92	

Tabela A.136: Tempo Perdido (s) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Fuzzy

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	20.365,13	21.630,75	22.775,82	24.537,72	28.842,62	-
EV2	2680,68	3.049,82	3.622,44	3.764,99	4.074,78	4.386,40	4.926,04
EV3	-	1.431,94	1.540,43	1.771,78	1.985,56	2.551,80	2.886,59
EV4	-	19.564,22	20.069,11	21.170,00	22.853,54	24.516,00	-

Tabela A.137: Tempo Perdido (s) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Filas

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	19.066,44	20.370,13	25.662,97	27.683,23	30.865,61	-
EV2	-	3.905,63	4.768,15	5.259,35	5.896,62	7.085,80	7.838,20
EV3	-	716,24	909,73	1.065,26	1.257,96	1.711,91	-
EV4	-	18.077,82	20.207,81	22.805,39	25.907,96	28.950,72	-

Tabela A.138: Tempo Perdido (s) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Onda de Choque

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	17.118,46	18.115,47	19.356,91	20.577,97	22.372,74	-
EV2	-	206,89	229,23	255,93	275,29	342,36	510,66
EV3	-	61,34	84,84	214,37	223,10	390,72	-
EV4	-	16.864,39	17.983,65	18.889,27	20.150,97	22.081,83	-

Tabela A.139: Tempo Perdido (s) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Onda de Choque + TPN*

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	23.747,52	26.374,01	27.882,88	29.569,31	33.685,17	-
EV2	-	69,42	87,59	145,53	1.136,83	2.172,32	3.334,51
EV3	-	39,81	44,75	48,28	55,53	56,70	196,98
EV4	-	17.014,37	18.663,40	19.298,55	20.957,74	23.193,69	-

Tabela A.140: Tempo Perdido (s) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Onda Verde

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	14.581,32	16.047,51	17.287,06	19.134,60	21.284,22	-
EV2	-	65,63	74,90	93,40	117,97	166,17	-
EV3	-	1.242,68	1.274,29	1.397,39	1.441,09	1.555,76	-
EV4	-	16.914,40	17.991,72	19.002,75	20.549,57	22.439,81	-

Tabela A.141: Tempo de Preempção Médio (s) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - RFID

EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	230,07	230,07	230,07	230,09	230,09	-
EV2	-	142,00	142,00	142,00	142,00	142,00	-
EV3	-	149,04	149,15	149,17	149,24	149,35	-
EV4	-	184,00	184,01	184,07	184,07	184,07	-

Tabela A.142: Tempo de Preempção Médio (s) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Fuzzy

	EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-		45,90	46,12	46,62	46,77	47,38	-
EV2	-		28,71	28,71	28,71	28,71	28,71	-
EV3	29,89		29,95	30,03	30,05	30,11	30,11	-
EV4	20,50		36,74	36,79	36,81	37,12	37,12	-

Tabela A.143: Tempo de Preempção Médio (s) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Filas

	EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	7.784,02	8.877,81	9.377,26	9.755,75	10.323,80	11.439,95	
EV2	-	704,73	799,20	940,97	1.071,27	1.409,93		-
EV3	-	361,83	399,83	454,30	503,20	590,87	688,87	
EV4	-	5.680,35	7.753,15	8.573,52	9.194,21	10.203,79		-

Tabela A.144: Tempo de Preempção Médio (s) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Onda de Choque

	EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	2.395,00	2.647,29	2.833,15	2.955,40	3.324,05	3.607,09	
EV2	-	116,40	121,70	125,50	132,83	145,33	159,20	
EV3	-	60,70	63,11	80,91	86,07	112,87		-
EV4	-	1.285,00	1.351,55	1.538,41	1.731,92	2.085,26		-

Tabela A.145: Tempo de Preempção Médio (s) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Onda de Choque + TPN*

	EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	793,93	930,02	1.041,43	1.140,15	1.318,27	1.903,60	
EV2	-	185,00	200,67	220,50	245,47	292,73	360,20	
EV3	-	232,52	234,59	240,09	255,28	275,74		-
EV4	-	422,47	471,78	509,88	580,83	723,51		-

Tabela A.146: Tempo de Preempção Médio (s) - Terceira Solução - Cenário Centro Expandido (Metrô OD SP 2017) - Onda Verde

	EV	Min	L. inferior	Q1	Mediana	Q3	L. superior	Max
EV1	-	15.463,80	16.926,73	18.167,37	20.014,36	22.162,98		-
EV2	-	563,00	573,00	594,37	618,60	665,00		-
EV3	-	1.907,30	1.940,61	2.063,67	2.106,78	2.222,26		-
EV4	-	17.568,09	18.645,09	19.654,16	21.202,14	23.092,19		-