



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE AUTOMAÇÃO E
SISTEMAS

Georgios Joannis Ninos Neto

**Análise do Gerenciamento de Veículos Automatizados com Exploração Livre da
Área da Interseção**

Florianópolis - SC
2021

Georgios Joannis Ninos Neto

**Análise do Gerenciamento de Veículos Automatizados com Exploração Livre da
Área da Interseção**

Dissertação submetida ao Programa de Pós-Graduação em Engenharia de Automação e Sistemas da Universidade Federal de Santa Catarina para a obtenção do título de mestre em Engenharia de Automação e Sistemas.

Orientador: Prof. Rodrigo Castelan Carlson, Dr.

Coorientadores: Prof. Rodrigo Tacla Saad, Dr. e Prof. Fábio Luis Baldissera, Dr.

Florianópolis - SC

2021

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

Neto, Georgios

Análise do Gerenciamento de Veículos Automatizados com Exploração Livre da Área da Interseção / Georgios Neto ; orientador, Rodrigo Castelan Carlson, coorientador, Rodrigo Tacla Saad, coorientador, Fábio Luís Baldissera, 2021.

100 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Automação e Sistemas, Florianópolis, 2021.

Inclui referências.

1. Engenharia de Automação e Sistemas. 2. Engenharia de Automação e Sistemas. 3. Veículos automatizados. 4. Gerenciamento de Interseção. 5. Modelagem Discreta. I. Castelan Carlson, Rodrigo. II. Tacla Saad, Rodrigo. III. Luís Baldissera, Fábio IV. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia de Automação e Sistemas. V. Título.

Georgios Joannis Ninos Neto

**Análise do Gerenciamento de Veículos Automatizados com Exploração Livre da
Área da Interseção**

O presente trabalho em nível de mestrado foi avaliado e aprovado por banca
examinadora composta pelos seguintes membros:

Prof. Max Hering de Queiroz, Dr.
PPGEAS/UFSC

Prof. Werner Kraus Junior, Dr.
PPGEAS/UFSC

Prof. Manoel Mendonça de Castro, Dr.
DET/UFSC

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi
julgado adequado para obtenção do título de mestre em Engenharia de Automação e
Sistemas.

Coordenação do Programa de
Pós-Graduação

Prof. Rodrigo Castelan Carlson, Dr.
Orientador

Florianópolis - SC, 2021.

Este trabalho é dedicado a toda a minha família, em especial a minha irmã e aos meus queridos pais.

AGRADECIMENTOS

Agradeço à todas as pessoas especiais que estiveram ao meu lado durante esta jornada.

A minha mãe Graça, que me deu apoio emocional e financeiro para que eu pudesse me manter em Florianópolis, principalmente no primeiro ano de mestrado, onde não tinha bolsa de estudos.

Ao meu pai João, que igualmente me deu suporte na busca pela pós-graduação, sempre incentivando para que eu não desistisse. A minha irmã Carla, que me ajudou a tomar a decisão de mudar de cidade e investir neste projeto, saindo da zona de conforto em que me encontrava.

A meus tios, tias, primos e primas, os quais amo gigantesicamente e vibraram comigo, mesmo de longe.

Aos Professores Rodrigo Carlson, Rodrigo Saad e Fábio Baldissera, que sempre estiveram disponíveis e dispostos a ajudar.

A meus amigos do mestrado, pelos momentos divididos juntos, especialmente a turma dos “mestrados anônimos” José Diogo, Gabriel Bruzaca e Pedro Martins, que me acompanharam em diversas reuniões de estudos. Foi excelente contar com vocês.

A meus amigos José Silvan e Gisllane, que toparam o desafio de se mudarem junto comigo para Florianópolis e dividir todas as angústias e dificuldades financeiras.

A meus amigos Fernando, Luiz e Rafaela, que me escutaram sempre nos momentos mais difíceis quando pensei em desistir.

Agradeço, também, à CAPES pelo apoio financeiro durante parte da pesquisa. À Universidade Federal de Santa Catarina e ao Departamento de Automação e Sistemas por abrirem as portas para que eu pudesse realizar este sonho que era a minha dissertação de mestrado. Proporcionaram-me mais que a busca de conhecimento técnico e científico. OBRIGADO A TODOS!

RESUMO

O tráfego em grandes cidades e rodovias tem sua operação impactada pelos congestionamentos com grande gasto de tempo, consumo de combustível e erros humanos na condução de veículos. Neste trabalho, aborda-se o uso de um Gerenciamento de Interseção Cooperativa que coordene um ambiente composto de Veículos Automatizados e que atenda a requisitos como eficiência e segurança. Nesse sistema de tomada de decisão, a troca de informações entre veículos é usada para realizar movimentos que não necessariamente seguem os caminhos convencionados pelas leis de trânsito atuais, por meio de um modelo de exploração flexível em uma área discretizada da interseção. Desta forma, estabeleceu-se uma coordenação capaz de organizar veículos que precisam cruzar o mesmo local em instantes de tempo relativamente próximos. Assim, construiu-se abstrações de um espaço de estados, com ferramentas de verificação formal, usando sistemas de transição para executar as especificações de manobras dos Veículos Automatizados e a lista de prioridades de passagem a ser seguida. Para analisar a alcançabilidade de estados, optou-se pela implementação de técnicas de tratamento de grafos, a fim de identificar os menores caminhos seguros possíveis que um conjunto de veículos leva para sair do setor de recursos compartilhados. Os testes iniciais do processo de tomada de decisão passaram por reduções no modelo, a fim de as limitações em termos de processamento de dados e de memória. Posteriormente, analisa-se a evolução dos modelos finais para variadas condições iniciais e explora-se visualmente os comportamentos de uso da área de interseção. Ao final, a exploração exaustiva e automática dos estados atingíveis gerou de uma gama de alternativas de caminhos em diversos cenários, com a possibilidade de escalar o modelo para outras condições iniciais. Estes resultados devem contribuir para a adoção de uma nova perspectiva em relação às pesquisas de coordenação de interseções.

Palavras-chave: Veículos automatizados. Gerenciamento de Interseção. Modelagem discreta.

ABSTRACT

Traffic in large cities and highways has its operation impacted by traffic jams, with great expenditure of time, fuel consumption, and human errors in driving vehicles. This work addresses the use of a Cooperative Intersection Management that coordinates an environment composed of Automated Vehicles and that meets requirements such as efficiency and safety. In this decision-making system, the exchange of information between vehicles is used to perform movements that do not necessarily follow the paths established by current traffic laws, through a flexible exploration model in a discrete area of the intersection. In this way, a coordination system capable of organizing vehicles that need to cross the same place in relatively close moments was established. Thus, state-space abstractions were constructed, with formal verification tools, using transition systems to execute the Automated Vehicle maneuver specifications and the pass priority list to be followed. To analyze the reachability of states, it was decided to implement graph treatment techniques in order to identify the smallest possible safe paths that a set of vehicles takes to leave the shared resources sector. Initial tests of the decision-making process underwent reductions in the model to address limitations in terms of data processing and memory. Subsequently, the evolution of the final models for various initial conditions is analyzed and the use behaviors of the intersection area are visually explored. In the end, the exhaustive and automatic exploration of the attainable states generated a range of alternative paths in different scenarios, with the possibility of scaling the model to other initial conditions. These results should contribute to the adoption of a new perspective concerning intersection coordination research.

Keywords: Automated Vehicles. Intersection Management. Discrete Modelling.

LISTA DE FIGURAS

Figura 1 – Etapas realizadas no gerenciamento de coordenação proposto . . .	34
Figura 2 – Ilustrações de uma área de interseção discretizada a) uma interseção de quatro vias com tráfego único em cada direção, dividida em vinte células. b) uma interseção de quatro vias com tráfego em linha reta e com curvas à esquerda e à direita, dividida em sessenta e quatro blocos iguais c) Uma rotatória discretizada em vinte e quatro partes, com as seções do meio configuradas de acordo com o contorno do caminho circular.	35
Figura 3 – Ilustração da interseção discretizada e as opções de passos dos VAs. Os espaços próximos às esquinas não podem ser reservados por serem muito pequenos, mas assume-se que são utilizados momentaneamente quando os VAs precisam fazer conversões de pista. a) As setas em vermelho indicam uma alocação proibida, seja por se tratar de uma célula ocupada por outro VA, seja porque ser célula em uma via diferente do seu destino, ou por ser uma calçada. b) Exemplos de trajetórias seguidas pelos veículos de acordo com suas reservas em cada tempo discreto.	36
Figura 4 – Ilustração do esquema de reserva de nós. O eixo horizontal é a distância discretizada, enquanto o eixo vertical é o tempo discretizado. A reserva do VA é mostrada passo a passo, e as reservas do seu sucessor e do seu predecessor na lista de prioridade são destacadas pelas duas linhas vermelhas.	38
Figura 5 – Ilustração do FCFS. a) Escolha de id de acordo com o número de células até o destino. b) Escolha de id de forma randômica.	39
Figura 6 – Representação da coordenação em lotes. A área de influência da Unidade Central de Interseção (UCI) é destacada em amarelo, e os id são indicados nos círculos brancos (lote atual) e vermelhos (próximo lote).	39
Figura 7 – Representação de um gráfico de alcançabilidade, onde o estado final e_f desejado é destacado em verde, os estados de <i>deadlock</i> são ilustrados em laranja, e $k = 0, 1, 2, \dots, n$, dependendo da ramificação.	43
Figura 8 – Representação de um processo com estados a , b e c em forma de autômatos. a) Processo determinístico onde a transita para b , que por sua vez evolui c , o qual finaliza o ciclo ao atingir o estado a novamente. b) Processo não determinístico, onde a tem duas opções de transição, o que aumenta a gama de combinações de sequências.	45
Figura 9 – Esquema de <i>Model Checking</i> realizado pelo modelo FIACRE. . . .	47

Figura 10 – Aplicação do método de Dijkstra, cujo objetivo é alcançar o estado e_f . O caminho com menor custo é representado em setas verdes, os caminhos opcionais em preto, e os caminhos que findam em <i>deadlocks</i> em laranja.	49
Figura 11 – Processo de higienização, tratamento e análise dos dados.	50
Figura 12 – Representação do Cenário de interseção, com os tamanhos da malha (<i>grid</i>) discretizada, das células e dos veículos.	51
Figura 13 – Cenários verificados. a) Cenário 1 - Interseção de quatro fluxos. b) Cenário 2 - Interseção de três fluxos.	51
Figura 14 – Cenário 1. a) Esquema de reservas com o modelo convencional. b) Esquema de reservas com o modelo flexível.	52
Figura 15 – Cenário 2 com esquema de reservas no modelo flexível para a avenida principal.	53
Figura 16 – Cenário 2. a) Esquema de reservas com o modelo flexível para a avenida secundária com conversão à esquerda. b) Esquema de reservas com o modelo flexível para a avenida secundária com conversão à direita.	54
Figura 17 – Cenário 2. a) Esquema de reservas com o modelo convencional para a avenida principal. b) Esquema de reservas com o modelo convencional para a avenida secundária.	54
Figura 18 – Fluxograma de decisão utilizado pela UCI para exploração de estados.	56
Figura 19 – Verificação da variável M . a) Indexação a partir do último movimento realizado por um VA. b) Duas manobras seguras (em azul) e uma manobra brusca (em vermelho) entre os instantes t_1 e t_2	57
Figura 20 – Relação dos autômatos executados na modelagem FIACRE.	59
Figura 21 – Exemplo de informações iniciais e análise de alcançabilidade feita pelo TINA.	62
Figura 22 – Marcações geradas pelo TINA	62
Figura 23 – Representação dos estados e_k em forma de vetores, após o processo de higienização.	63
Figura 24 – Exemplo de uma lista de transições que relacionam todos os estados alcançáveis.	63
Figura 25 – Exemplo de uma lista de transições higienizada, pronta para ser consumida pelo NetworkX.	64
Figura 26 – Exemplo de como funciona as ramificações do grafo de alcançabilidade para o esquema de reservas da UCI.	65
Figura 27 – Exemplo de um dicionário Python contendo as informações de um caminho computado pelo NetworkX.	65

Figura 28 – Configurações de testes, onde os VAs são representados por quadradinhos e cada cor representa um fluxo de origem diferente. a) Posições iniciais de teste no Cenário 1. b) Posições iniciais de teste no Cenário 2.	66
Figura 29 – Configurações de testes, onde os VAs são representados por quadradinhos e cada cor representa um fluxo de origem diferente. a) Posições iniciais de teste para 17 VAs no Cenário 1. b) Posições iniciais de teste para 17 VAs no Cenário 2.	67
Figura 30 – Imagem do gerenciador de tarefas mostrando o desempenho do uso da memória RAM durante a execução modelo FIACRE com uma interseção com 17 VAs no Cenário 1 com movimentos flexíveis.	68
Figura 31 – Imagem do gerenciador de tarefas mostrando o desempenho do uso da memória RAM durante a construção do grafo de alcançabilidade, via a biblioteca Python NetworkX, para uma interseção com 14 VAs no Cenário 1 com movimentos flexíveis.	70
Figura 32 – 10 posições iniciais para o Cenário 1 com 14 veículos.	71
Figura 33 – 10 posições iniciais para o Cenário 2 com 14 veículos.	71
Figura 34 – Evolução de transições do Cenário 1 com o modelo de reserva flexível, a partir de 10 condições iniciais diferentes.	72
Figura 35 – Evolução de estados do Cenário 1 com o modelo de reserva flexível, a partir de 10 condições iniciais diferentes.	73
Figura 36 – Evolução dos estados de <i>deadlocks</i> no Cenário 1 com o modelo de reserva flexível, a partir de 10 condições iniciais diferentes	73
Figura 37 – Evolução de transições do Cenário 1 com o modelo de reserva convencional, a partir de 10 condições iniciais diferentes.	74
Figura 38 – Evolução de estados do Cenário 1 com o modelo de reserva convencional, a partir de 10 condições iniciais diferentes.	75
Figura 39 – Evolução de transições do Cenário 2 com o modelo de reserva flexível, a partir de 10 condições iniciais diferentes.	76
Figura 40 – Evolução de estados do Cenário 2 com o modelo de reserva flexível, a partir de 10 condições iniciais diferentes.	76
Figura 41 – Evolução dos estados de <i>deadlocks</i> no Cenário 2 com o modelo de reserva flexível, a partir de 10 condições iniciais diferentes	77
Figura 42 – Comparação entre as condições iniciais para 12 (à esquerda) e 14 VAs (à direita) na linha de evolução 3. Os VAs adicionais são destacados com um quadrado amarelo.	77
Figura 43 – Evolução de estados do Cenário 2 com o modelo de reserva convencional, a partir de 10 condições iniciais diferentes, para 8, 10, 12 e 14 VAs.	78

Figura 44 – Evolução de estados do Cenário 2 com o modelo de reserva convencional, a partir de 10 condições iniciais diferentes, para 8, 10, 12 e 14 VAs.	79
Figura 45 – Menor número de passos para esvaziar o <i>grid</i> para 14 VAs no Cenário 1.	80
Figura 46 – Menor número de rodadas para esvaziar o <i>grid</i> para 14 VAs no Cenário 1.	80
Figura 47 – Menor número de passos para esvaziar o <i>grid</i> para 14 VAs no Cenário 2.	81
Figura 48 – Menor número de rodadas para esvaziar o <i>grid</i> para 14 VAs no Cenário 2.	81
Figura 49 – Exemplo de uma sequência de reservas no Cenário 1 com 76 passos e 7 rodadas.	83
Figura 50 – Exemplo de uma sequência de reservas no Cenário 1 com 77 passos e 7 rodadas.	84
Figura 51 – Exemplo de uma sequência de reservas no Cenário 1 com 78 passos e 7 rodadas.	84
Figura 52 – Exemplo de uma sequência de reservas no Cenário 2 com 49 passos e 6 rodadas.	86
Figura 53 – Exemplo de uma sequência de reservas no Cenário 2 com 50 passos e 6 rodadas.	86
Figura 54 – Exemplo de uma sequência de reservas no Cenário 2 com 51 passos e 6 rodadas.	87
Figura 55 – Distribuição de passos no Cenário 1 para os caminhos com 7 rodadas. O 'X' vermelho indica a posição inicial do VA.	88
Figura 56 – Distribuição de passos no Cenário 2 para os caminhos com 6 rodadas. O 'X' vermelho indica a posição inicial do VA.	89
Figura 57 – Distribuição de passos no Cenário 2 para os caminhos com 8 rodadas. O 'X' vermelho indica a posição inicial do VA.	89
Figura 58 – Distribuição de caminhos com o modelo flexível no Cenário 1 para os caminhos com 76 passos e distribuídos por número de mudanças de faixa durante o trajeto.	90
Figura 59 – Distribuição de caminhos no Cenário 2 para os caminhos com 49 passos e distribuídos por número de mudanças de faixa durante o trajeto na avenida principal.	91

LISTA DE QUADROS

Quadro 1 – Uso da interseção nos modelos com discretização no tempo e espaço.	30
Quadro 2 – Uso da interseção nos modelos com planejamento de trajetória.	31
Quadro 3 – Uso da interseção nos modelos com regiões de colisões.	32
Quadro 4 – Processo determinístico no FIACRE.	46
Quadro 5 – Processo não determinístico no FIACRE.	46
Quadro 6 – Exemplo do emprego de uma fórmula LTL no FIACRE.	46
Quadro 7 – Sintaxe FIACRE que representa um exemplo de matriz <i>a</i> inicial.	58
Quadro 8 – Representação FIACRE do processo de verificação.	60
Quadro 9 – Exemplo de uma escolha no processo de tomada de decisão.	61
Quadro 10 – Sintaxe utilizada FIACRE para verificar o atingimento do estado <i>done</i>	61
Quadro 11 – Script básico para criação de um grafo no NetworkX.	64

LISTA DE TABELAS

Tabela 1 – Evolução de estados e transições para alcance do estado final nos primeiros testes do modelo flexível.	67
Tabela 2 – Evolução de estados e transições para alcance do estado final no primeiros testes do modelo convencional com critério de parada. . .	69
Tabela 3 – Segmentação de caminhos por passos para o Cenário 1.	82
Tabela 4 – Segmentação de caminhos por passos para o Cenário 2.	82
Tabela 5 – Segmentação de caminhos de caminhos por passos e rodadas. . .	83

LISTA DE ABREVIATURAS E SIGLAS

CCAC	Controle de Cruzeiro Adaptativo Cooperativo
FCFS	<i>First Come, First Served</i>
GIC	Gerenciamento de Interseção Cooperativo
GPS	<i>Global Positioning System</i>
I2I	Infraestrutura a Infraestrutura
LIDAR	<i>Light Detection And Ranging</i>
LTL	Lógica Temporal Linear
REGEX	<i>Regular Expression</i>
SDA	Sistemas de Direção Automatizada
SED	<i>Stream Editor</i>
SELT	<i>State-Event LTL</i>
SIT	Sistemas Inteligentes de Transporte
ST	Sistema de Transição
TINA	<i>Time Petri Net Analyzer</i>
UCI	Unidade Central de Interseção
V2I	Veículo a Infraestrutura
V2V	Veículo a Veículo
V2X	Veículo a X - <i>Qualquer Coisa</i>
VA	Veículo Automatizado
VMEI	<i>Vehicular Mutual Exclusion for Intersections</i>

SUMÁRIO

1	INTRODUÇÃO	17
1.1	OBJETIVOS	20
1.2	HIPÓTESE DE PESQUISA	20
1.3	ESTRUTURA DO TRABALHO	21
2	COORDENAÇÃO DE TRÁFEGO E MODELAGEM DE INTERSEÇÃO PARA VAS	22
2.1	MODELOS COM DISCRETIZAÇÃO DO TEMPO E ESPAÇO	24
2.2	MODELOS COM PLANEJAMENTO DE TRAJETÓRIA	25
2.3	MODELOS COM REGIÕES DE COLISÃO	27
2.4	USO DO ESPAÇO COMPARTILHADO NAS INTERSEÇÕES	29
3	MÉTODOS UTILIZADOS PARA IMPLEMENTAÇÃO DA UCI	34
3.1	MODELAGEM DISCRETA E SISTEMA DE RESERVAS	34
3.2	PROTOCOLOS DE COMUNICAÇÃO E SISTEMA DE PRIORIZAÇÃO	37
3.3	MODEL CHECKING	39
3.3.1	Sistemas de transição	41
3.3.2	Grafo de Alcançabilidade	42
3.3.3	Lógica Temporal Linear	44
3.3.4	Modelagem FIACRE	44
3.4	ANÁLISE DO GRAFO DE ALCANÇABILIDADE	47
4	MODELAGEM DO SISTEMA	51
4.1	FORMATOS DE EXPLORAÇÃO DA ÁREA DE INTERSEÇÃO	52
4.2	PREMISSAS PARA TOMADA DE DECISÃO	55
4.3	ESPECIFICAÇÕES DOS SISTEMAS DE TRANSIÇÃO	56
4.3.1	Configuração dos estados	59
4.3.2	Atualização das variáveis	60
4.3.3	Propriedade LTL	61
4.4	HIGIENIZAÇÃO DOS DADOS	61
4.5	CONSTRUÇÃO DO GRAFO DE ALCANÇABILIDADE	64
5	RESULTADOS E DISCUSSÕES	66
5.1	CONFIGURAÇÕES DE TESTE	66
5.2	ABSTRAÇÕES DO MODELO	69
5.3	EVOLUÇÃO DOS CENÁRIOS	70
5.3.1	Condições Iniciais	71
5.3.2	Cenário 1 com modelo flexível	72
5.3.3	Cenário 1 com modelo convencional	74
5.3.4	Cenário 2 com modelo flexível	75
5.3.5	Cenário 2 com modelo convencional	78

5.3.6	Passos e rodadas nos menores caminhos	79
5.4	ANÁLISE DE CAMINHOS	82
5.4.1	Segmentação por número de passos e rodadas	82
5.4.2	Sequências de reservas	83
5.4.3	Distribuição de passos	88
5.4.4	Variação de faixas	90
5.4.5	Definição de caminho	91
6	CONCLUSÃO E PERSPECTIVAS	93
	REFERÊNCIAS	95

1 INTRODUÇÃO

Hoje em dia, dirigir tornou-se uma tarefa bastante custosa, tanto em relação ao tempo dispendido para esta atividade, quanto em termos de consumo de combustível, principalmente nos engarrafamentos cotidianos que ocorrem nas ruas de grandes cidades e rodovias movimentadas. Além disso, a ocorrência de erros humanos na condução de veículos, geralmente relacionados a atos de imprudência, estresse ou cansaço, pode resultar em colisões com vítimas fatais. A Organização Mundial da Saúde (2018) aponta que o número de acidentes continua aumentando ao longo dos anos, tornando-se a oitava causa de morte de pessoas em todas as faixas de idade. Neste contexto, soluções para amenizar essas situações inconvenientes tem sido o objetivo de muitas pesquisas na academia e na indústria, onde algumas previsões apontam que o uso de Sistemas Inteligentes de Transporte (SIT) proporcionará um ambiente conectado entre os veículos, capaz de resultar em grandes avanços na segurança e mobilidade das pessoas (GREER *et al.*, 2018).

Explorar a implantação dos SIT pode contribuir para uma integração otimizada entre informação e comunicação em meio ao gerenciamento e operação de diversos veículos, permitindo que os condutores saibam as condições de tráfego antecipadamente, e, conseqüentemente, tomem melhores decisões. Ao se analisar, em tempo real, uma quantidade extensa de dados coletada de veículos, infraestruturas de trânsito e até de *smartphones* dos usuários de vias públicas, é possível extrair diversos benefícios. Por exemplo, em um cenário repleto de veículos equipados com tecnologias anti-colisão e com *Global Positioning System* (GPS) (traduzido do inglês por Sistema de Posicionamento Global), garante-se uma redução de riscos para todos que trafegam em uma determinada região, ao se estimar o tempo de viagem ao cruzar informações históricas com dados de velocidade, rota, posição, e indicações de potenciais ameaças nos arredores.

Quando se fala de SIT, é importante ressaltar que este conceito só foi viabilizado a partir da evolução da tecnologia da informação aplicada à automação veicular (GONZÁLEZ *et al.*, 2015). De acordo com a SAE International (2018), existem seis níveis de direção automatizada: sem automação (nível 0), assistência ao condutor (nível 1), automação parcial (nível 2), automação condicional (nível 3), automação elevada (nível 4) e automação completa (nível 5).

As características dos níveis 3, 4 e 5 englobam os chamados Sistemas de Direção Automatizada (SDA), que combinam hardwares e softwares capazes de realizar tarefas de direção dinâmica, como movimentos laterais, movimentos longitudinais, monitoramento do ambiente ao redor, e resposta operacional e tática, considerando diversas situações operacionais, como condições climáticas boas ou ruins, estradas esburacadas ou em perfeito estado, fluxo de veículos variável, e assim por diante. No

entanto, apenas os veículos que possuem os dois últimos níveis são considerados como SDA dedicados, ou seja, que operam sem a necessidade de uma interface de usuário.

Atualmente, os modelos de mercado se encontram no nível 3, onde o veículo é capaz de se guiar sozinho, mas o motorista precisa assumir a direção quando necessário. Contudo, diversas empresas automotivas já estão projetando lançamentos de sistemas de níveis 4 e 5 nos próximos anos (GIANINNI, 2021).

Para seu correto funcionamento, os SDA precisam de comunicação e cooperação com entidades externas, por isso, a SAE International (2018) recomenda cautela no uso do termo “autônomo” para descrever direção automatizada, visto que autonomia sugere a capacidade de governança própria. Então, neste trabalho, usa-se a nomenclatura Veículo Automatizado (VA), que alia tecnologias de sensoriamento, reconhecimento de padrões, inteligência de controle e compartilhamento de dados entre veículos para suportar todos os aspectos da tomada de decisão em uma condução segura dos passageiros. O advento dos VAs pode impactar também na qualidade de experiência dos usuários, que poderão focar em atividades recreativas, como ler, descansar ou ouvir música, em vez de se condicionar a situações de estresse provocadas pela necessidade de prestar atenção ao trânsito no caminho de casa ou do trabalho, por exemplo. Outros ganhos atrelados à implementação dos VAs são as reduções de congestionamentos, de consumo de energia, de emissões de gases, de custos com acidentes, além da possibilidade de uma locomoção independente para pessoas que não podem dirigir, seja por terem algum tipo de restrição física ou de habilitação (LITMAN, 2018).

Em relação ao controle dos VAs, as estratégias devem estabelecer uma condução eficiente satisfazendo restrições dinâmicas e desviando de obstáculos durante a trajetória (QIAN, 2016). Em outras palavras, é de suma importância planejar uma rota segura que considere uma mudança constante das circunstâncias do ambiente, como o movimento de outros entes na via. Por isso, o poder público, a indústria de transportes e a sociedade em geral, se preocupam constantemente com o status de segurança das pesquisas realizadas nesta área de estudo. Mesmo para um veículo nível 5, com automação completa, funcional e confiável, será preciso um tempo adicional para testes e aprovação regulatória, com avaliações de performance em condições reais de trânsito. Portanto, testes devem ser feitos exaustivamente antes da disponibilização em massa para a população, considerando, principalmente o cenário mais crítico para o desempenho dos VAs, que se trata do controle de fluxo nas interseções de vias urbanas e rodovias (MAHMASSANI, 2016). Isso se deve pelo fato dos cruzamentos serem responsáveis por uma parte significativa dos acidentes de trânsito, mesmo sendo uma parte relativamente pequena de todo o sistema viário (CHEN; ENGLUND, 2015).

Uma maneira de reverter este cenário é repensar os tipos de modelagem e os

métodos de coordenação de tráfego, com um Gerenciamento de Interseção Cooperativo (GIC). A ideia de um GIC é construir uma operação altamente inteligente, capaz de organizar veículos que desejam cruzar o mesmo local em horários relativamente próximos. Desta forma, os veículos se sujeitam a algumas regras para obter a melhor sequência de travessia. Para isso, existem algumas opções amplamente utilizadas, como a discretização de espaço e tempo, a otimização de trajetórias e o evitamento de regiões de colisão. É relevante dizer que não há limites estritos entre tais abordagens, podendo ocorrer o compartilhamento de alguns subconjuntos de teorias.

Os problemas de controle de cruzamentos e convergência de vias (dois fluxos se juntando em um única saída) são muito semelhantes e podem ser facilmente tratados com as mesmas abordagens (RIOS-TORRES; MALIKOPOULOS, 2015). Normalmente, existe uma área delimitada antes da interseção, onde os veículos trocam dados e definem uma lista de prioridades para a passagem de fato. Embora seja possível considerar a presença de semáforos em um GIC, a maioria das pesquisas concentra-se em interseções não semaforizadas, onde as comunicações sem fio (*wireless*) possuem papel vital na coordenação do tráfego (CHEN; ENGLUND, 2015). Assim, tão importante quanto definir os caminhos a serem seguidos, é estabelecer uma troca de informações entre os VAs, de forma a evitar falhas no envio e recepção de mensagens. Neste quesito, adota-se um ou mais sistemas de transmissão em meio as opções Veículo a Veículo (V2V), Veículo a Infraestrutura (V2I), Infraestrutura a Infraestrutura (I2I) ou Veículo a X - *Qualquer Coisa* (V2X).

Dada a gama de interações possíveis em um GIC, é essencial definir se a coordenação será centralizada, na qual há pelo menos uma tarefa geral decidida por uma unidade central, ou descentralizada, em que cada veículo determina sua política de governança. Em alguns casos, é possível estabelecer um formato híbrido, onde haja tanto processos distribuídos, quanto direcionados a partir de um infraestrutura auxiliar, como em um cenário onde não há nenhum sistema de coordenação fixa, mas há um veículo líder, selecionado por meio de negociações V2V, responsável por gerenciar o tráfego de forma temporária. Sendo assim, pensando em um contexto onde todos os motoristas sejam substituídos pelos SDA, alguns elementos como semáforos, placas de trânsito e leis atuais precisam ser reconsiderados, visto que são projetados para a ótica humana e não aproveitam a maior sensibilidade e precisão de agentes computadorizados (DRESNER; STONE, 2008). Com as diversas discussões em volta de um GIC e VAs, vários artigos propõem uma revisão dos conceitos gerais e tendências futuras de ambos os setores como em (RIOS-TORRES; MALIKOPOULOS, 2015; CHEN; ENGLUND, 2015; MAHMASSANI, 2016).

Diante do impacto que as interseções têm no desempenho global do tráfego nas cidades, este trabalho apresenta um modelo de GIC composto por veículos totalmente automatizados, onde o sistema de tomada de decisão utiliza a troca de informações

entre carros para realizar movimentos que não necessariamente seguem os caminhos convencionais atuais. Assim, durante a passagem dos VAs pela interseção, eles podem realizar movimentos que normalmente seriam considerados como "contra-mão", desde que possam cruzar sem riscos de colisões. A fim de tornar isso possível, construiu-se abstrações de um espaço de estados, usando sistemas de transição para executar as especificações de manobras dos VAs. Dessa forma, cada estado representa um arranjo único da interseção e cada transição de estado representa diferentes opções de movimento que os VAs podem adotar. Utilizou-se ferramentas de verificação formal para gerar um grafo de alcançabilidade de estados, e, posteriormente, os dados foram tratados com o uso de bibliotecas Python para identificar os comportamentos apresentados.

1.1 OBJETIVOS

Esta dissertação avalia uma utilização mais flexível das áreas de interseção em um tráfego composto exclusivamente por VAs, que atenda aos principais objetivos propostos na literatura para esses sistemas, como eficiência e segurança. Assim, os seguintes pontos nortearam o trabalho durante o desenvolvimento do processo de coordenação:

- Implementação de um modelo de tomada de decisão onde uma infraestrutura central aplica reservas prioritárias e regras de movimentação para ocupação de recursos compartilhados entre os VAs.
- Desenvolvimento de técnicas de abstração para reduzir esforços computacionais.
- Execução de cenários de interseção com número de VAs e condições variáveis de início e de destino.
- Análise e comparação dos diferentes caminhos seguros gerados.

1.2 HIPÓTESE DE PESQUISA

Este trabalho visa validar a seguinte hipótese: para modelar um GIC seguro, não necessariamente é preciso seguir as regulamentações de trânsito atuais, ou seja, os VAs não devem se restringir às faixas convencionais da interseção, para ampliar seu leque de opções. Uma premissa importante é que os VAs possuam SDA projetados para operarem adequadamente em interseções, com alta performance de sensores ópticos, radares, sistemas de *Light Detection And Ranging* (LIDAR), controles automatizados (direção, aceleração, frenagem, etc.), softwares, servidores, fontes de alimentação, redes sem fio, comunicações de curto alcance, relatórios rodoviários e GPS.

1.3 ESTRUTURA DO TRABALHO

Este trabalho possui seis capítulos, que são descritos a seguir. O Capítulo 2 cobre a revisão da literatura, incluindo as principais soluções propostas para a problemática de interseções em vias urbanas e rodovias, com um gerenciamento automatizado.

O Capítulo 3 demonstra o tipo de modelagem escolhida e as técnicas escolhidas para a exploração de todos os estados possíveis do sistema de coordenação. Além disso, aborda-se o método de Dijkstra, responsável por selecionar os melhores caminhos a partir da análise do grafo de alcançabilidade de estados.

No Capítulo 4 são detalhadas as premissas adotadas para o GIC e as especificações de movimentos dos VAs, ambas elaboradas por meio da linguagem de programação formal FIACRE. Adicionalmente, são apresentadas as aplicações do método de Dijkstra através da linguagem de programação Python.

No Capítulo 5 são evidenciadas as condições do ambiente de verificação utilizadas para os testes e, em seguida, expõe-se os resultados obtidos.

O Capítulo 6 traz as ponderações finais e sugestões de aprimoramentos para trabalhos futuros.

2 COORDENAÇÃO DE TRÁFEGO E MODELAGEM DE INTERSEÇÃO PARA VAS

Quando se considera a utilização de SIT e metodologias de coordenação de VAs, as principais premissas levadas em conta para avaliação da performance são a segurança e a eficiência operacional desses sistemas (CHEN; ENGLUND, 2015; RIOS-TORRES; MALIKOPOULOS, 2015; GONZÁLEZ *et al.*, 2015; KATRAKAZAS *et al.*, 2015; MAHMASSANI, 2016). Segurança é um requisito obrigatório, visto que tais meios tratarão diretamente do transporte de vidas humanas. Por isso, esse tipo de abordagem busca minimizar, ou até mesmo evitar, a intervenção manual no controle dos veículos, a fim de se impedir quaisquer riscos com acidentes ocasionados por atos falhos dos condutores. Em relação a eficiência operacional, muitos pontos podem ser debatidos, como a maximização do fluxo de veículos, a minimização do consumo de combustível e a emissão de gases poluentes, além do bem-estar dos passageiros. Alguns fatores focados no comportamento e nas características do modelo, também são tratados pela literatura, como a ausência de configurações com bloqueios permanentes de um ou mais veículos, a possibilidade de uma divisão justa de tomadas de decisão para todos os elementos envolvidos, a responsividade dos VAs à obstáculos inesperados, o grau de confiabilidade do sistema e até mesmo o esforço computacional para implementação dos mesmos. Entretanto, mesmo os pontos mencionados anteriormente sendo contornados, existe uma última barreira a ser quebrada até os VAs serem incorporados ao nosso cotidiano: a percepção que a sociedade como um todo terá sobre a condução automatizada.

A adoção em massa de inovações tecnológicas para meios de transportes costuma depender bastante delas estarem, ou não, disponíveis aos consumidores finais por um preço acessível (LITMAN, 2018). Com os VAs não seria diferente, e um bom parâmetro para analisar esse aspecto financeiro é o custo de desenvolvimento de sistemas de detecção de luz e alcance, ou LIDAR (do inglês *Light Detection and Ranging*), que são componentes de sensoriamento considerados essenciais pela maioria das empresas que pesquisam sobre VAs. Em 2014, um LIDAR custava de \$30.000,00 a \$85.000,00 (SHCHETKO, 2014), e, mesmo que atualmente gigantes do mercado, como Google e Velodyne, indiquem que suas tecnologias de LIDAR possuam despesas em torno de \$7.500,00 e \$8.000,00, respectivamente (AMADEO, 2017), os valores ainda permanecem altos para uma rápida adesão comercial. Adicionalmente, para uma operação ótima dos VAs, as fabricantes ainda devem considerar o investimento em tecnologias para outras funções como comunicação, condução e softwares.

Para medir outros impactos sociais dos VAs, Fagnant e Kockelman (2015) fazem um levantamento sobre regulamentações, ciberataques, confiança do público e privacidade de dados, e como as políticas públicas podem minimizar quaisquer consequências negativas da transição para uma frota amplamente automatizada de

veículos. Do ponto de vista moral e legal, Hevelke e Nida-Rümelin (2015) discutem como atribuir responsabilidades quando ocorrerem acidentes com carros totalmente automatizados, visto que serão sistemas livres de interferência humana. Por outro lado, Kalra e Paddock (2016) questionam em seu trabalho se os testes de direção realizados atualmente podem fornecer evidências suficientes para demonstrar a segurança dos VAs, antes de inseri-los no mercado. Estes estudos demonstram como ainda há muito o que se discutir até os VAs virarem uma realidade viável para a população mundial.

A fim de transformar os VAs em alternativas confiáveis perante o julgamento coletivo, os adeptos da ideia precisam apresentar indícios técnicos irrefutáveis, principalmente em estratégias de controle e de conectividade dos veículos. Segundo Talebpour *et al.* (2016), idealizar uma estrutura abrangente, que considere tanto tomadas de decisão relacionadas a aceleração dos veículos e mudanças de faixa, quanto a obtenção de informações em tempo real com comunicações V2V e V2I, terá um papel crucial na segurança e no desempenho dos VAs. Assim, os autores apresentam um modelo de simulação de tráfego que alia eventos de comunicação discreta e controle de direção em uma rede de veículos conectados para apontar como as falhas de comunicação impactam na performance de VAs. Visando discutir a evolução da conectividade de VAs em rede, Gerla *et al.* (2014) mostram um estudo de caso completo para destacar as vantagens e os desafios em relação à troca de informações e a prevenção de possíveis ataques de *hackers*. Os resultados do estudo apontam que uma arquitetura de conexão em nuvem será imprescindível para a performance dos VAs, que tendem a ser os maiores beneficiários dessa tecnologia, conforme ela tenha maiores avanços.

Focando nas possibilidades de gerenciamento de cruzamentos e junções de rodovias, Rios-torres e Malikopoulos (2015) destacam as principais linhas de desenvolvimento, separando-as em abordagens de controle centralizado e descentralizado, além de discutir possíveis tendências de pesquisa, tratando algumas das questões não respondidas. Já Chen e Englund (2015) sumarizam os trabalhos que projetam uma gestão totalmente cooperativa entre VAs e infraestruturas de interseção, classificando as modelagens em três tipos: discretização de espaço e tempo, modelagem de trajetória e modelagem de regiões de colisão.

Para fins de organização conceitual, optou-se por utilizar a classificação de Chen e Englund (2015) na revisão bibliográfica deste trabalho, seguindo a mesma interpretação dos autores de que não há uma separação clara entre os segmentos, visto que uma região de colisão pode ser considerada um subconjunto da discretização de tempo e espaço, ao passo que uma trajetória pode ser modelada como um número de blocos discretos consecutivos. Escolheu-se priorizar referências de gerenciamento de interseções isoladas, sem comunicação I2I, publicadas nos últimos dez anos, mas não deixando de mencionar publicações relevantes datadas em anos anteriores. Além das especificações de coordenação, também observou-se como a literatura explora o

uso da área compartilhada pelos VAs, tanto em cruzamentos, quanto em convergência de vias em um único percurso.

2.1 MODELOS COM DISCRETIZAÇÃO DO TEMPO E ESPAÇO

Discretizações são utilizadas principalmente para alocação agendada de recursos, movimentos e/ou tomadas de decisão. Essa abordagem simplifica a resolução de problemas contínuos, uma vez que é possível restringir a passagem de um indivíduo por vez em cada fração de espaço. Com a implementação de uma política que determina quais blocos de reserva serão ocupados pelo veículos, é possível reduzir drasticamente o tempo de atraso de cada veículo em relação aos tráfegos regidos por semáforos e placas de parada, pois necessitam de menos passos discretos para cruzarem a interseção. Geralmente, uma divisão mais granular pode ajudar a se ter um modelo mais detalhado, porém, ao custo de alto impacto na memória dos computadores utilizados na implementação dos mesmos.

Um exemplo clássico de sistema de reservas é o trabalho de Dresner e Stone (2005), que utiliza sistemas multiagentes para estabelecer uma dinâmica de comunicação entre agentes “condutores”, que precisam requerer a sua entrada em espaços específicos, e agentes “gerentes”, incumbidos de autorizar, ou não, essas requisições. Em um trabalho posterior, Dresner e Stone (2008) estenderam sua modelagem de reserva, adicionando uma política de *First Come, First Served* (FCFS), uma expressão em inglês que significa dar preferência para o agente que primeiro apresentou um pedido de travessia. O resultados mostraram que o método conseguiu reduzir drasticamente o tempo de passagem de cada carro em duas ordens de magnitude em relação a regimes semaforicos com tempos de espera variados. A modelagem ainda considerou uma priorização a mais para veículos de emergência como ambulâncias, carros de bombeiros e viaturas de polícia. Au *et al.* (2012) propõem uma abordagem preventiva onde um agente central reserva espaços suficientes para que os VAs possam executar ações evasivas, ao se depararem com riscos emergentes. Como resultado, uma vez que um plano de evasão seja executado, não haverá colisão mesmo que um dos veículos apresente falha mecânica e pare subitamente. Quem também utiliza um esquema multiagente é Jin *et al.* (2013), porém, o trabalho analisa a performance na formação de pelotões pelo VAs ao cruzarem o espaço compartilhado. Comparado com uma simulação sem pelotões, esta abordagem conseguiu reduzir o tempo de chegada em 30% e o consumo de combustível e emissões de dióxido de carbono em cerca de 23%.

Para diminuir o esforço computacional, diversos trabalhos optam por ter poucas células, ou até mesmo uma célula única, como é o caso da modelagem de Loos e Platzer (2011), onde argumentam que cruzamentos grandes e complexos são apenas uma compilação de duas configurações de pista básicas de cruzamentos e entroc-

mentos. Desta forma, analisam a interação de dois VAs, considerando-os como pontos iguais em cada uma das pistas, com uma condição de exclusão mútua na região espacial, garantindo um sistema sem colisões. Ahmane *et al.* (2013) desenvolvem uma coordenação onde cada via é dividida em três seções (zona de espera, zona de conflito e zona de saída), e cada seção é um subconjunto de espaços em uma Rede de Petri, onde os veículos, são representados por *tokens* que só podem transitar entre seções se atenderem alguns critérios. Assim, um controle central permite que dois VAs compartilhem a zona de conflito, se ambos estiverem na mesma faixa da rodovia e a uma distância segura, ou se estiverem em faixas contrárias, sem risco de colisão. O algoritmo de coordenação foi testado tanto em simulações, quanto em experimentos em ambientes reais, se mostrando uma técnica tão eficiente quanto o esquema de reserva tradicional.

Wuthishuwong e Traechtler (2013) trabalham com uma interseção de quatro vias simétricas, discretizada em nós e com comunicação V2I, quebrando uma programação dinâmica recursiva em sub-problemas, de forma a encontrar o menor caminho entre dois nós, ao passo que busca maximizar o fluxo de veículos considerando pesos que indicam o tempo absoluto para um nó ser liberado para outros VAs. Esta configuração garantiu ausência de colisões e todos os veículos se movimentaram continuamente pelo cruzamento, sem necessidade de paradas, um fenômeno conhecido como *Onda Verde*. Complementando este trabalho, Wuthishuwong *et al.* (2015) exploram com mais detalhes as velocidades médias em cada sentido de pista, as trajetórias, os tempos de cruzamento e a relação a velocidade média dos VAs e o fluxo da interseção. O modelo de Tlig *et al.* (2014) também se apresenta como livre de paradas ao ajustar as velocidades dos VAs para que eles passem alternativamente pelo recurso único do cruzamento, alocado por um agente central, garantindo segurança e um menor de tempo de passagem.

2.2 MODELOS COM PLANEJAMENTO DE TRAJETÓRIA

Este tipo de modelagem considera que os VAs sempre seguem trajetórias predefinidas ao cruzarem pelas interseções, logo, é possível implementar um controle que evite a sobreposição de trajetórias, fazendo ajustes finos em parâmetros como aceleração e frenagem. Ao planejar e garantir que os VAs executem esses padrões seguros, esta metodologia atende aos critérios de maximização da eficiência e minimização de colisões. Seguindo essas premissas, Lee e Park (2012) elaboram uma interseção com faixa única para reduzir a proximidade entre os caminhos que podem se sobrepor. Desta forma, um algoritmo de coordenação substitui um semáforo e consegue diminuir as paradas totais e, conseqüentemente, o tempo total de viagem. Utilizando processos de decisão de Markov, Brechtel *et al.* (2014) apresentam um sistema onde, basicamente, um VA em uma rua secundária regula sua velocidade para se encaixar

em uma região entre dois veículos em uma rua preferencial. Como premissa, o VA deve se manter a pelo menos 12 m da interseção e deve seguir as melhores opções em um *ranking* de pontos que avalia as opções de decisões como boas ou ruins. Para reduzir a complexidade do modelo, os autores escolheram tratar o espaço contínuo com uma discretização equidistante.

Onieva *et al.* (2012) coordenam a interação de um VA com um veículo dirigido por um ser humano, onde o primeiro se depara com uma gama de situações randômicas, que representam as decisões do segundo, e, usando um método de otimização baseado em algoritmo genético, sempre se adapta para garantir uma trajetória segura para ambos. Como resultados, este processo gerou 161 controladores distintos com lógica *fuzzy*, divididos em três níveis hierárquicos: detecção de manobra, cálculo de velocidade ideal e atuação. Makarem e Gillet (2013) abordam uma otimização descentralizada que produz uma sequência ideal de ajustes de aceleração em cada VA na zona de controle, com base na trajetória predita dos demais. O controle preditivo utilizou uma função custo quadrática e restrições lineares para garantir a redução do gasto de energia e trajetórias suaves, para cenários com e sem peloteamento de veículos, considerando um fluxo de 2400 VAs por hora.

Zhang *et al.* (2016) focam em uma mistura entre alocação com mútua exclusão da área da interseção com uma trajetória planejada em função de ajustes de aceleração dos VAs de forma descentralizada, desde a sua entrada na zona pré interseção, até a sua saída da zona de conflito. Diferente dos trabalhos citados anteriormente, aqui são considerados dois cruzamentos interligados, que aprimoraram em 42% o consumo de combustível e reduziram em 37,3% do tempo de viagem, comparado com um modelo de coordenação por semáforo. Já Rios-Torres e Malikopoulos (2016) optam por um cenário de entrelaçamento entre duas rodovias onde os VAs, ao entrarem na zona de controle, tem suas posições na fila de passagem selecionadas aleatoriamente por um controlador central. Após a ordenação dos VAs, um problema de controle ótimo é simulado, permitindo que os veículos avancem sem criar congestionamentos e sob a forte restrição de evitamentos de colisões, ao passo que reduz significativamente o consumo de combustível e o tempo de viagem.

Gregoire *et al.* (2014) propõem um planejamento de movimento para multi-robôs, a fim de permitir mais robustez e distribuição em futuros sistemas automatizados de gerenciamento de interseções. O modelo considera tanto restrições cinemáticas, quanto dinâmicas, em conjunto com um sistema de alocação por priorização, para evitar que haja a ocupação de uma região comum. Qian *et al.* (2015) aperfeiçoam a implementação citada anteriormente, e a utilizam em uma coordenação de interseção descentralizada focada em prever possíveis estados de colisão e em garantir a segurança dos envolvidos, mesmo que um estado não previsto venha a ocorrer. Para calcular e implementar sua melhor sequência de controles, é assumido que cada veículo se

move apenas para frente e sabe os estados preditos dos demais veículos.

Utilizando um acesso alternado ao espaço compartilhado, o trabalho de Wu *et al.* (2014) garante os parâmetros de segurança, tanto de forma centralizada, quanto de forma distribuída, onde os VAs negociam entre si as prioridades de travessia. A sua modelagem *Vehicle Mutual Exclusion for Intersections* (VMEI) (traduzida do inglês por exclusão mútua veicular para interseções) assegura uma eficiência maior em relação a coordenação via semáforos. Alonso *et al.* (2011) utilizam uma estrutura lógica na forma de um grafo, que relaciona arcos (estradas) e nós (interseções). Assim, a rota que um veículo deve seguir é definida por uma sucessão de nós, cujos arcos que os interligam são dotados de informações sobre toda a cadeia, e, portanto, pode-se determinar quais carros estão localizados nesses arcos e antecipar qualquer conflito em potencial. Os testes foram feitos na interação entre dois carros com direção manual e um VA, os quais respeitaram os sistemas de priorização impostos pelo modelo. Utilizando redes Bayesianas e filtros de janela de tempo, Noh (2018) buscou uma solução experimental em um veículo de testes que permita uma navegação segura em um cruzamento, sem ser excessivamente conservadora, mesmo em face de veículos que violam as regras de tráfego. Para prever todos os movimentos futuros possíveis, o modelo usa um mapa digital com informações geométricas e topológicas, e, a partir disso, determinar manobras apropriadas para a passagem dos VAs.

2.3 MODELOS COM REGIÕES DE COLISÃO

As regiões de colisões podem ser tratadas como união das características de discretização e de planejamento de trajetória, afinal ela ainda assume que os VAs são obrigados a seguirem percursos fixados, mas dessa vez, somente os pontos onde há uma sobreposição de caminhos, são considerados como recursos a serem reservados, sem necessidade de segmentar 100% da área de interseção. Desta forma, esta opção tende a retirar complexidades atreladas aos demais tipos de modelagem. O controle distribuído proposto por Naumann *et al.* (1998) associa *tokens* individuais, que possuem um conjunto de direções que cada VA pode tomar, a um único espaço de conflito na interseção. Quando um movimento ocorre, esses *tokens* são removidos de seus locais originais e repassados a um novo destino, em meio a uma negociação entre os VAs que sempre certifica a adoção de prioridades FCFS e que não há mais de um *token* em cada região crítica.

Marinescu *et al.* (2012) apresenta uma simulação onde o curso de uma avenida secundária se une ao de uma avenida principal, na qual os VAs tem preferência e não podem parar. Assim, os VAs na via secundária precisam se inserir em um ponto seguro na via principal, determinado por uma infraestrutura central, que evita a possibilidade de veículos na estrada principal de se moverem para aquele potencial espaço de conflito. Focando no controle de aceleração e frenagem, Hafner *et al.* (2013) apresen-

tam uma técnica validada experimentalmente para projetar controladores individuais que usam comunicação V2V para negociar a entrada na interseção, além de aplicar o controle automático apenas quando for absolutamente necessário para evitar uma colisão. O cenário utilizado foi a junção de duas pistas, onde um VA tem intenção de seguir reto, e o outro pretende fazer uma conversão à direita, gerando um ponto de colisão no meio da trajetória de ambos. Zohdy e Rakha (2016) utilizam um sistema que ajusta o perfil de velocidade de um pelotão de VAs equipados com Controle de Cruzeiro Adaptativo Cooperativo (CCAC), de modo que todos atravessem a interseção em sua respectiva velocidade máxima permitida. Ao considerar 16 pontos de conflitos, foi possível reduzir significativamente, a um nível global, o atraso médio e o nível de consumo de combustível em 90% e 45%, respectivamente, além analisar o impacto de mudanças climáticas no desempenho dos VAs.

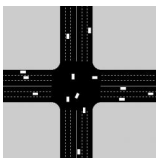
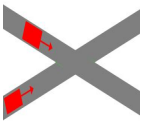
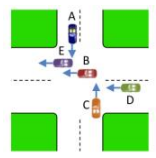
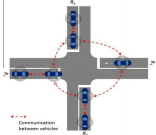
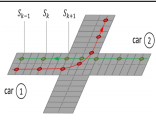
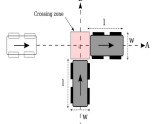
Com foco na qualidade de experiência do usuário e maximização de fluxo, o trabalho de Dai *et al.* (2016) se destaca por implementar um algoritmo de controle que analisa características cinemáticas, a fim de diminuir pequenas variações e paradas bruscas. A ideia é que um controlador central priorize a alocação de recursos (áreas de colisão entre rotas) para os veículos que estão há mais tempo na interseção, utilizando um algoritmo preditivo para determinar a melhor ação para o próximo instante de tempo, considerando diversas condições de tráfego. Lin *et al.* (2017) implementam um mecanismo centralizado de atribuições de passagem em uma interseção separada em *buffer area*, espaço anterior à interseção onde os VAs ajustam suas velocidades de acordo com as instruções recebidas, e *core area*, que é o núcleo da interseção composto pelas zonas de conflito. Em meio aos VAs, os cenários testados consideram a presença de carros dirigidos por seres humanos, que tem seu tempo de travessia reduzido quando atendem aos sinais de coordenação da unidade central.

Ahn e Del Vecchio (2017) demonstram uma solução que determina a existência de futuras regiões de colisão, e os atrasos são minimizados ao tratar o problema não-linear de reserva de recursos como dois sub-problemas lineares. A modelagem é simulada e validada em termos de segurança, desde um contexto mais simples com três VAs, até um cruzamento envolvendo vinte veículos e quarenta e oito áreas de conflito. No trabalho de Müller (2018) também é proposto uma estratégia de escalonamento ótimo, relacionado ao horário de chegada dos VAs, que permite uma grande eficiência viária e redução do tempo de execução, possibilitando uma implementação em tempo real mesmo com uma demanda de tráfego elevada. Para isso, o modelo aplica controles ótimos para balancear o tempo de travessia da interseção e o consumo de combustível.

2.4 USO DO ESPAÇO COMPARTILHADO NAS INTERSEÇÕES

Como visto nas seções anteriores, os modelos propostos na literatura usam como premissa o seguimento rigoroso das leis atuais de trânsito, com pouca flexibilização em relação ao uso de trajetos não convencionais em meio a interseções. Esta tendência também é apontada em trabalhos que fazem uma revisão mais profunda dos esforços de pesquisas para o gerenciamento de interseções automatizadas, como em Chen e Englund (2015), Rios-torres e Malikopoulos (2015), Katrakazas *et al.* (2015) e González *et al.* (2015). Como esta dissertação busca testar a hipótese de um uso mais maleável do espaço compartilhado entre os VAs em cruzamentos, elencou-se nos Quadro 1, Quadro 2 e Quadro 3, um resumo dos requisitos e restrições de movimento adotados pelas referências de modelagem levantadas.

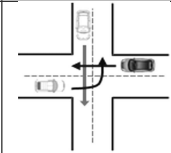
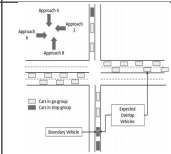
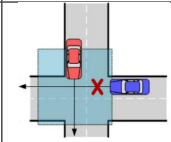
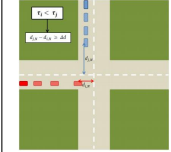
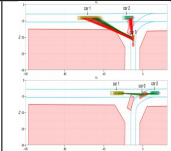
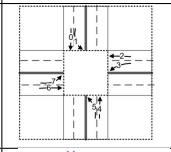
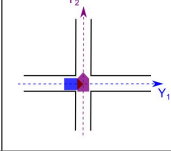
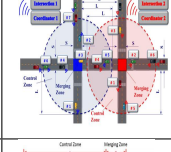
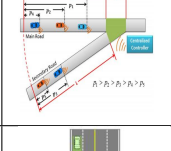

Quadro 1 – Uso da interseção nos modelos com discretização no tempo e espaço.

Autor(es)	Esquema gráfico feito pelo(s) autor(es)	Quantidade de células discretas (Máximo de células por veículo)	Faixas por via	Todos os veículos possuem movimento contínuo?	É permitido movimentos fora das trajetórias convencionais?	É permitida a conversão à esquerda/direita?
Dresner e Stone (2005, 2008)		576(8)	6	Não. Se o carro não ganha uma reserva de espaço, ele pode ficar parado, inclusive no meio da interseção	É permitido variar as faixas ao fazer uma conversão à direita ou à esquerda	Sim
Loos e Platzer (2011)		1(1)	1	Não. Se o carro não ganha uma reserva de espaço, ele pode ficar parado.	Não	Não
Au et al. (2012)		Não especificado	2	Não. Se o carro não ganha uma reserva de espaço, ele pode ficar parado, inclusive no meio da interseção	É permitido fazer manobras evasivas	Sim
Ahmane et al. (2013)		4(1)	2	Não. Se o carro não ganha uma reserva de espaço, ele pode ficar parado.	Não	Sim
Jin et al. (2013)	Sem esquema gráfico disponível	Recurso único, veículos passam em pelotões	1	Não. Se o pelotão não ganha uma reserva de espaço, ele pode ficar parado	Não	Não
Wuthishuwong e Traechtler (2013) e Wuthishuwong et al. (2015)		Discretização é iterativa (1)	2	Sim. Ajustam a velocidade para não conflitarem as reservas	Não	Sim
Tlig et al. (2014)		1(1)	2	Sim. Ajustam a velocidade para não conflitarem a reserva do recurso único	Não	Não

Fonte: Adaptado de Dresner e Stone (2005, 2008), Loos e Platzer (2011), Au et al. (2012), Ahmane et al. (2013), Jin et al. (2013), Wuthishuwong e Traechtler (2013) e Wuthishuwong et al. (2015) e Tlig et al. (2014).

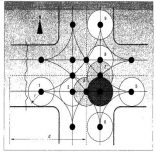
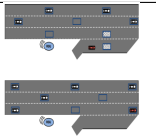
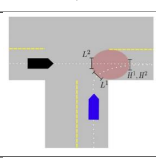
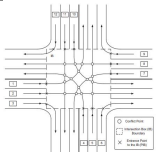
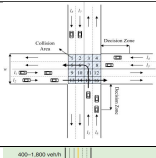
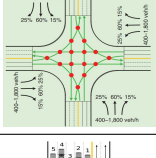
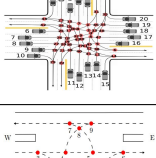
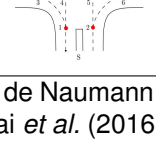
Os quadros resumos mostram que alguns modelos discretos permitem exceções quanto ao uso de caminhos predefinidos, como em Dresner e Stone (2005, 2008), que consideram a conversão de pista utilizando qualquer faixa, pois não há nada inerente ao sistema de reserva que exija o seguimento de uma faixa específica. Os testes realizados indicaram que ao fixar as faixas de conversão, a média de tempo é menor, porém, os autores ressaltam que não incluíram na modelagem a possibilidade

Quadro 2 – Uso da interseção nos modelos com planejamento de trajetória.

Autor(es)	Esquema gráfico feito pelo(s) autor(es)	Faixas por via	Todos os veículos possuem movimento contínuo?	É permitido movimentos fora das trajetórias convencionais?	É permitida a conversão à esquerda/direita?
Alonso <i>et al.</i> (2011)		2	Não. Só entram na interseção, quando não há risco de colisão na trajetória	Não	Sim
Lee e Park (2012)		4	Sim. Ajustam a velocidade para não conflitarem trajetórias	Não	Não
Onieva <i>et al.</i> (2012)		2	Não. Só entram na interseção, quando não há risco de colisão na trajetória	Não	Sim
Makarem e Gillet (2013)		2	Sim. Ajustam a velocidade para não conflitarem trajetórias	Não	Não
Brechtel <i>et al.</i> (2014)		1	Apenas os carros na via principal	Não	Duas vias convergem em uma
Wu <i>et al.</i> (2014)		4	Não. Só entram na interseção, quando não há risco de colisão na trajetória	Não	Sim
Gregoire <i>et al.</i> (2014) e Qian <i>et al.</i> (2015)		1	Não. Só entram na interseção, quando não há risco de colisão na trajetória	Não	Não
Zhang <i>et al.</i> (2016)		2	Sim. Ajustam a velocidade para não conflitarem trajetórias	Não	Não
Rios-Torres e Malikopoulos (2016)		1	Sim. Ajustam a velocidade para não conflitarem trajetórias	Não	Duas vias convergem em uma
Noh (2018)		4	Não. Só entram na interseção, quando não há risco de colisão na trajetória	Não	Sim

Fonte: Adaptado de Alonso *et al.* (2011), Lee e Park (2012), Onieva *et al.* (2012), Makarem e Gillet (2013), Brechtel *et al.* (2014), Wu *et al.* (2014), Gregoire *et al.* (2014), Qian *et al.* (2015), Zhang *et al.* (2016), Rios-Torres e Malikopoulos (2016) e Noh (2018).

Quadro 3 – Uso da interseção nos modelos com regiões de colisões.

Autor(es)	Esquema gráfico feito pelo(s) autor(es)	Quantidade de regiões de colisão	Faixas por via	Todos os veículos possuem movimento contínuo?	É permitido movimentos fora das trajetórias convencionais?	É permitida a conversão à esquerda/direita?
Naumann <i>et al.</i> (1998)		16	2	Não. Só entram na interseção, quando não há risco de colisão na trajetória	Não	Sim
Marinescu <i>et al.</i> (2012)		Variável	Principal - 3, Secundária - 1	Sim. Ajustam a velocidade para alocarem pontos seguros na trajetória	Não	Duas vias convergem em uma
Hafner <i>et al.</i> (2013)		1	2	Sim. Ajustam a velocidade para alocar o único ponto seguro na trajetória	Não	Duas vias convergem em uma
Zohdy e Rakha (2016)		16	6	Não. Só entram na interseção, quando não há risco de colisão na trajetória	Não	Sim
Dai <i>et al.</i> (2016)		16	4	Sim. Ajustam a velocidade para alocarem pontos seguros na trajetória	Não	Sim
Lin <i>et al.</i> (2017)		16	4	Sim. Ajustam a velocidade para alocarem pontos seguros na trajetória	Não	Sim
Ahn e Del Vecchio (2017)		48	6	Sim. Ajustam a velocidade para alocarem pontos seguros na trajetória	Não	Sim
Müller (2018)		9	2	Sim. Ajustam a velocidade para alocarem pontos seguros na trajetória	Não	Sim

Fonte: Adaptado de Naumann *et al.* (1998), Marinescu *et al.* (2012), Hafner *et al.* (2013), Zohdy e Rakha (2016), Dai *et al.* (2016), Lin *et al.* (2017), Ahn e Del Vecchio (2017) e Müller (2018).

do VAs realizarem mudanças de faixa antes da interseção, o que poderia impactar na performance final. Au *et al.* (2012), que também segue o modelo discretizado de Dresner e Stone (2005), apresentou um GIC com planos de *backup*, considerando a possibilidade de falhas mecânicas de cada veículo durante um intervalo de tempo. Os resultados foram atualizados de forma incremental e armazenados em um banco de dados como estratégias de evasão. Os modelos com planejamento de trajetória e regiões de colisão, por suas características intrínsecas de seguimento de caminhos padrões, obviamente não apresentaram alternativas de uso da zona de interseção.

Outros aspectos levantados nos quadros resumos são as variadas formas de configuração que cada autor considera em sua modelagem relacionadas ao número de faixas e a conversão de pista à direita ou esquerda, fatores que impactam diretamente no grau de complexidade da implementação, uma vez que mais faixas indicam um fluxo maior de veículos, enquanto que múltiplas opções de tomada de decisão implicam em mais variáveis de coordenação. Observa-se também como algumas referências eliminam a possibilidade de paradas durante os trajetos para minimizar o tempo de passagem pela interseção, realizando ajuste nos perfis de velocidade ou otimizando a sequência de reserva de recursos.

3 MÉTODOS UTILIZADOS PARA IMPLEMENTAÇÃO DA UCI

Para testar a hipótese de pesquisa, optou-se por uma modelagem do tipo discreta, pois como visto anteriormente, este formato permite certa flexibilidade na ocupação dos espaços e nas tomadas de decisão de movimento de cada veículo. Em relação à reserva de células, considerou-se que os VAs respeitam uma política de um veículo por área discretizada durante um período predeterminado, de forma que nenhum acidente ocorra, mesmo com mudanças constantes de faixas no trajeto. Adicionalmente, adotou-se como premissa que os veículos saiam da área de interseção sem paradas ao longo do percurso.

O GIC possui uma comunicação V2I, onde os VAs, ao se aproximarem da zona de influência de uma UCI, devem indicar uma intenção de passagem e se submeter a todas as diretrizes da infraestrutura gerenciadora, que terá total acesso às suas informações. Com base nos dados de cada solicitante, a UCI calcula todos os caminhos seguros possíveis e escolhe um deles como referência para fazer o agendamento de recursos. Esta exploração de trajetórias é feita em cima de diversas ramificações do estado inicial, criadas a partir da combinação das opções individuais de movimento. Para garantir reservas mutuamente exclusivas, a UCI segue uma sequência de prioridades, e, ao final da verificação, se encarrega de transmitir as manobras que devem ser seguidas. Os VAs atualizam suas posições com base nas reservas alocadas, até que completem uma travessia segura dentro daquele cenário. Uma visão macroscópica da coordenação é mostrada na Figura 1.

Figura 1 – Etapas realizadas no gerenciamento de coordenação proposto



Fonte: Autoria própria.

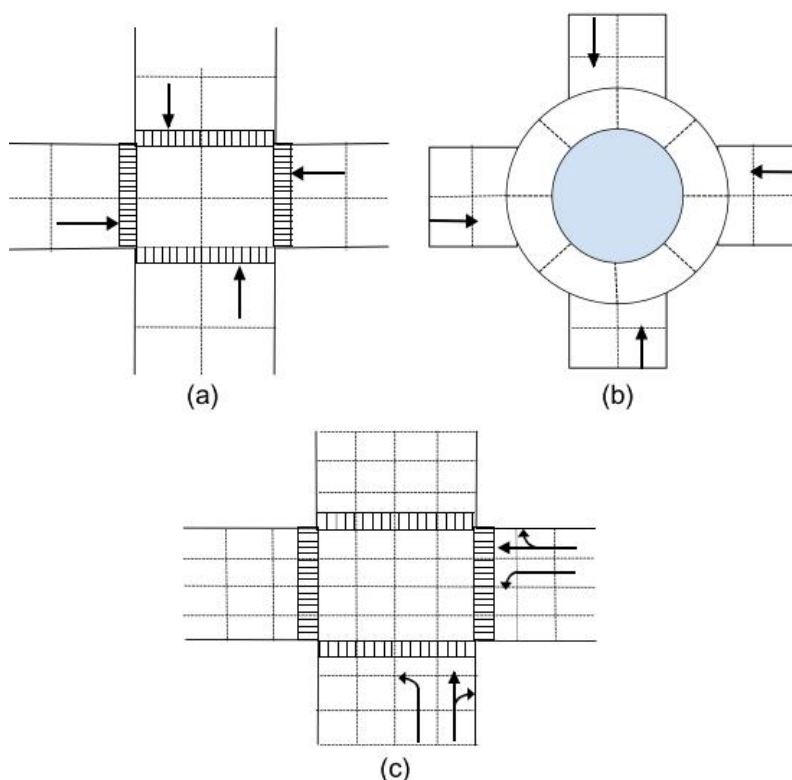
Embora se busque que os VAs façam movimentos livres de padrões predefinidos, por questão de organização, definiu-se que eles precisam entrar na zona de influência da UCI pelo sentido convencional, não podendo surgir na interseção pela contramão.

3.1 MODELAGEM DISCRETA E SISTEMA DE RESERVAS

Como se trata de um sistema de alocação de recursos, um ponto importante a se definir é como realizar a divisão dos componentes espaciais e temporais do modelo. O nível de granularidade da área de interseção é determinado pelo número e pelo tamanho das células que o espaço compartilhado pelos VAs possui. Geralmente,

adotam-se células quadradas de proporções iguais e grandes o suficiente para serem ocupadas por um carro de medidas padrões, mas outros moldes podem ser implementados, como visto na Figura 2.

Figura 2 – Ilustrações de uma área de interseção discretizada a) uma interseção de quatro vias com tráfego único em cada direção, dividida em vinte células. b) uma interseção de quatro vias com tráfego em linha reta e com curvas à esquerda e à direita, dividida em sessenta e quatro blocos iguais c) Uma rotatória discretizada em vinte e quatro partes, com as seções do meio configuradas de acordo com o contorno do caminho circular.



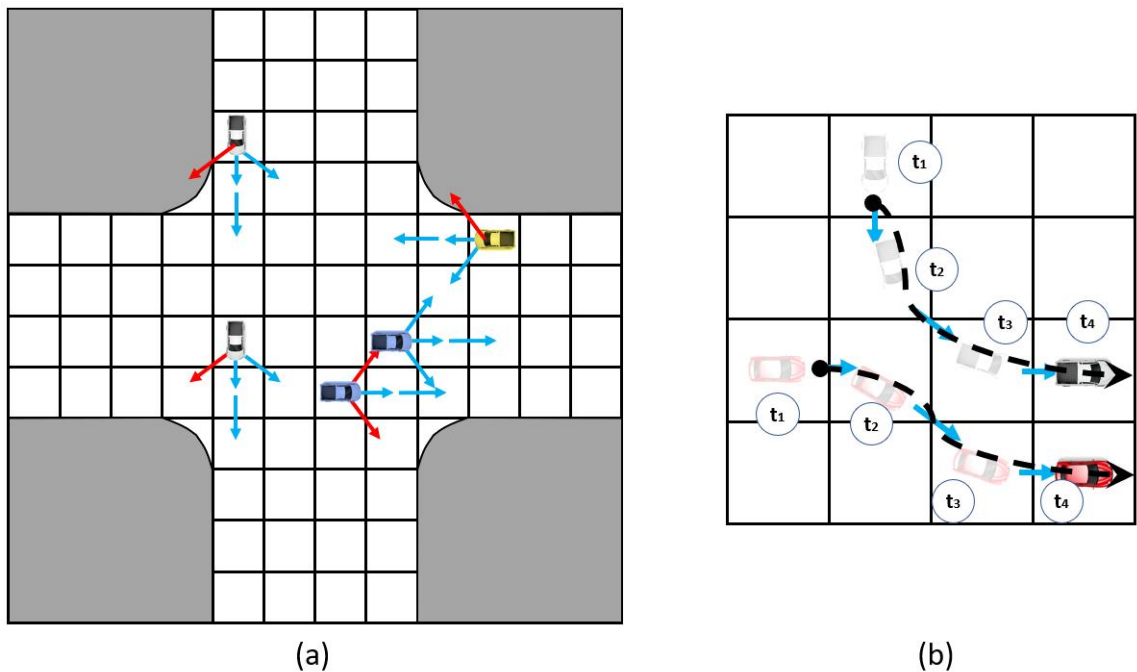
Fonte: Chen e Englund (2015).

O número de parâmetros para se efetuar uma reserva abre margem para uma infinidade de sequências que um veículo pode seguir. Por exemplo, em alguns cenários, a variação de velocidade pode fazer com que um VA colida com o outro, enquanto em outros pode deixar o veículo passar com segurança. Por esse motivo, a aceleração na interseção deve ser restringida pela política de controle da interseção (DRESNER; STONE, 2008).

Neste trabalho, um passo discreto significa uma mudança de célula, e em cada oportunidade de movimento os veículos podem avançar no máximo duas células em linha reta, uma célula na diagonal para a direita ou uma célula na diagonal para a esquerda, conforme exemplificado na Figura 3. É importante frisar que a possibilidade de andar uma ou duas casas não são excludentes, ou seja, se um veículo pode realizar as duas ações, duas ramificações diferentes serão criadas, pois ambas as opções são

consideradas pela UCI .

Figura 3 – Ilustração da interseção discretizada e as opções de passos dos VAs. Os espaços próximos às esquinas não podem ser reservados por serem muito pequenos, mas assume-se que são utilizados momentaneamente quando os VAs precisam fazer conversões de pista. a) As setas em vermelho indicam uma alocação proibida, seja por se tratar de uma célula ocupada por outro VA, seja porque ser célula em uma via diferente do seu destino, ou por ser uma calçada. b) Exemplos de trajetórias seguidas pelos veículos de acordo com suas reservas em cada tempo discreto.



Fonte: Autoria Própria.

Cada seção discreta pode ser representada como um nó inserido em uma rede interligada, que possui coordenadas de posição x e y específicas. Então, a distância percorrida é calculada pela soma das distâncias equivalentes de cada célula discreta:

$$\sum_{k=0}^f s_{i,k} = S_{i,f} \quad (1)$$

onde i é o índice de identificação (id) de um VA na lista de prioridades da UCI, k é um passo discreto dado ao mudar de célula, f é o passo final de um VA dentro da zona de influência da UCI, $s_{i,k}$ é a distância discreta do passo k da trajetória de um VA e $S_{i,f}$ é a distância total feita por um VA.

A posição de um VA, atualizada quando há uma mudança de célula, é dada por:

$$P_{i,k} = [x_i(s_{i,k}), y_i(s_{i,k})]; P_{i,k} \subseteq S_{i,f} \quad (2)$$

onde $P_{i,k}$ é a posição discreta de um VA em relação a um plano cartesiano.

O tempo que um VA percorre os espaços reservados é discreto e constante, sendo atualizado a cada passo k dado. Já a mudança de espaço de todos os veículos é sincronizada. Os critérios adotados para discretização do tempo são:

$$\dots < t_{k-1} < t_k < t_{k+1} < \dots; t_k \leq T_f, \quad (3)$$

$$t_{k+1} - t_k = \Delta t, \quad (4)$$

$$\sum_{k=0}^f t_k = T_f, \quad (5)$$

onde t_k é o tempo discreto em cada passo k , T_f é o tempo total de travessia e Δt é o tamanho de cada tempo discreto.

O método de discretização implementado neste trabalho é baseado no modelo apresentado em Wuthishuwong e Traechtler (2013) e Wuthishuwong *et al.* (2015), onde o processo de reserva é feito através do cálculo iterativo dos parâmetros dos nós e cada veículo só define sua reserva de espaço após a escolha do veículo predecessor na lista de prioridades. Na Figura 4, ilustra-se os esquemas de reserva de nós, que alia as coordenadas de distância e tempo de cada veículo.

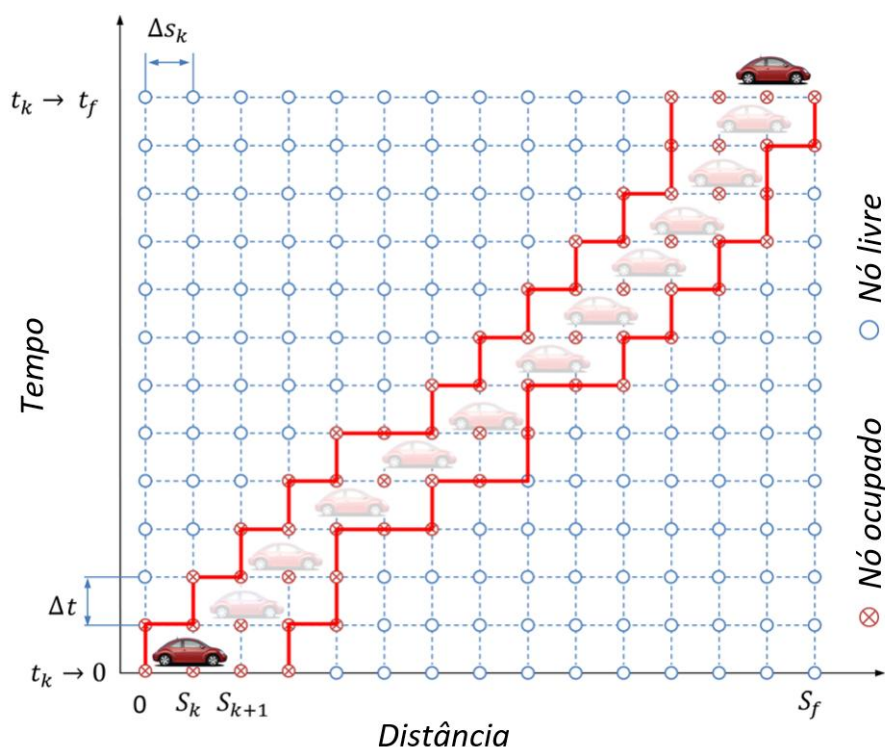
Como o foco é analisar o uso do espaço de interseção, apenas assume-se que os VAs seguirão o esquema de reserva estipulado pela UCI, não importando suas velocidades ou acelerações instantâneas, logo, equações cinemáticas e dinâmicas não são consideradas na modelagem. Desta forma, os melhores caminhos serão aqueles que possuírem os menores valores de T_f , considerando as distâncias $S_{i,f}$ somadas.

3.2 PROTOCOLOS DE COMUNICAÇÃO E SISTEMA DE PRIORIZAÇÃO

Dentro e ao redor da interseção, admite-se a existência de uma rede *wireless* de comunicação, na qual os veículos compartilham suas informações com a UCI, ao se aproximarem da sua zona de influência. Assim, o veículo envia uma requisição de entrada e indica por qual célula entrará e qual o seu destino final, e, após o aceite, ele receberá uma *id* única atribuída pela UCI.

Para um cruzamento com quatro fluxos de entrada, uma política de priorização do tipo FCFS é seguida, dando preferência àqueles que entraram primeiro na zona de influência e estão mais próximos do seu destino final. Caso haja uma rodovia principal em sentido único e outras duas secundárias de mão dupla que permitem conversão à direita ou à esquerda, há uma priorização para via primária e depois um esquema de FCFS para as demais. A fim de construir a ordenação FCFS, a UCI considera um cenário individual para cada VA, sem a presença dos demais, e aquele que tiver menos células até o destino desejado, estará em posições mais privilegiadas

Figura 4 – Ilustração do esquema de reserva de nós. O eixo horizontal é a distância discretizada, enquanto o eixo vertical é o tempo discretizado. A reserva do VA é mostrada passo a passo, e as reservas do seu sucessor e do seu predecessor na lista de prioridade são destacadas pelas duas linhas vermelhas.



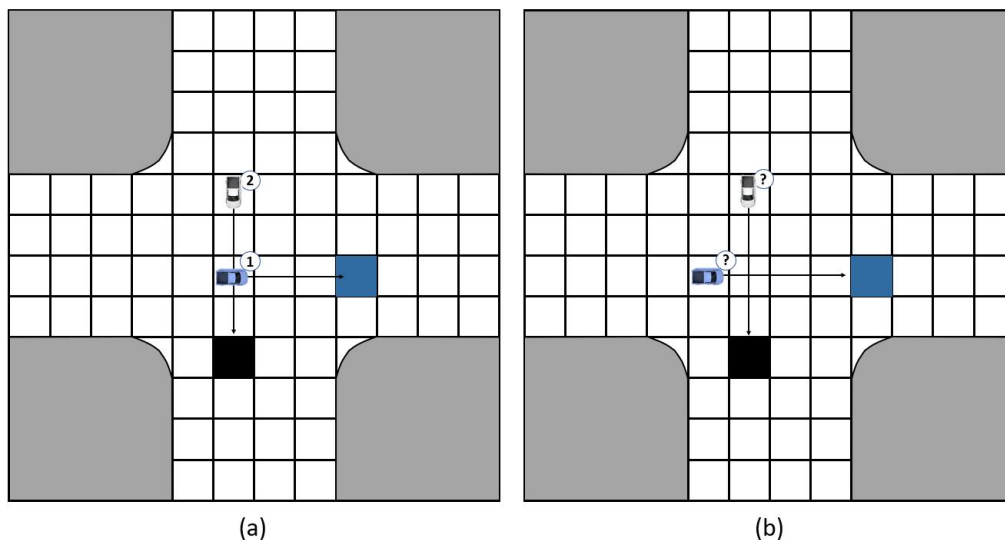
Fonte: Adaptado de Wuthishuwong *et al.* (2015).

na lista. Se nesta projeção dois ou mais VAs possuírem o mesmo número de células, a prioridade é decidida randomicamente. As diretrizes FCFS são exemplificadas na Figura 5.

Como visto na Figura 5, em a) o carro azul será atendido primeiro, pois precisa avançar duas células para sair da interseção, enquanto o carro preto precisa avançar três células. Já em b), ambos possuem o mesmo número de células até saírem da interseção, desta forma a prioridade é atribuída de forma aleatória. É importante ressaltar que a escolha das *id* é feita em lotes, ou seja, há uma limitação no número de veículos a cada verificação. Entretanto, os VAs podem pertencer a dois lotes diferentes, visto que é preciso levar em conta que há um fluxo contínuo na interseção. Assim, aqueles que estão no fim da fila de prioridades de um lote serão os primeiros na próxima sequência de verificação. Esta configuração é ilustrada na Figura 6.

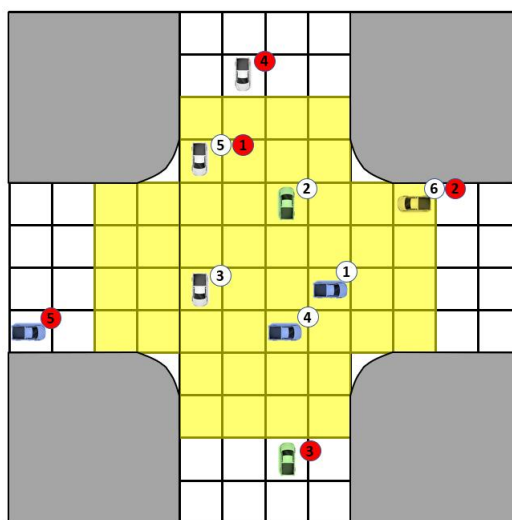
Na Figura 6, os VAs com *id* 5 e 6, do lote em verificação, serão os primeiros no grupo subsequente, assim que novos VAs entrarem. Para calcular o conjunto de manobras, a UCI repassa um *token* de priorização de um *id* para o outro, assim que as possibilidades de reservas são armazenadas iterativamente. Finalizada a exploração das trajetórias, uma mensagem é disparada para os VAs, contendo um sequência de

Figura 5 – Ilustração do FCFS. a) Escolha de *id* de acordo com o número de células até o destino. b) Escolha de *id* de forma randômica.



Fonte: Autoria Própria.

Figura 6 – Representação da coordenação em lotes. A área de influência da UCI é destacada em amarelo, e os *id* são indicados nos círculos brancos (lote atual) e vermelhos (próximo lote).



Fonte: Autoria Própria.

reservas seguras. Considera-se que a possibilidade de uma mensagem ser corrompida é relativamente baixa e o protocolo de comunicação é robusto à perda de mensagens.

3.3 MODEL CHECKING

Pela natureza não determinística das escolhas de movimento, onde a cada rodada de atualização de posições há múltiplas combinações de reservas, escolheu-se uma técnica de verificação formal para implementação do modelo, o *Model checking*

(traduzida do inglês por checagem de modelos), que quando aplicado a um sistema de estados finitos, explora de maneira exaustiva se o mesmo satisfaz ou não uma propriedade predeterminada. A formalidade matemática aplicada no *Model Checking* oferece uma integração ágil da etapa de verificação em projetos de sistemas que utilizam tecnologia da informação e comunicação, proporcionando eficácia e redução do tempo gasto na avaliação.

De acordo com Baier e Katoen (2008):

A atratividade desse método vem do fato de que é totalmente automático e a curva de aprendizado do usuário é muito suave. Além disso, oferece contra-exemplos no caso de um modelo não atender a uma especificação que serve como informação de depuração essencial.

No contexto deste trabalho, pode-se considerar que o *Model Checking* é um método de força bruta para identificar todos os possíveis estados de um sistema de gerenciamento de reservas. Para isso, usou-se as seguintes etapas, conforme sugerido por Baier e Katoen (2008):

- Especificação de propriedades - definição dos atributos qualitativos que compõem o sistema e, concomitantemente, como eles se comportarão. A descrição das propriedades precisa ter uma forma precisa e inequívoca, para que a verificação seja a mais rigorosa possível. Geralmente, representa-se as propriedades como conjuntos de estados e transições, usando autômatos de estados finitos.
- Execução - testagem dos elementos da etapa anterior, que irão compor uma abordagem algorítmica que fará a verificação do sistema.
- Análise - validação ou não do modelo testado. Em caso de resposta negativa, um contra-exemplo é gerado para auxiliar na correção dos erros. Uma terceira opção é o fenômeno da explosão do espaço de estados, que ocorre quando o modelo acaba sendo muito grande para caber nos limites físicos da memória do computador.

A exposição de erros do *Model Checking* não é probabilística, contrastando com testes e simulações que visam rastrear os defeitos mais prováveis. E isso torna o diagnóstico bastante atrativo pelo seu grau de acuracidade. Por isso, destaca-se a seguir alguns aspectos da etapa de análise e suas respectivas tratativas recomendadas:

- Propriedade Satisfeita - checar a próxima propriedade (se houver).
- Propriedade Violada - analisar o contra-exemplo, que mostra o passo a passo de transições que levou até aquele estado indesejado, e, posteriormente, refinar o modelo e repetir as etapas anteriores.

- Memória Insuficiente - realizar abstrações no modelo para tentar reduzi-lo e executa-lo novamente.

Os métodos formais são amplamente utilizados para verificação do comportamento e segurança de VAs. Entretanto, o estado da arte carece de referências específicas para coordenação destes em interseções, priorizando cenários onde eles se movimentam em uma via expressa de fluxo contínuo. Isso pode ser um reflexo da explosão do espaço de estados citada anteriormente, que requer um grau de complexidade para achar a redução mais adequada do modelo e definir novas propriedades, dentro do formalismo lógico utilizado. Este ponto possui um maior aprofundamento no Capítulo 5, quando se apresenta os resultados e as abstrações feitas para contornar este fenômeno.

3.3.1 Sistemas de transição

O modelo é feito a partir de um Sistema de Transição (ST), frequentemente usado para descrever comportamentos por meio de grafos direcionados, onde os nós representam estados e os arcos são as mudanças de estado. Desta forma, um estado é qualquer uma das possibilidades de reservas que a UCI pode alocar para cada veículo, considerando todas as combinações com os demais VAs envolvidos, e as transições são os passos individuais dados por eles.

Um *ST* pode ser estruturado como a tupla:

$$ST = \{E, A, \rightarrow, I, P, R\} \quad (6)$$

onde:

- E é um conjunto de estados e ,
- A é um conjunto de ações α , que descrevem as transições,
- $\rightarrow \subseteq E \times A \times E$ é uma relação de transição,
- $I \subseteq E$ é um conjunto de condições iniciais do sistema,
- P é um conjunto de proposições atômicas p , cuja função é formalizar características temporais e expressar intuitivamente fatos sobre os estados do sistema,
- $R : E \rightarrow 2^P$ é uma função de rótulo que relaciona um conjunto de proposições atômicas a qualquer estado $e \in E$

Quando se trata da modelagem de mecanismos de comunicação, a nomeação de ações α é essencial. Entretanto, este não é o caso do modelo proposto, onde o protocolo de comunicação é assumido como uma premissa externa às reservas.

Desta forma, a atualização dos estados, por ser um processo interno, pode ter sua identificação omitida ou generalizada, sem impacto no resultado final:

$$e_k \xrightarrow{\text{atualiza}} e_{k+1}. \quad (7)$$

Como as proposições atômicas de um sistema de transição dependem das propriedades do mesmo, é comum deixar as características de estado agirem como tais. Alguns exemplos de proposições que podem ser atribuídas aos estados do GIC são “as posições x dos VAs são $[2, 3, 7]$ ”, “a contagem de passos está em 3” ou “o VA que está na prioridade é o que possui a *id* 5”. Na representação de sistemas de transição, o conjunto P muitas vezes não é explicitamente indicado e assume-se que $P \subseteq E$ e $R(e) = \{e\} \cap P$ (BAIER; KATOEN, 2008).

3.3.2 Grafo de Alcançabilidade

Para um determinado ST , é possível definir um conjunto de ações α posteriores a um estado e como:

$$Post(e, \alpha) = \{e' \in E \mid e \xrightarrow{\alpha} e'\}, Post(e) = \bigcup_{\alpha \in A} Post(e, \alpha). \quad (8)$$

De forma semelhante, um conjunto de ações α predecessoras a um estado e pode ser dado por:

$$Pre(e, \alpha) = \{e' \in E \mid e' \xrightarrow{\alpha} e\}, Pre(e) = \bigcup_{\alpha \in A} Pre(e, \alpha). \quad (9)$$

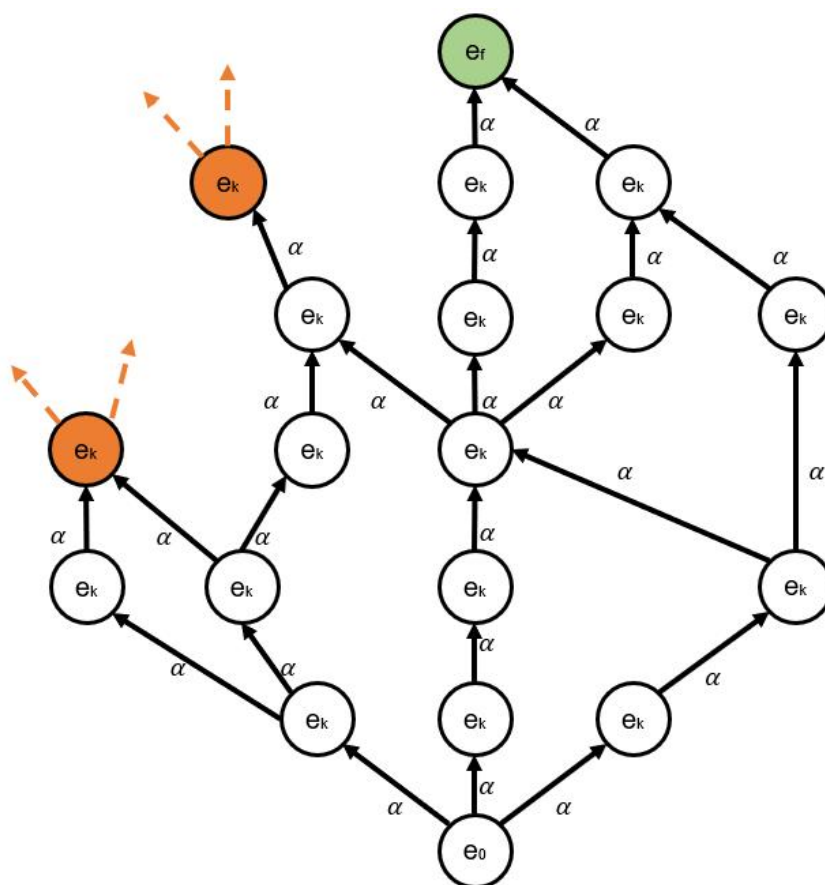
Um estado $e \in E$ é considerado alcançável se existe um fragmento de execução finita com um número n de elementos que o compõem:

$$e_0 \xrightarrow{\alpha_1} e_1 \xrightarrow{\alpha_2} \dots \xrightarrow{\alpha_n} e_n = e. \quad (10)$$

Desta forma, a infinidade de variações de estados e transições se tornam componentes de um grafo de alcançabilidade, constituído por diversos caminhos paralelos, os quais podem ser interrompidos ao se depararem com um estado que não possui nenhuma transição posterior, ou seja, $Post(e) = \emptyset$. Essa situação pode ocorrer quando o ST atinge o estado final desejado, ou por conta de alguma configuração de *deadlock* (traduzido livremente do inglês como “bloqueio permanente”).

O *deadlock* é uma das propriedades inerentes de um sistema com comportamento linear e com processos paralelos, e ocorre em um ST quando todos seus componentes não podem avançar em direção ao seu destino porque não há recursos disponíveis para serem solicitados, e aguardam o progresso uns dos outros para darem seus próximos passos. A representação de um grafo de alcançabilidade, com seus estados desejados e de *deadlock* é exemplificada na Figura 7.

Figura 7 – Representação de um gráfico de alcançabilidade, onde o estado final e_f desejado é destacado em verde, os estados de *deadlock* são ilustrados em laranja, e $k = 0, 1, 2, \dots, n$, dependendo da ramificação.



Fonte: Autoria Própria.

No GIC, um *deadlock* vem à tona quando a UCI não consegue alocar as reservas de células seguindo a lista de prioridades, gerando um estado que impossibilita a atualização dos estados seguros e, por consequência, ocasionando um congestionamento no cruzamento. Quando uma interseção não apresentar essa situação indesejada, ela é considerada *deadlock free*.

Um *ST* também está sujeito a propriedade de *livelock* (traduzido livremente do inglês como "bloqueio vivo"), que ocorre quando um processo nunca atinge um estado terminal, pois todos os canais necessários para isso estão ocupados, e então entra em um ciclo eterno ao redor de um nó desejado durante a sua execução. Obviamente, visa-se erradicar essa propriedade, mesmo que ela atenda critérios de segurança, para evitar que os VAs se movimentem em círculos na interseção e resultem em impactos negativos no desempenho global do sistema.

Por outro lado, a característica de *fairness* (traduzida do inglês por "justiça") é bastante requerida para um *ST*, uma vez que seu princípio é garantir que nenhum processo será preterido. Um exemplo de *fairness* em modelagem de interseções é

quando se assegura que todos os VAs terão acesso ao recursos sem esperar infinitamente pela sua vez de movimento, situação conhecida tecnicamente como *starvation free*. Desta forma, a estratégia de tomadas de decisões da UCI deve, eventualmente, responder a qualquer solicitação de entrada e também considerar transições de estado para todos os VAs, rodada a rodada.

3.3.3 Lógica Temporal Linear

A Lógica Temporal Linear (LTL) é um formalismo usado para tratar a validação de sistemas reativos através de uma notação muito intuitiva, mas matematicamente precisa, a fim de expressar as propriedades da relação entre os rótulos de estado. O termo temporal possui um sentido abstrato, referente a especificação da ordem relativa dos eventos, não remetendo a um comportamentos em tempo real, ou seja, nem a duração da transição, nem os tempos de permanência de um estado podem ser precisados.

A composição básica de uma sintaxe LTL possui ingredientes como proposições atômicas p , conectores booleanos (conjunção \wedge , negação \neg , *falso*, *verdadeiro*, etc.) e modalidades temporais. Em relação a esta última, a maioria delas incluem os seguintes operadores:

\square	“Sempre”	(agora e sempre no futuro)	
\diamond	“Eventualmente”	(eventualmente no futuro)	
\cup	“Até ”	(até atingir algo)	
\bigcirc	“Próximo passo”	(próximo passo é algo)	(11)

Assim, dada uma fórmula LTL φ , ela pode ser expressada de acordo com o seguinte exemplo:

$$\varphi ::= \textit{verdadeiro} \mid p \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \bigcirc\varphi. \quad (12)$$

Ao se combinar modalidades temporais, novas modalidades podem ser obtidas. Por exemplo, $\square \diamond p$ ("sempre eventualmente p ") descreve uma propriedade na qual, a qualquer momento k , existe um passo $k + 1 \geq k$, onde um caminho atinge uma proposição atômica p .

3.3.4 Modelagem FIACRE

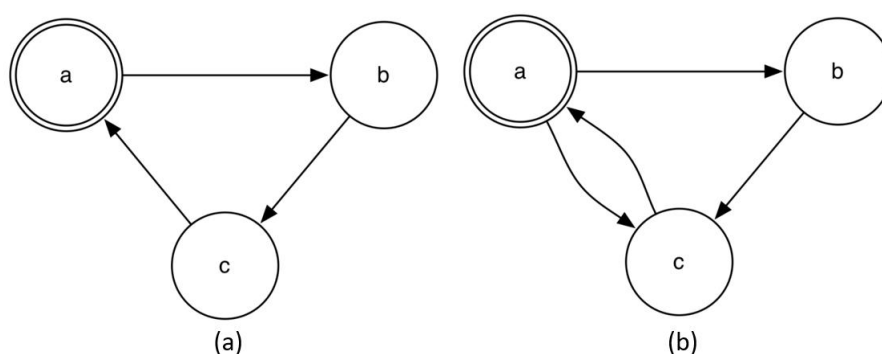
Com base nas premissas do *Model Checking*, diversos projetos reunindo indústrias e um grupo de equipes de cientistas do *Laboratoire d'analyse et d'architecture des systèmes - Centre national de la recherche scientifique (LAAS-CNRS)*, *Institut National de Recherche en Sciences et Technologies du Numérique (INRIA)*, e do *Institut de Recherche en Informatique de Toulouse (IRIT)*, desenvolveram uma linguagem chamada FIACRE, como uma ferramenta focada em verificação formal de propriedades LTL.

FIACRE é um acrônimo para *Format Intermédiaire pour les Architectures de Composants Répartis Embarqués*, uma sentença em francês que significa Formato Intermediário para Arquiteturas de Componentes Distribuídos Embarcados (BERTHO-MIEU *et al.*, 2008). Basicamente, esta ferramenta lida com processos, que descrevem o comportamento de componentes sequenciais, cujos estados de controle podem evoluir de forma determinística ou não determinística, dependendo das diretrizes de programação escolhidas.

A sintaxe do FIACRE suporta expressões *booleanas*, constantes e variáveis numéricas (inteiras e naturais), matrizes, estruturas dinâmicas e funções nativas. Também é possível criar novos tipos de variáveis, e, caso duas expressões possuam tipos iguais, então elas podem ser usadas nos mesmos contextos de declarações. Assim como em outras linguagens de programação, permite-se fazer comentários dentro das linhas de código, e isto é feito entre a combinação de caracteres `/**/`.

Um processo básico do FIACRE pode ser representado na forma de autômatos, um conjunto de células que interagem entre si em função de uma coleção finita de condições predefinidas. Os autômatos são usualmente representados na forma de um grafo dirigido, onde os estados são representados por círculos, sendo que estados finais são representados por círculos duplos, e as transições por arestas entre os dois estados conectados, conforme visto na Figura 8.

Figura 8 – Representação de um processo com estados *a*, *b* e *c* em forma de autômatos. a) Processo determinístico onde *a* transita para *b*, que por sua vez evolui *c*, o qual finaliza o ciclo ao atingir o estado *a* novamente. b) Processo não determinístico, onde *a* tem duas opções de transição, o que aumenta a gama de combinações de sequências.



Fonte: Autoria Própria.

A descrição de transições no FIACRE é relativamente simples e bastante intuitiva. Por exemplo, para compilar as relações de autômatos vistas na Figura 8, pode-se expressá-las conforme o Quadro 4 e o Quadro 5:

Quadro 4 – Processo determinístico no FIACRE.

```
/* Processo determinístico */
process ABC is /* Define o nome do processo */
  states a, b, c /* Define os estados */
  from a to b /* 'from' define as transições */
  from b to c
  from c to a
  /* indica qual processo é a entrada principal da verificação */
  ABC
```

Fonte: Autoria própria.

Quadro 5 – Processo não determinístico no FIACRE.

```
/* Processo não determinístico */
process ABC is
  states a, b, c
  from a select /* seleciona uma instrução a ser executada */
    to b
    [] to c /* '[' indica possibilidade de escolha */
  end
  from b to c
  from c to a
  ABC
```

Fonte: Autoria própria.

Ao fim do código, adiciona-se também as relações LTL que farão a validação do sistema. Um exemplo de fórmula LTL é visto no Quadro 6:

Quadro 6 – Exemplo do emprego de uma fórmula LTL no FIACRE.

```
/* Fórmula para verificar se o processo ABC sempre alcança c*/
/* Neste caso, os caracteres '[' indicam o operador LTL 'sempre'*/
property nome_da_propriedade is ltl ([] (ABC/state c))
assert nome_da_propriedade
```

Fonte: Autoria própria.

Em relação à execução das especificações do modelo, inicialmente utiliza-se o compilador FRAC, que traduz o formalismo do modelo FIACRE em um arquivo no formato *.tts* contendo uma descrição do sistema de transição temporal gerado, e um arquivo *.ltl*, com as propriedades LTL do modelo que precisam de verificação.

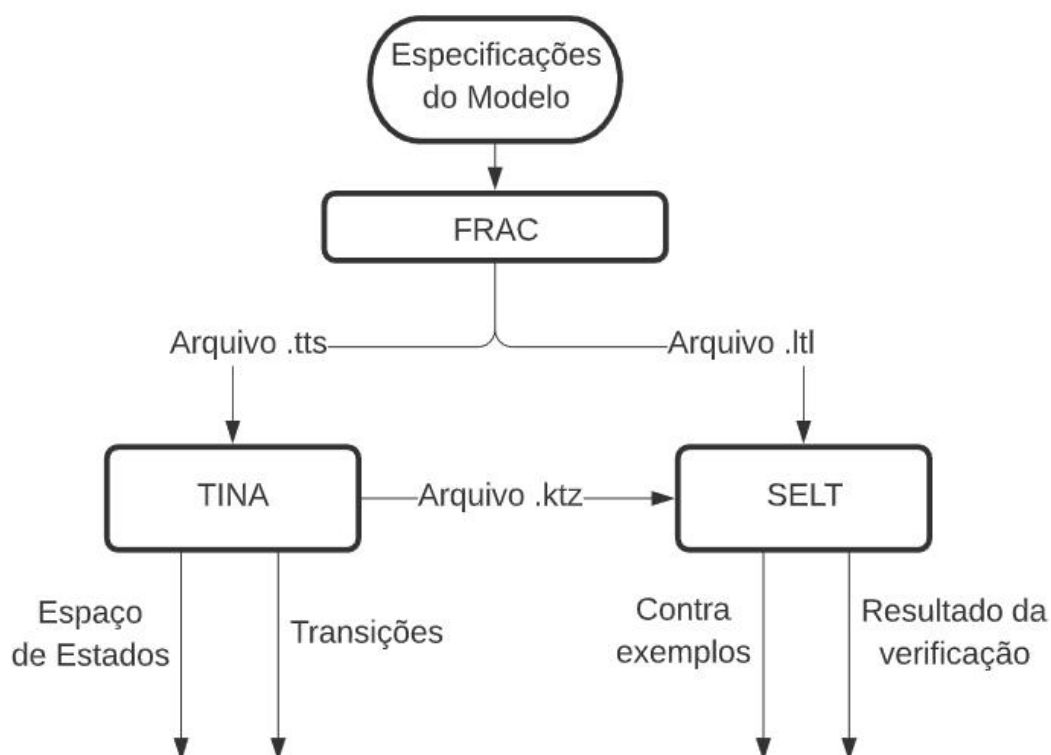
Em seguida, a ferramenta *Time Petri Net Analyzer* (TINA) (traduzido do inglês como Analisador de Redes de Petri Temporais) usa os arquivos *.tts* para criar o espaço de estados através de uma técnica eficiente de exploração dos limites de grandes redes de Petri, cujas representações podem ser feitas por meio de autômatos, cujas

transições representam as ações realizadas pelo sistema. O TINA constrói um grafo de alcançabilidade, cuja execução finda quando um estado é considerado sem limites, ou quando um limite opcional de marcações é atingido. Para aprofundamento no uso do TINA, recomenda-se o trabalho de Berthomieu *et al.* (2004).

A etapa final fica por conta do verificador de *State-Event LTL* (SELT) (abreviação cuja tradução livre do inglês é "espaço-evento LTL"), responsável por validar se as propriedades desejadas foram satisfeitas em pelo menos uma das ramificações. O SELT tem como variáveis de entrada um arquivo *.ltl* e um arquivo *.ktz* (um formato compactado do *ST* gerado pelo TINA). O resultado da avaliação é impresso em um *log* de acordo com o modo de saída e sinalizadores determinados, e, assim, a verificação formal do sistema é finalizada.

Todas as etapas descritas acima são mostradas na Figura 9.

Figura 9 – Esquema de *Model Checking* realizado pelo modelo FIACRE.



Fonte: Autoria Própria.

3.4 ANÁLISE DO GRAFO DE ALCANÇABILIDADE

Apesar do TINA provisionar um editor gráfico para visualização do grafo de alcançabilidade gerado, o mesmo se torna ineficaz para uma análise mais detalhada, focada na mudança de células dos VAs em meio a uma interseção, visto que o TINA constrói apenas uma representação em forma de autômatos. Desta forma, uma vez

que o estado final é atingido e o espaço de estados e transições são gerados, é possível se montar um grafo de alcançabilidade focado nas posições individuais dos VAs através de uma linguagem de alto nível que trate o arquivo de *log* gerado pelo TINA.

Além disso, para que a UCI possa ter opções mais refinadas e transmitir uma sequência de alocações de reservas para os VAs é necessário filtrar os melhores caminhos, eliminando aqueles que terminaram em *deadlock* ou *livelock*. Assim, seguiu-se a seguintes etapas:

1. Conversão do espaço de estados e transições em um arquivo de leitura
2. Higienização dos dados do arquivo de leitura para obter apenas as informações dos estados e transições
3. Formatação dos estados e transições em vetores
4. Construção do grafo de alcançabilidade, com a transformação dos estados em nós e das transições em arcos de peso 1, visto que não há transições prioritárias
5. Aplicação de um método para identificação dos caminhos com menos arcos
6. Implementação de técnicas de visualizações para acompanhar o comportamento dos VAs em diferentes cenários e condições iniciais

A conversão do espaço de estados e transições em um arquivo de leitura é feita nativamente dentro do TINA ao se configurar o formato de saída do resultado. A higienização dos dados é realizada por meio do uso de *Regular Expression* (REGEX) (traduzido do inglês por expressão regular) dentro do *Stream Editor* (SED) (traduzido do inglês por editor de fluxo), uma GNU/Linux para tratamento textual.

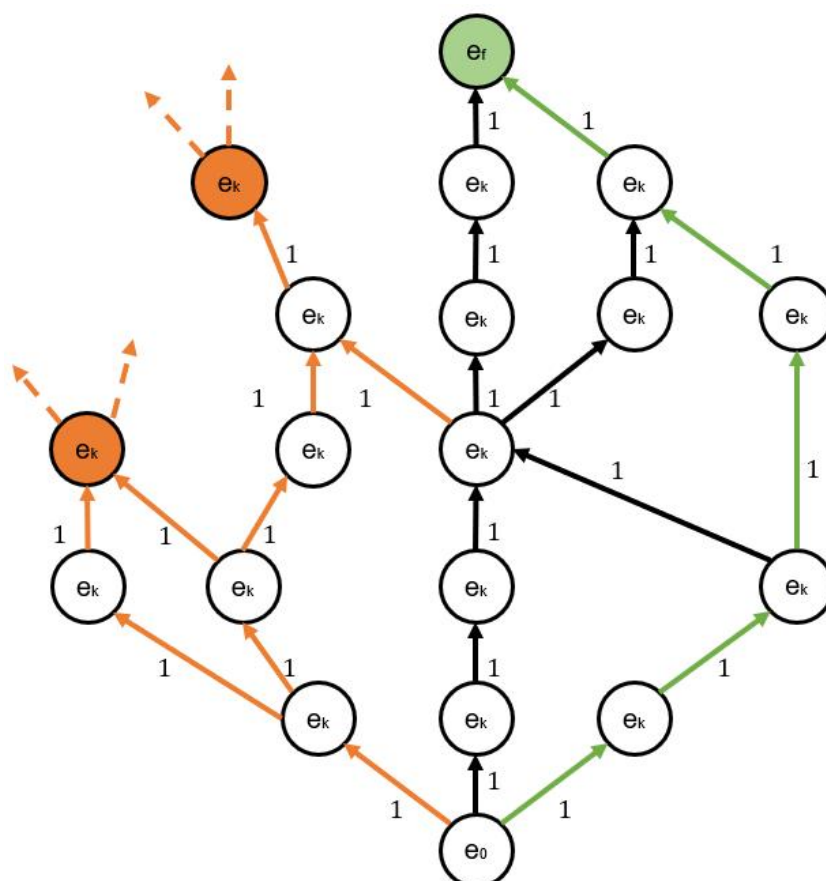
As demais etapas são efetuadas com o auxílio da linguagem Python, onde a construção e a análise do grafo de alcançabilidade foram feitas com a biblioteca NetworkX (NETWORKX, 2020), a qual permite a criação, manipulação e estudo de estruturas, dinâmicas e funções de redes complexas. Um grafo no NetworkX é uma coleção de nós (vértices) junto com pares identificados de nós (chamados arestas ou arcos). Os nós podem ser qualquer objeto utilizável como chave, por exemplo, uma cadeia de texto (*string*), uma imagem, um objeto XML, ou até mesmo outro gráfico.

Como o grafo de alcançabilidade possui todos os arcos com valor unitário, escolheu-se o método de Dijkstra, para computar os menores caminhos até o estado final, pois se trata de uma solução desenvolvida para encontrar o menor caminho entre dois nós de um grafo, onde todas as arestas devem ter pesos não negativos. Este algoritmo contabiliza o custo mínimo entre um nó inicial e os demais, e, a cada

passo seleciona no conjunto de nós aquele mais perto da origem, e assim sucessivamente, sempre atualizando o menor caminho ao passar por novos nós (DIJKSTRA *et al.*, 1959).

O método de Dijkstra está presente nas rotinas do NetworkX, nas funções `shortest_path(G, source, target, method='dijkstra')` e `all_shortest_path(G, source, target, method='dijkstra')`, onde G armazena os nós e os arcos do grafo, $source$ indica o nó com a condição inicial e $target$ é o nó que se deseja atingir. Enquanto a `shortest_path` retorna uma única lista de nós que pertencem a um dos caminhos mais curtos da origem ao destino, a `all_shortest_path` constrói uma estrutura Python do tipo *generator*, que não retorna um único valor, em vez disso, gera um objeto iterador com uma sequência de valores, correspondentes a todos os caminhos entre a origem e o destino. A análise de um grafo pelo método de Dijkstra é mostrada na Figura 10.

Figura 10 – Aplicação do método de Dijkstra, cujo objetivo é alcançar o estado e_f . O caminho com menor custo é representado em setas verdes, os caminhos opcionais em preto, e os caminhos que findam em *deadlocks* em laranja.



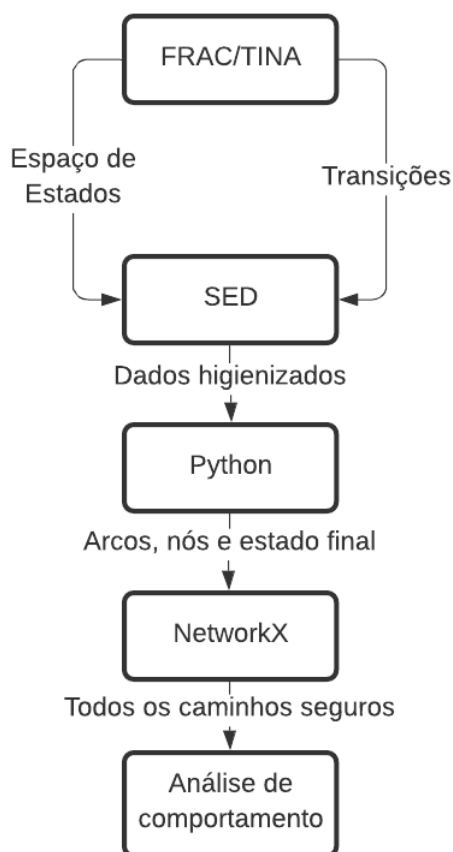
Fonte: Autoria Própria.

Após essas etapas, garante-se um rol de opções de travessias seguras e *deadlock-free*, pois as ramificações onde ocorrem bloqueios permanentes são eliminadas, e apenas aquelas que atingem o estado desejado são mantidas. Adicionalmente, pode-se explorar o conjunto de rotas que não possuem os menores T_f , e também segmentar o avanço por rodadas de atualização, a fim de se expandir as rotas alternativas computadas pela UCI.

A análise de comportamento foca em apresentar de forma visual a exploração da área de interseção pelas coordenadas x e y dos VAs, seja através de seus caminhos individuais ou pela distribuição dos passos dados de forma global, utilizando mapas de calor para identificar os caminhos mais utilizados.

O passo a passo do processo de análise é ilustrado na Figura 11.

Figura 11 – Processo de higienização, tratamento e análise dos dados.

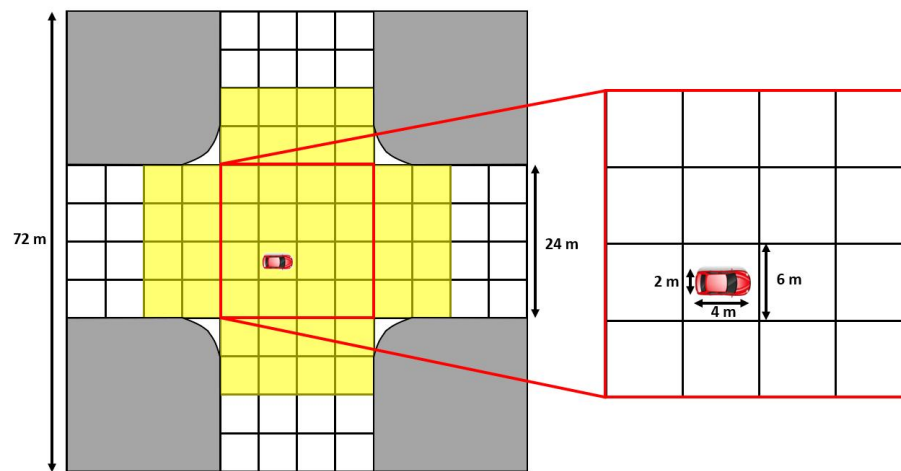


Fonte: Autoria Própria.

4 MODELAGEM DO SISTEMA

Todos os testes são executados em cruzamentos com formato de cruz, conforme Cenário mostrado na Figura 12, onde apenas as 48 células quadradas e discretas da zona de influência da UCI, representada em amarelo, são consideradas na modelagem. Assume-se que as células possuem áreas capazes de comportar as medidas de um carro popular.

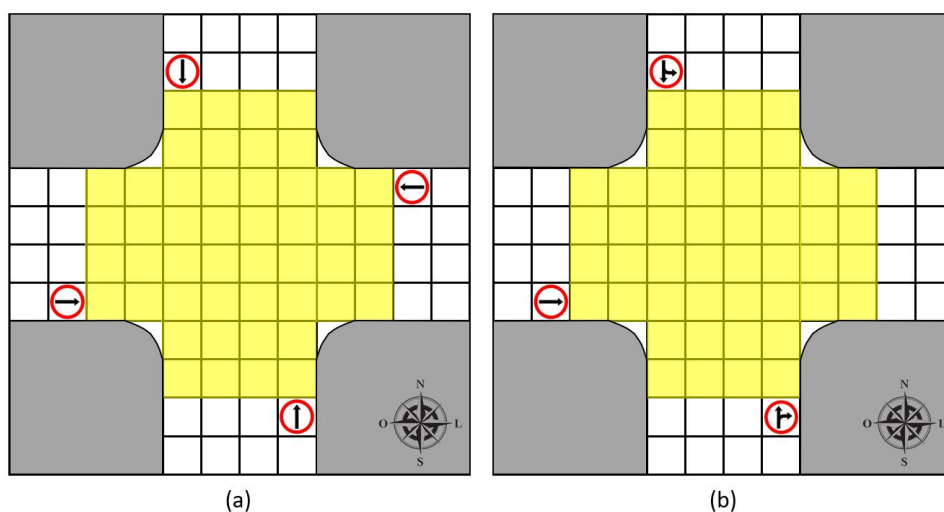
Figura 12 – Representação do Cenário de interseção, com os tamanhos da malha (*grid*) discretizada, das células e dos veículos.



Fonte: Autoria Própria.

A Figura 13 detalha os fluxos de movimentos adotados na verificação do modelo.

Figura 13 – Cenários verificados. a) Cenário 1 - Interseção de quatro fluxos. b) Cenário 2 - Interseção de três fluxos.



Fonte: Autoria Própria.

O primeiro cenário apresentado na Figura 13 é uma interseção de quatro fluxos de veículos vindos de norte, sul, leste e oeste. Já o Cenário 2 exibe a interseção de

uma avenida principal com mão única no sentido oeste-leste e uma avenida secundária com mão dupla nos sentidos norte-sul, que podem fazer conversão para a via principal.

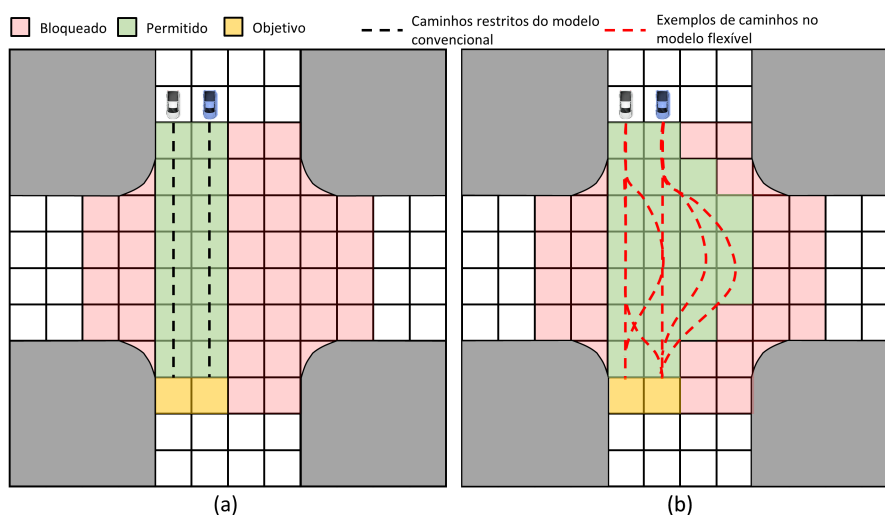
4.1 FORMATOS DE EXPLORAÇÃO DA ÁREA DE INTERSEÇÃO

Para testar o uso do espaço nos Cenários 1 e 2, optou-se por fazer uma comparação entre dois modelos distintos de reserva: Flexível (reservas possuem poucas restrições de mudanças de faixa, conforme hipótese de pesquisa) e Convencional (movimentos restritos a caminhos tradicionais, de acordo com as leis de trânsito, e sem possibilidade de mudança entre faixas).

Vale ressaltar que utilizou-se um modelo convencional de reservas apenas com o propósito de ter um referência em relação à evolução de alguns parâmetros como estados e transições do modelo flexível. Não faz parte do escopo deste trabalho comprovar que o uso mais livre da interseção é melhor que os padrões utilizados atualmente. O foco deste trabalho é analisar o comportamento do modelo flexível e se com o uso do mesmo é possível realizar uma travessia segura de interseção. Os esquemas de reserva de Dresner e Stone (2005, 2008) e Au *et al.* (2012) foram usados como referência para os modelos convencionais.

Os detalhes de cada modelos são evidenciados nas Figuras 14, 15, 16 e 17, onde as linhas pretas tracejadas indicam opções únicas de caminhos convencionais e as linhas vermelhas tracejadas mostram apenas alguns dos caminhos livres possíveis.

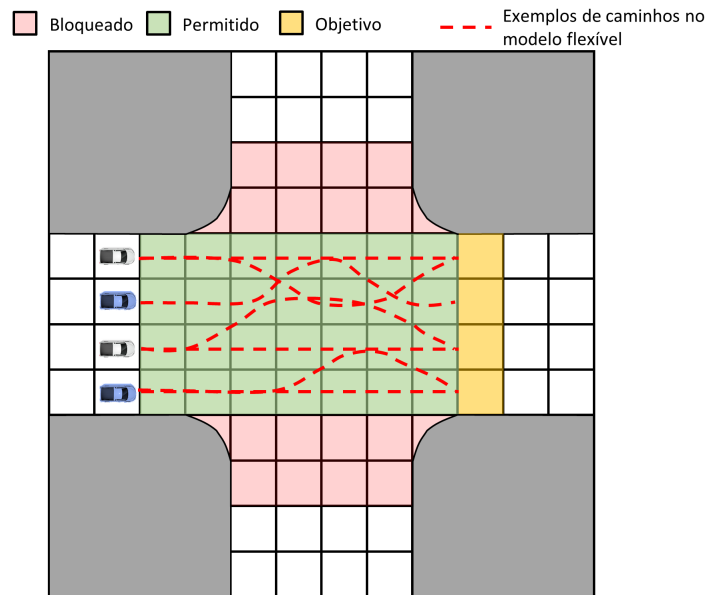
Figura 14 – Cenário 1. a) Esquema de reservas com o modelo convencional. b) Esquema de reservas com o modelo flexível.



Fonte: Autoria Própria.

Para o Cenário 1, os VAs adeptos do modelo convencional só podem andar reto e se manter na mesma faixa, conforme a linha preta tracejada na Figura 14. Já no modelo flexível, os VAs podem explorar outras faixas, dentro das células permitidas (em verde), mas precisam retornar a uma das faixas de seu fluxo original (células amareladas).

Figura 15 – Cenário 2 com esquema de reservas no modelo flexível para a avenida principal.

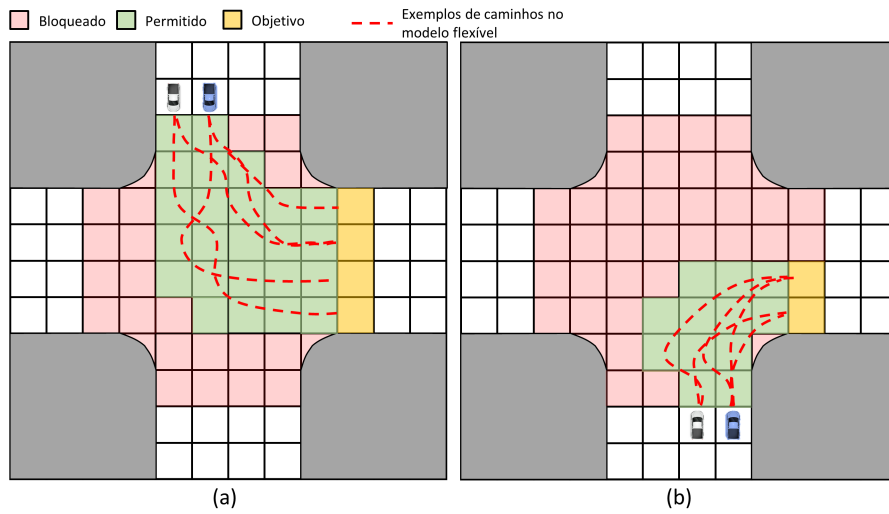


Fonte: Autoria Própria.

Em relação ao Cenário 2, o uso livre da interseção varia de acordo com o destino de cada VA. Para aqueles na avenida principal, que seguem em mão única, a Figura 15 demonstra que eles podem percorrer qualquer uma das quatro faixas para fazer a travessia e atingir seus objetivos, que podem ser qualquer uma das células amareladas.

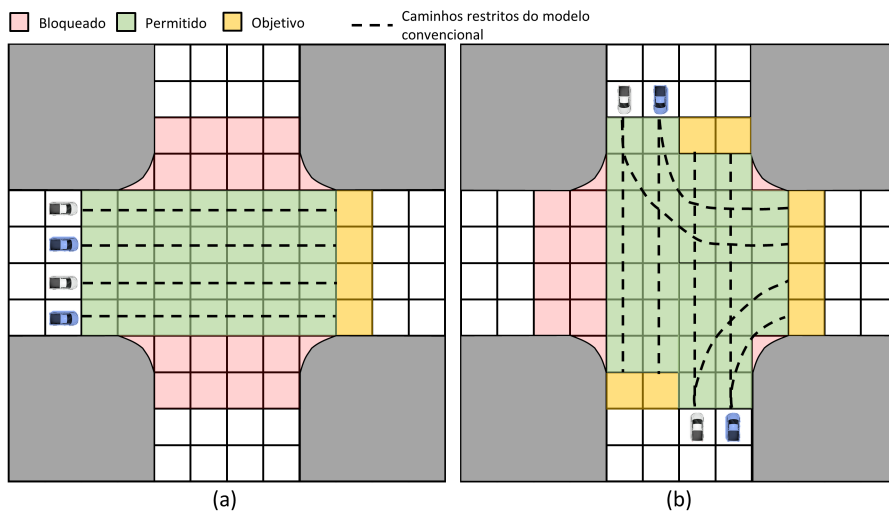
Os VAs que têm origem na avenida secundária do Cenário 2 exploram a área de interseção, em células que estejam alinhadas a seu destino. Por exemplo, um VA que tenha o destino na mesma via, ele terá uma esquema de reserva igual à parte b) da Figura 14, já aqueles que pretendem fazer a conversão para a via principal, terão um sistema de alocação igual o mostrado na Figura 16.

Figura 16 – Cenário 2. a) Esquema de reservas com o modelo flexível para a avenida secundária com conversão à esquerda. b) Esquema de reservas com o modelo flexível para a avenida secundária com conversão à direita.



Fonte: Autoria Própria.

Figura 17 – Cenário 2. a) Esquema de reservas com o modelo convencional para a avenida principal. b) Esquema de reservas com o modelo convencional para a avenida secundária.



Fonte: Autoria Própria.

Por último, temos o Cenário 2 no modelo convencional, os VAs precisam ocupar as células ao longo das linhas pretas tracejadas na Figura 17, de acordo o destino desejado.

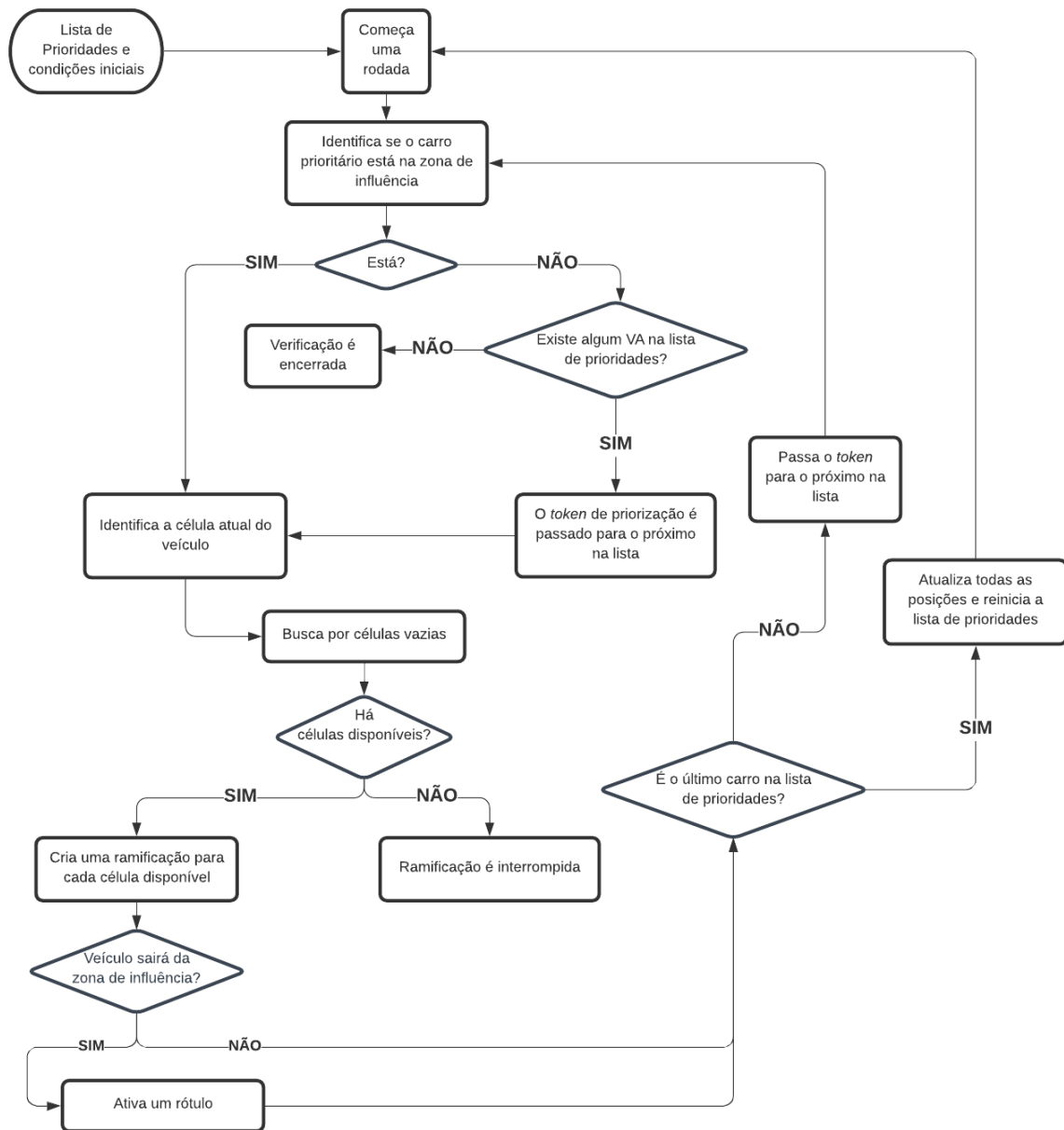
4.2 PREMISSAS PARA TOMADA DE DECISÃO

Para todos os modelos, a exploração de estados da UCI adota sempre algumas premissas a fim de garantir uma rota segura:

- A lista de prioridades é seguida a risca;
- Se não houver reservas disponíveis para o carro prioritário, a ramificação em verificação é interrompida e gera um estado de *deadlock*;
- Quando há um indicativo de que um VA sairá da zona de influência em uma rodada de verificação, a UCI ativa um rótulo na sua *id*, para que ela não seja mais considerada na lista de prioridades;
- Para garantir que haja a propriedade de *fairness*, uma rodada só finaliza quando o último veículo na lista de prioridades for verificado;
- Um lote de verificação só termina quando não há mais veículos na lista de prioridades.

A Figura 18 mostra o fluxograma de decisões adotado pela UCI para definir a alocação de recursos da interseção.

Figura 18 – Fluxograma de decisão utilizado pela UCI para exploração de estados.



4.3 ESPECIFICAÇÕES DOS SISTEMAS DE TRANSIÇÃO

Um estado e_k de uma ramificação do grafo de alcançabilidade é composto da seguinte forma:

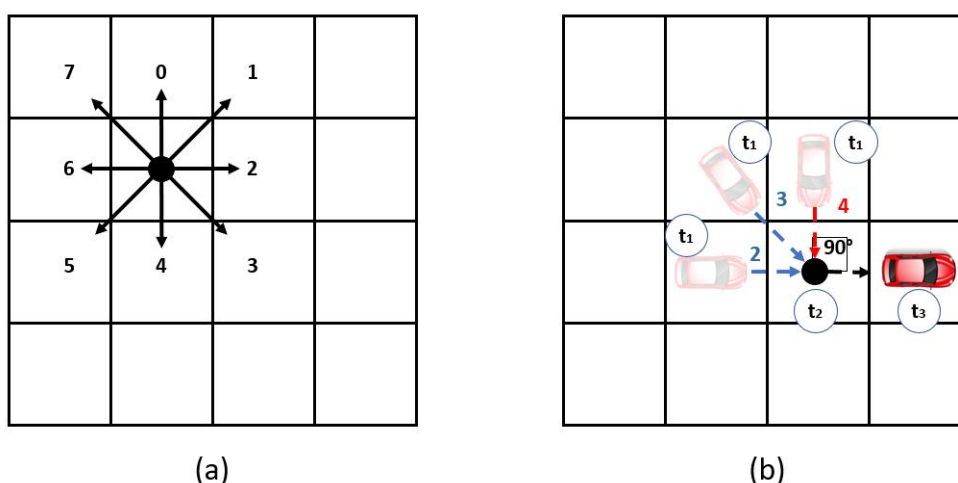
$$e_k = \{a, X, Y, ID, M\} \tag{13}$$

onde a é uma matriz que representa a configuração geral da interseção em cada passo k , X é o conjunto de coordenadas x de todos os veículos, Y é o conjunto de coordenadas y de todos os veículos, ID é uma variável que representa o *token* de priorização repassado de um id para outro, e M é o conjunto dos últimos movimentos

m dados por todos os veículos. Desta forma, é possível que dois estados e_k possuam a , X e Y iguais, mas um ID diferente, sendo assim dois estados distintos, mesmo que as posições sejam as mesmas no *grid*. É importante essa diferenciação, pois o ID é essencial para que a UCI faça as reservas seguindo a lista de prioridades.

A variável M se torna necessária apenas no Cenário 2, para evitar que veículos realizem movimentos bruscos ao efetuarem uma mudança de via. Com isso, adotou-se um referencial estático para rotular o sentido adotado na última reserva de cada veículo, independente do sentido original, conforme retratado na Figura 19.

Figura 19 – Verificação da variável M . a) Indexação a partir do último movimento realizado por um VA. b) Duas manobras seguras (em azul) e uma manobra brusca (em vermelho) entre os instantes t_1 e t_2 .



Fonte: Autoria própria.

Das três hipóteses mostradas na Figura 19, aquela onde o VA se move no sentido Norte-Sul entre um instante t_1 e t_2 (em vermelho), é a única que não poderá avançar no sentido oeste-leste entre t_2 e t_3 , pois seu índice m indicará que o movimento se configura em uma manobra brusca, onde o VA viraria bruscamente a 90° para alcançar a célula desejada em um instante t_3 , pois será descartada pelo algoritmo de verificação. M pode ser configurado como um vetor de tamanho n (quantidade de VAs), alocando a atualização das indexações m_i de acordo com a lista de prioridades:

$$M = [m_0, m_1, m_2, \dots, m_n]; m_i = 0, \dots, 7 \quad (14)$$

A condição inicial de M , assim como a lista de prioridades e as informações de origem e destino de cada VA dentro da sua zona de influência, é repassado à UCI pelo conjunto I .

Para representar o *grid* inicial de coordenação em FIACRE, usou-se uma variável do tipo *array*, que encapsula um determinado número de componentes do mesmo tipo em um único valor, para montar uma matriz 8×8 . Cada elemento da matriz a pode

ser indexado com uma coordenada x e y , iniciando sempre do índice 0, conforme a sintaxe mostrada no Quadro 7.

Quadro 7 – Sintaxe FIACRE que representa um exemplo de matriz a inicial.

```

a      : array 8 of array 8 of uint_8bit := [
                                                    /*y*/
[6,6,1,0,0,0,6,6], /*0*/
[6,6,1,1,0,0,6,6], /*1*/
[0,2,0,0,2,0,0,0], /*2*/

[2,0,2,1,0,0,0,0], /*3*/

[0,2,0,0,0,3,0,0], /*4*/
[2,0,0,2,0,0,0,0], /*5*/
[6,6,0,0,3,3,6,6], /*6*/
[6,6,0,0,3,0,6,6]] /*7*/
/*-----*/
/*x*/ /*0 1 2 3 4 5 6 7 */

```

Fonte: Autoria própria.

O valor 0 representa uma célula vazia e o número 6 indica uma célula ocupada por uma calçada. Como cada sentido de fluxo possui uma série de especificações de movimento única, as células ocupadas pelos VAs possuem numerações l que correspondem às suas vias de origem, conforme a seguir.

1. Sentido Norte-Sul
2. Sentido Oeste-Leste
3. Sentido Sul-Norte
4. Sentido Leste-Oeste

A Equação (15) mostra um vetor L de tamanho n (quantidade de VAs), que armazena as numerações l_j e que serve como a lista de prioridades, seguindo uma ordem da esquerda para a direita.

$$L = [l_0, l_1, l_2, \dots, l_n]; l_j = 1, 2, 3, 4. \quad (15)$$

Como visou-se emular um fluxo contínuo de VAs, uma matriz a inicial de verificação pode conter veículos no meio da interseção. Porém, cada teste possui um objetivo isolado, que foca em fazer a matriz a conter apenas os valores 0 e 6, indicando que todos os VAs atravessaram o cruzamento de forma segura.

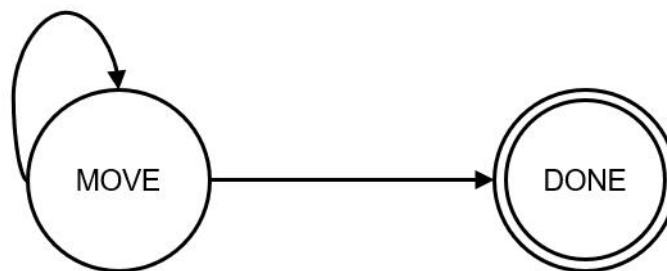
Para garantir que o VA atingirá o seu destino desejado, o vetor D contém a numeração da via de saída d_j de cada veículo, que é sempre checado pela UCI para averiguar se está condizente com a id preferencial.

$$D = [d_0, d_1, d_2, \dots, d_n]; d_i = 1, 2, 3, 4. \quad (16)$$

4.3.1 Configuração dos estados

Durante o processo exploratório de reservas, os estados podem assumir diversos rótulos $R(e)$, porém, instituiu-se apenas dois autômatos na modelagem FIACRE, um que indica a possibilidade de atualização dos movimentos (*move*) dos veículos no *grid*, e outro que indica a paralisação de movimentos (*done*), quando todos os veículos atingiram seus respectivos destinos. A execução do modelo perdura enquanto não se alcançar um rótulo composto pela proposição atômica "*grid vazio*", que qualifica a atualização de um estado *move* para um estado *done*. A representação simplificada das interações dos autômatos é exemplificada na Figura 20.

Figura 20 – Relação dos autômatos executados na modelagem FIACRE.



Fonte: Autoria própria.

O grafo de alcançabilidade sempre inicia com um estado inicial *move* e é composto por outras instâncias *move* em suas ramificações. Então, é possível identificar os diversos caminhos gerados a partir das combinações que levaram ao atingimento do estado final *done*.

Para iniciar a alocação de recursos da interseção, os componentes do estado e_k são inseridos como variáveis no *script* FIACRE, juntamente com as condições iniciais do conjunto I . Antes de efetuar uma transição, uma função auxiliar (f_{aux}) do processo valida se a *id* está de acordo com a lista de prioridades, e, caso esteja tudo certo, as informações do VA são selecionadas e a execução das escolhas não determinísticas começa.

Dentro da sintaxe do FIACRE é possível garantir que uma instrução de transição é realizada se, e somente se, um conjunto de proposições atômicas for atendido. Para isso, utiliza-se a declaração de bloqueio *on*, que assegura o seguimento das premissas do modelo, como as mostradas na Figura 18 anteriormente. Assim, cada $R(e)$ pode determinar uma atualização única para o estado e_k .

Um representação simplificada do processo é mostrada no Quadro 8.

Quadro 8 – Representação FIACRE do processo de verificação.

```

process intersection is
  states move, done
  var /* determina as variáveis do processo e condições iniciais */
      a = a0;
      X = X0;
      Y = Y0;
      ID = i0;
      M = M0
  from move
    /* funções f que selecionam as informações do VA preferencial*/
    ID' = f(X,Y,ID); /* verifica a id */
    pos = f(X,Y,ID'); /* seleciona a posição */
    x' = pos[0]; /* seleciona a coordenada x */
    y' = pos[1]; /* seleciona a coordenada y */
    i = f(ID') /* seleciona o sentido de origem */
    d = f(ID') /* seleciona o destino */
    select
      on R_objetivo to done /* possui p = "grid vazio" */
      [] on R1; atualiza ek;
        to move
      [] on R2; atualiza ek;
        to move
      .
      .
      .
      [] on Rn; atualiza ek;
        to move

```

Fonte: Autoria própria.

A quantidade de declarações de verificação depende da complexidade dos fluxos de movimentos permitidos da interseção. Desta forma, a modelagem do Cenário 2, devido às conversões à direita e à esquerda executadas pelos VAs da via secundária, demanda um montante maior de linhas de código para abarcar todas as premissas do modelo.

4.3.2 Atualização das variáveis

Para cada um dos cenários testados, há um conjunto de opções não determinísticas e restrições associadas aos locais de origem dos VAs. O Cenário 1, por exemplo, não permite que os VAs façam conversão à direita ou à esquerda, logo, a declaração *on* limita as células que podem ser consideradas, a fim de que haja uma convergência para o objetivo desejado. A partir dessas premissas, verifica-se as quatro possibilidades de reservas que os VAs podem adquirir na matriz *a*, de acordo com suas posições *x* e *y*, e também se d_j e i_j correspondem à *id* prioritária.

No Quadro 9, um exemplo evidencia a verificação da reserva de uma célula na diagonal direita para um VA preferencial.

Quadro 9 – Exemplo de uma escolha no processo de tomada de decisão.

```

[] on ((y < 5 and x > 2) or (y = 5 and (x = 3 or x =4)))
  and a[y][x] = ID'
  and a[y+1][x-1] = 0
  and i = 1
  and d = 3
  and M = M'; /* M' é um número entre 0 e 7 */

/* Caso positivo, atualiza*/
a[y+1][x-1]:=ID';
a[y][x]:=0;
y[ID']:=y+1;
x[ID']:=x-1;
M[ID']:= M'' /* M'' é um número entre 0 e 7 */
/* Se o ID for o último na lista*/
/* Ele reinicia as prioridades*/
if ID' < n-1 then ID := ID' + 1 else ID := 0 end;
to move

```

Fonte: Autoria própria.

O comando condicional *if* é requerido ao fim da atualização de variáveis para identificar se o VA em questão é o último na lista de prioridades. Caso positivo, a lista é reiniciada do zero e uma nova rodada se inaugura.

4.3.3 Propriedade LTL

O SELT é utilizado apenas para garantir que o estado *done* é atingido. Nenhuma outra propriedade é analisada. Desta forma, ao fim do *script*, é adicionado um ponto de entrada para identificar o processo sendo verificado e a propriedade LTL é composta pela modalidade temporal "*eventualmente*", conforme o Quadro 10.

Quadro 10 – Sintaxe utilizada FIACRE para verificar o atingimento do estado *done*.

```

/* Ponto de entrada */
intersection

/* Propriedade */

property target is ltl (<> (intersection/state done))
assert target

```

Fonte: Autoria própria.

4.4 HIGIENIZAÇÃO DOS DADOS

Os arquivos de leitura gerados na compilação do modelo contêm informações bem detalhadas sobre a construção do espaço de estados feita pelo TINA, que mos-

tra isso através de estruturas chamadas de marcações de estados, correspondentes aos diferentes rótulos que e_k assume em função de a, X, Y, ID, M . A Figura 21 apresenta um exemplo de como o TINA recebe as informações do FRAC para realizar a construção das marcações e transições.

Figura 21 – Exemplo de informações iniciais e análise de alcançabilidade feita pelo TINA.

```
initial state properties
intersection_1_vt(0) {intersection_1_vpx.at[0]}(2) {intersection_1_vpx.at[1]}(3) {intersection_1_vpx.at[2]}(4) {intersection_1_vpx.at[3]}(5) {intersection_1_vpx.at[4]}(3) {intersection_1_vpx.at[5]}(1) {intersection_1_vpx.at[6]}(5) {intersection_1_vpx.at[7]}(6)
{intersection_1_vpx.at[8]}(2) {intersection_1_vpx.at[9]}(0) {intersection_1_vpx.at[10]}(4) {intersection_1_vpx.at[11]}(7)
{intersection_1_vpx.at[12]}(2) {intersection_1_vpx.at[13]}(0) {intersection_1_vpy.at[0]}(3) {intersection_1_vpy.at[1]}(4)
{intersection_1_vpy.at[2]}(5) {intersection_1_vpy.at[3]}(3) {intersection_1_vpy.at[4]}(1) {intersection_1_vpy.at[5]}(4)
{intersection_1_vpy.at[6]}(6) {intersection_1_vpy.at[7]}(2) {intersection_1_vpy.at[8]}(1) {intersection_1_vpy.at[9]}(5)
{intersection_1_vpy.at[10]}(6) {intersection_1_vpy.at[11]}(3) {intersection_1_vpy.at[12]}(0) {intersection_1_vpy.at[13]}(4)
{intersection_1_va.at[0].at[0]}(6) {intersection_1_va.at[0].at[1]}(6) {intersection_1_va.at[0].at[2]}(1) {intersection_1_va.at[0].at[3]}(0)
{intersection_1_va.at[0].at[4]}(0) {intersection_1_va.at[0].at[5]}(0) {intersection_1_va.at[0].at[6]}(6) {intersection_1_va.at[0].at[7]}(6)
{intersection_1_va.at[1].at[0]}(6) {intersection_1_va.at[1].at[1]}(6) {intersection_1_va.at[1].at[2]}(1) {intersection_1_va.at[1].at[3]}(1)
{intersection_1_va.at[1].at[4]}(0) {intersection_1_va.at[1].at[5]}(0) {intersection_1_va.at[1].at[6]}(6) {intersection_1_va.at[1].at[7]}(6)
{intersection_1_va.at[2].at[0]}(0) {intersection_1_va.at[2].at[1]}(0) {intersection_1_va.at[2].at[2]}(0) {intersection_1_va.at[2].at[3]}(0)
{intersection_1_va.at[2].at[4]}(0) {intersection_1_va.at[2].at[5]}(0) {intersection_1_va.at[2].at[6]}(4) {intersection_1_va.at[2].at[7]}(0)
{intersection_1_va.at[3].at[0]}(0) {intersection_1_va.at[3].at[1]}(0) {intersection_1_va.at[3].at[2]}(1) {intersection_1_va.at[3].at[3]}(0)
{intersection_1_va.at[3].at[4]}(0) {intersection_1_va.at[3].at[5]}(4) {intersection_1_va.at[3].at[6]}(0) {intersection_1_va.at[3].at[7]}(4)
{intersection_1_va.at[4].at[0]}(2) {intersection_1_va.at[4].at[1]}(2) {intersection_1_va.at[4].at[2]}(0) {intersection_1_va.at[4].at[3]}(2)
{intersection_1_va.at[4].at[4]}(0) {intersection_1_va.at[4].at[5]}(0) {intersection_1_va.at[4].at[6]}(0) {intersection_1_va.at[4].at[7]}(0)
{intersection_1_va.at[5].at[0]}(2) {intersection_1_va.at[5].at[1]}(0) {intersection_1_va.at[5].at[2]}(0) {intersection_1_va.at[5].at[3]}(0)
{intersection_1_va.at[5].at[4]}(3) {intersection_1_va.at[5].at[5]}(0) {intersection_1_va.at[5].at[6]}(0) {intersection_1_va.at[5].at[7]}(0)
{intersection_1_va.at[6].at[0]}(6) {intersection_1_va.at[6].at[1]}(6) {intersection_1_va.at[6].at[2]}(0) {intersection_1_va.at[6].at[3]}(0)
{intersection_1_va.at[6].at[4]}(3) {intersection_1_va.at[6].at[5]}(3) {intersection_1_va.at[6].at[6]}(6) {intersection_1_va.at[6].at[7]}(6)
{intersection_1_va.at[7].at[0]}(6) {intersection_1_va.at[7].at[1]}(6) {intersection_1_va.at[7].at[2]}(0) {intersection_1_va.at[7].at[3]}(0)
{intersection_1_va.at[7].at[4]}(0) {intersection_1_va.at[7].at[5]}(0) {intersection_1_va.at[7].at[6]}(6) {intersection_1_va.at[7].at[7]}(6)

REACHABILITY ANALYSIS -----
13091 marking(s), 14145 transition(s)
```

Fonte: Autoria própria.

A Figura 22 mostra a composição de e_k como uma marcação de estado para o Cenário 1 (à esquerda) e para o Cenário 2 (à direita).

Figura 22 – Marcações geradas pelo TINA

MARKINGS:	MARKINGS:
<pre>0 : intersection_1_smove { intersection_1_vid=0, intersection_1_vpx=[2,3,4,5,3,1,5,6,2,0,4,7,2,0], intersection_1_vpy=[3,4,5,3,1,4,6,2,1,5,6,3,0,4], intersection_1_va=[[6,6,1,0,0,0,6,6], [6,6,1,0,0,0,6,6], [0,0,0,0,0,0,4,0], [0,0,1,0,0,4,0,4], [2,2,0,2,0,0,0,0], [2,0,0,0,3,0,0,0], [6,6,0,0,3,3,6,6], [6,6,0,0,0,0,6,6]]</pre>	<pre>0 : intersection_1_smove { intersection_1_vid=0, intersection_1_vpx=[2,3,4,5,3,1,5,6,2,0,4,7,2,0], intersection_1_vpy=[3,4,5,3,1,4,6,2,1,5,6,3,0,4], intersection_1_vm= [0,0,0,0,0,0,0,0,0,0,0,0,0,0], intersection_1_va=[[6,6,1,0,0,0,6,6], [6,6,1,0,0,0,6,6], [0,0,0,0,0,0,4,0], [0,0,0,0,0,0,4,0], [0,0,1,0,0,4,0,4], [2,2,0,2,0,0,0,0], [2,0,0,0,3,0,0,0], [6,6,0,0,3,3,6,6], [6,6,0,0,0,0,6,6]]</pre>

Fonte: Autoria própria.

Como o objetivo deste trabalho é analisar a exploração da área da interseção, os conjuntos X e Y são os únicos que importam para dar um contexto mais visual aos resultados, visto que ID e M já exerceram sua função na tomada decisão e a é apenas uma representação do *grid* a partir das atualizações de X e Y .

Assim sendo, com auxílio de identificação de padrões via REGEX, elimina-se os detalhes textuais desnecessários, deixando cada marcação de estado apenas com os conjuntos X e Y em forma de vetores, como mostrado na Figura 23.

Figura 23 – Representação dos estados e_k em forma de vetores, após o processo de higienização.

1	[2,3,4,5,3,1,5,6,2,0,4,7,2,0]	[3,4,5,3,1,4,6,2,1,5,6,3,0,4]
2	[2,3,4,5,3,1,5,6,2,0,4,7,2,0]	[4,4,5,3,1,4,6,2,1,5,6,3,0,4]
3	[2,3,4,5,3,1,5,6,2,0,4,7,2,0]	[5,4,5,3,1,4,6,2,1,5,6,3,0,4]
4	[2,4,4,5,3,1,5,6,2,0,4,7,2,0]	[4,4,5,3,1,4,6,2,1,5,6,3,0,4]
5	[2,4,4,5,3,1,5,6,2,0,4,7,2,0]	[5,4,5,3,1,4,6,2,1,5,6,3,0,4]
6	[2,4,4,5,3,1,5,6,2,0,4,7,2,0]	[4,4,5,3,1,4,6,2,1,5,6,3,0,4]
7	[2,4,4,5,3,1,5,6,2,0,4,7,2,0]	[5,4,5,3,1,4,6,2,1,5,6,3,0,4]
8	[2,4,4,4,3,1,5,6,2,0,4,7,2,0]	[4,4,5,3,1,4,6,2,1,5,6,3,0,4]
9	[2,4,4,4,3,1,5,6,2,0,4,7,2,0]	[5,4,5,3,1,4,6,2,1,5,6,3,0,4]
10	[2,4,4,4,3,1,5,6,2,0,4,7,2,0]	[4,4,5,3,2,4,6,2,1,5,6,3,0,4]
11	[2,4,4,4,3,1,5,6,2,0,4,7,2,0]	[4,4,5,3,3,4,6,2,1,5,6,3,0,4]
12	[2,4,4,4,3,1,5,6,2,0,4,7,2,0]	[5,4,5,3,2,4,6,2,1,5,6,3,0,4]
13	[2,4,4,4,3,1,5,6,2,0,4,7,2,0]	[5,4,5,3,3,4,6,2,1,5,6,3,0,4]
14	[2,4,4,4,3,1,5,6,2,0,4,7,2,0]	[4,4,5,3,2,4,6,2,1,5,6,3,0,4]
15	[2,4,4,4,3,1,5,6,2,0,4,7,2,0]	[4,4,5,3,3,4,6,2,1,5,6,3,0,4]
16	[2,4,4,4,3,2,5,6,2,0,4,7,2,0]	[5,4,5,3,2,4,6,2,1,5,6,3,0,4]
17	[2,4,4,4,3,2,5,6,2,0,4,7,2,0]	[5,4,5,3,3,4,6,2,1,5,6,3,0,4]
18	[2,4,4,4,3,1,5,6,2,0,4,7,2,0]	[4,4,5,3,2,4,5,2,1,5,6,3,0,4]
19	[2,4,4,4,3,1,5,6,2,0,4,7,2,0]	[4,4,5,3,2,4,4,2,1,5,6,3,0,4]
20	[2,4,4,4,3,1,5,6,2,0,4,7,2,0]	[4,4,5,3,3,4,5,2,1,5,6,3,0,4]
21	[2,4,4,4,3,1,5,6,2,0,4,7,2,0]	[4,4,5,3,3,4,4,2,1,5,6,3,0,4]
22	[2,4,4,4,3,2,5,6,2,0,4,7,2,0]	[5,4,5,3,2,4,5,2,1,5,6,3,0,4]
23	[2,4,4,4,3,2,5,6,2,0,4,7,2,0]	[5,4,5,3,2,4,4,2,1,5,6,3,0,4]
24	[2,4,4,4,3,2,5,6,2,0,4,7,2,0]	[5,4,5,3,3,4,5,2,1,5,6,3,0,4]
25	[2,4,4,4,3,2,5,6,2,0,4,7,2,0]	[5,4,5,3,3,4,4,2,1,5,6,3,0,4]
26	[2,4,4,4,3,1,5,5,2,0,4,7,2,0]	[4,4,5,3,2,4,5,2,1,5,6,3,0,4]
27	[2,4,4,4,3,1,5,5,2,0,4,7,2,0]	[4,4,5,3,2,4,4,2,1,5,6,3,0,4]
28	[2,4,4,4,3,1,5,5,2,0,4,7,2,0]	[4,4,5,3,3,4,5,2,1,5,6,3,0,4]

Fonte: Autoria própria.

As transições de estados também passam por tratamento semelhante. O arquivo original relaciona um estado a outro e evidencia a transição que foi disparada, conforme mostrado na Figura 24. Desta forma, utilizou-se padrões REGEX para retirar os nomes das interseções para facilitar a adição das transições como arcos de entrada do NetworkX. Um exemplo de documento tratado é destacado na Figura 25.

Figura 24 – Exemplo de uma lista de transições que relacionam todos os estados alcançáveis.

```
des(0,14145,13091)
(0,"intersection_1_t6",1)
(0,"intersection_1_t7",2)
(1,"intersection_1_t9",3)
(2,"intersection_1_t9",4)
(3,"intersection_1_t10",5)
(4,"intersection_1_t10",6)
(5,"intersection_1_t14",7)
(6,"intersection_1_t14",8)
(7,"intersection_1_t6",9)
(7,"intersection_1_t7",10)
(8,"intersection_1_t6",11)
(8,"intersection_1_t7",12)
(9,"intersection_1_t8",13)
(10,"intersection_1_t8",14)
(11,"intersection_1_t9",15)
(12,"intersection_1_t9",16)
(13,"intersection_1_t11",17)
(13,"intersection_1_t12",18)
(14,"intersection_1_t11",19)
(14,"intersection_1_t12",20)
(15,"intersection_1_t11",21)
(15,"intersection_1_t12",22)
(16,"intersection_1_t11",23)
(16,"intersection_1_t12",24)
```

Fonte: Autoria própria.

Figura 25 – Exemplo de uma lista de transições higienizada, pronta para ser consumida pelo NetworkX.

```
1 (0,6,1)
2 (0,7,2)
3 (1,9,3)
4 (2,9,4)
5 (3,10,5)
6 (4,10,6)
7 (5,14,7)
8 (6,14,8)
9 (7,6,9)
10 (7,7,10)
11 (8,6,11)
12 (8,7,12)
13 (9,8,13)
14 (10,8,14)
15 (11,9,15)
16 (12,9,16)
17 (13,11,17)
18 (13,12,18)
19 (14,11,19)
20 (14,12,20)
21 (15,11,21)
22 (15,12,22)
23 (16,11,23)
24 (16,12,24)
```

Fonte: Autoria própria.

4.5 CONSTRUÇÃO DO GRAFO DE ALCANÇABILIDADE

A montagem do grafo é relativamente simples no NetworkX, uma vez que precisa apenas de um conjunto de nós e outro de arcos identificando a relação entre os nós. Um *script* básico para a criação do grafo é mostrado no Quadro 11.

Quadro 11 – Script básico para criação de um grafo no NetworkX.

```
import networkx as nx

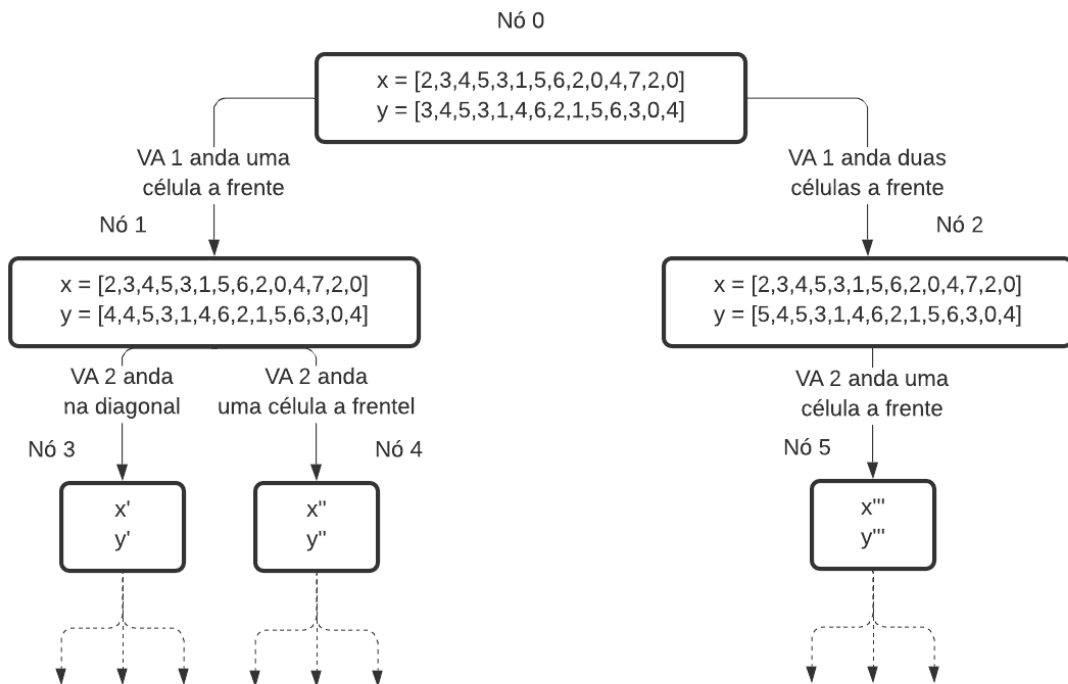
nos = ['0','1','2',...] # Índices dos nós
arcos = [['0','1'],['0','2'],['1','3'],...] # transições entre estados

G = nx.Graph()
G.add_nodes_from(nos)
G.add_edges_from(arcos)
```

Fonte: Autoria própria.

Desta forma, as posições *X* e *Y* das marcações de estados são associadas aos índices dos nós, e a cada nova atualização de estado, novas ramificações surgem, conforme exemplificado na Figura 26.

Figura 26 – Exemplo de como funciona as ramificações do grafo de alcançabilidade para o esquema de reservas da UCI.



Fonte: Autoria Própria.

Com os menores caminhos estabelecidos por meio da função *shortest_path*, é possível percorrer os seus nós e identificar aqueles que correspondem a um estado onde houve uma rodada completa, ou seja, o último veículo na lista de prioridades foi verificado. Assim, as informações de menor número de passos, menor número de rodadas e a lista de nós de cada caminho é armazenada em forma de dicionários Python.

Figura 27 – Exemplo de um dicionário Python contendo as informações de um caminho computado pelo NetworkX.

Chave	Tipo	Tamanho	Valor
path	list	10	['0', '137', '756', '1643', '3572', '6667', '9302', '10781', '11291', ...]
rounds	int	1	9
steps	int	1	82

Fonte: Autoria própria.

5 RESULTADOS E DISCUSSÕES

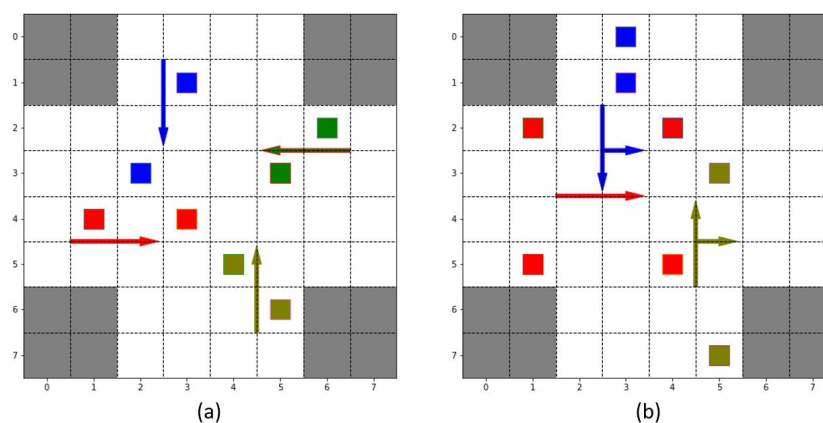
Neste capítulo são apresentados os resultados obtidos para os dois cenários e as duas formas de uso da interseção definidas no capítulo 4. Para executar o modelo FIACRE optou-se pelo uso do compilador FRAC 2.5.0 e o TINA 3.5.0 em uma distribuição Ubuntu 2004.2020.812, um sistema operacional de código aberto, construído a partir de um núcleo GNU/Linux. A construção e análise do grafo de alcançabilidade foi feita no Spyder 3.7, um ambiente de desenvolvimento integrado de código aberto em Python, em um sistema operacional Windows 10 Pro. Todos os testes foram feitos em um computador com processador intel i7-9700K 3,6 GHz, e memória RAM de 32 GB.

A seção 5.1 apresenta os testes iniciais do processo de tomada de decisão da UCI, seguindo um modelo flexível nos dois cenários estabelecidos no capítulo 4. A seção 5.2 detalha as abstrações utilizadas para reduzir o modelo, a fim de contornar os problemas com o processamento de dados. A seção 5.3 foca na evolução dos modelos finais para variadas condições iniciais I , e explora visualmente os comportamentos de uso da área de interseção. A seção 5.4 foca em analisar de forma detalhada caminhos alternativos além do menor T_f para uma determinada condição inicial I .

5.1 CONFIGURAÇÕES DE TESTE

Para verificar os limites de desempenho do modelo flexível do GIC em relação ao uso da memória disponível, testou-se disposições iniciais com um número de 8 VAs, onde, no Cenário 2, o destino de VAs com origem na vias secundárias foi variado, com alguns seguindo em linha reta e outros fazendo conversão para a via principal. As posições iniciais dos 8 VAs são configuradas conforme visto na Figura 28.

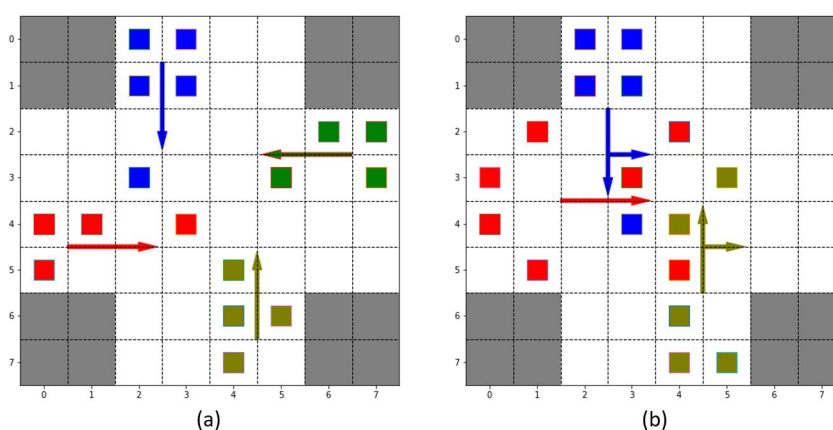
Figura 28 – Configurações de testes, onde os VAs são representados por quadrados e cada cor representa um fluxo de origem diferente. a) Posições iniciais de teste no Cenário 1. b) Posições iniciais de teste no Cenário 2.



Fonte: Autoria própria.

A partir da circunstância dada na Figura 28, populou-se o *grid* com novos VAs em posições *x* e *y* no início de cada fluxo de entrada, como forma de averiguar o desempenho em um dos piores cenários possível, onde os últimos VAs prioritários precisassem andar um número maior de células, resultando em um número maior de rótulos de estados. Assim, testou-se o processo de reservas até que a interseção contasse com 17 VAs, pois a partir deste número, os testes com o Cenário 1 resultaram na explosão de estados. As posições iniciais usadas para os cenários com 17 VAs são demonstradas na Figura 29.

Figura 29 – Configurações de testes, onde os VAs são representados por quadrados e cada cor representa um fluxo de origem diferente. a) Posições iniciais de teste para 17 VAs no Cenário 1. b) Posições iniciais de teste para 17 VAs no Cenário 2.



Fonte: Autoria própria.

No Cenário 1, o esforço computacional começou a se elevar com a verificação de reservas para 14 VAs, passando da ordem de 20 milhões de estados e transições. Já para o Cenário 2, a curva de crescimento foi mais suave, passando dos 20 milhões apenas com 17 VAs. A evolução dos estados e transições é resumida na Tabela 1.

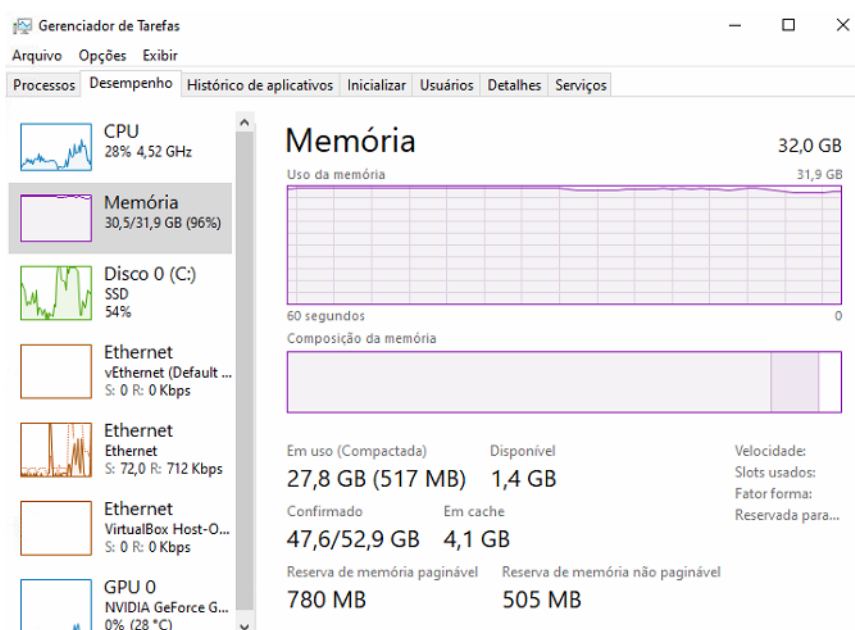
Tabela 1 – Evolução de estados e transições para alcance do estado final nos primeiros testes do modelo flexível.

N	Cenário 1 - Estados	Cenário 2 - Estados	Cenário 1 - Transições	Cenário 2 - Transições
8	365.141	198.195	501.489	323.354
10	1.811.002	1.334.905	2.485.612	2.058.324
12	6.451.143	1.904.930	7.814.037	2.799.559
14	21.932.454	3.217.733	26.530.479	4.967.065
15	59.770.975	3.657.215	72.834.245	5.345.375
16	83.384.751	12.998.598	100.153.874	19.264.287
17	-	23.308.769	-	33.458.778

Fonte: Autoria própria.

Conforme visto no Tabela 1, o Cenário 1 atinge uma escala muito maior de estados e transições, e isso se justifica pelo fato de existir um fluxo de entrada a mais que o Cenário 2. Além disso, no Cenário 2, alguns VAs no sentido sul-norte saem mais rápido da interseção, pois são necessárias poucas células para se fazer a conversão a direita na via principal, e, conseqüentemente, com menos carros participando do processo de tomada de decisão, há menos combinações de movimento e menos estados gerados. A explosão de estados no Cenário 1 ocorreu quando a memória disponível já não era suficiente para se computar as verificações do modelo, conforme exemplificado na Figura 30.

Figura 30 – Imagem do gerenciador de tarefas mostrando o desempenho do uso da memória RAM durante a execução modelo FIACRE com uma interseção com 17 VAs no Cenário 1 com movimentos flexíveis.



Fonte: Autoria própria.

Em relação ao modelo convencional, testou-se mesmas conjunturas do modelo flexível nos Cenários 1 e 2, entretanto, notou-se que partir de 12 VAs, o estado final nunca era atingido. Isso se deveu pela redução de células permitidas neste modelo, resultando em várias situações com *deadlocks* e sem uma única opção de acesso a um recurso da interseção, seguindo a lista de prioridades da UCI. Desta forma, para o sistema de reservas convencional, permitiu-se que os VAs pudessem parar quando não houvesse possibilidades de movimento na sua vez, como forma de prover o atingimento do estado final. Esta opção de parada pode ser feita mesmo no meio da interseção, assim como em Dresner e Stone (2005, 2008) e Au *et al.* (2012), diferente das demais estratégias de discretização detalhadas no Quadro 1 da seção 2.4, onde os carros ficam parados antes da interseção e só adentram quando há garantias de reserva.

Um ponto importante de enfatizar é que, mesmo não havendo uma reserva e o carro prioritário permaneça na mesma posição, uma transição é disparada, pois ocorre a atualização do *ID* para o próximo na sequência, gerando assim um novo estado, com as mesmas posições *X* e *Y* anteriores. Nesse sentido, embora a chance de parar não seja considerada no modelo flexível, devido às diversas alternativas de movimento, é razoável imaginar que a implementação deste artifício elevaria ainda mais a quantidade de estados e transições atingíveis, caso fosse adotado.

Com esta alteração no modelo convencional, montou-se a evolução dos estados e transições na Tabela 2.

Tabela 2 – Evolução de estados e transições para alcance do estado final no primeiros testes do modelo convencional com critério de parada.

N	Cenário 1 - Estados	Cenário 2 - Estados	Cenário 1 - Transições	Cenário 2 - Transições
8	39.305	432	54.085	482
10	248.550	1.602	338.572	1.799
12	1.134.044	4.148	1.488.519	4.751
14	2.778.373	9.995	3.573.789	11.559
15	3.709.931	26.066	4.669.234	30.288
16	6.649.968	47.067	8.362.413	55.232
17	11.941.801	78.512	14.873.919	92.004

Fonte: Autoria própria.

Comparando a Tabela 1 e a Tabela 2, percebe-se que o modelo convencional, mesmo com o impacto do critério de parada no aumento de estados e transições, ainda requer menos esforço do que o modelo flexível para realizar as sequências de travessia seguras, sem ocorrência de explosão de estados para o mesmo número de VAs usados no modelo flexível. Tal observação já era esperada, devido a menor ocorrência de combinações com menos células disponíveis para reservas.

5.2 ABSTRAÇÕES DO MODELO

Dentre os ambientes testados, o Cenário 1 com modelo flexível precisou ser repensado, por causa do fenômeno de explosão de estados com 17 VAs. Mesmo que tenha se imaginado premissas para um dos piores cenários possíveis, é razoável pensar que outras combinações podem passar por desempenhos semelhantes. Além disso, a configuração de teste com 14 VAs, com 21.932.454 estados e 26.530.479 transições, gerou um arquivo higienizado de leitura com 1,84 GB de tamanho total.

Arquivos com esta ordem grandeza são usualmente bastante complicados de se ler e de se tratar. Mesmo assim, testou-se a leitura e construção do grafo de alcançabilidade com o arquivo em questão, e a etapa de leitura, embora custosa em questão de tempo e memória, foi superada. Entretanto, ao se construir o grafo de alcançabilidade com o NetworkX, demandou-se mais poder de computação, visto que a partir

do leque de milhões de transições pode-se contabilizar uma infinidade de caminhos alternativos até o estado final desejado, impossibilitando o armazenamento de todas as combinações em memória.

A construção dos caminhos seguros através do NetworkX para o Cenário 1 com 14 VAs no modelo flexível, se tornou inatingível, uma vez que consumiu muito recurso de memória disponível, conforme visto na Figura 31.

Figura 31 – Imagem do gerenciador de tarefas mostrando o desempenho do uso da memória RAM durante a construção do grafo de alcançabilidade, via a biblioteca Python NetworkX, para uma interseção com 14 VAs no Cenário 1 com movimentos flexíveis.

The screenshot shows the Windows Task Manager Performance tab. The 'Memória' (Memory) section is highlighted in red, showing 98% usage. Below it, a table lists the memory usage for various processes:

Nome	Status	CPU	Memória	Disco
Python		15,1%	25.120,9 ...	62,1 MB/s
TeamViewer (32 bits)		0,4%	95,5 MB	0,1 MB/s
Python (4)		0,1%	73,4 MB	0 MB/s

Fonte: Autoria própria.

Devido ao estresse de memória computacional demonstrado na Figura 31, optou-se por realizar uma abstração do modelo flexível para o Cenário 1, retirando a possibilidade dos carros no fluxos oeste-leste e leste-oeste de avançarem duas células em linha reta, como uma solução para reduzir o número de estados, e, conseqüentemente, permitir a construção e análise visual do comportamento dos VAs em meio à interseção. Esta abstração se estendeu para o modelo convencional também, para manter os parâmetros de comparação nivelados. Os modelos para o Cenário 2, por apresentarem menos desgaste em relação uso do recursos, tanto para a compilação do modelo, quanto para a construção do grafo, se mantiveram com as disposições originais.

5.3 EVOLUÇÃO DOS CENÁRIOS

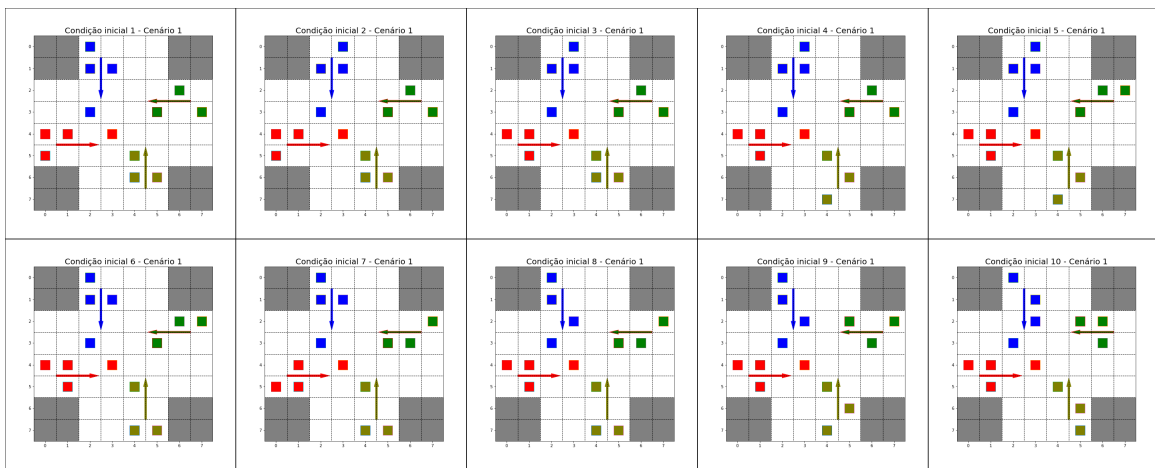
Com os ajustes feitos nos modelos, explorou-se a evolução dos cenários para outras premissas, não só em relação ao número de estados e transições, mas também para o menor número de passos e o menor número de rodadas em que os VAs de cada lote de verificação saem do *grid* em segurança, já considerando o método de Dijkstra, para a análise do menor caminho entre os arcos do grafo de alcançabilidade.

5.3.1 Condições Iniciais

Escolheu-se analisar os 2 cenários com 10 evoluções diferentes em relação às posições iniciais de cada VA no *grid*, iniciando com um número de 8 VAs, e adicionando 2 VAs por etapa até que se completasse 14. A inserção de novos veículos segue um padrão igual de distribuição entre os fluxos de cada cenário.

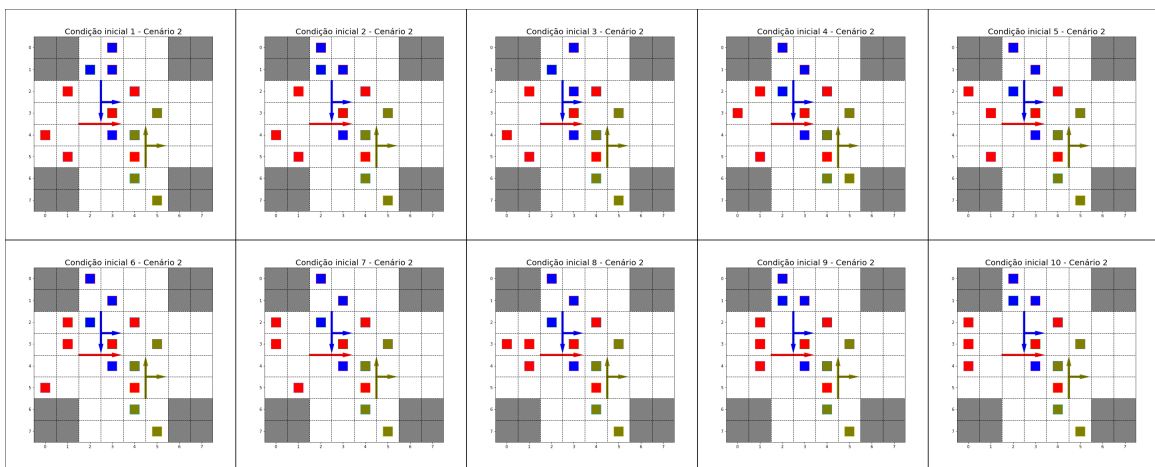
As posições iniciais para o estágio com 14 VAs foram mais distribuídas ao longo dos fluxos e não tão concentradas no início de cada fluxo, como nas condições de teste da Seção 5.1. As configurações utilizadas para o caso máximo são exemplificadas nas Figuras 32 e 33.

Figura 32 – 10 posições iniciais para o Cenário 1 com 14 veículos.



Fonte: Autoria própria.

Figura 33 – 10 posições iniciais para o Cenário 2 com 14 veículos.



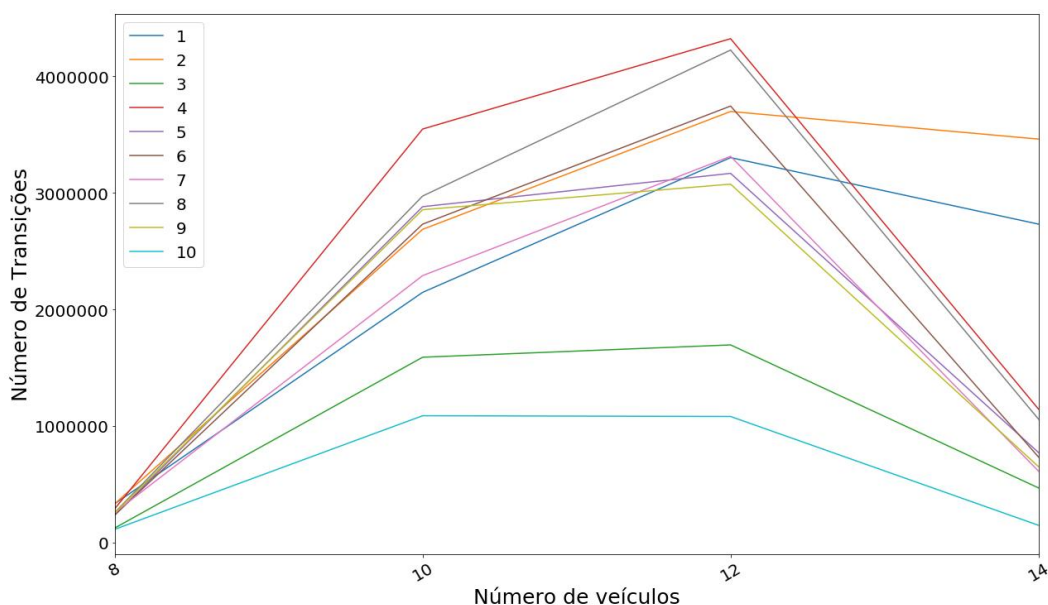
Fonte: Autoria própria.

A partir das circunstâncias estabelecidas, elaborou-se gráficos de evolução de estados e transições. Obviamente, a quantidade de VAs impacta diretamente na ordem de grandeza do espaço estados, e dependendo das condições iniciais escolhidas, pode haver uma diminuição entre uma faixa e outra subsequente, em vez de uma elevação.

5.3.2 Cenário 1 com modelo flexível

Ao se analisar o desenvolvimento do número de transições, estados e deadlocks nas Figuras 34, 35 e 36, respectivamente, percebe-se que há uma relação direta entre os perfis de ambos, para todas as linhas de evolução, no que se refere à ordem de grandeza. Entretanto, nem sempre há um aumento das variáveis quando o número de veículos no *grid* se eleva.

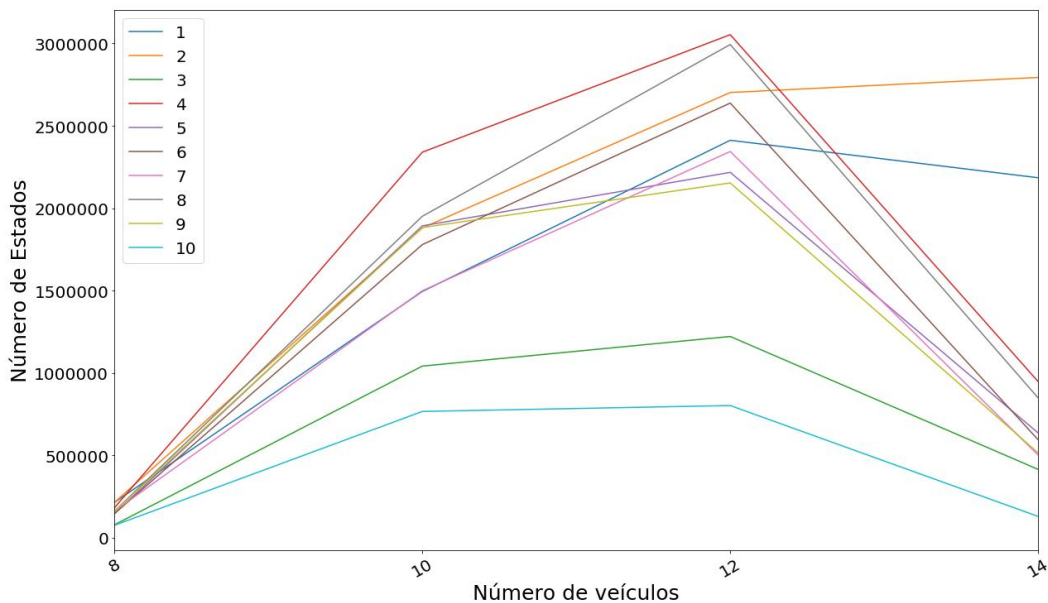
Figura 34 – Evolução de transições do Cenário 1 com o modelo de reserva flexível, a partir de 10 condições iniciais diferentes.



Fonte: Autoria própria.

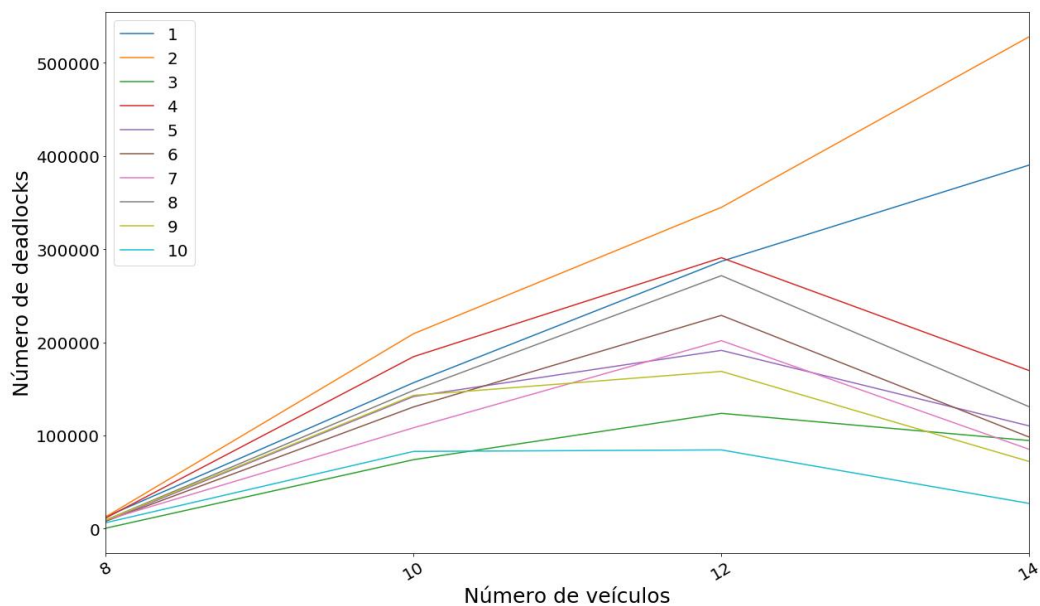
É notório que a abstração feita no modelo, reduziu consideravelmente a composição do espaço de estados e transições, uma vez que agora as ramificações que antes se criavam com o avanço de duas células nos sentidos oeste-leste e leste-oeste não existem mais. Em alguns casos, como o da condição inicial 10, há uma redução por volta de 99% no número de estados para 14 VAs, em relação às condições de teste da Seção 5.1. Agora com um progresso limitado a um recurso por rodada, se um VA iniciar em uma célula nas adjacências do *grid*, ele precisará executar no mínimo 7 rodadas até chegar à fronteira da área de influência da UCI, do lado oposto à sua origem.

Figura 35 – Evolução de estados do Cenário 1 com o modelo de reserva flexível, a partir de 10 condições iniciais diferentes.



Fonte: Autoria própria.

Figura 36 – Evolução dos estados de *deadlocks* no Cenário 1 com o modelo de reserva flexível, a partir de 10 condições iniciais diferentes



Fonte: Autoria própria.

Observa-se que a evolução das linhas 1 (em azul escuro) e 2 (em laranja), tanto em estados, quanto em transições, apresentam uma ordem de grandeza entre 2.000.000 e 4.000.000 para a faixa de 14 VAs, bem superior às demais, que ficam em torno ou abaixo de 1.000.000. Com o aumento discrepante no número de *deadlocks* de ambas, conforme apresentado na Figura 36, infere-se que o número de *deadlocks* impacta tanto no número de estados totais atingidos, quanto nas transições disparadas.

Nota-se como no desenvolvimento dos estados ocorre um decréscimo entre 12 e 14 VAs em todas as linhas de evolução, exceto na linha 2, onde há um aumento de 2.702.188 para 2.793.780. Neste caso, é razoável supor que as posições iniciais resultaram em menos recursos para serem reservados nas rodadas iniciais para um volume maior de carros em travessia simultânea. Contudo, à medida que em algumas ramificações os VAs prioritários nos fluxos norte-sul e sul-norte avançam mais rápido, os recursos vão ficando disponíveis.

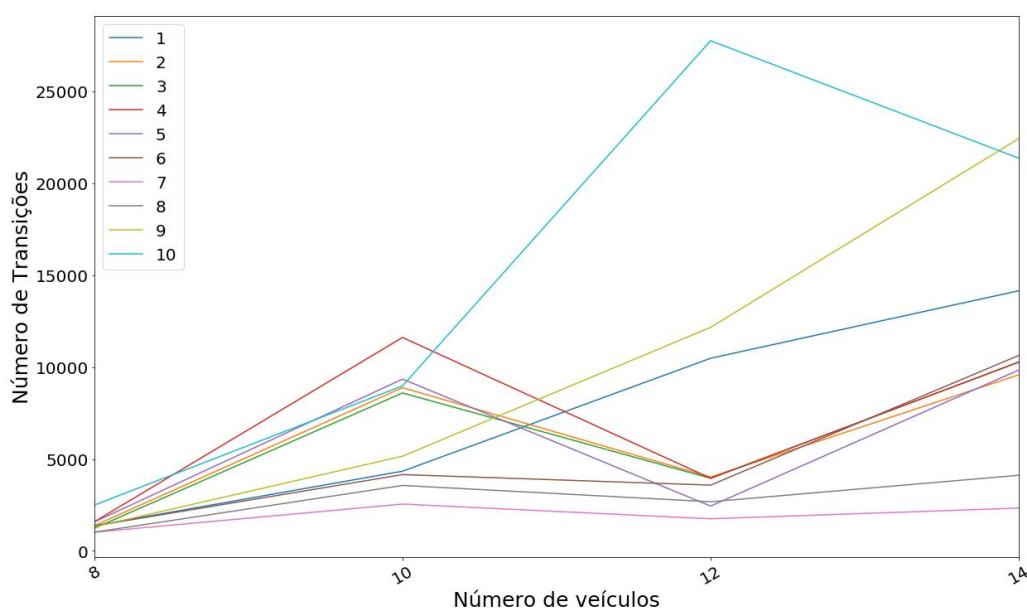
Desta forma, assume-se que para estas condições onde ocorrem há uma redução na linha evolução, não há muita ocorrência de *deadlocks* nas etapas finais do processo de alocação de células, enquanto que para 12 veículos, os *deadlocks* tendem a se acumular ao longo de toda a verificação, pela maior disponibilidade de área compartilhada, resultando em mais arranjos entre os conjuntos e variáveis de e_k .

5.3.3 Cenário 1 com modelo convencional

Da mesma forma que no modo flexível, o modelo convencional também não possui um padrão único de desenvolvimento do número de estados e de transições, variando de acordo com as posições iniciais dos VAs.

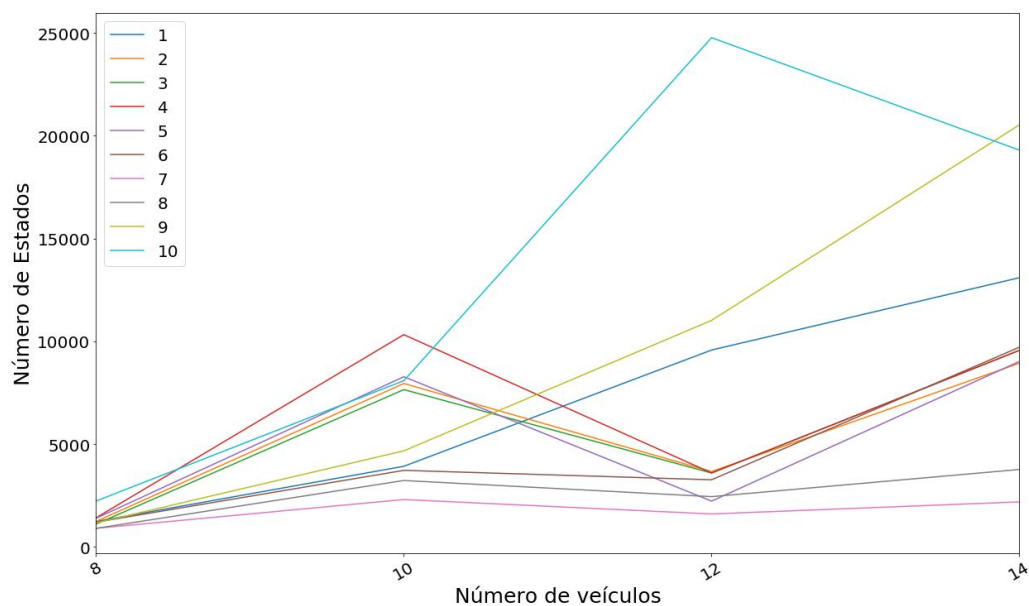
Devido ao critério de parada adotado, não há *deadlocks* neste contexto, exceto pelo estado final desejado. Assim, o número de paralisações é o único parâmetro que impacta na quantidade de variáveis geradas. As evoluções das transições e dos estados são mostradas nas Figuras 37 e 38.

Figura 37 – Evolução de transições do Cenário 1 com o modelo de reserva convencional, a partir de 10 condições iniciais diferentes.



Fonte: Autoria própria.

Figura 38 – Evolução de estados do Cenário 1 com o modelo de reserva convencional, a partir de 10 condições iniciais diferentes.



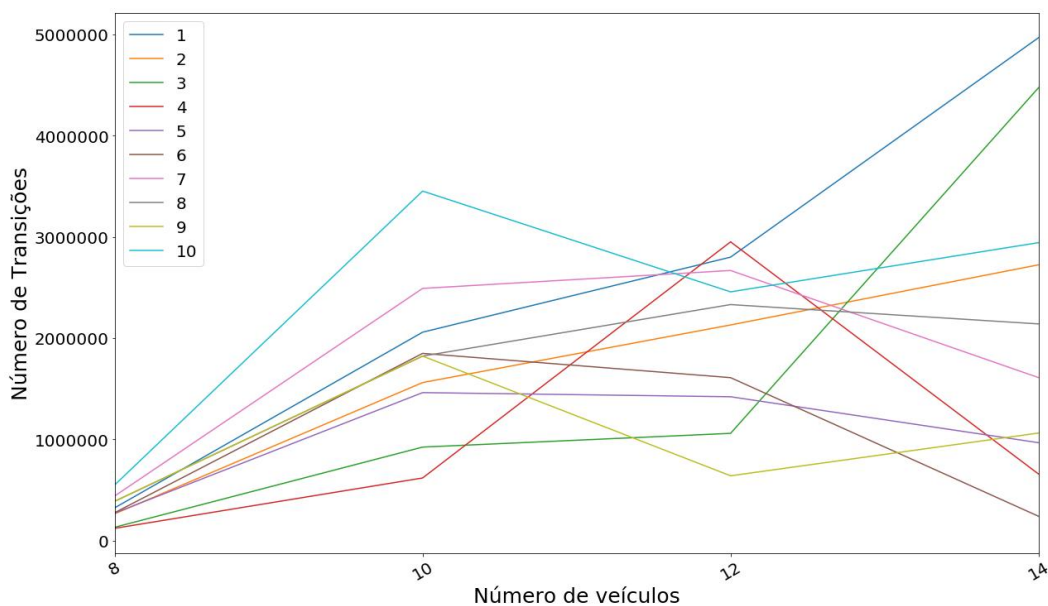
Fonte: Autoria própria.

5.3.4 Cenário 2 com modelo flexível

Para o Cenário 2, é necessário considerar que o destino individual de cada VA é um fator preponderante na geração dos estados, uma vez que nas vias secundárias pode se seguir reto ou fazer a conversão para o fluxo da avenida principal, diferente do Cenário 1, onde os veículos só podem seguir no mesmo fluxo.

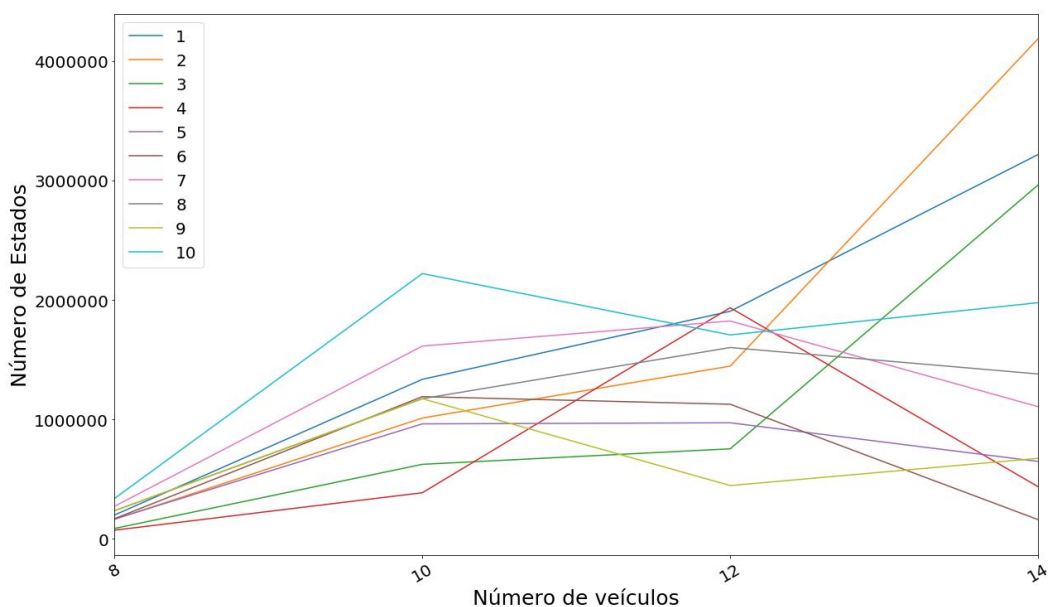
Outro fator adicional de influência é a diversidade de células à disposição, já que a avenida, por ser mão única, possui quatro faixas. Com esses aspectos, a evolução do modelo flexível difere bastante entre todas as condições iniciais, conforme mostrado nas Figuras 39, 40 e 41.

Figura 39 – Evolução de transições do Cenário 2 com o modelo de reserva flexível, a partir de 10 condições iniciais diferentes.



Fonte: Autoria própria.

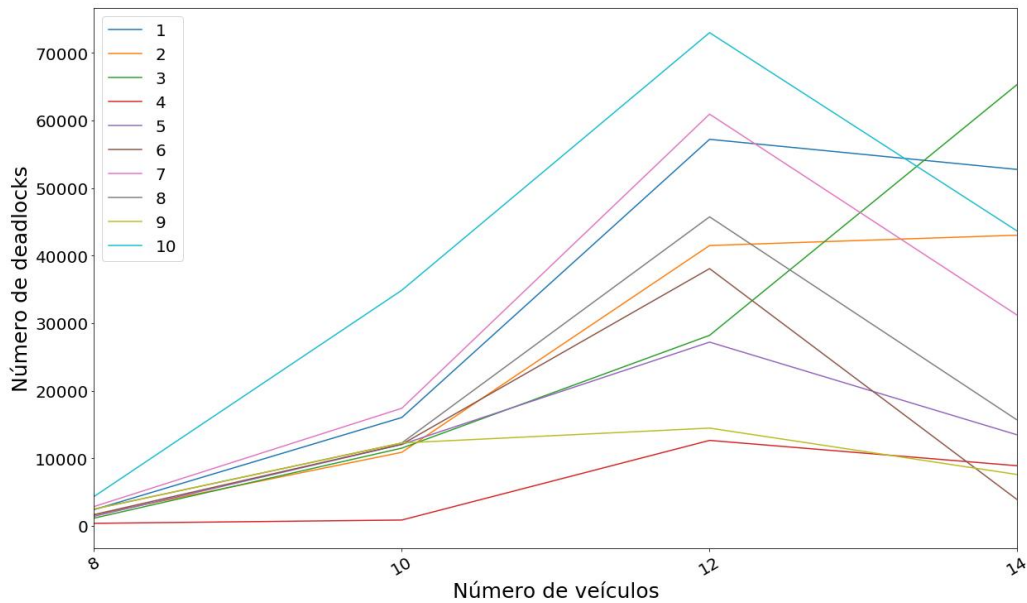
Figura 40 – Evolução de estados do Cenário 2 com o modelo de reserva flexível, a partir de 10 condições iniciais diferentes.



Fonte: Autoria própria.

Dentre os comportamentos vistos na Figura 40, o da linha de evolução 3 (em verde) apresenta um salto agressivo de 753.097 para 2.963.143, quando se aumenta de 12 para 14 VAs no *grid*. Analisando as condições iniciais de ambos, conforme Figura 42, assume-se que a adição de um VA na posição [2,1] pode ser o grande responsável pela grande alteração observada, pois o destino do mesmo foi definido no mesmo fluxo,

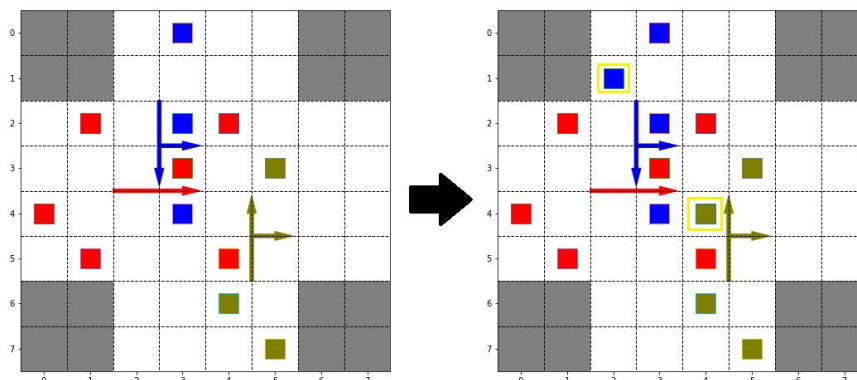
Figura 41 – Evolução dos estados de *deadlocks* no Cenário 2 com o modelo de reserva flexível, a partir de 10 condições iniciais diferentes



Fonte: Autoria própria.

sem conversão de via. Desta forma, ao longo das rodadas deve se deparar com alguns *deadlocks*, conforme evidenciado na Figura 41, pois os VAs prioritários (quadrados vermelhos) nas posições [1,2], [1,5] e [0,4] que seguem pela avenida principal, estarão passando a sua frente.

Figura 42 – Comparação entre as condições iniciais para 12 (à esquerda) e 14 VAs (à direita) na linha de evolução 3. Os VAs adicionais são destacados com um quadrado amarelo.



Fonte: Autoria própria.

Em contrapartida, a condição 5 (em roxo) apresenta um declínio na mudança de 12 para 14 VAs, com apenas cinco alterações sutis em relação à condição mostrada na Figura 42, o que reforça o dinamicismo do modelo. Com a escolha correta da

condição inicial é possível reduzir bastante o esforço de memória de processamento para computar os espaço de estados e transições.

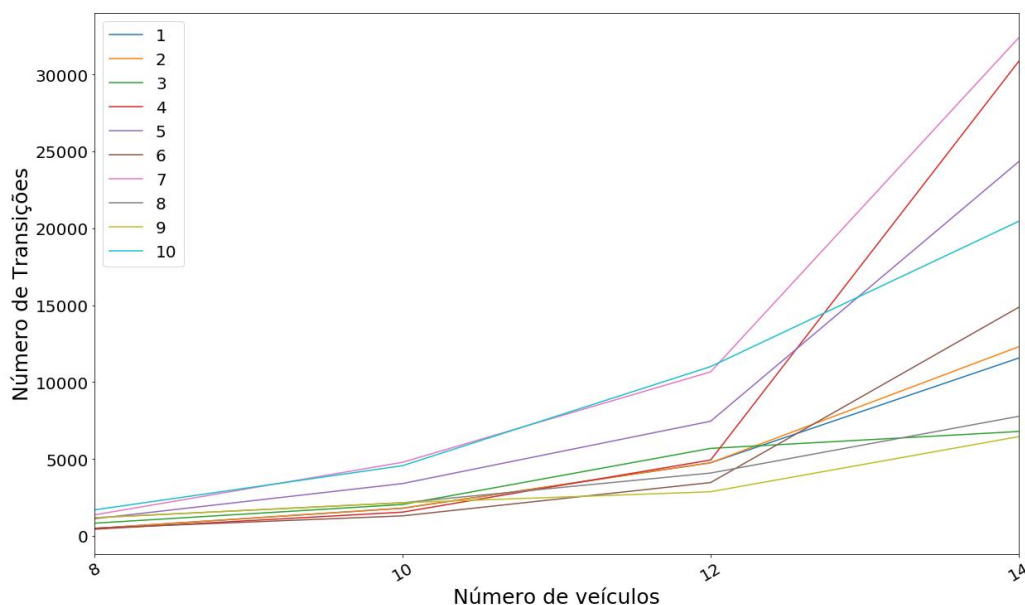
Examinado mais detalhadamente a Figura 41, percebe-se que, diferente do Cenário 1 com modelo flexível, o comportamento dos *deadlocks* não aparenta ter um impacto muito incisivo no número de estados e transições. Por exemplo, com base na faixa de 14 VAs, a condição 3 é a que tem mais *deadlocks*, com 65.347, mas é apenas a terceira com mais estados e a segunda com maior número de transições, enquanto que a condição 2 é a quarta com mais *deadlocks*, possuindo 43.026 atingimentos, mas é a que tem o maior número de estados, além de ser a quarta com mais transições.

De forma geral, para o modelo flexível, o Cenário 2 possui bem menos *deadlocks* que o Cenário 1, visto que o maior número obtido é 72.995, com a 12 VAs na condição 10, enquanto para o Cenário 1, a média para a mesma faixa de VAs é superior a 200.000.

5.3.5 Cenário 2 com modelo convencional

Pelo fato do Cenário 2 com o modelo de reserva convencional ter um menor número de fluxos de entrada e haver um seguimento de padrões definidos de caminhos, os perfis de evolução apresentam acréscimos contínuos em relação a estados e a transições, conforme se aumenta a quantidade de VAs no *grid*. Isso se reflete em todas as 10 condições testadas, conforme mostrado nas Figuras 43 e 44.

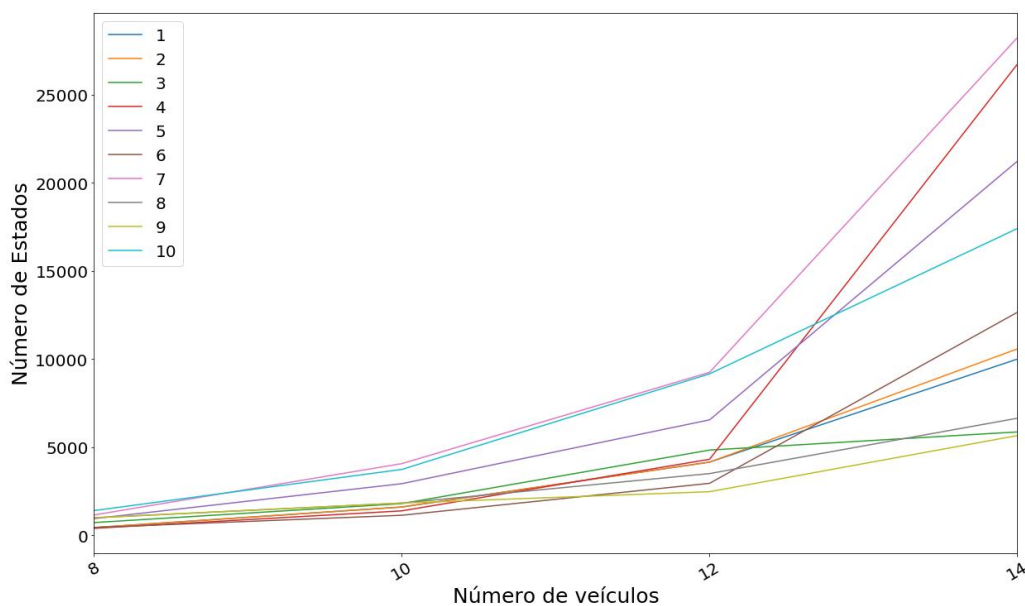
Figura 43 – Evolução de estados do Cenário 2 com o modelo de reserva convencional, a partir de 10 condições iniciais diferentes, para 8, 10, 12 e 14 VAs.



Fonte: Autoria própria.

Além de seguir esse progresso sempre crescente, este cenário é também o que apresenta o menor esforço computacional para todas as condições, mesmo para o

Figura 44 – Evolução de estados do Cenário 2 com o modelo de reserva convencional, a partir de 10 condições iniciais diferentes, para 8, 10, 12 e 14 VAs.



Fonte: Autoria própria.

cenário com maior número de transições e estados, com 32.383 transições e 28.235 estados, na linha de evolução 7 (em rosa).

5.3.6 Passos e rodadas nos menores caminhos

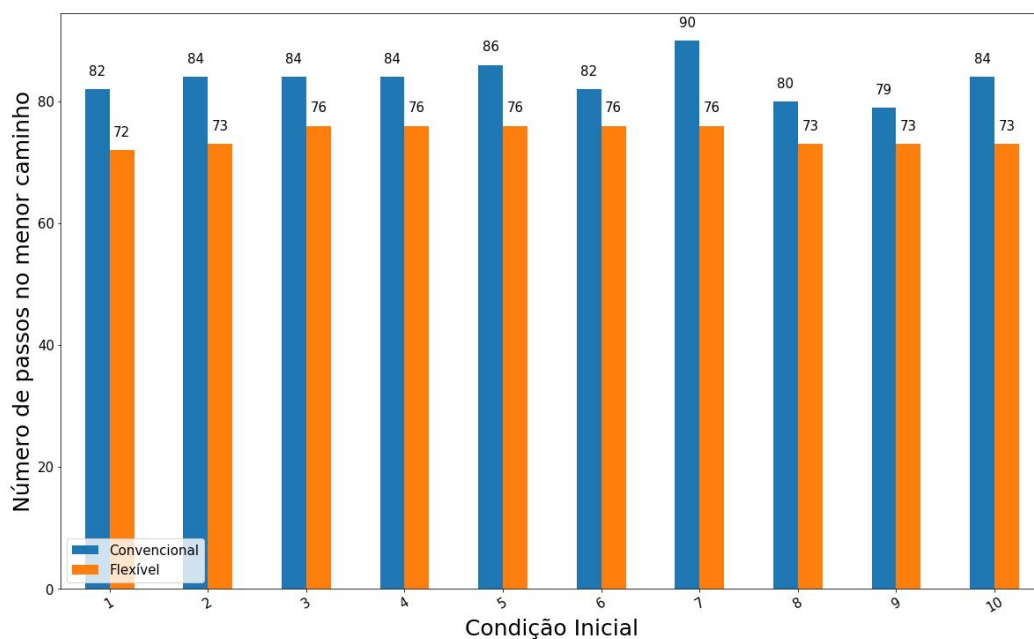
Dentre as condições testadas, escolheu-se as configurações com um lote de 14 VAs para se analisar os menores números de passos discretos e rodadas necessárias para o escoamento total nos dois cenários. Neste caso, considerou-se uma amostra de menor T_f para cada cenário, obtida com o *shortest_path*.

A comparação entre os dois modelos para o Cenário 1 é vista nas Figuras 45 e 46.

Os dois gráficos, de passos e rodadas, mostram que em todas as condições testadas, o modelo flexível consegue garantir que os VAs cruzem as interseções com segurança em um menor número de passos e rodadas por lote, em relação ao modelo convencional, indicando uma capacidade de se ter um fluxo maior de veículos. Visto que se trata de uma comparação entre dois sistemas de reservas e considerando que o modelo flexível consegue explorar mais recursos e traçar caminhos adicionais, e o modelo convencional é mais restrito, esperava-se algo semelhante ao resultado obtido.

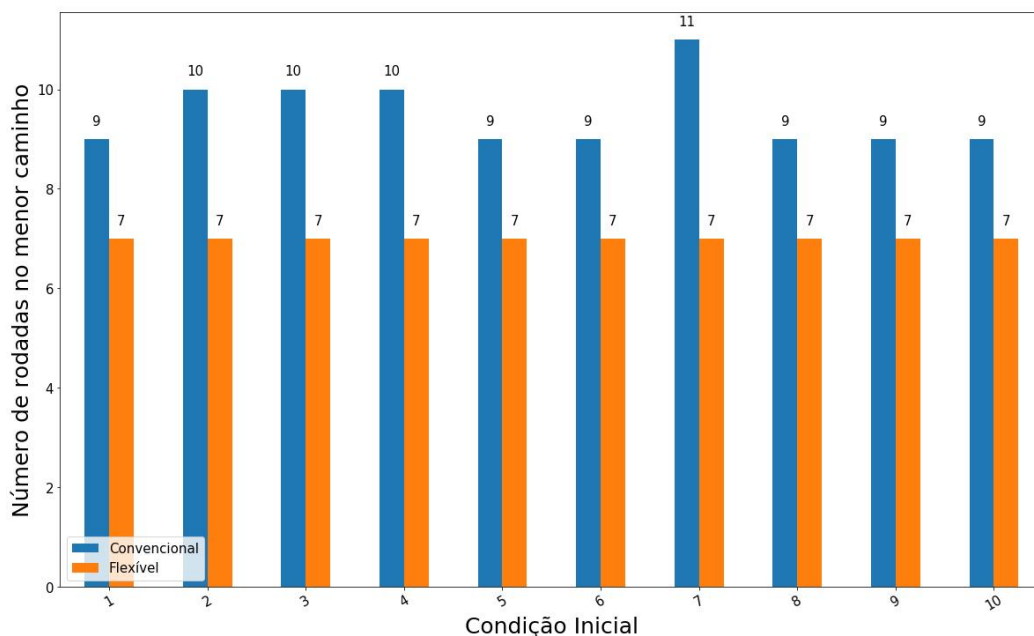
Como não se considera equações cinemáticas e dinâmicas, a análise de passos e rodadas é bastante preliminar quanto a eficiência em relação aos métodos convencionais, e este trabalho não se propõe a aprofundar este ponto. Entretanto, esta abordagem se mostra eficaz quanto a ratificar que a travessia dos veículos será feita no menor tempo discreto possível e sem conflito nas alocações.

Figura 45 – Menor número de passos para esvaziar o *grid* para 14 VAs no Cenário 1.



Fonte: Autoria própria.

Figura 46 – Menor número de rodadas para esvaziar o *grid* para 14 VAs no Cenário 1.

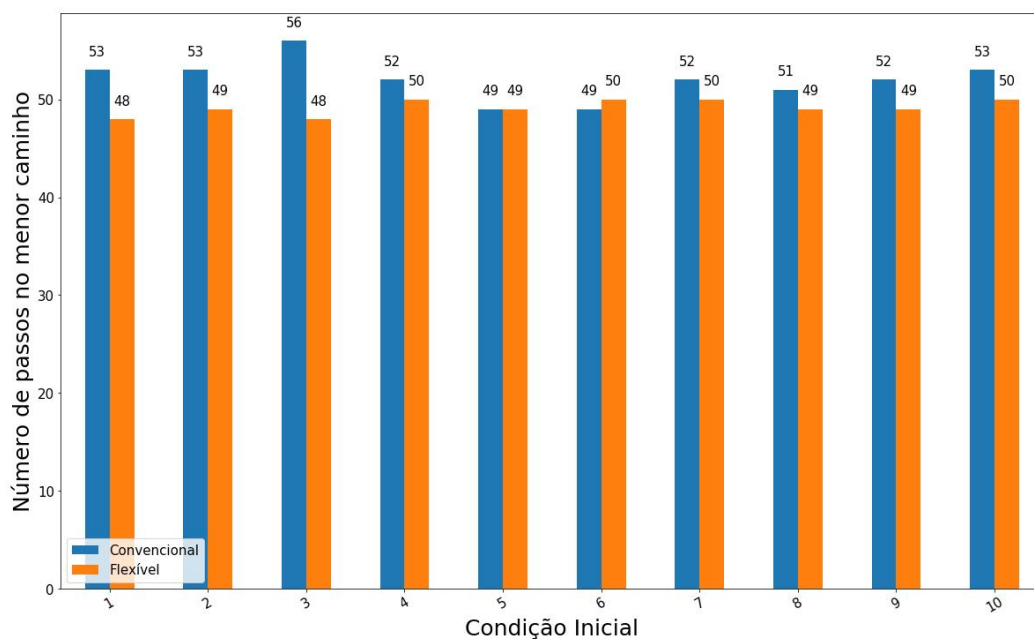


Fonte: Autoria própria.

Uma observação a ser feita em relação a Figura 46 é que ela apresenta uma contagem de rodadas similar para todas as condições. E isso se justifica, já que o último VA a sair é sempre um que está na célula mais à direita do *grid* e que anda no sentido leste-oeste, logo, levará sete passos para cobrir todo o *grid*.

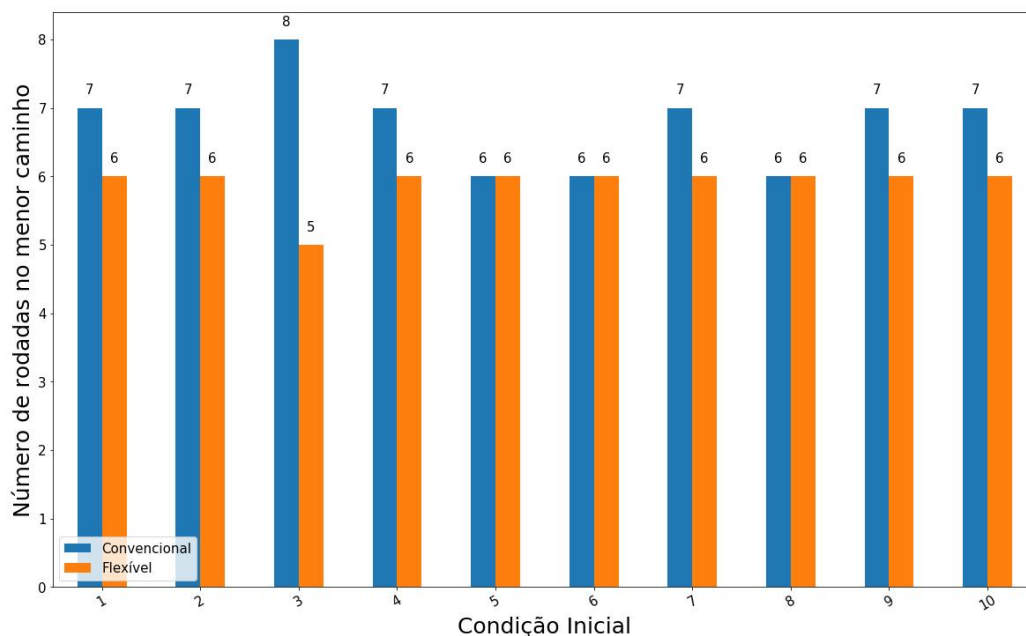
A comparação entre os dois modelos para o Cenário 2 é vista nas Figuras 47 e 48.

Figura 47 – Menor número de passos para esvaziar o *grid* para 14 VAs no Cenário 2.



Fonte: Autoria própria.

Figura 48 – Menor número de rodadas para esvaziar o *grid* para 14 VAs no Cenário 2.



Fonte: Autoria própria.

Assim como no cenário anterior, o modelo flexível se sobressaiu em relação ao modelo convencional, mas em algumas condições chegou até a ter números maiores, como na condição 6.

Com menos fluxos de entrada e mais liberdade para os VAs na avenida principal, o Cenário 2 acabou dando uma vazão maior em relação à alocação de reservas comparado a Cenário 1, tanto no modelo flexível, quanto no convencional.

Investigando o desempenho de passos e rodadas, nota-se que não há uma correlação aparente entre estes parâmetros e a evolução de estados, transições e *deadlocks*. Por exemplo, para o Cenário 1 com modelo flexível, as condições 1 e 2 se destacaram por ter mais estados que os demais, mas ambas se apresentaram com os menores números de passos no caminho com menor T_f .

5.4 ANÁLISE DE CAMINHOS

Para averiguar de forma visual como os caminhos de uso livre gerados pela UCI se distribuem pela área de interseção, escolheu-se a condição inicial 5 com 14 VAs como amostra. Além dos caminhos com menores T_f , explorou-se outras faixas, utilizando a função *all_shortest_paths* para selecionar aqueles que atingem o estado final em 76, 77, 78 passos no Cenário 1, e 49, 50 e 51 passos no Cenário 2.

5.4.1 Segmentação por número de passos e rodadas

O número de caminhos obtidos para o Cenário 1 e para o Cenário 2 foram 12.758.016 e 201.963, respectivamente. Como o Cenário 1 possui um maior T_f é natural que possua muito mais caminhos seguros em relação ao Cenário 2, visto que possibilita mais combinações de movimentos. A distribuição dos caminhos por cada faixa é vista na Tabela 3 e na Tabela 4.

Tabela 3 – Segmentação de caminhos por passos para o Cenário 1.

Passos	Caminhos
76	199.680
77	2.696.448
78	9.861.888

Fonte: Autoria própria.

Tabela 4 – Segmentação de caminhos por passos para o Cenário 2.

Passos	Caminhos
49	604
50	15.788
51	185.571

Fonte: Autoria própria.

O repartimento de caminhos mostra que há um leque vasto de opções a ser explorado além das com menor T_f . Além de focar no número absoluto de passos, também dividiu-se cada faixa apresentadas em rodadas. Todas as três divisões por passos do Cenário 1 completaram o percurso em 7 rodadas, que é o menor possível.

Tabela 5 – Segmentação de caminhos de caminhos por passos e rodadas.

Caminhos por número de rodadas				
Passos	6	7	8	
49		604	0	0
50		13.588	2.200	0
51		114.780	66.711	4.080

Fonte: Autoria própria.

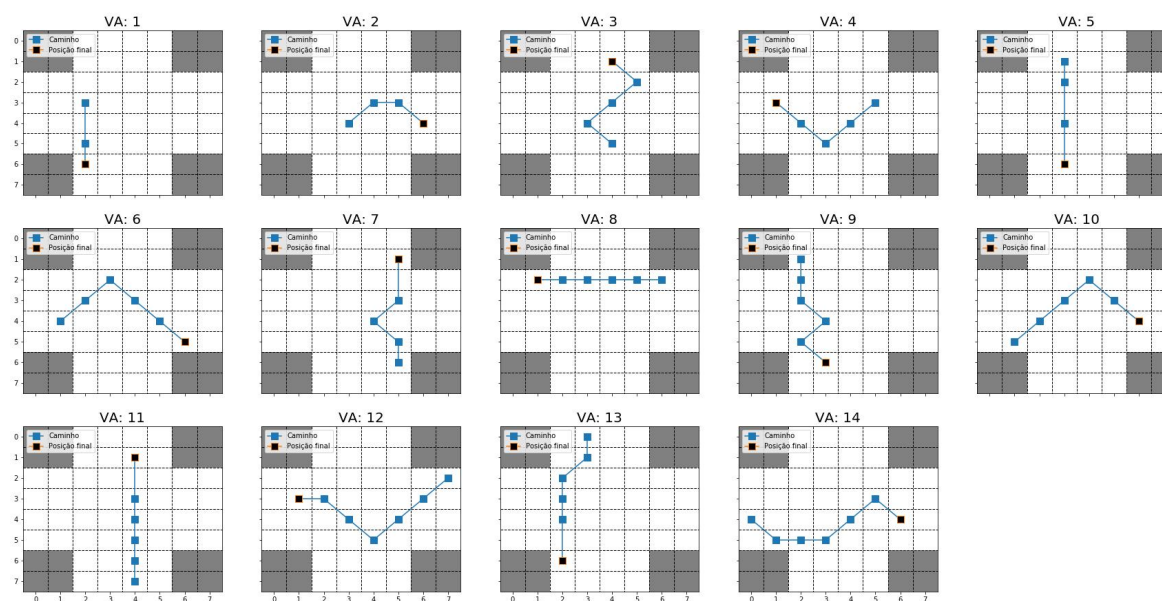
Já no Cenário 2, o desdobramento de caminhos por passos e rodadas é visto na Tabela 5.

Observando os números da Tabela 5, vê-se que analisar apenas o número de passos acaba ocultando outras boas possibilidades de escolha, pois mesmo com um número maior de passos, existem alguns casos onde os VAs sairão no mesmo número de rodadas que a faixa de menores T_f . Este tipo de informação pode ficar armazenado na UCI para que ela trace rotas alternativas que, globalmente, atenderão o sistema de reservas no menor tempo discreto possível.

5.4.2 Sequências de reservas

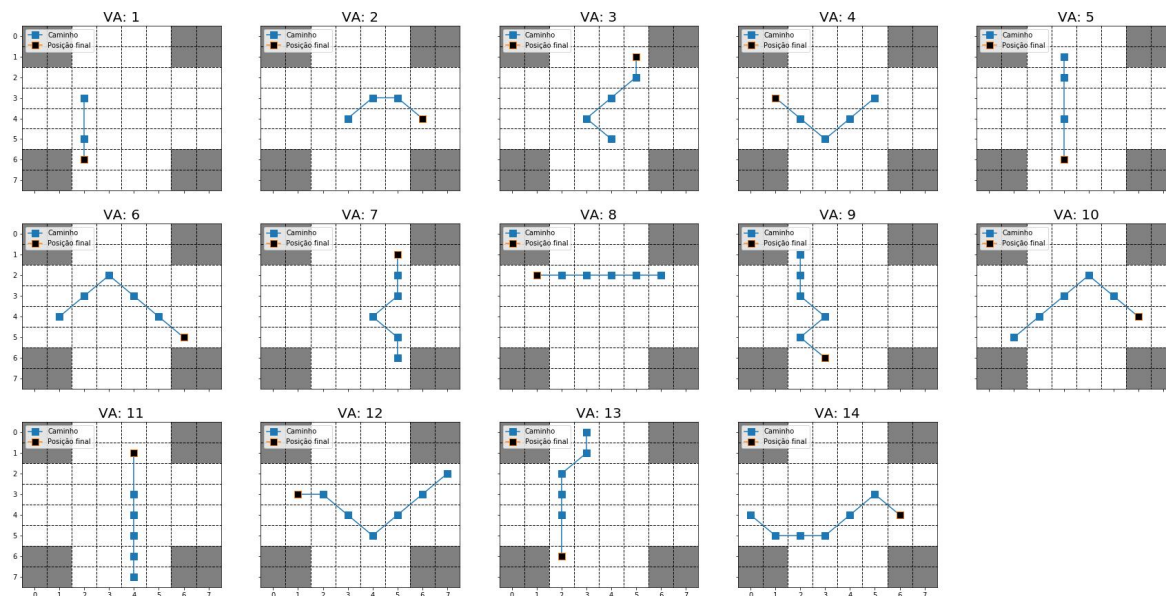
Para ilustrar os caminhos seguidos no Cenário 1, retirou-se uma exemplo de cada classificação da Tabela 3. Para as três amostras apresentadas, alguns VAs possuem uma trajetória bastante diferenciada, realizando manobra dentro das suas possibilidades. As trajetórias individuais são vistas nas Figuras 49, 50 e 51.

Figura 49 – Exemplo de uma sequência de reservas no Cenário 1 com 76 passos e 7 rodadas.



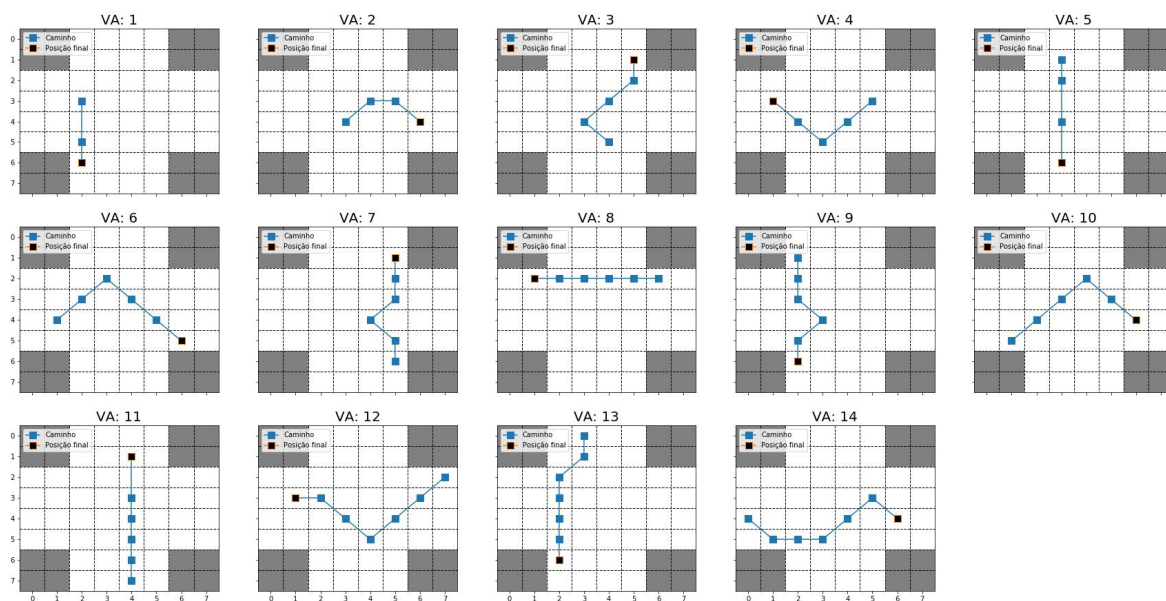
Fonte: Autoria própria.

Figura 50 – Exemplo de uma sequência de reservas no Cenário 1 com 77 passos e 7 rodadas.



Fonte: Autoria própria.

Figura 51 – Exemplo de uma sequência de reservas no Cenário 1 com 78 passos e 7 rodadas.



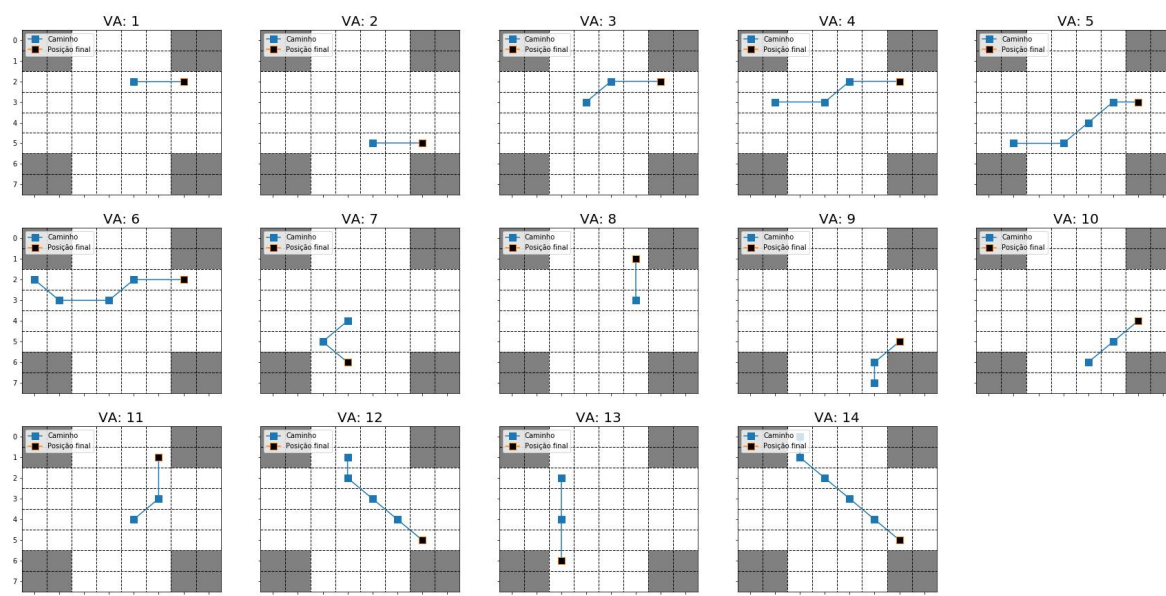
Fonte: Autoria própria.

Os caminhos representados ilustram como seria um cenário onde os VAs pudessem usar livremente a área de interseção. Nestes exemplos, observa-se que alguns VAs permanecem fazendo trajetórias tradicionais, enquanto outros percorrem em direções nada usuais para a atualidade, conforme desejado. Isto pode gerar uma impressão de penalidade para alguns veículos, entretanto, ao se pensar de forma global na redução de permanência no trânsito, explorar estes novos atalhos seguros, tendem

a ser interessantes.

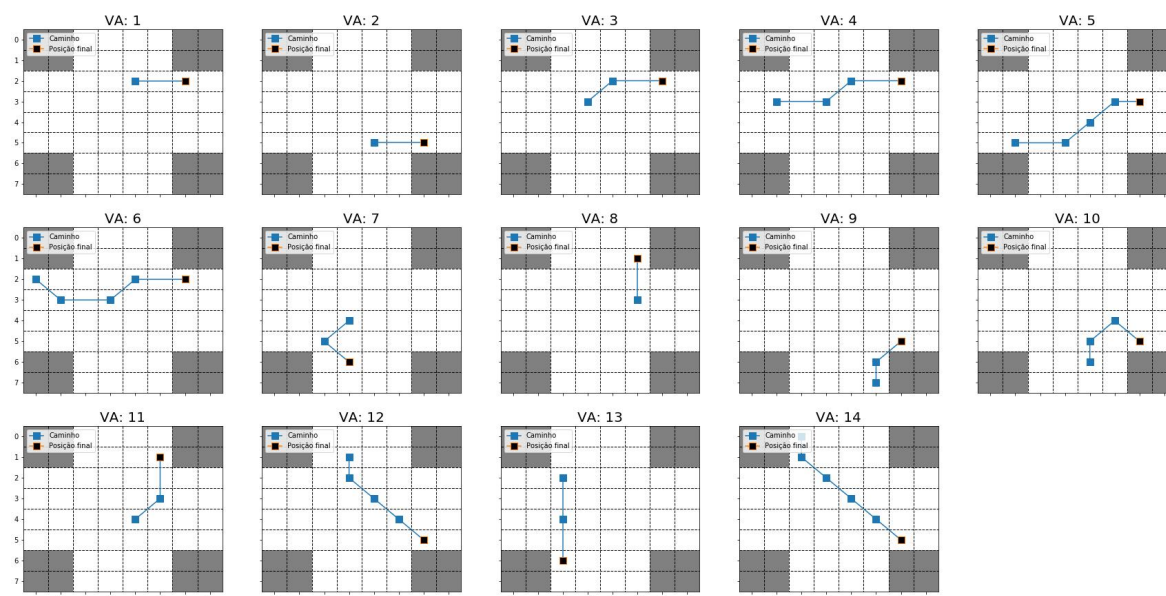
Da mesma forma, desenhou-se três exemplos para o Cenário 2, vistos nas Figuras 52, 53 e 54.

Figura 52 – Exemplo de uma sequência de reservas no Cenário 2 com 49 passos e 6 rodadas.



Fonte: Autoria própria.

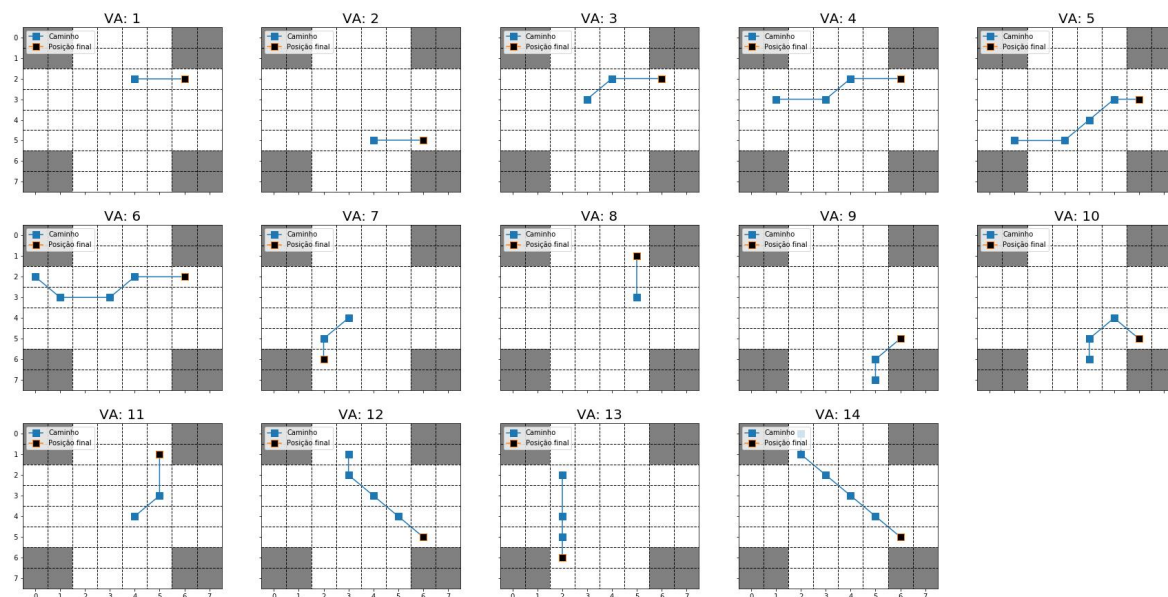
Figura 53 – Exemplo de uma sequência de reservas no Cenário 2 com 50 passos e 6 rodadas.



Fonte: Autoria própria.

Com as amostras analisadas, percebe-se menos diferenciação nos movimentos, uma vez que os VAs na avenida primária tem prioridade e mais liberdade de escolha, resulta que eles sem mantêm em caminhos mais convencionais. Por outro lado, VAs

Figura 54 – Exemplo de uma sequência de reservas no Cenário 2 com 51 passos e 6 rodadas.



Fonte: Autoria própria.

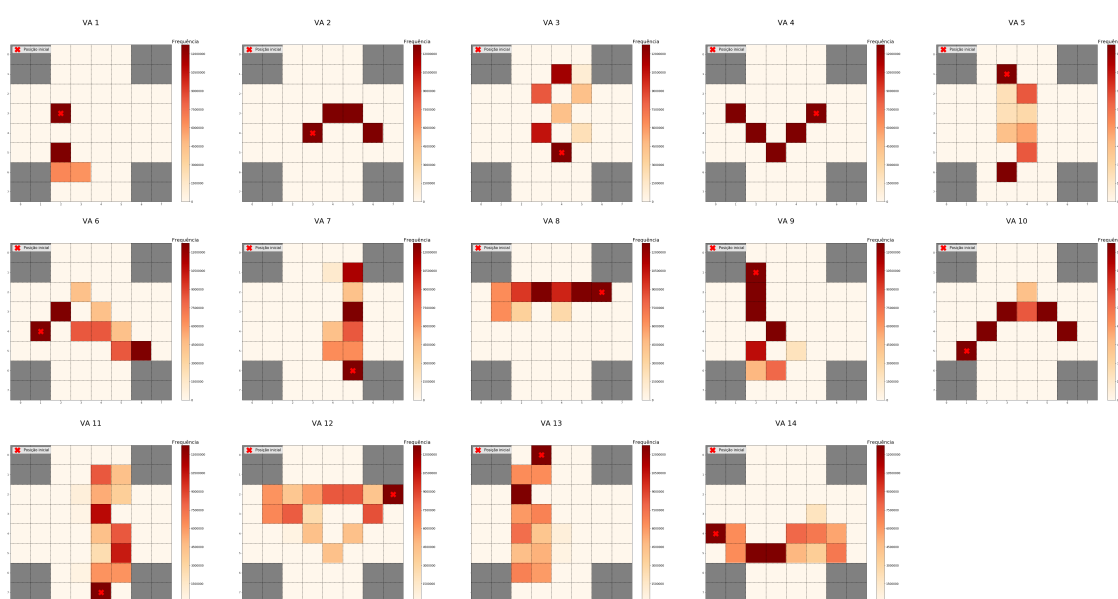
no sentido norte-sul e com um destino com necessidade de conversão à esquerda são os que apresentam manobras mais diversificadas.

5.4.3 Distribuição de passos

Para se ter uma noção ainda maior da ocupação dos espaços na interseção, plotou-se a distribuição de passos através de um mapa de calor. Este mapa identifica quais células foram mais ocupadas por mais tempo, levando em conta um número determinado de caminhos percorridos.

Testou-se o Cenário 1 para todos os 12.758.016 caminhos que garantiram a travessia segura em 7 rodadas. O resultado é visto na Figura 55 de forma individual para a condição inicial 5 com 14 veículos.

Figura 55 – Distribuição de passos no Cenário 1 para os caminhos com 7 rodadas. O 'X' vermelho indica a posição inicial do VA.



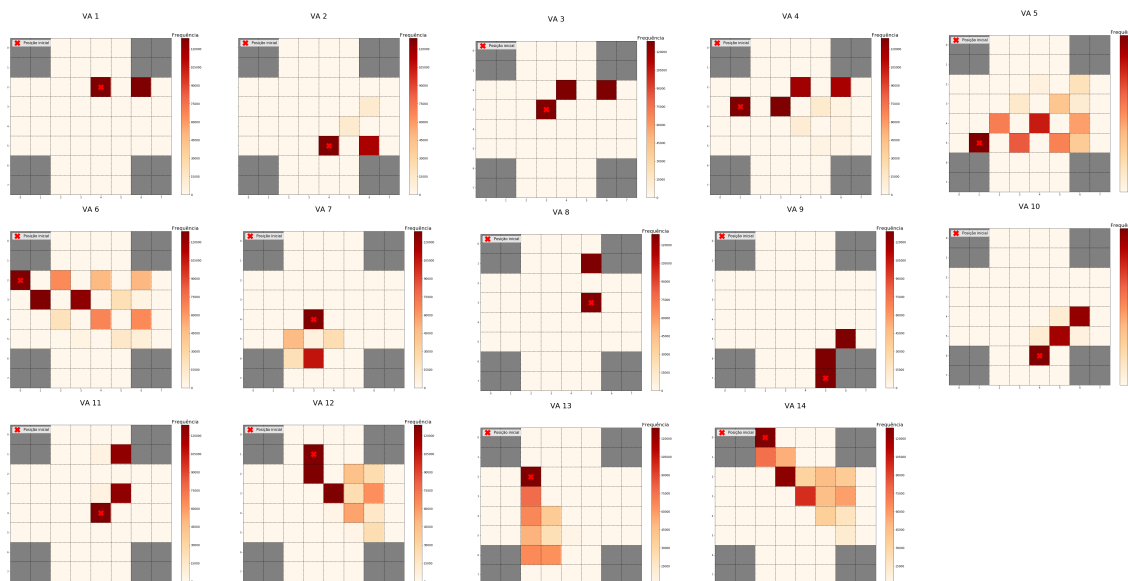
Fonte: Autoria própria.

Na Figura 55 é possível perceber que, exceto pelo VAs 2 e 4, que apresentam um seguimento regular de células, os demais variam bastante a exploração de células, com destaque o VA 12.

Em relação ao Cenário 2, por ter mais variedade de segmentações, verificou-se a distribuição de passos para caminhos com saída em 6 e 8 rodadas, respectivamente o menor e o maior valor apresentados na Tabela 5. Os mapas de calor para essas duas configurações são vistos nas Figuras 56 e 57.

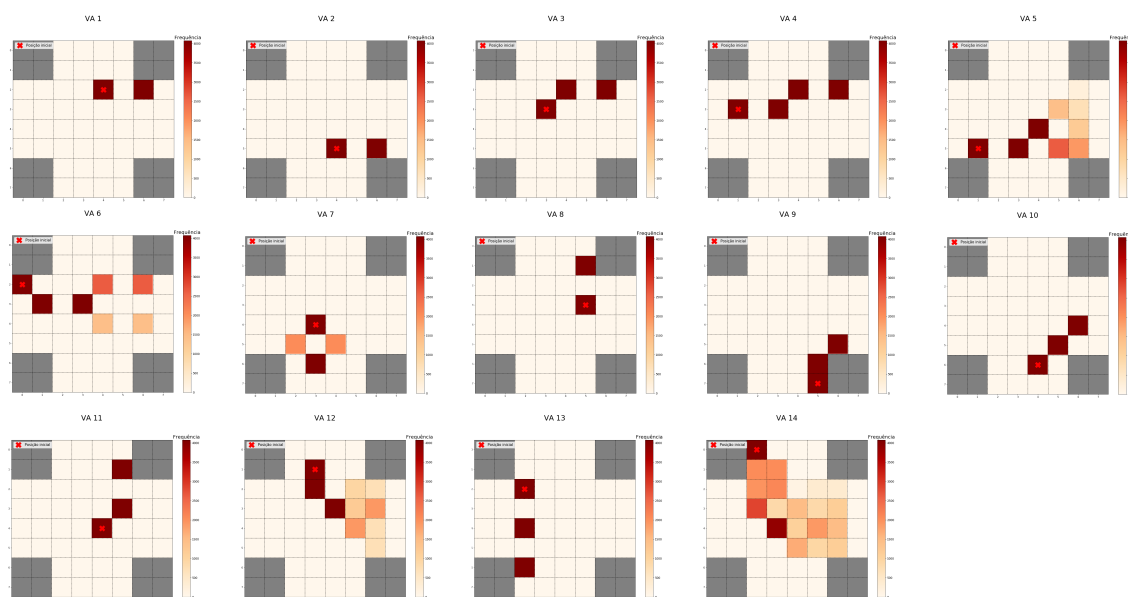
Tanto na Figura 56, quanto na Figura 57, percebe-se que a maioria dos VAs não varia os caminhos seguidos, exceto pelos VAs 5, 6, 7, 12, 13 e 14. Aqui vale relembrar que o Cenário 2 possui uma T_f bem menor, pois vários VAs saem mais rápido *grid* por conta da sua posição e do seu destino. Vale destacar o comportamento dos VAs 12 e 14, uma vez que, por precisarem fazer a conversão à esquerda, encontram muitos

Figura 56 – Distribuição de passos no Cenário 2 para os caminhos com 6 rodadas. O 'X' vermelho indica a posição inicial do VA.



Fonte: Autoria própria.

Figura 57 – Distribuição de passos no Cenário 2 para os caminhos com 8 rodadas. O 'X' vermelho indica a posição inicial do VA.



Fonte: Autoria própria.

mais obstáculos que os demais, e por isso apresentam uma maior variabilidade nos caminhos possíveis.

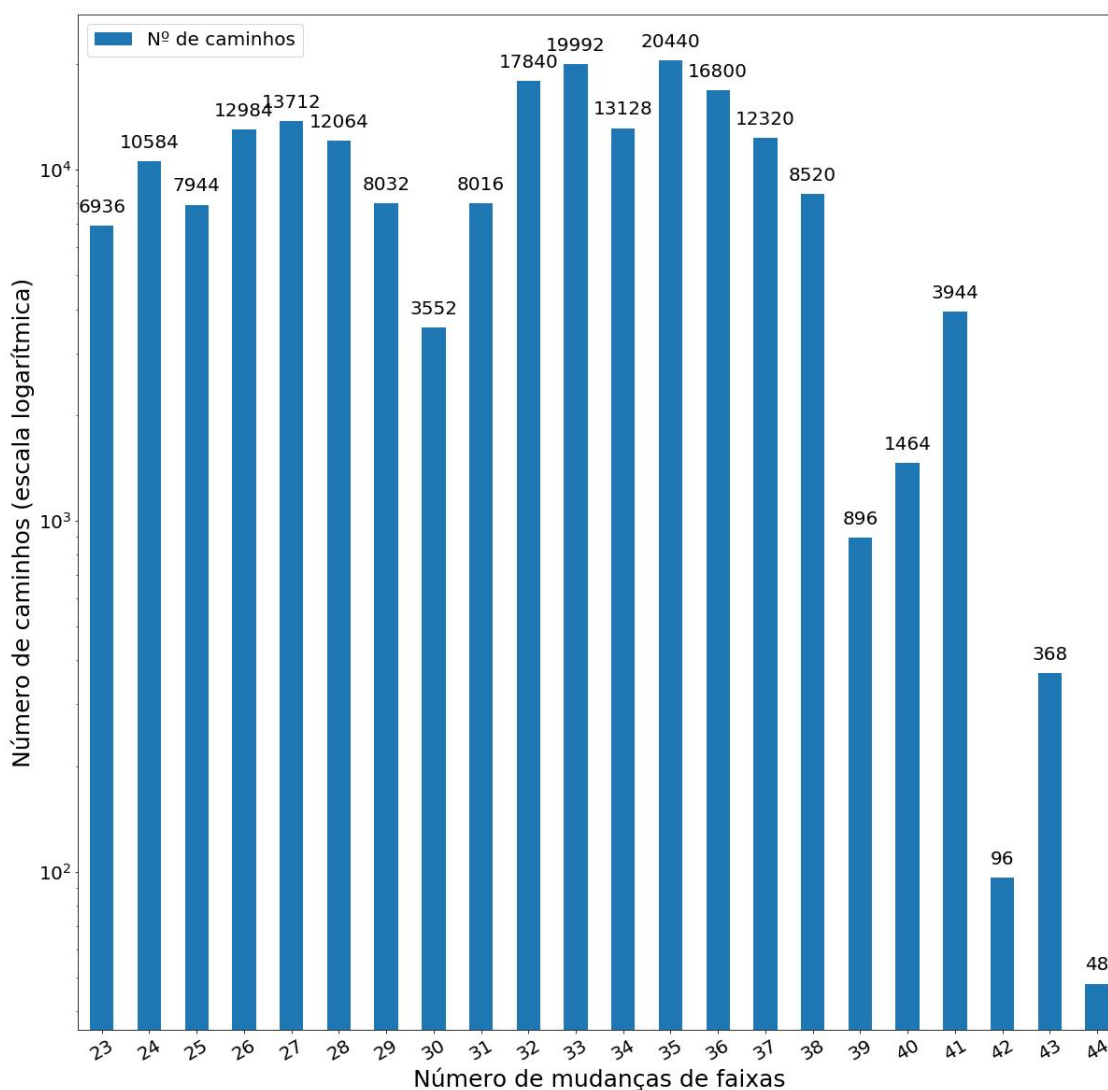
Em todos os três mapas de calor é possível notar diversos padrões de comportamentos, onde mesmo com as possibilidades maleáveis de movimentos, alguns carros tendem a sempre seguir por caminhos conservadores, enquanto outros conseguem diversificar bastante suas rotas seguras.

5.4.4 Variação de faixas

Como a intensa mudança de faixas pode causar estranheza ao usuário final, também pensou-se em uma segmentação com base no número de vezes que os veículos fazem o intercambiamento de células na diagonal, ou seja, separar aqueles caminhos onde os carros conseguem andar mais de forma retilínea. Desta forma, pegou-se as execuções menores T_f e contou-se o número de trocas.

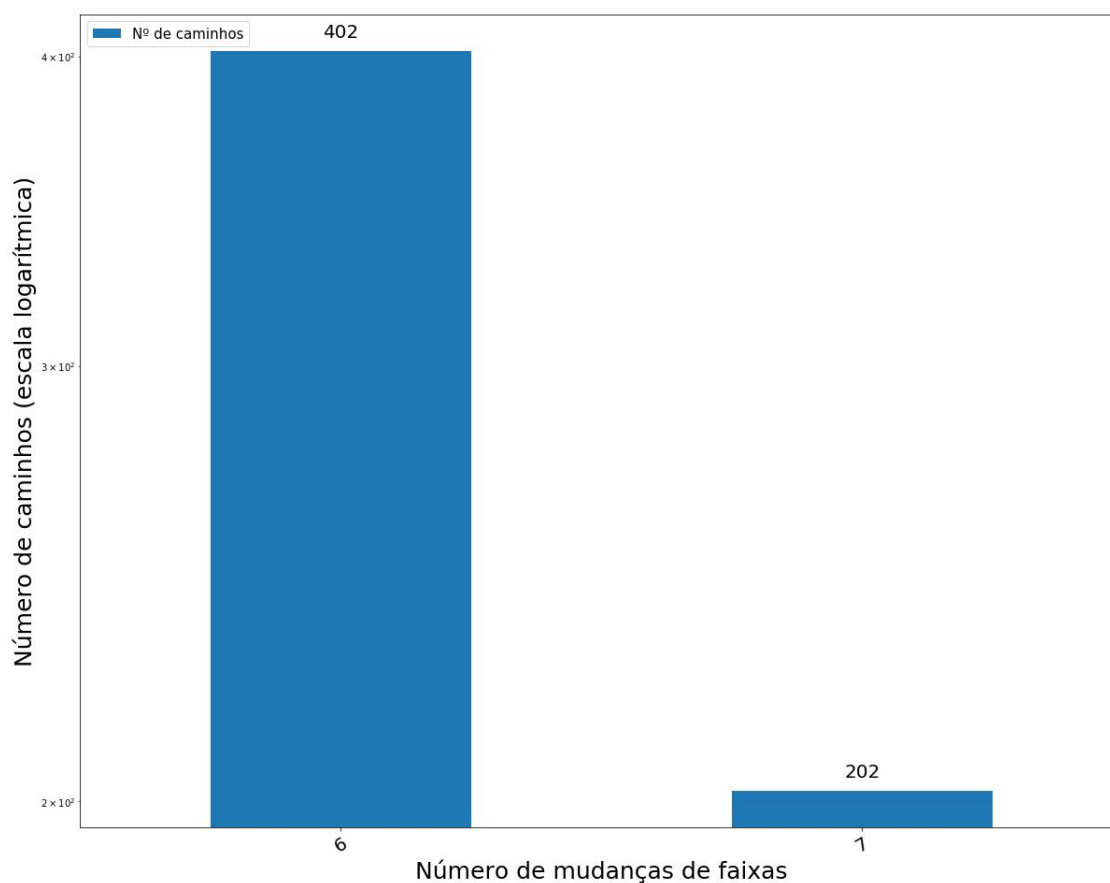
Para o Cenário 1, aferiu-se as mudanças em todos os quatro fluxos, e para o Cenário 2, mediu-se apenas o fluxo da avenida principal. As distribuições de caminhos por cada limite de trocas de faixas são vistas nas Figuras 58 e 59.

Figura 58 – Distribuição de caminhos com o modelo flexível no Cenário 1 para os caminhos com 76 passos e distribuídos por número de mudanças de faixa durante o trajeto.



Fonte: Autoria própria.

Figura 59 – Distribuição de caminhos no Cenário 2 para os caminhos com 49 passos e distribuídos por número de mudanças de faixa durante o trajeto na avenida principal.



Fonte: Autoria própria.

5.4.5 Definição de caminho

Com os caminhos gerados e restrições de segurança respeitadas, a UCI pode selecionar a sequência ideal de passagem em meio a milhares, ou até milhões de alternativas, dependendo do cenário adotado.

Foge do escopo deste trabalho definir uma estratégia de escolha do caminho a ser seguido. Entretanto, apresenta-se alguns quesitos que podem ser explorados na hora de definir, inclusive alguns de forma concomitante:

- Esforço computacional - Dado que se adota como premissa que, ao se aproximarem da zona de influência da UCI, os VAs devem indicar uma intenção de passagem e se submeter a todas as diretrizes da infraestrutura gerenciadora, é possível se pensar em uma forma de orquestrar o posicionamento de veículos para que eles assumam coordenadas mais favoráveis em relação à geração do espaço de estados.
- Desempenho Global - Pode se adotar uma escolha baseada no menor número

de rodadas, visando uma maior vazão de tráfego, mesmo que algum VA seja penalizado.

- Desempenho Individual - Como destacado anteriormente, focar só no número de passos pode ser uma análise muito fria, mas caso precise de um nível a mais de prioridade, para em casos emergenciais, pode se focar em um caminho que favoreça mais um VA do que outro.
- Qualidade de experiência - Conforme mostrado nas Figuras 58 e 59, caminhos com o mesmo T_f podem variar bastante quanto ao número de mudanças de faixa. Assim, é possível colocar esta medida como filtro.

6 CONCLUSÃO E PERSPECTIVAS

Implementar estratégias cooperativas no gerenciamento de interseções tornou-se essencial para solucionar e aprimorar o desempenho global do tráfego nas cidades e nas grandes rodovias. Alguns elementos de trânsito atualmente vigentes precisam ser reconsiderados para tentar abarcar a entrada gradual de veículos automatizados no dia a dia das pessoas. Desta forma, este trabalho surge como uma provocação às áreas de pesquisas de sistemas inteligentes de transporte e traz um novo olhar em relação ao processo de tomada de decisão em um sistema de reservas de recursos compartilhados em uma interseção.

A hipótese proposta, onde VAs fazem a passagem do cruzamento de forma livre, sem necessariamente seguir as regulamentações de trânsito atuais, e se restringindo às faixas convencionais, foi validada via um método exaustivo e automático de exploração de estados de um sistema de transição. Houve a geração de uma gama de alternativas de caminhos seguros em diversos cenários, com a possibilidade de escalar o modelo para outras condições não testadas neste trabalho.

Ainda que a abordagem de modelo flexível cause estranheza inicialmente, deve-se imaginar que VAs logo se tornarão uma realidade e é preciso expandir os horizontes de pesquisa para este cenário que se desenha para os sistemas de transportes. A hipótese tomou como premissa que os VAs chegarão a um ponto tão avançado de segurança, que os passageiros focarão em atividades recreativas e estarão alheios às condições externas do ambiente, pouco importando a trajetória que os VAs tomarão, desde que sejam acompanhados de movimentos suaves que não comprometam as suas experiências como usuários.

Como cada estado representa uma configuração única da interseção e cada transição de estado representa diferentes opções de movimento que os veículos podem adotar, o principal obstáculo no desenvolvimento do projeto foi a quantidade enorme de memória exigida para a obtenção do grafo de alcançabilidade por meio da ferramenta TINA. Os diferentes perfis de desempenho mostrados evidenciaram que mudanças mínimas nas características iniciais do modelo têm grande impacto na forma de exploração da interseção. Até por isso, pelo menos no formato apresentado, este processo de tomada de decisão não seria recomendado para execuções *online*, mesmo para os modelos com abstração, uma vez que análise do grafo dentro de uma ferramenta de alto nível como o Python também demanda muito esforço computacional.

Uma alternativa viável seria uma execução abrangente de várias condições iniciais e de forma *offline*, para posterior armazenamento em banco de dados. Depois de tratados, o espaço de estados pode ser reduzido apenas para elementos que representam rodadas cheias, ou seja, quando todos os veículos ainda presentes no *grid* tiveram atualizações efetivadas. A partir destas opções geradas, a UCI pode ter

um padrão de seleção das sequências de reservas com base em pré-requisitos pré estabelecidos e inclusive preparada com rotas alternativas para eventuais mudanças.

Com os exemplos gerados, pode-se citar um ponto da metodologia que podem ser reimaginado, como a necessidade de uma lista de prioridade variável e não fixa. Ademais, novas regras de movimentação e cenários podem ser averiguados, a fim de que o modelo seja adaptável à diversas circunstâncias.

Além dos ajustes e adições ao modelo, com a conclusão deste trabalho, algumas perspectivas e projeções de trabalhos futuros são propostas. A complementação deste trabalho, a partir dos resultados obtidos, pode ser feita através das seguintes contribuições:

- Implementar uma forma de se conciliar equações dinâmicas em meio à modelagem discreta.
- Orquestração das bateladas de verificação para se ter uma visão macro de fluxo contínuo.
- Utilização de técnicas de *machine learning* para identificar padrões na base de dados gerada e auxiliar na escolha de sequência de reservas.

REFERÊNCIAS

AHMANE, Mourad; ABBAS-TURKI, Abdeljalil; PERRONNET, Florent; WU, Jia; EL MOUDNI, Abdellah; BUISSON, Jocelyn; ZEO, Renan. Modeling and controlling an isolated urban intersection based on cooperative vehicles. **Transportation Research Part C: Emerging Technologies**, Elsevier, v. 28, p. 44–62, 2013.

AHN, Heejin; DEL VECCHIO, Domitilla. Safety verification and control for collision avoidance at road intersections. **IEEE Transactions on Automatic Control**, IEEE, v. 63, n. 3, p. 630–642, 2017.

ALONSO, Javier; MILANÉS, Vicente; PÉREZ, Joshué; ONIEVA, Enrique; GONZÁLEZ, Carlos; PEDRO, Teresa De. Autonomous vehicle control systems for safe crossroads. **Transportation Research Part C**, Elsevier Ltd, v. 19, n. 6, p. 1095–1110, 2011. ISSN 0968-090X. DOI: 10.1016/j.trc.2011.06.002. Disponível em: <http://dx.doi.org/10.1016/j.trc.2011.06.002>.

AMADEO, Ron. **Google's Waymo invests in LIDAR technology, cuts costs by 90 percent**. [S.l.: s.n.], set. 2017. Arstechnica. Disponível em: <https://arstechnica.com/cars/2017/01/googles-waymo-invests-in-lidar-technology-cuts-costs-by-90-percent/>.

AU, Tsz-Chiu; FOK, Chien-Liang; VISHWANATH, Sriram; JULIEN, Christine; STONE, Peter. Evasion planning for autonomous vehicles at intersections. *In*: IEEE. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. [S.l.: s.n.], 2012. P. 1541–1546.

BAIER, Christel; KATOEN, Joost-Pieter. **Principles of model checking**. [S.l.]: MIT press, 2008.

BERTHOMIEU, Bernard; BODEVEIX, Jean-Paul; FARAIL, Patrick; FILALI, Mamoun; GARAVEL, Hubert; GAUFILLET, Pierre; LANG, Frederic; VERNADAT, François. Fiacre: an intermediate language for model verification in the topcased environment. *In*: 4TH European Congress on EMBEDDED REAL TIME SOFTWARE. [S.l.: s.n.], 2008.

BERTHOMIEU, Bernard; RIBET, P-O; VERNADAT, François. The tool TINA—construction of abstract state spaces for Petri nets and time Petri nets. **International journal of production research**, Taylor & Francis, v. 42, n. 14, p. 2741–2756, 2004.

BRECHTEL, Sebastian; GINDELE, Tobias; DILLMANN, Rüdiger. Probabilistic Decision-Making under Uncertainty for Autonomous Driving using Continuous POMDPs. **17th International IEEE Conference on Intelligent Transportation Systems (ITSC)**, p. 392–399, 2014. DOI: 10.1109/ITSC.2014.6957722.

CHEN, Lei; ENGLUND, Cristofer. Cooperative Intersection Management : A Survey. **IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS**, p. 1–17, 2015.

DAI, Penglin; LIU, Kai; ZHUGE, Qingfeng; SHA, Edwin H-M; LEE, Victor Chung Sing; SON, Sang Hyuk. Quality-of-experience-oriented autonomous intersection control in vehicular networks. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 17, n. 7, p. 1956–1967, 2016.

DIJKSTRA, Edsger W *et al.* A note on two problems in connexion with graphs. **Numerische mathematik**, v. 1, n. 1, p. 269–271, 1959.

DRESNER, Kurt; STONE, Peter. A multiagent approach to autonomous intersection management. **Journal of artificial intelligence research**, v. 31, p. 591–656, 2008.

DRESNER, Kurt; STONE, Peter. Multiagent traffic management: An improved intersection control mechanism. *In*: PROCEEDINGS of the fourth international joint conference on Autonomous agents and multiagent systems. [S.l.: s.n.], 2005. P. 471–477.

FAGNANT, Daniel J; KOCKELMAN, Kara. Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. **Transportation Research Part A: Policy and Practice**, Elsevier, v. 77, p. 167–181, 2015.

GERLA, Mario; LEE, Eun-Kyu; PAU, Giovanni; LEE, Uichin. Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds. *In*: IEEE. 2014 IEEE world forum on internet of things (WF-IoT). [S.l.: s.n.], 2014. P. 241–246.

GIANINNI, Alessandro. **Carros semiautônomos são uma realidade cada vez mais próxima do consumidor**. [S.l.: s.n.], mar. 2021. Revista VEJA. Disponível em: <https://veja.abril.com.br/tecnologia/carros-semiautonomos-sao-uma-realidade-cada-vez-mais-proxima-do-consumidor/>.

- GONZÁLEZ, David; PÉREZ, Joshué; MILANÉS, Vicente; NASHASHIBI, Fawzi. A review of motion planning techniques for automated vehicles. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 17, n. 4, p. 1135–1145, 2015.
- GREER, L.; FRASER, J.; HICKS, D.; MERCER, M.; THOMPSON, K. **Intelligent Transportation Systems Benefits, Costs, and Lessons Learned 2018 Update Report**. Washington, 2018.
- GREGOIRE, Jean; BONNABEL, Silvere; DE LA FORTELLE, Arnaud. Priority-based intersection management with kinodynamic constraints. *In*: IEEE. 2014 European Control Conference (ECC). [S.l.: s.n.], 2014. P. 2902–2907.
- HAFNER, M. R.; CUNNINGHAM, D.; CAMINITI, L.; VECCHIO, D. Del. Cooperative Collision Avoidance at Intersections: Algorithms and Experiments. **IEEE Transactions on Intelligent Transportation Systems**, v. 14, n. 3, p. 1162–1175, set. 2013. ISSN 1524-9050. DOI: 10.1109/TITS.2013.2252901.
- HEVELKE, Alexander; NIDA-RÜMELIN, Julian. Responsibility for crashes of autonomous vehicles: an ethical analysis. **Science and engineering ethics**, Springer, v. 21, n. 3, p. 619–630, 2015.
- JIN, Q.; WU, G.; BORIBOONSOMSIN, K.; BARTH, M. Platoon-based multi-agent intersection management for connected vehicle. *In*: 16TH International IEEE Conference on Intelligent Transportation Systems (ITSC 2013). [S.l.: s.n.], out. 2013. P. 1462–1467. DOI: 10.1109/ITSC.2013.6728436.
- KALRA, Nidhi; PADDOCK, Susan. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? **Transportation Research Part A: Policy and Practice**, v. 94, p. 182–193, dez. 2016. DOI: 10.1016/j.tra.2016.09.010.
- KATRAKAZAS, Christos *et al.* Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. **Elsevier**, p. 416–442, set. 2015.
- LEE, Joyoung; PARK, Byungkyu. Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 13, n. 1, p. 81–90, 2012.

LIN, Peiqun; LIU, Jiahui; JIN, Peter J; RAN, Bin. Autonomous vehicle-intersection coordination method in a connected vehicle environment. **IEEE Intelligent Transportation Systems Magazine**, IEEE, v. 9, n. 4, p. 37–47, 2017.

LITMAN, Todd. **Autonomous Vehicle Implementation Predictions Implications for Transport Planning**. Victoria, 2018.

LOOS, Sarah M; PLATZER, André. Safe intersections: At the crossing of hybrid systems and verification. *In*: IEEE. 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC). [S.l.: s.n.], 2011. P. 1181–1186.

MAHMASSANI, Hani. Autonomous Vehicles and Connected Vehicle Systems: Flow and Operations Considerations. **Transportation Science**, p. 1–23, out. 2016.

MAKAREM, Laleh; GILLET, Denis. Model predictive coordination of autonomous vehicles crossing intersections. *In*: IEEE. 16TH International IEEE Conference on Intelligent Transportation Systems (ITSC 2013). [S.l.: s.n.], 2013. P. 1799–1804.

MARINESCU, D.; ČURN, J.; BOUROCHE, M.; CAHILL, V. On-ramp traffic merging using cooperative intelligent vehicles: A slot-based approach. *In*: 2012 15th International IEEE Conference on Intelligent Transportation Systems. [S.l.: s.n.], set. 2012. P. 900–906. DOI: 10.1109/ITSC.2012.6338779.

MÜLLER, Eduardo. **Optimal arrival time scheduling of automated vehicles at intersections**. 2018. Diss. (Mestrado) – PGEAS - UFSC.

NAUMANN, Rolf; RASCHE, Rainer; TACKEN, Jürgen. Managing autonomous vehicles at intersections. **IEEE Intelligent Systems and their Applications**, IEEE, v. 13, n. 3, p. 82–86, 1998.

NETWORKX. **Network Analysis in Python**. [S.l.: s.n.], ago. 2020. Disponível em: <https://networkx.org/documentation/stable/index.html>.

NOH, Samyeul. Decision-making framework for autonomous driving at road intersections: Safeguarding against collision, overly conservative behavior, and violation vehicles. **IEEE Transactions on Industrial Electronics**, IEEE, v. 66, n. 4, p. 3275–3286, 2018.

- ONIEVA, Enrique; MILANÉS, Vicente; VILLAGRA, Jorge; PÉREZ, Joshué; GODOY, Jorge. Genetic optimization of a vehicle fuzzy decision system for intersections. **Expert Systems with Applications**, Elsevier, v. 39, n. 18, p. 13148–13157, 2012.
- ORGANIZAÇÃO MUNDIAL DA SAÚDE. **Global Status Report on Road Safety 2018**. Geneva, 2018.
- QIAN, Xiangjun. **Model predictive control for autonomous and cooperative driving**. Dez. 2016. Diss. (Mestrado) – PSL Research University, Paris, FR.
- QIAN, Xiangjun; GREGOIRE, Jean; DE LA FORTELLE, Arnaud; MOUTARDE, Fabien. Decentralized model predictive control for smooth coordination of automated vehicles at intersection. *In*: IEEE. 2015 European control conference (ECC). [S.l.: s.n.], 2015. P. 3452–3458.
- RIOS-TORRES, Jackeline; MALIKOPOULOS, Andreas A. Automated and cooperative vehicle merging at highway on-ramps. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 18, n. 4, p. 780–789, 2016.
- RIOS-TORRES, Jackeline; MALIKOPOULOS, Andreas A. A Survey on the Coordination of Connected and Automated Vehicles at Intersections and Merging at Highway On-Ramps. **IEEE Transactions On Intelligent Transportation Systems**, p. 1–12, 2015.
- SAE INTERNATIONAL. **SAE J3016™: Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems**. Warrendale, 2018.
- SHCHETKO, Nick. **Laser Eyes Pose Price Hurdle for Driverless Cars**. [S.l.: s.n.], jul. 2014. The Wall Street Journal. Disponível em: <https://www.wsj.com/articles/laser-eyes-pose-price-hurdle-for-driverless-cars-1405969441>.
- TALEBPOUR, A.; MAHMASSANI, H.; BUSTAMANTE, Fabián. Modeling Driver Behavior in a Connected Environment: Integrated Microscopic Simulation of Traffic and Mobile Wireless Telecommunication Systems. **Transportation Research Record: Journal of the Transportation Research Board**, p. 75–86, 2016.
- TLIG, Mohamed; BUFFET, Olivier; SIMONIN, Olivier. Decentralized traffic management: A synchronization-based intersection control. *In*: IEEE. 2014

International Conference on Advanced Logistics and Transport (ICALT). [S.l.: s.n.], 2014. P. 109–114.

WU, Weigang; ZHANG, Jiebin; LUO, Aoxue; CAO, Jiannong. Distributed mutual exclusion algorithms for intersection traffic control. **IEEE Transactions on Parallel and Distributed Systems**, IEEE, v. 26, n. 1, p. 65–74, 2014.

WUTHISHUWONG, Chairit; TRAECHTLER, Ansgar. Vehicle to infrastructure based safe trajectory planning for Autonomous Intersection Management. *In*: IEEE. 2013 13th international conference on ITS telecommunications (ITST). [S.l.: s.n.], 2013. P. 175–180.

WUTHISHUWONG, Chairit; TRAECHTLER, Ansgar; BRUNS, Torsten. Safe trajectory planning for autonomous intersection management by using vehicle to infrastructure communication. **EURASIP Journal on Wireless Communications and Networking**, Springer, v. 2015, n. 1, p. 33, 2015.

ZHANG, Yue J; MALIKOPOULOS, Andreas A; CASSANDRAS, Christos G. Optimal control and coordination of connected and automated vehicles at urban traffic intersections. *In*: IEEE. 2016 American Control Conference (ACC). [S.l.: s.n.], 2016. P. 6227–6232.

ZOHDY, Ismail H; RAKHA, Hesham A. Intersection management via vehicle connectivity: The intersection cooperative adaptive cruise control system concept. **Journal of Intelligent Transportation Systems**, Taylor & Francis, v. 20, n. 1, p. 17–32, 2016.