

UNIVERSIDADE FEDERAL FLUMINENSE

EDGAR BARBOSA LIMA

**Modelos microscópicos para simulação do tráfego
baseados em autômatos celulares.**

NITERÓI

2007

UNIVERSIDADE FEDERAL FLUMINENSE

EDGAR BARBOSA LIMA

**Modelos microscópicos para simulação do tráfego
baseados em autômatos celulares.**

Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Mestre em Computação.

Orientadora:
Regina Célia Paula Leal Toledo

NITERÓI

2007

Modelos microscópicos para simulação do tráfego baseados em autômatos celulares.

Edgar Barbosa Lima

Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Mestre.

Aprovada por:

Profa. D.Sc. Regina Célia de Paula Leal Toledo / IC-UFF
(Presidenta)

Prof. D.Sc. Nelson Francisco Ebecken / COPPE-UFRJ

Prof. D.Sc. Mauricio Kischinhevsky / IC-UFF

Niterói, 29 de agosto de 2007.

*Dedico este trabalho aos meus pais,
que sempre me estimularam
a perseguir meus sonhos.*

“O rio atinge seus objetivos
porque aprendeu a contornar
obstáculos”. - *Lao Tsé*

Agradecimentos

Agradeço a minha orientadora Regina Célia de Paula Leal, que me fez superar minhas dificuldades e explorar ao máximo minhas qualidades.

Agradeço aos meus professores da graduação que me estimularam a continuar meus estudos. Aos meus amigos de graduação que me seguiram até o mestrado (Augusto, Diego, Marcio e Ildenir) e aqueles, que mesmo seguindo outros caminhos, sempre estiveram presentes nessa caminhada (Arnaldo, Fábio-Pará, Darlison Fernando-Formiga, Joana, Elaine, Vinícios-Vela, Beth e outros).

Agradeço também aos amigos que encontrei na UFF (Kennedy, Jaques, Sanderson, Warleitoca, Rafael-Guto, Renata, Stênio, Alexandre, Edwin, Helder, Jhonny, etc.).

Agradeço a minha família que me apoiou nos momentos difíceis e comemorou com minhas vitórias. Ao meu pai que me incentivou a buscar vôos mais altos e especialmente a minha mãe onde sempre busquei forças para continuar.

Resumo

Neste trabalho, propõe-se um modelo microscópico baseado em autômatos celulares para simular o comportamento do tráfego de veículos em uma auto-estrada. Para atualizar a velocidade de cada veículo considera-se, além do espaço disponível para seu deslocamento em um certo instante de tempo, o possível movimento do veículo à sua frente. Um procedimento iterativo é utilizado para tornar o algoritmo explícito no tempo e evitar colisões irreais causadas pelo procedimento antecipatório adotado, caso algum veículo se desloque menos que o estimado. Um coeficiente é utilizado para simular diferentes comportamentos dos motoristas ao dirigir. Além disso propõe-se um esquema multi-célula possibilitando a consideração tanto de diferentes incrementos da aceleração quanto diferentes tipos de veículos. Um procedimento de mudança de faixa também é utilizado. Resultados são apresentados e comparados com os descritos na bibliografia.

Palavras-chave: autômatos celulares, simulação microscópica do tráfego, modelo antecipatório, modelo multi-célula, modelo multi-faixa.

Abstract

This work proposes a microscopic model based on cellular automaton in order to simulate the behavior of vehicle traffic in a highway. Aiming towards updating the velocity of each vehicle, we consider the available space for a vehicle displacement in a certain instant as well as the possible movement of the vehicle in front of it. The algorithm proposed is time explicit and an iterative procedure is used. Such iterative procedure was developed to avoid unreal collisions. These collisions occur when vehicles move less than expected as well as because of the adopted anticipatory scheme. A coefficient is used in order to simulate different behaviors of the drivers. Moreover, the algorithm proposed was expanded by a multicell scheme. This multicell scheme permits considering several acceleration increments and also different types of vehicles. Furthermore, a procedure of lane change is used. Results are produced and compared with the ones described in the literature with different anticipation behaviors.

Keywords: cellular automata, microscopic traffic simulation, anticipatory model, multicell, multi-lane.

Sumário

Lista de Figuras	x
1 Introdução	1
2 Alguns conceitos sobre teoria de fluxo do tráfego em auto-vias	3
2.1 Conceitos e definições	3
2.2 Diagramas	6
2.2.1 Diagrama fundamental	7
2.2.2 Fluxo-velocidade	8
2.2.3 Densidade-velocidade	9
2.3 Alguns modelos para simulação do tráfego veicular	9
2.4 Modelos microscópicos	10
3 Autômatos Celulares	13
3.1 Definição	13
3.1.1 Espaço e dimensão de um autômato celular	14
3.1.2 Vizinhaça e contorno.	15
3.1.3 Estado de uma célula e regras de transição	16
3.2 Autômato celular unidimensional	16
3.3 Jogo da vida	17
4 Modelos em autômatos celulares para simulação do tráfego veicular	21
4.1 Modelagem do tráfego por autômatos celulares	21

4.2	Simulação do detector de movimento	23
4.3	Modelos determinísticos	24
4.3.1	Modelo regra 184	24
4.3.2	Modelo Fukui-Ishibashi	26
4.4	Modelos probabilísticos	27
4.4.1	O modelo NaSch	27
4.4.2	Modelos <i>slow-to-start</i>	29
4.4.2.1	Modelo <i>velocity dependent randomization</i> (VDR)	29
4.4.2.2	Modelo Takayasu-Takayasu (TT)	30
4.4.2.3	Modelo BJH	31
4.4.3	Modelo de ajuste de fluxo	32
4.4.3.1	Modelo H. Emmerich e E. Rank	33
5	Modelo de antecipação proposto	35
5.1	Proposta de antecipação	35
5.1.1	Escolha de α	36
5.2	Evitando colisões irreais	37
5.2.1	Definição dos <i>clusters</i>	39
5.2.2	Custo computacional	40
5.3	Simulações	41
6	Modelo multi-célula	46
6.1	O modelo de Helbing e Schreckenberg(HS)	47
6.2	Modelo luz de freios	48
6.3	O modelo de Kerner, Klenov e Wolf	50
6.4	Modelo proposto com multi-célula	52
6.4.1	Simulações	53

7	Modelo multi-faixas	58
7.1	Mudança de faixas	58
7.1.1	Inversão do uso de faixa	59
7.2	Efeito ping-pong	60
7.3	Modelos de mudanças de faixas	60
7.3.1	Modelo multi-faixas com compensação de velocidade	62
7.3.2	Modelo luz de freio multi-faixas.	63
7.4	Modelo proposto	66
7.5	Simulações realizadas	69
7.5.1	Simulação 1	69
7.5.2	Simulação 2	71
8	Conclusões	73
	Referências	75
	Apêndice A - Código computacional.	78

Lista de Figuras

2.1	Diagrama espaço-tempo.	4
2.2	Média espacial.	5
2.3	Média temporal.	6
2.4	Diagrama fundamental do tráfego	7
2.5	Efeito de histerese do tráfego.	8
2.6	Esquema teórico da relação fluxo-velocidade [1].	9
2.7	Esquema teórico da relação densidade-velocidade [1].	9
2.8	Esquema do modelo de perseguição.	10
3.1	Dimensões de um autômato celular.	14
3.2	Diferentes topologias de células de autômatos celulares bidimensionais.	15
3.3	Tipos mais usuais de vizinhanças para AC bidimensionais.	15
3.4	Condições de contorno	16
3.5	Vizinhança de um AC unidimensional.	17
3.6	Regra 30.	17
3.7	Simulação da regra 30.	18
3.8	Comportamentos estáveis do jogo da vida.	19
3.9	Figuras oscilatórias do jogo da vida.	19
3.10	Naves espaciais.	19
3.11	Estruturas simples que desaparecem depois de vários passos.	20
3.12	Configuração de crescimento ilimitado.	20
4.1	Regra 184.	25
4.2	Diagrama fundamental do modelo 184.	25

4.3	Diagrama fundamental do Fukui-Ishibashi.	26
4.4	Diagrama fundamental do Fukui-Ishibashi para $v_{max} \rightarrow \infty$	27
4.5	Modelo NaSch com diferentes valores de p	28
4.6	Comparação entre resultados do modelo NaSch com $p = 0,40$ com dados do tráfego real [2].	29
4.7	Modelo VDR com $p_s = 0,75$ e $p_m = 0,01$	30
4.8	Modelo TT com $p_s = 0,75$ e $p_m = 0,01$	31
4.9	Modelo BJH com $p_s = 0,75$ e $p_m = 0,01$	32
4.10	Modelo Emmerich e Rank para diferentes valores de p	34
5.1	Representação da distância efetiva.	36
5.2	Comportamentos	37
5.3	Técnica da rejeição.	38
5.4	Distribuição de comportamentos.	38
5.5	Região da pista que deve ser recalculada.	40
5.6	Número médio de reavaliações da velocidade por densidade.	41
5.7	Baixa agressividade.	43
5.8	Média agressividade.	44
5.9	Alta agressividade.	45
6.1	Veículo ocupando duas células.	46
6.2	Diagrama fundamental do modelo HS [3].	48
6.3	Diagrama fundamental do modelo luz de freio [4].	50
6.4	Diagrama fundamental do modelo KKW [5].	52
6.5	Multi-célula: simulação 1, $p = 0,35$	54
6.6	Multi-célula: simulação 1, $p = 0,50$	55
6.7	Multi-célula: simulação 1, $p = 0,70$	56
6.8	Multi-célula: simulação 2, $p = 0,35$	57

7.1	Fenômeno de inversão de pista.	60
7.2	Variáveis: veículos da mesma faixa.	61
7.3	Variáveis: veículos da faixa à direita.	62
7.4	Variáveis: veículos da faixa à esquerda.	62
7.5	Resultados do modelo multi-faixa com compensação da velocidade [6].	64
7.6	Representação da variável <i>gap</i>	65
7.7	“Distância temporal”.	65
7.8	Resultados do modelo luz de freio multi-faixa [7].	66
7.9	Distância efetiva à frente.	67
7.10	Distância efetiva à esquerda.	67
7.11	Distância efetiva à direita.	69
7.12	Frequência de mudanças de faixa.	70
7.13	Distribuição do uso das faixas.	71
7.14	Diagrama fundamental para as duas faixas da pista.	71
7.15	Exemplo de acidente em uma das pistas.	72
A.1	Exemplo de arquivo de entrada de dados.	79
A.2	Fluxograma.	82

Capítulo 1

Introdução

O contínuo aumento do tráfego rodoviário vem se tornando um dos principais problemas da sociedade moderna, devido ao grande impacto gerado pela saturação da malha viária, refletindo diretamente na diminuição da qualidade de vida da população. Nesse contexto, a simulação computacional pode ter um papel importante, possibilitando avaliar diferentes políticas para controle e gerenciamento do tráfego. Nessas simulações os modelos microscópicos, baseados em autômatos celulares, vêm se destacando dentre os principais modelos computacionais de simulação de tráfego veicular, devido a sua capacidade de representar as principais características do tráfego, com a utilização de regras simples [8, 9, 10, 11, 12]. Na Alemanha, por exemplo, utiliza-se uma modelagem baseada em autômatos celulares para prever a situação do tráfego na próxima hora, a partir dos dados obtidos por sensores espalhados pela rede de estradas do país [13].

Um dos principais modelos de autômatos celulares probabilísticos é o modelo proposto por Nagel e Schreckenberg [14], chamado de modelo NaSch. Nesse modelo pioneiro, espaço e velocidade de cada veículo são atualizados sincronamente baseados apenas na informação da sua velocidade e do espaço à sua frente no instante anterior, gerando com isso um resultado conservador no que diz respeito à relação fluxo-densidade. Uma das razões do fluxo máximo do modelo NaSch estar abaixo do desejado é que ao se calcular a velocidade de cada veículo, em um determinado instante de tempo, não se considera que o veículo à frente irá também se movimentar. Desta forma muitas vezes, o deslocamento do veículo em questão, será limitado por essa distância à sua frente e não pela velocidade que este poderia alcançar ou pela distância real.

Existem, na bibliografia, modelos propondo diferentes formas de se considerar esse movimento [4, 15, 16]. Em alguns desses modelos a velocidade de cada veículo é recalculada seqüencialmente no mesmo intervalo de tempo [2, 17], o que implica em um algoritmo implí-

cito.

Neste trabalho, baseado em um esquema de antecipação proposto em [18], propõe-se um modelo explícito com um mecanismo iterativo para evitar colisões irreais, recalculando a velocidade de um veículo sempre que existe algum risco de colisão. Para evitar um grande aumento no custo computacional do modelo proposto, o mecanismo iterativo é realizado para recalcular somente as velocidades dos veículos necessários, utilizando para isso, uma divisão dos veículos da pista em grupos, no que chamou-se de *clusters*. Para permitir diferentes acelerações e tamanhos de veículos propõe-se uma discretização multi-célula adaptando-se, o modelo proposto, para considerar essas novas características. Finalmente uma extensão da modelagem é feita para considerar uma pista com múltiplas faixas e assim simular situações mais reais, incluindo a dinâmica de mudança de faixas.

Para isso apresenta-se no Capítulo 2 uma introdução aos fundamentos da teoria de fluxo do tráfego e algumas de suas principais técnicas de modelagem. No Capítulo 3 define-se teoricamente autômatos celulares, os elementos que os compõem e alguns exemplos de aplicações dessa técnica são mostrados. A modelagem de tráfego por autômatos celulares (TCA) e alguns dos principais modelos são apresentados no Capítulo 4. No Capítulo 5 propõe-se um modelo de antecipação e seus resultados são apresentados, discutidos e comparados com outros encontrados na bibliografia. No Capítulo 6 apresentam-se tanto os principais modelos originalmente concebidos com multi-célula, como também uma adaptação para consideração de um esquema multi-célula no modelo de antecipação, anteriormente proposto. Resultados são apresentados variando-se tanto a discretização da velocidade quanto o tamanho dos veículos. No Capítulo 7 propõem-se regras de mudança de faixa e alguns resultados preliminares para validar o procedimento descrito e finalmente no Capítulo 8 são apresentadas algumas conclusões e propostas para continuidade da trabalho aqui apresentado.

Capítulo 2

Alguns conceitos sobre teoria de fluxo do tráfego em auto-vias

O bom entendimento da dinâmica do tráfego é fundamental para auxiliar na escolha da estratégia mais eficaz a ser adotada para cada tipo de problema a ser tratado. Nesse contexto, simulações que possam reproduzir o efeito que eventuais mudanças terão no tráfego, predizendo seu comportamento, podem ser fundamentais na definição de estratégias adequadas a serem adotadas para melhoria do tráfego em questão.

Nesse capítulo apresentam-se conceitos básicos da teoria de fluxo do tráfego. Primeiramente mostram-se as principais grandezas físicas envolvidas na descrição dos problemas de tráfego rodoviário, bem como diferentes enfoques para as expressões dessas grandezas. Em seguida apresentam-se os diagramas relacionais entre essas grandezas e seus comportamentos teóricos. Encerrando o capítulo, alguns modelos microscópicos de simulação do tráfego rodoviário são apresentados.

2.1 Conceitos e definições

O comportamento do tráfego pode ser avaliado por grandezas tais como: fluxo (J), dado em veículos por unidade de tempo, que é a quantidade de veículos que passam por uma seção da via em um determinado período de tempo; velocidade média (\bar{v}) dada pelo unidade de espaço percorrido por esses veículos por unidade de tempo; e a densidade (ρ) dada pelo número de veículos por unidade espaço [19].

Nesta seção apresentam-se diferentes formas de se calcular essas grandezas. Essas diferentes expressões fornecem a base necessária para sua utilização no decorrer deste trabalho.

Define-se o comportamento dessas grandezas a partir do diagrama espaço-tempo, que representa a trajetória de cada veículo ao longo do tempo, como por exemplo, os descritos na Figura 2.1, onde A e B são trajetórias válidas para veículos e C não, pois um veículo não pode estar em mais de uma posição do espaço em um mesmo instante de tempo. Como as características do tráfego variam no tempo e no espaço, é comum adotar-se valores médios para essas grandezas, sendo que estas médias podem ser temporais ou espaciais. Assim, existem diferentes expressões para essas grandezas medidas quando se considera uma seção da pista em um determinado intervalo de tempo, denominado no presente trabalho de média temporal, ou um trecho da pista em um instante de tempo, chamado de média espacial.

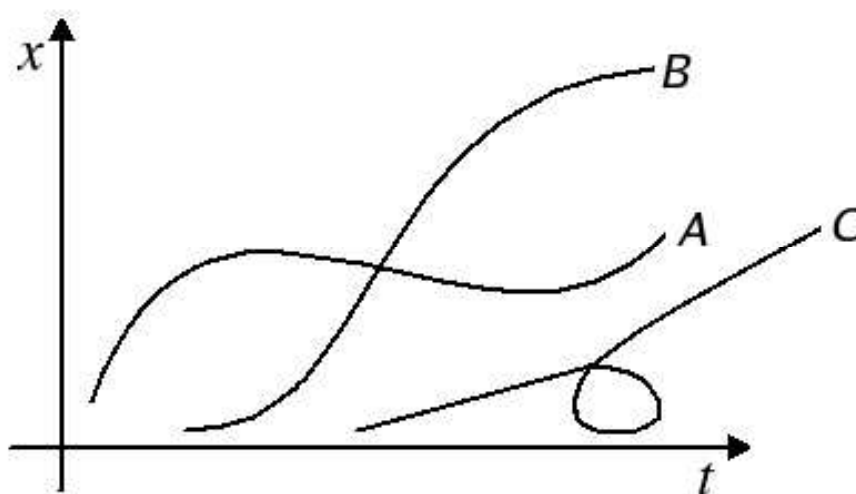


Figura 2.1: Diagrama espaço-tempo.

Médias espaciais

A definição da densidade em um trecho da pista (L) em um instante de tempo (dt), como apresentado na Figura 2.2, é o número de trajetórias (veículos) nesse trecho no instante de tempo t_i dividido pelo comprimento L considerado [20]:

$$\rho = \frac{n}{L}, \quad (2.1)$$

com n sendo o número de veículos presentes nesse trecho L .

A velocidade média dos veículos nesse trecho da via pode ser expressa pela seguinte equação:

$$\bar{v} = \frac{\sum_{i=1}^n v_i}{n}, \quad (2.2)$$

onde v_i é a velocidade instantânea do i -ésimo veículo no trecho considerado. É fácil verificar que em regime permanente a velocidade pode também ser dada por:

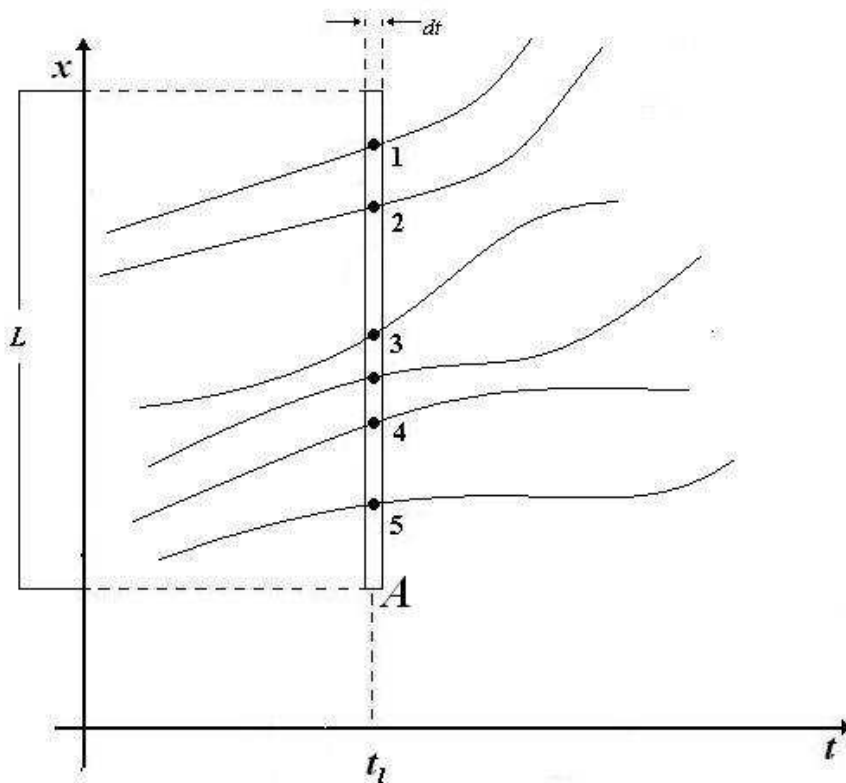


Figura 2.2: Média espacial.

$$v = \frac{J}{\rho}, \quad (2.3)$$

ou seja:

$$J = \rho v. \quad (2.4)$$

Como nesse caso considera-se a velocidade dos veículos em uma seção (\bar{v}), tem-se:

$$J = \rho \bar{v}, \quad (2.5)$$

Substituindo as Equações 2.1 e 2.2, na Equação 2.5, a equação do fluxo pode ser escrita como:

$$J = \frac{\sum_{i=1}^n v_i}{L}. \quad (2.6)$$

Médias temporais

No caso anterior a informação que se dispõe é de um trecho da pista em um único instante de tempo, como em uma foto aérea. Normalmente, quando se quer obter as grandezas a partir de dados reais, utilizam-se detectores de movimento instalados em alguma seção da pista. Por

isso, faz-se necessário a definição de expressões que considerem várias medições na mesma seção da pista x_k . Dessa forma a média temporal, consiste de se considerar uma seção da via x_k em um período tempo T , como mostrado na Figura 2.3. Como definido anteriormente, o fluxo é dado pelo número de veículos que cruzam uma determinada seção da via (m) em um dado intervalo de tempo (T), ou seja:

$$J = \frac{m}{T}. \quad (2.7)$$

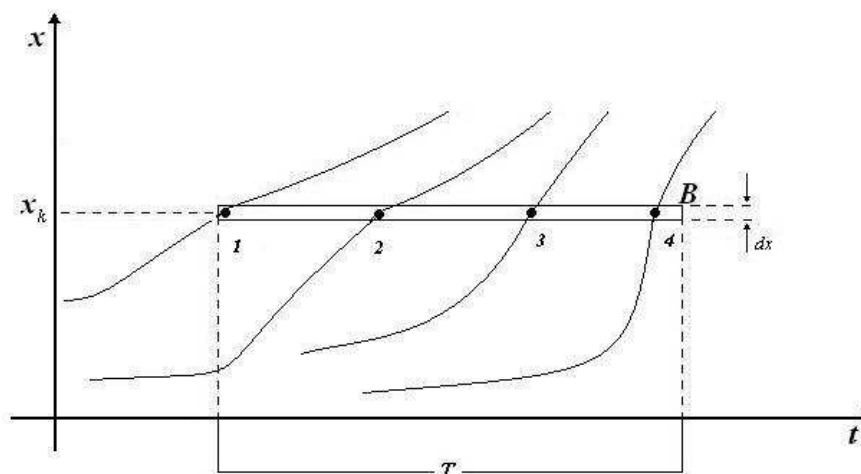


Figura 2.3: Média temporal.

Nesse caso a média das velocidades dos veículos que cruzam uma seção da via é dada por:

$$\bar{v} = \frac{\sum_{j=1}^m v_j}{m}, \quad (2.8)$$

com v_j sendo a velocidade do j -ésimo veículo a cruzar esta seção. Substituindo-se as Equações 2.7 e 2.8 na Equação 2.5 tem-se a densidade média de veículos, nessa seção da pista, dada por:

$$\rho = \frac{m^2}{T \sum_{j=1}^m v_j}. \quad (2.9)$$

2.2 Diagramas

Vários estudos empíricos têm sido realizados para entender o comportamento dos veículos em estradas, como as grandezas apresentadas na seção anterior se relacionam e como elas podem afetar uma direção confortável. Nessa seção apresentam-se os principais diagramas que descrevem o comportamento relacional dessas grandezas no tráfego real, tais como o que relaciona o fluxo com a densidade, o que relaciona o fluxo com a velocidade e a densidade com a velocidade.

2.2.1 Diagrama fundamental

O diagrama que relaciona o fluxo com a densidade é o principal diagrama utilizado, devido à sua forte ligação com o nível de saturação da pista. O diagrama que representa essa relação chama-se diagrama fundamental e apresenta três diferentes fases [21] como apresentado na Figura 2.4:

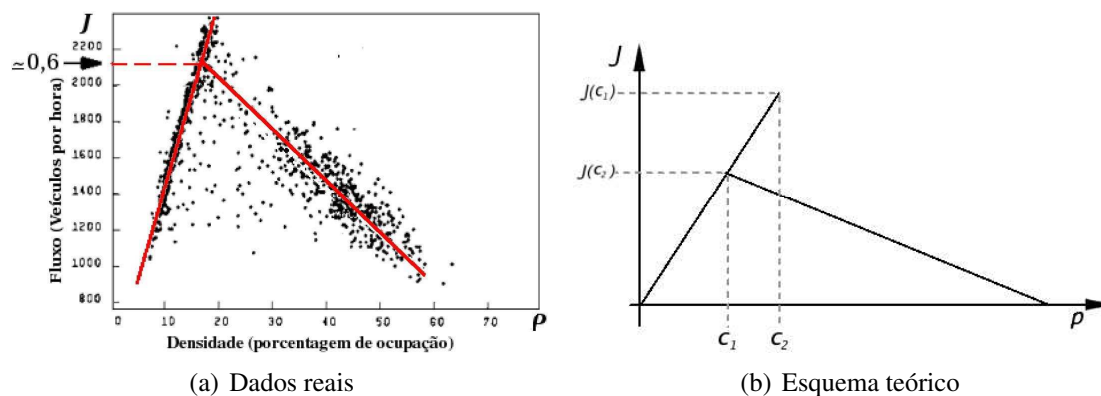


Figura 2.4: Diagrama fundamental do tráfego

(i) uma região de baixa densidade, chamada de fluxo livre. Essa fase permite que os motoristas andem na velocidade desejada e de forma geral, se aproximando da velocidade máxima permitida. Com este comportamento o diagrama fundamental apresenta um crescimento linear, com densidade variando de $0 < \rho < c_1$, como apresentado da Figura 2.4(b). Pode-se observar ainda, na Figura 2.4(a), que o fluxo máximo dessa região ocorre com aproximadamente 2100 veículos por hora ou 0,6 veículos por segundo;

(ii) uma região de média densidade, $c_1 < \rho < c_2$, Figura 2.4(b), onde o fluxo não é unicamente definido pela densidade. Nessa região a configuração do tráfego influencia diretamente no seu fluxo, podendo gerar um fluxo livre ou congestionado. Esta região de densidade intermediária é denominada região de *meta-estabilidade*;

(iii) uma região de alta densidade, com $\rho > c_2$, onde o fluxo decai à medida que a densidade aumenta. Nesta região a grande concentração de veículos faz com que estes se agrupem em engarrafamentos e um veículo que deixa um destes engarrafamentos encontra um outro logo à frente. Este tipo de tráfego é caracterizado pelo comportamento *para e anda*.

A região de *meta-estabilidade*, de média densidade, é caracterizada por um efeito de histerese do tráfego. Numa primeira fase os motoristas conseguem manter suas velocidades com $\rho > c_1$ (sentido *a*, Figura 2.5). Esse comportamento se mantém até uma determinada densidade ($\rho \leq c_2$) quando a distância entre os veículos deixa de ser suficiente para seu deslocamento. A

partir desse instante, quando um motorista desacelera, faz com que os veículos que o seguem também desacelerem gerando instantaneamente um engarrafamento (sentido *b*, Figura 2.5). Por outro lado, sentido *c* da Figura 2.5, se o fluxo caminha de engarrafado para livre ou seja, a densidade diminui, o fluxo aumenta linearmente até que se torne um fluxo livre (ponto *d* da Figura 2.5). Este efeito se dá devido ao fato conhecido, de que o fluxo de saída de um engarrafamento é baixo em relação ao de entrada.

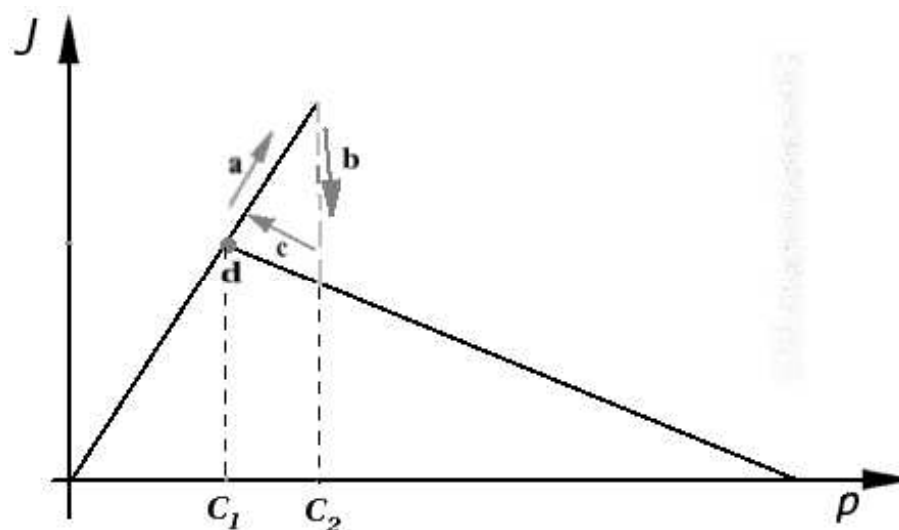


Figura 2.5: Efeito de histerese do tráfego.

2.2.2 Fluxo-velocidade

Outra relação importante é a do fluxo com a velocidade apresentado na Figura 2.6, onde V_f é a velocidade média final, ou seja, é um valor um pouco abaixo da velocidade máxima da pista já que em um fluxo livre nem todos os veículos estão o tempo todo na velocidade máxima permitida, e V_0 é a velocidade que confere ao tráfego o fluxo máximo J_{max} , chamada de velocidade ótima.

Pode-se observar, na Figura 2.6 que, tanto em alta velocidade quanto em baixa velocidade o fluxo é baixo. Isso ocorre porque, quando os veículos estão em baixa velocidade, formam-se engarrafamentos, que geram um baixo fluxo de veículos, apesar da densidade ser alta. Quando os veículos estão em alta velocidade significa que existe pouca interação entre eles e consequentemente a densidade é baixa, o que gera um pequeno fluxo. Por essa razão, o fluxo máximo (J_{max}) ocorre com uma velocidade abaixo da máxima, em um ponto de maior densidade de veículos, mas antes que comece aparecer engarrafamentos.

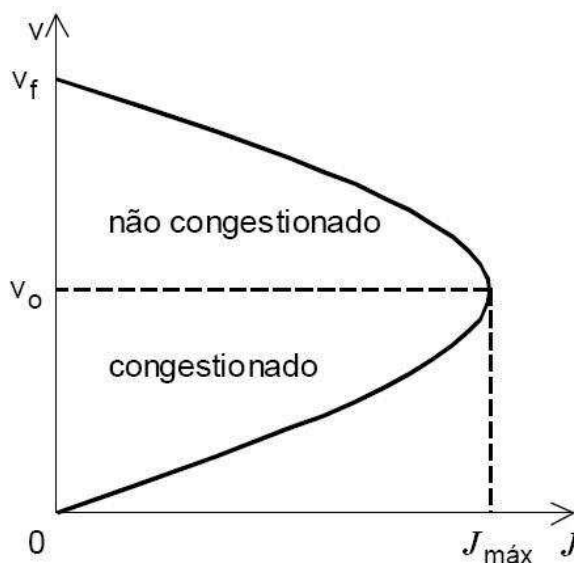


Figura 2.6: Esquema teórico da relação fluxo-velocidade [1].

2.2.3 Densidade-velocidade

Esse diagrama, apresentado na Figura 2.7, mostra como a velocidade diminui com a saturação da via. Devido a seu comportamento monotonamente decrescente, alguns modelos matemáticos representam esse diagrama como uma função linear.

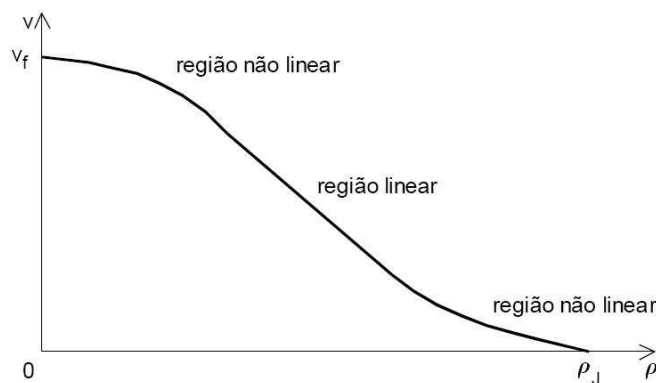


Figura 2.7: Esquema teórico da relação densidade-velocidade [1].

2.3 Alguns modelos para simulação do tráfego veicular

Os modelos matemáticos para simulação de tráfego [21] podem ser divididos em duas abordagens básicas:

- a macroscópica, descreve o comportamento das correntes de tráfego. As análises macros-

cópicas do tráfego baseiam-se na consideração de que as correntes de tráfego são meios contínuos. Para estudar seu comportamento a abordagem macroscópica aplica Leis da Hidrodinâmica, motivo pelo qual a abordagem é conhecida também como Analogia Hidrodinâmica do Tráfego. Por suas características e considerações, os modelos macroscópicos aplicam-se com sucesso ao estudo de tráfego com grande densidade, mas não se prestam facilmente às situações de tráfego rarefeito [22];

- a microscópica, se baseia no estudo da interação entre dois veículos consecutivos numa corrente de tráfego;

Apresenta-se na próxima seção uma breve descrição de modelos microscópicos utilizados, que fornecem uma base para os desenvolvimentos realizados nesse trabalho.

2.4 Modelos microscópicos

Um dos tipos de modelo microscópico de simulação do tráfego mais estudado é o modelo de perseguição (*car following*) que foi desenvolvido no final dos anos 1950s e busca traduzir a variação de velocidade de um veículo, chamado perseguidor, como resposta ao estímulo representado pela diferença de velocidade entre ele e o veículo que se desloca à sua frente, chamado veículo líder.

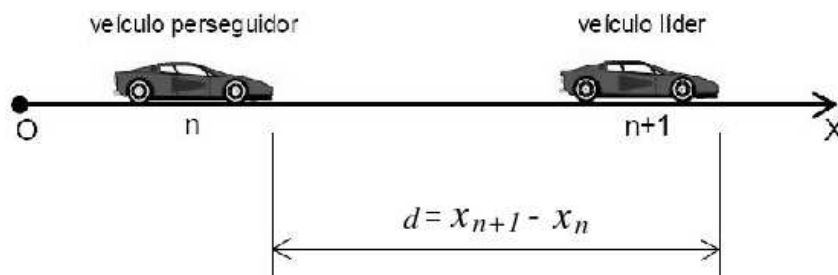


Figura 2.8: Esquema do modelo de perseguição.

Em geral os modelos de perseguição se baseiam na seguinte relação [23]:

$$[\text{Resposta}]_n \propto [\text{Estímulo}]_n \quad (2.10)$$

para o n -ésimo veículo ($n = 1, 2, \dots$). Esse veículo só poderá acelerar ou desacelerar como resposta a diferentes condições do fluxo em questão. As equações dos modelos de perseguição podem levar em consideração a velocidade do veículo analisado (v_n), a diferença de velocidade entre ele e o veículo líder (Δv_n) e a distância entre estes veículos (d_n), dentre outras. Em geral tem-se:

$$\ddot{x}_n(t) = f(v_n(t), d_n(t), \Delta v_n(t)) \quad (2.11)$$

A função f é a função estímulo que assume diferentes interpretações para cada modelo de perseguição. O modelo mais simples é o *seguindo o líder* (*follow-the-leader*) [24] que faz uso somente da diferença entre as velocidades entre o veículo perseguidor e o veículo líder, sendo representado pela seguinte equação:

$$\ddot{x}_n(t) = \frac{1}{\tau}(\dot{x}_{n+1}(t) - \dot{x}_n(t)) \quad (2.12)$$

onde τ é o parâmetro de configuração da escala de tempo. Podemos interpretar $1/\tau$ como sendo a sensibilidade do veículo à variações de velocidade do veículo líder.

Um outro modelo considera que a velocidade no tempo $t + T$ deve ser dependente do estímulo sofrido por ele no tempo t , sendo T esse tempo de resposta [25]. Utilizando-se deste fato, a Equação 2.12 é reescrita como:

$$\ddot{x}_n(t + T) = \frac{1}{\tau}(\dot{x}_{n+1}(t) - \dot{x}_n(t)) \quad (2.13)$$

Esses modelos têm a característica de acelerar ou desacelerar o veículo linearmente em função da velocidade do veículo líder, sem levar em consideração a distância entre eles, ou seja, os veículos lentos são arrastados pelos veículos rápidos. Esse comportamento é diferente do observado em dinâmica de grupos de veículos no tráfego real. Uma solução para esse problema é considerar-se que, quando um veículo está muito próximo do veículo líder, ele é mais sensível às variações de velocidade [26], o que pode ser representado pela seguinte equação

$$\ddot{x}_n(t + T) = K(\dot{x}_{n+1}(t) - \dot{x}_n(t)) \quad (2.14)$$

onde

$$K = \frac{k}{x_{n+1}(t) - x_n(t)} \quad (2.15)$$

sendo k uma constante. Assim, neste modelo, quanto mais próximo um veículo está do outro, $x_{n+1}(t) \simeq x_n(t)$, mais sensível este será em relação à variações de velocidade, simulando uma maior atenção dos motoristas quando em pequenas distâncias.

Outro tipo de modelo de perseguição é o chamado *modelo de velocidade ótima* [21] que parte de uma reformulação da Equação 2.13 e representa a variação da velocidade a partir de uma função chamada de “velocidade desejada”:

$$\ddot{x}_n(t + T) = \frac{1}{\tau}(V_n^d(t) - v_n(t)) \quad (2.16)$$

onde $V_n^d(t)$ é a velocidade desejada para o n -ésimo veículo no tempo $t + T$. Em todos os modelos *seguindo o líder* a distância entre os veículos é escolhida em função da velocidade do veículo líder, o que leva a uma velocidade desejada igual à velocidade do líder, ou seja, $V_n^d(t) = \dot{x}_{n+1}$. Uma possível alternativa à dependência somente da velocidade do veículo líder [27] é se fazer V_n^d dependente também da distância ou seja, $V_n^d(t) = V^o(d_n(t))$, e a Equação 2.16 é reescrita como:

$$\ddot{x}_n(t + T) = \frac{1}{\tau} (V^o(d_n(t)) - v_n(t)) \quad (2.17)$$

onde $V^o(d)$ é a velocidade otimizada para o veículo n e em geral $V^o(0) \rightarrow 0$ quando $d \rightarrow 0$ e para a velocidade máxima (v_{max}) quando $d \rightarrow \infty$. Por exemplo, definindo-se:

- (i) distância de segurança como sendo a menor distância que pode haver entre veículos consecutivos sem que haja risco de acidentes, ou seja, se a distância entre os veículos se tornar menor que a distância de segurança, o veículo perseguidor deve diminuir sua velocidade;
- (ii) distância de interação como a maior distância em que um veículo poderá influenciar na decisão da velocidade de outro;

a expressão de $V^o(d)$ pode ser dada como em [28]:

$$V^o = \begin{cases} 0 & \text{para } d < d_A \\ f(d) & \text{para } d_A < d < d_B \\ v_{max} & \text{para } d_B < d \end{cases} \quad (2.18)$$

onde d_A é a distância de segurança, d_B é a distância de interação entre os veículos e $f(d)$ é uma função contínua onde $f(d) \rightarrow 0$ quando $d \rightarrow d_A$ e $f(d) \rightarrow v_{max}$ quando $d \rightarrow d_B$. Sendo assim, um veículo que tenha uma distância maior que d_B à sua frente, poderá andar na velocidade máxima.

Capítulo 3

Autômatos Celulares

Um autômato celular (AC) é um modelo matemático discreto que tenta simular comportamentos complexos a partir de regras simples. Nesse tipo de modelagem o espaço é discretizado em uma grade regular de células, onde cada uma dessas células é um autômato, ou seja uma máquina de estados, que só pode assumir um número finito de estados.

Nos 1950's John von Neumann propôs um dos primeiros modelos de autômato celular, que consistia de um modelo matemático abstrato que simulava as rotinas de um robô que podia se replicar a partir de um conjunto de peças separadas. Este modelo ficou conhecido como *universal copier and constructor* e é baseado numa grade de duas dimensões onde cada célula dessa grade podia estar em um conjunto de 29 estados [29, 30].

Nos anos 70 os autômatos celulares se popularizaram graças ao modelo bidimensional de dois estados proposto por John Conway, chamado de jogo da vida (*game of life*) [31].

Nos anos 80 e 90, Stephen Wolfram estudou autômatos celulares unidimensionais utilizando mecanismos determinísticos simples. A grande complexidade dos resultados por ele encontrada, o levou a concluir que mecanismos similares poderiam modelar fenômenos complexos. Esses resultados culminaram no livro lançado em 2001 chamado *A New Kind of Science* [32], onde ele descreve sistematicamente seus experimentos e suas conclusões.

Nesse capítulo apresentam-se as definições básicas de autômatos celulares e alguns de seus modelos clássicos.

3.1 Definição

Seja:

- L é uma grade regular, d -dimensional, formada de células;

- S é um conjunto finito de estados;
- N , a vizinhança, é um conjunto finito, de tamanho $|N| = n$ e dimensão d , tal que para todo $c \in L \Rightarrow N(c) \in L$;
- $f : (S, N) \rightarrow S$ uma função de transição.

Chama-se o conjunto (L, S, N, f) de um autômato celular, onde L representa a discretização do espaço e S é o conjunto dos diferentes estados em que cada célula pode estar. A vizinhança N é o conjunto de células que influenciam na definição do novo estado da célula considerada e a função f , chamada de regra de transição que, a partir da vizinhança e do estado da célula, define seu próximo estado. Além desses 4 elementos, a definição da condição de contorno e a vizinhança destas células do contorno do domínio considerado, também são partes importantes na descrição de um autômato celular. A seguir apresentam-se mais detalhadamente cada um desses componentes de um autômato celular.

3.1.1 Espaço e dimensão de um autômato celular

Em um modelo de autômato celular o espaço é discretizado em uma grade regular de células. Define-se uma grade regular como sendo uma discretização do domínio de dimensão d , em que seus elementos, as células, preenchem o domínio completamente, e ao se transladar a grade em d direções independentes, obtém-se a mesma grade.

Essa grade pode ser de diferentes dimensões, uni, bi ou tri-dimensional, e diferentes topologias como apresentado nas Figuras 3.1 e 3.2 respectivamente.

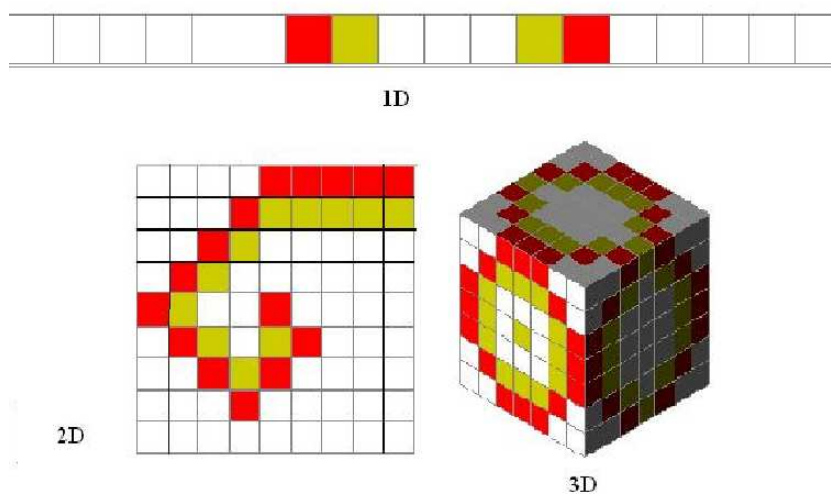


Figura 3.1: Dimensões de um autômato celular.

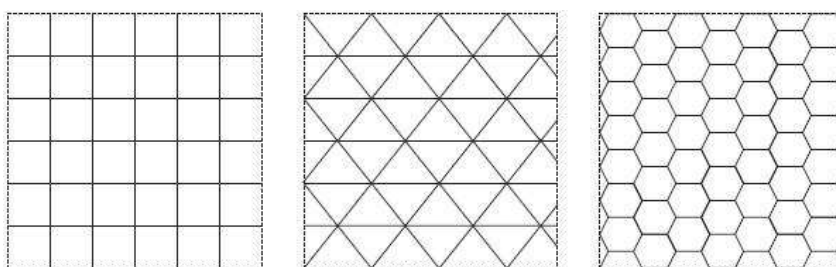


Figura 3.2: Diferentes topologias de células de autômatos celulares bidimensionais.

3.1.2 Vizinhança e contorno.

O conjunto de vizinhos da célula c é o conjunto

$$N(c) = \{i \in L; (c + i) \in N\} \quad (3.1)$$

A definição dos vizinhos de cada célula vai depender do problema a ser resolvido, levando-se em consideração as características do problema a ser tratado, escolhendo a representação mais adequada. Algumas definições de vizinhança de autômatos celulares levam em consideração o estado atual da célula a ser modificada. Portanto, a configuração da célula atual pode fazer ou não parte do argumento da função de transição. A Figura 3.3 apresenta tipos de vizinhanças usuais em autômatos celulares bidimensionais, com diferentes áreas de influência, definidas pelo raio.

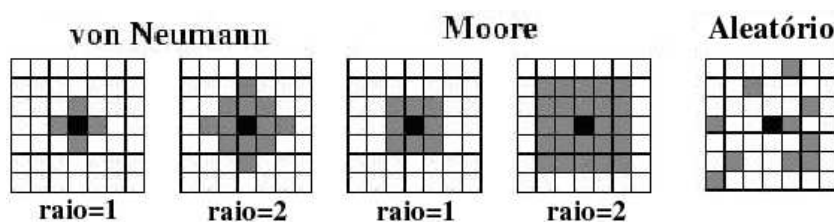


Figura 3.3: Tipos mais usuais de vizinhanças para AC bidimensionais.

Para as células que estão nas fronteiras do domínio, há necessidade de se definir sua vizinhança ou seja, as células cujos estados influenciam no resultado da aplicação da regra de transição. As principais condições de contorno encontradas na bibliografia são: condições de contorno periódicas, reflexivas e constantes.

A condição de contorno periódica, Figura 3.4(a), simula um domínio circular, como por exemplo a superfície de um toro, enquanto que a condição de contorno reflexiva, Figura 3.4(b), e a constante, Figura 3.4(c), simulam condições de contorno de Neumann e de Dirichlet, respectivamente.

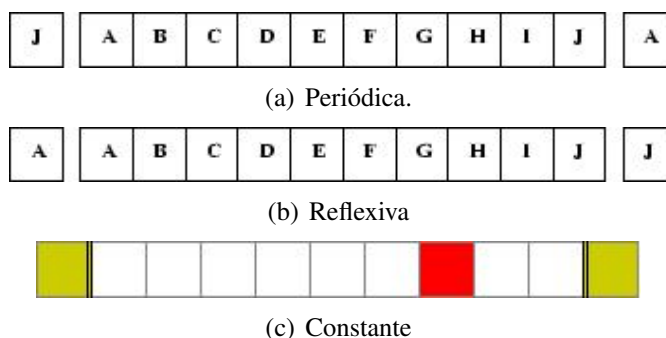


Figura 3.4: Condições de contorno

3.1.3 Estado de uma célula e regras de transição

Em um autômato celular, S é o conjunto de estados que cada célula pode assumir, sendo este um conjunto finito de qualquer tipo de dados ou informações. O valor atribuído a cada célula para iniciar a simulação é o valor inicial do problema e o estado de cada célula após alguns passos será o resultado desta simulação.

Uma configuração $C_t : L \rightarrow S$ é uma função que associa um estado a cada célula da grade. O efeito da função ou regra de transição f muda a configuração C_t para uma nova configuração C_{t+1} de acordo com a seguinte expressão:

$$C_{t+1} = f(C_t) \quad (3.2)$$

Essas funções de transição podem ser determinísticas ou probabilísticas:

- uma função de transição determinística é uma função onde, dada uma configuração para a vizinhança considerada, existe um único estado possível para célula analisada;
- uma função de transição probabilística é uma função onde as regras são definidas com a introdução de probabilidades de ocorrência, ou seja, dada uma configuração para a vizinhança existe um conjunto de possíveis de estados para célula, que serão escolhidos com base em variáveis aleatórias.

3.2 Autômato celular unidimensional

Os autômatos celulares unidimensionais em que cada célula pode assumir dois estados foram primeiramente analisados por Wolfram [32]. Esse tipo de AC apesar de ser muito simples é capaz de gerar comportamentos complexos. Esses AC têm regras de transição que consideram como vizinhança as duas células mais próximas e o estado da própria célula, Figura 3.5. Uma

vez que o conjunto estados só tem dois valores, isso fornece 2^3 configurações possíveis.

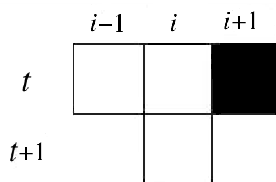


Figura 3.5: Vizinhança de um AC unidimensional.

Com isso, para definir a função de transição f , tem-se que definir uma regra para cada configuração de estados possíveis da vizinhança. Chamando-se as células pretas de 1 e as brancas de 0, tem-se que as diferentes vizinhanças podem ser representadas por um número em binário de 3 bits, ou seja, variando de $(000)_2 = (0)_{10}$, as três células brancas, até $(111)_2 = (7)_{10}$, as três células pretas. Usando desse fato cada configuração de vizinhança é normalmente representada ordenando-se essas possíveis configurações, em ordem decrescente, conforme apresentado na Figura 3.6. O número, em binário, formado pelo resultado da aplicação dessas regras fornece o nome da função. O exemplo de regra de transição mostrado na Figura 3.6, é conhecida como regra 30 pelo fato do número em binário formado pelo resultado da aplicação das regras ser o número 30, ou seja, $(00011110)_2 = (30)_{10}$.

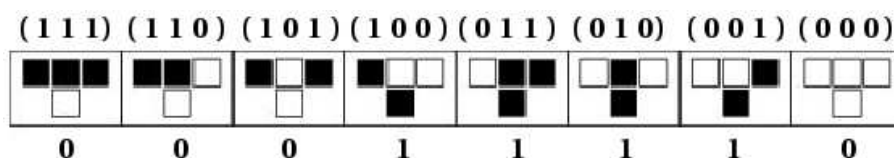


Figura 3.6: Regra 30.

A Figura 3.7 apresenta um exemplo de aplicação da regra 30 tendo como condição inicial apenas a célula central preta (valor 1) e as outras brancas (valor 0) e condições de contorno constantes e iguais a 0.

Como para cada configuração de vizinhança pode-se definir dois resultados diferentes, para a célula considerada (0 ou 1), então existe a possibilidade de se definirem $2^{2^3} = 256$ funções de transição diferentes.

3.3 Jogo da vida

Dentre os autômatos celulares bidimensionais mais conhecidos está o Jogo da Vida, desenvolvido pelo matemático John Horton Conway em 1970 [31].

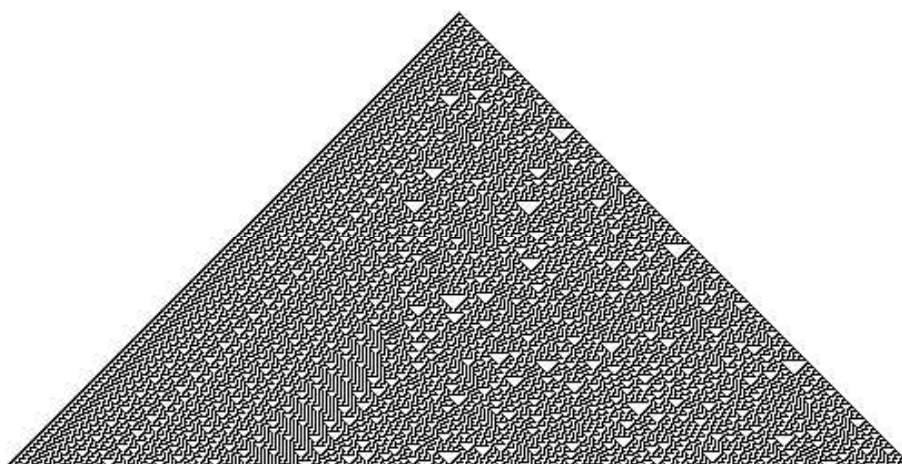


Figura 3.7: Simulação da regra 30.

O jogo da vida, *game of life*, é um autômato celular com dois estados $S = (\text{vivo}, \text{morto})$ e com vizinhança de Moore de raio 1 (Figura 3.3), com a função de transição sendo composta pelas seguintes regras:

- uma célula viva continua viva se existem 2 ou 3 vizinhos vivos;
- uma célula morta vive se existem 3 vizinhos vivos;
- uma célula morre se existem mais de 3 ou menos de 2 células vivas.

Alguns comportamentos podem ser observados no jogo da vida. Estes comportamentos podem ser estáticos, que não apresentam variação no decorrer do tempo; oscilatórios, que possuem um ciclo de estados; caminhantes, também conhecidos como naves espaciais, que movem-se pelo domínio com um ciclo de estados.

A Figura 3.8 apresenta alguns exemplos de configurações estáticas, como o *bloco* e o *bote*. São apresentados na Figura 3.9, oscilando entre as configurações “a” e “b” alternadamente, o *blinker* e o *sapo* e a Figura 3.10 apresenta exemplos dos caminhantes, o *Gliter* e o *LWSS*.

Além desses comportamentos também surgem no jogo da vida configurações que desaparecem depois de vários passos como os *Diehard* e *Acorn* apresentados na Figura 3.11.

É possível também encontrar configurações que crescem indefinidamente, como na Figura 3.12. Essas configurações incluem estruturas estacionárias e atiram caminhantes indefinidamente.

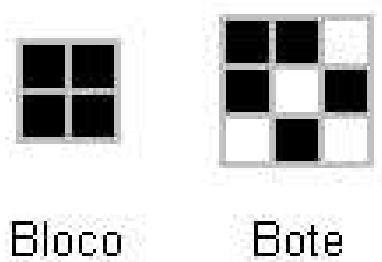


Figura 3.8: Comportamentos estáveis do jogo da vida.

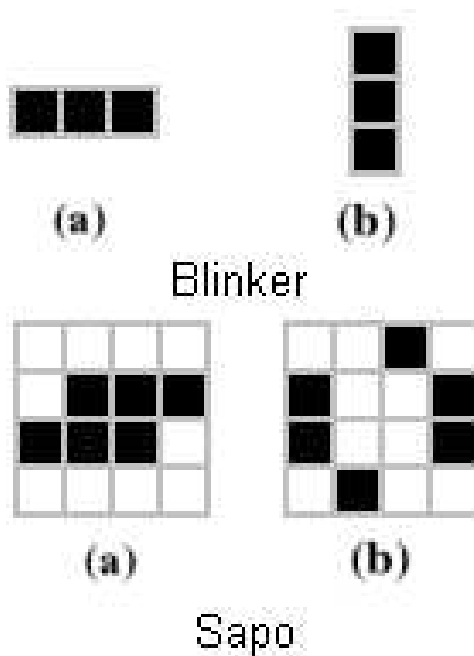


Figura 3.9: Figuras oscilatórias do jogo da vida.

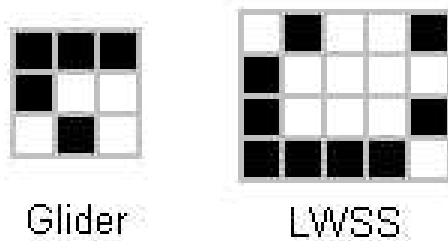


Figura 3.10: Naves espaciais.

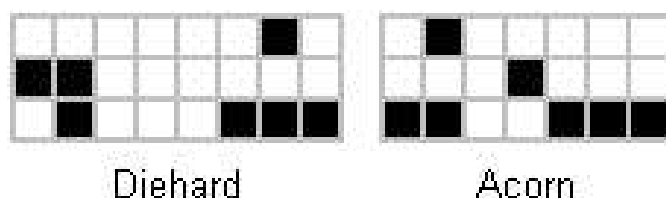


Figura 3.11: Estruturas simples que desaparecem depois de vários passos.



Figura 3.12: Configuração de crescimento ilimitado.

Capítulo 4

Modelos em autômatos celulares para simulação do tráfego veicular

Um tipo de modelagem microscópica que vem sendo utilizada amplamente para simulação de tráfego rodoviário é a baseada em autômatos celulares, devido a sua grande versatilidade e simplicidade. O primeiro modelo probabilístico que reproduziu as condições básicas do tráfego com o uso de regras de transição simples foi o proposto por K. Nagel e M. Schreckenberg em 1992 [14]. Partindo desse modelo vários autores têm proposto alterações e expansões de suas regras, para melhor reproduzir toda a variação de comportamento do tráfego.

Nesse capítulo apresenta-se primeiramente na Seção 4.1 a modelagem para o tráfego em autômato celular, seus parâmetros e definições. Em seguida apresentam-se na Seção 4.2, diferentes formas de se simular um detector de movimento. Isso se faz necessário para que se possa calcular, nas pistas simuladas, as grandezas físicas envolvidas na discussão do problema em questão, tais como fluxo e densidade. Finalmente faz-se uma revisão dos principais modelos de autômato celular para o tráfego (TCA). Para melhor entendimento estes modelos foram divididos em determinísticos, Seção 4.3, e probabilísticos, Seção 4.4, sendo os probabilísticos subdivididos em modelos *slow-to-start* e de ajuste de fluxo. Resultados destes modelos são apresentados e discutidos.

4.1 Modelagem do tráfego por autômatos celulares

Em um TCA cada faixa da estrada é discretizada como um autômato celular, com os seguintes elementos:

- L , uma grade que representa a via, onde cada célula representa um pequeno trecho da

pista;

- S , o conjunto de estados que a célula pode assumir e é composto de informações tais como: ocupação da célula, velocidade do veículo que a ocupa, tamanho do veículo, entre outros. A quantidade e o tipo de informação contida em cada célula vai depender do modelo;
- N , a vizinhança, é um conjunto de células à frente da célula avaliada, no sentido do tráfego. Essa vizinhança pode variar de acordo com o modelo adotado.
- f , a função de transição é o conjunto de regras que decidirá sobre a movimentação dos veículos, e também varia para cada modelo.

Na maioria dos trabalhos, para testar os modelos, utilizam-se condições de contorno periódicas mantendo o número de veículos fixo. Apesar de ser um modelo de via simplificado, condições realistas de tráfego podem ser simuladas baseadas nessas condições [8, 33, 34]. Além disso utiliza-se, como usual, o tamanho da célula de $7,5m$ ($tam = 7,5$), que corresponde a ocupação média de cada veículo em um engarrafamento, considerando tamanho do carro e a distância até o veículo da frente e cada período de tempo, discreto, corresponde a um segundo, que é o tempo médio de resposta de cada motorista [3]. Com isso pode-se concluir que se o veículo está com velocidade $v = 1$ célula por segundo, este veículo está a $7,5m/s$ ou $27km/h$. A maioria dos modelos encontrados na bibliografia utiliza velocidade máxima $v_{max} = 5$, o que corresponde a uma velocidade de $135km/h$.

Pode-se observar que o valor da velocidade significa o número de células que o veículo irá se deslocar. Assim, se a velocidade é $v = v_i$, por exemplo, isso significa que este veículo se movimentará v_i células, ou seja, se o veículo i está na posição x_i no tempo t (x_i^t) e sua velocidade é v_i^t , pode-se dizer que a nova posição desse veículo no tempo $t + 1$ será:

$$x_i^{t+1} = x_i^t + v_i^t \quad (4.1)$$

Para extrair as informações necessárias para o cálculo das grandezas físicas de interesse, existe a necessidade de simular um detector de movimento, que é o equipamento utilizado para colher essas informações.

4.2 Simulação do detector de movimento

Para avaliar os resultados das simulações de tráfego realizadas, há a necessidade de se medir, na pista simulada, o movimento dos veículos para que se possa gerar os diagramas relacionais de interesse e compará-los com dados reais. Para isso apresenta-se a seguir, como as expressões obtidas no Capítulo 2 são utilizados com esse tipo de discretização da via.

- Utilizando-se a média espacial, como na Seção 2.1, para o cálculo da densidade (Equação 2.1) e velocidade (Equação 2.2) tem-se:

$$\rho = \frac{N_i}{L} \quad (4.2)$$

$$v = \frac{\sum_{j=1}^{N_i} v_j}{N_i} \quad (4.3)$$

onde N_i é o número de veículos presentes em uma seção da pista de tamanho L e v_i é a velocidade do veículo i . Para a avaliação dessas grandezas ao longo do tempo deve-se considerar a densidade média ocorrida durante os T intervalos de tempo, ou seja:

$$\rho = \frac{1}{T} \sum_{i=1}^T \left(\frac{N_i}{L} \right) \quad (4.4)$$

Da mesma forma, para a velocidade tem-se:

$$\bar{v} = \frac{1}{T} \sum_{i=1}^T \left(\frac{\sum_{j=1}^{N_i} v_j}{N_i} \right) \quad (4.5)$$

e com isso pode-se calcular o fluxo médio pela expressão:

$$J = \bar{v} \cdot \rho \quad (4.6)$$

como descrito na Equação 2.5. Com isso, podem-se fazer os diagramas relacionais para a pista simulada.

- Considerando-se a média temporal, como descrita na Seção 2.1, tem-se que T é o tempo de medição, dx a seção da pista onde está o detector de movimento, que para a discretização utilizada é uma célula que chamou-se “célula detectora”. Nesse caso o fluxo é dado por:

$$J = \frac{N_p}{T} \quad (4.7)$$

onde N_p é a quantidade de veículos que passam numa determinada célula detetora em um período de tempo T . A densidade é dada por:

$$\rho = \frac{N_p^2}{T \sum_{k=1}^{N_p} v_k} + \frac{N_s}{T} \quad (4.8)$$

onde v_k é a velocidade do k -ésimo veículo a cruzar a célula detetora e N_s o número de instantes de tempo que um veículo permaneceu parado nesta célula detetora [35]. Como apresentado na Equação 2.9 a densidade é dada pela primeira parcela da Equação 4.8, e de fato é, se for considerado um fluxo livre onde não existem veículos parados, ou seja, a contribuição para a densidade média da célula detetora de cada veículo que passa por ela em um passo de tempo é de $1/(v_i T)$. Caso o veículo fique parado sobre a célula detetora, sua contribuição não pode ser calculada da mesma maneira, já que sua velocidade nesse passo é 0 (zero). Nesse caso, apesar do veículo não ter contribuído com o fluxo, ele contribui para a densidade já que este veículo ocupará a célula nesse passo de tempo, tendo uma contribuição de $1/T$ para a densidade média temporal da pista. Então é necessário somar-se a contribuição dos veículos em movimento com a dos veículos parados obtendo-se a Equação 4.8.

Neste trabalho optou-se por utilizar as Equações 4.5, 4.7 e 4.8 para cálculo da velocidade, fluxo e densidade respectivamente, uma vez que se assemelham mais a uma medição real em uma via.

A seguir apresentam-se alguns dos principais TCA's encontrados na bibliografia. Inicialmente, a título de comparação de resultados, apresentam-se dois modelos determinísticos: o modelo 184 [32] e o modelo Fukui-Ishibashi [36]. A seguir apresentam-se o modelo NaSch [14], que é um modelo probabilístico pioneiro, e uma família de modelos que tentam melhorar os resultados obtidos pelo NaSch: modelos *slow-to-start* e modelos ajuste de fluxo.

4.3 Modelos determinísticos

4.3.1 Modelo regra 184

O modelo 184 é o mais simples modelo de TCA e se baseia na regra 184 de autômato celular unidimensional. Esta regra é apresentada na Figura 4.1, onde a célula preta indica que está ocupada por um veículo e a branca, que está desocupada. Esse modelo considera somente que

um veículo move-se para célula à sua frente, se esta estiver vazia, considerando dessa forma uma velocidade máxima igual a 1 célula por segundo e pode ser representado pelo seguinte algoritmo:

Algoritmo 184:

Para todos os veículos faça

aceleração: $v_i = \min(d_i, 1)$;

atualização: $x_i = x_i + v_i$.

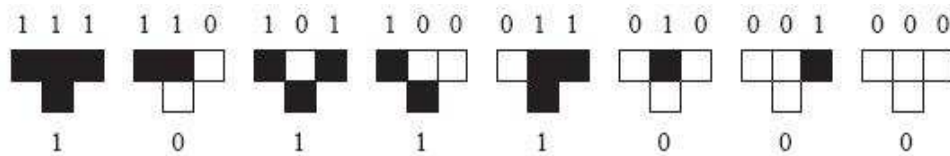


Figura 4.1: Regra 184.

onde v_i , d_i e x_i são a velocidade, o número de células vazias à frente do veículo considerado no sentido do fluxo.

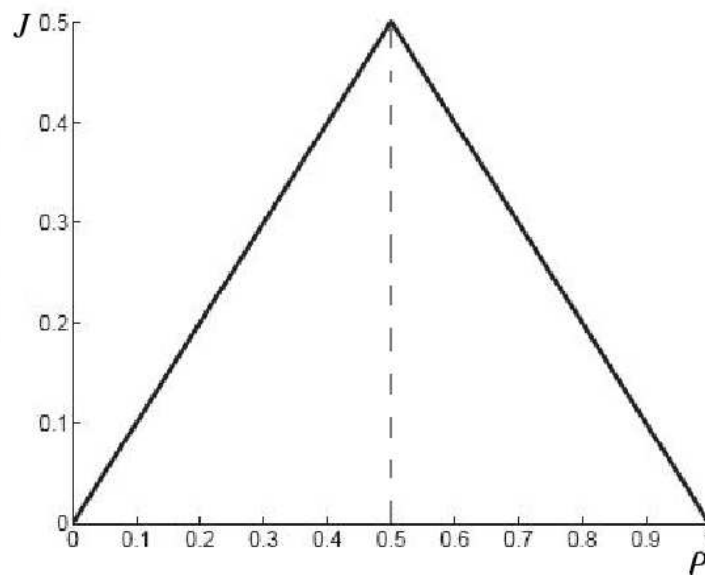


Figura 4.2: Diagrama fundamental do modelo 184.

Pode-se observar no diagrama fundamental desse modelo, Figura 4.2, duas fases distintas: uma de crescimento linear chamada de fluxo livre e outra de decrescimento linear chamada de fluxo congestionado. A mudança entre as fases ocorre exatamente quando a densidade é $\rho_0 = 0,5$.

Devido ao fato desse modelo ser determinístico e com velocidade máxima igual a 1, quando $\rho \leq 0,5$, mesmo uma configuração inicial de distribuição de veículos heterogênea, estabilizará

depois de vários passos, gerando uma configuração homogênea. Isso fará com que exista pelo menos um espaço vazio à frente de cada veículo. Nesse caso o modelo mantém o mesmo movimento para todos os veículos. Quando a densidade é maior que 0,5 então nem todos os veículos terão espaço para se mover. Logo, o fluxo cairá proporcionalmente ao número de veículos incluídos na pista.

4.3.2 Modelo Fukui-Ishibashi

Esse modelo [36] é simplesmente uma expansão do modelo 184 para considerar outras velocidades além de 0 e 1, realizando uma pequena alteração no algoritmo anterior. Chamando de v_{max} a velocidade máxima permitida, tem-se:

Algoritmo Fukui-Ishibashi:

Para todos os veículos faça

aceleração: $v_i = \min(d_i, v_{max})$;

atualização: $x_i = x_i + v_i$.

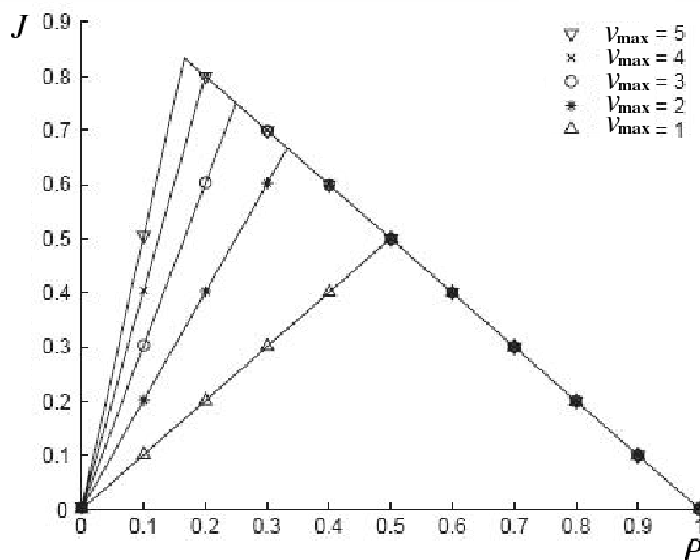


Figura 4.3: Diagrama fundamental do Fukui-Ishibashi.

Nesse modelo a divisão das duas fases se faz em $\rho_0 = \frac{1}{v_{max}+1}$ com fluxo máximo igual a $J_{max} = \rho_0 v_{max}$, onde $v_{max} + 1$ corresponde ao espaço mínimo que um veículo precisa para se deslocar na velocidade máxima, contando com a célula ocupada por ele mais v_{max} células vazias à sua frente, observado na Figura 4.3. Assim, quando a razão entre o número de células da pista e o número de veículos for menor que $v_{max} + 1$ significa que nem todos os veículos

poderão se manter na velocidade máxima, então o fluxo começa a diminuir. Um caso limite para este modelo é considerar $v_{max} = \infty$. Nesse caso $\rho_0 = 0$ e

$$J_{max} = \rho_0 v_{max} = \lim_{v_{max} \rightarrow \infty} \frac{v_{max}}{v_{max} + 1} = 1 \quad (4.9)$$

como mostrado na Figura 4.4.

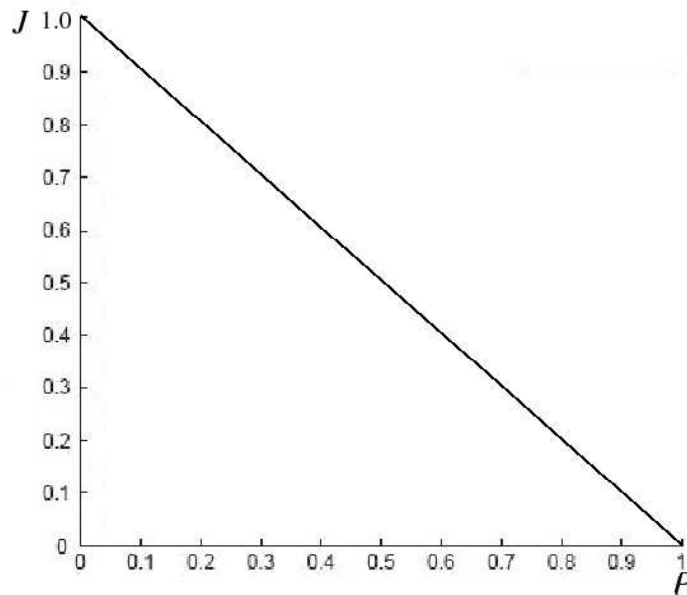


Figura 4.4: Diagrama fundamental do Fukui-Ishibashi para $v_{max} \rightarrow \infty$.

4.4 Modelos probabilísticos

Nesta seção são apresentados modelos que têm como principal característica, a adição de uma regra probabilística à função de transição, possibilitando que assumam comportamentos mais realistas.

4.4.1 O modelo NaSch

O modelo NaSch [14] é o modelo pioneiro de autômato celular probabilístico aplicado ao tráfego e parte dos seguintes princípios: todos os motoristas tentam andar na maior velocidade possível ou permitida; eventualmente, com probabilidade $p = p_m$, algum motorista não tem esse comportamento, diminuindo ou mantendo a velocidade sem nenhum motivo aparente; os motoristas regulam sua velocidade baseados na distância para o veículo da frente e na sua velocidade no instante de tempo anterior. Esses comportamentos são descritos pelo seguinte algoritmo:

Algoritmo NaSch:

Para todos os veículos faça

 aceleração: $v_i = \min(v_i + 1, v_{max});$

 frenagem: $v_i = d_i$ se $v_i > d_i$;

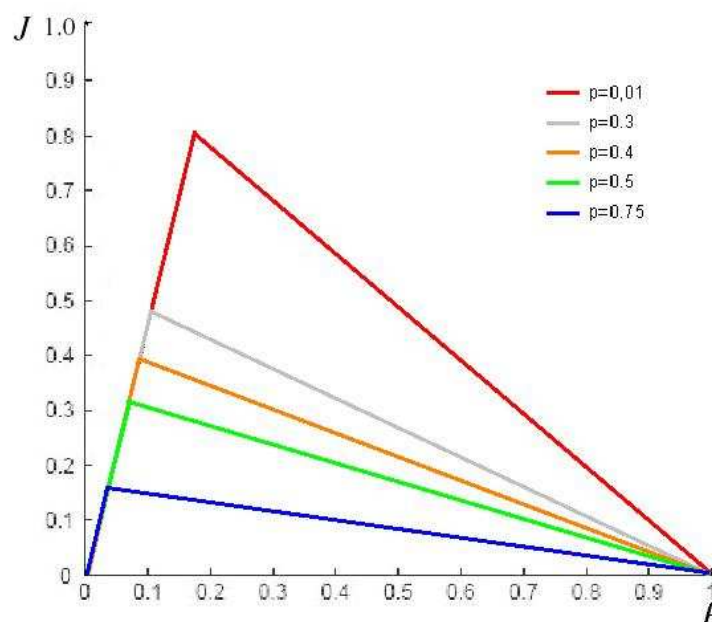
 randomização: $v_i = v_i - 1$ com probabilidade $p = p_m$ e se $v_i > 0$;

 atualização: $x_i = x_i + v_i.$

onde v_i e x_i são a velocidade e a posição do i -ésimo veículo respectivamente, v_{max} é a velocidade máxima permitida, $d_i = x_{i+1} - x_i$ é a distância do i -ésimo veículo para o veículo à sua frente e p a probabilidade do motorista alterar seu comportamento.

A Figura 4.5 apresenta os resultados para diferentes valores de p , utilizando pistas circulares com tamanho total de 300 células. O número de veículos foi incrementado de 3 em 3 chegando a 297 veículos. Em todas as simulações foram realizados 10.000 passos de tempo, descartando-se os 1.000 primeiros passos e coletando dados a cada T passos, com $T = 300$. Para se calcular as grandezas físicas de interesse utilizou-se as expressões 4.8 e 4.7 da média temporal, discutidos na Seção 4.2.

Pode-se observar que apesar da estrutura básica da dinâmica do tráfego ser corretamente reproduzida por esse modelo, ele não é suficiente para uma descrição precisa do tráfego, pois não consegue simular a meta-estabilidade e gera uma curva abaixo da observada no diagrama fundamental do tráfego real, Figura 4.6, que para valores de p entre 0,3 e 0,5 [2], devem ficar com valores próximos a 0,6 de fluxo (Figura 2.4(a)).


 Figura 4.5: Modelo NaSch com diferentes valores de p .

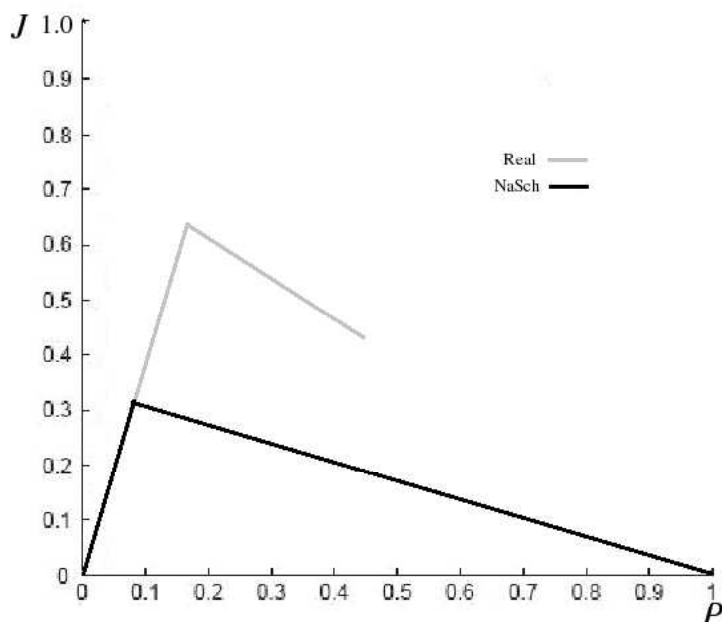


Figura 4.6: Comparação entre resultados do modelo NaSch com $p = 0, 40$ com dados do tráfego real [2].

Vários autores têm sugerido modificações no modelo NaSch, tanto tentando reproduzir a *meta-estabilidade*, chamados de modelos *slow-to-start* [37, 38, 39], como tentando melhorar a relação fluxo-densidade [2]. Neste trabalho são apresentados modelos *slow-to-start*, como *VDR* [37], *TT* [39] e *BJH* [38] e o modelo de *Emmerich e Rank* [2] de ajuste de fluxo.

4.4.2 Modelos *slow-to-start*

Todos os modelos *slow-to-start* simulam, de alguma forma, a lenta reação do motorista para restabelecer o movimento de um veículo parado. Mesmo não sendo causada por esse comportamento a meta-estabilidade é reproduzida por esse procedimento e aparece quando utilizam-se duas condições iniciais diferentes nas simulações [37, 38, 39]: uma homogênea, que é quando os veículos são distribuídos igualmente na pista, e outra congestionada, quando inicia-se a simulação com uma distribuição heterogênea, contendo grandes engarrafamentos.

4.4.2.1 Modelo *velocity dependent randomization* (VDR)

No modelo VDR [37] a probabilidade do motorista demorar mais para restabelecer o movimento é simulada alterando-se somente o fator de randomização, ou seja, se um veículo está parado ele tem uma probabilidade p_s ($p_s > p_m$) de continuar parado, onde p_m é a probabilidade de desaceleração descrita no modelo NaSch. Esse comportamento é implementado inserindo-se um passo, anterior ao passo de aceleração no Algoritmo NaSch, que altera a probabilidade de

desaceleração em função da velocidade do veículo, isto é:

Algoritmo VDR:

Para todos os veículos faça

$$\begin{array}{ll}
 \textit{slow-to-start} & p = \begin{cases} p_s & \text{se } v_i = 0; \\ p_m & \text{caso contrário;} \end{cases} \\
 \textit{aceleração:} & v_i = \min(v_i + 1, v_{max}); \\
 \textit{frenagem:} & v_i = d_i \text{ se } v_i > d_i; \\
 \textit{randomização:} & v_i = v_i - 1 \text{ com probabilidade } p \text{ e se } v_i > 0; \\
 \textit{atualização:} & x_i^t = x_i^{t-1} + v_i^t.
 \end{array}$$

Resultados do modelo VDR para $p_s = 0,75$ e $p_m = 0,01$ são apresentados na Figura 4.7. Pode-se observar que o modelo representa bem a meta-estabilidade, mas o fluxo máximo fica abaixo do encontrado nos dados reais para qualquer valor de p_s e p_m .

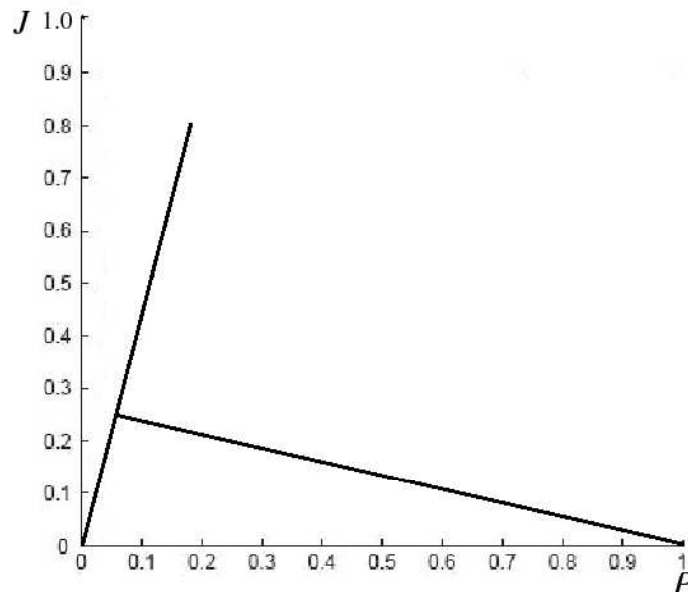


Figura 4.7: Modelo VDR com $p_s = 0,75$ e $p_m = 0,01$.

4.4.2.2 Modelo Takayasu-Takayasu (TT)

Esse modelo, proposto por Takayasu e Takayasu [39], foi um dos primeiros modelos propostos para simular o efeito *slow-to-start*. Ele parte do princípio que, estando em um engarrafamento, se o veículo logo à frente anda, o motorista tem uma probabilidade maior de continuar parado ($p_s > p_m$) caso esteja muito próximo dele ou seja, um veículo parado dificilmente ocupa um espaço à sua frente assim que ele aparece. Esse efeito também é simulado acrescentando-se um passo inicial ao Algoritmo NaSch, ou seja:

 Algoritmo TT:

Para todos os veículos faça

$$\textit{slow-to-start} \quad p = \begin{cases} p_s & \text{se } v_i = 0 \text{ e } d_i = 1; \\ p_m & \text{caso contrário;} \end{cases}$$

$$\textit{aceleração:} \quad v_i = \min(v_i + 1, v_{max});$$

$$\textit{frenagem:} \quad v_i = d_i \text{ se } v_i > d_i;$$

$$\textit{randomização:} \quad v_i = v_i - 1 \text{ com probabilidade } p \text{ e se } v_i > 0;$$

$$\textit{atualização:} \quad x_i^t = x_i^{t-1} + v_i^t.$$

No resultado mostrado na Figura 4.8 para o modelo TT com $p_s = 0,75$ e $p_m = 0,01$, pode-se observar que esse modelo apresenta uma boa representação da meta-estabilidade associada a uma boa relação fluxo densidade. No entanto, nesse modelo, a meta-estabilidade só é representada para uma grande diferença de valores entre p_s e p_m .

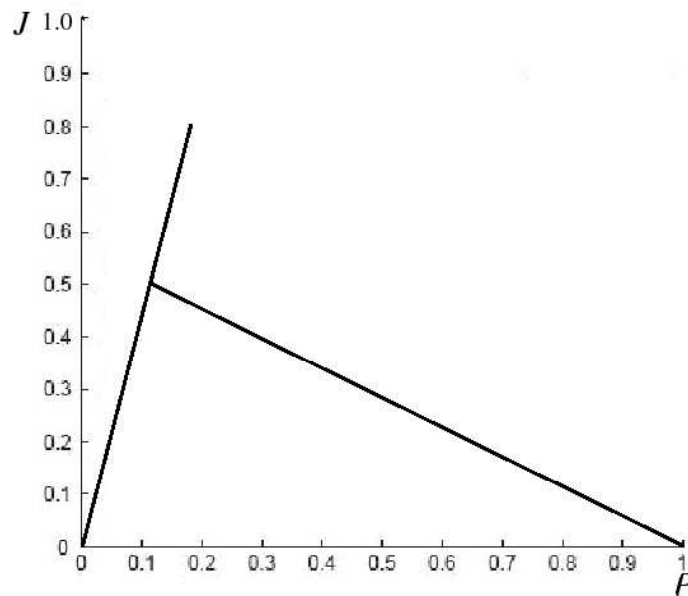


Figura 4.8: Modelo TT com $p_s = 0,75$ e $p_m = 0,01$.

4.4.2.3 Modelo BJH

Nesse modelo Benjamin-Johnson-Hui [38] propõem não se levar em consideração somente o espaço que existe à frente do veículo, considerando também se ele parou devido a outro veículo à sua frente. Se isso ocorre então o veículo continuará parado com probabilidade $p_s > p_m$. Isso é simulado adicionando-se uma outra variável ao algoritmo para informar se o veículo parou devido a outro à sua frente (*flag*), como apresentado a seguir:

Algoritmo BJH:

Para todos os veículos faça

slow-to-start $v_i = 0$ com probabilidade p_s se $flag_i = 1$;

aceleração: $v_i = \min(v_i + 1, v_{max})$;

frenagem: $v_i = d_i$ se $v_i > d_i$ e

$$flag_i = \begin{cases} 1 & \text{se } d_i = 0; \\ 0 & \text{caso contrário;} \end{cases}$$

randomização: $v_i = v_i - 1$ com probabilidade p_m e se $v_i > 0$;

atualização: $x_i^t = x_i^{t-1} + v_i^t$.

Na Figura 4.9 mostra-se o resultado do modelo BJH com $p_s = 0,75$ e $p_m = 0,01$. Pode-se observar que este modelo apresenta resultados equivalentes ao obtido pelo modelo TT, mas com as mesmas restrições em relação aos valores de p_s e p_m .

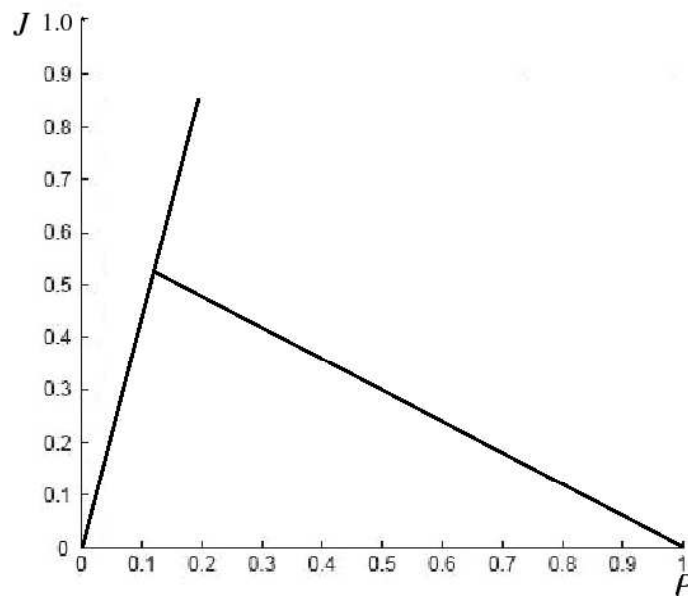


Figura 4.9: Modelo BJH com $p_s = 0,75$ e $p_m = 0,01$.

Em geral os modelos *slow-to-start* conseguem simular a região de meta-estabilidade mas, para isso, necessitam de uma grande diferença entre p_s e p_m . Quando esses valores, estão mais próximos, mesmo que pouco, a meta-estabilidade tende a não ser representada.

4.4.3 Modelo de ajuste de fluxo

Uma outra crítica feita ao modelo NaSch é que a curva gerada pelo diagrama fundamental, utilizando-se parâmetros reais de randomização entre 30% e 50% [2], fica abaixo dos dados reais disponíveis, como mostrado na Figura 2.4(a). Apresenta-se a seguir um modelo que tenta

aproximar a curva do diagrama fundamental do fluxo máximo real, considerando a movimentação que o veículo à frente fará no próximo passo.

4.4.3.1 Modelo H. Emmerich e E. Rank

Sugere-se em [2] um modelo onde cada veículo, para se movimentar, considera o quanto o veículo à sua frente se deslocará, chamado de propagação implícita de movimento. Esse comportamento é modelado da seguinte forma: primeiramente encontra-se o veículo que tenha o maior espaço livre à sua frente, movimenta-se este veículo e a seguir movimenta-se sucessivamente os outros veículos no sentido contrário ao do fluxo. Outra mudança proposta é que, no momento da desaceleração devido a presença de outro carro à frente, levar-se em consideração não somente o espaço mas também a velocidade do veículo considerado, fazendo com que veículos mais rápidos mantenham uma distância maior do veículo da frente com representado na matriz $M(v, d)$, onde d é a distância ao veículo da frente, e v é a velocidade do veículo analisado:

$$M(v, d) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 0 & 1 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ 0 & 1 & 2 & 3 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 0 & 1 & 2 & 3 & 4 & 4 & 4 & 4 & 4 & 4 & 5 \end{pmatrix} \quad (4.10)$$

Por exemplo, considerando-se a velocidade de um veículo igual a 5, se o espaço do veículo considerado para o da frente for igual a 8, então sua velocidade no próximo instante de tempo será $v = M(5, 8) = 4$.

Algoritmo ER:

Para todos os veículos em sequência, começando do com maior d_i , faça

aceleração: $v_i = \min(v_i + 1, v_{max});$

frenagem: $v_i = M(v_i, d_i)$ se $d_i < 10$;

randomização: $v_i = v_i - 1$ com probabilidade p e se $v_i > 0$;

atualização: $x_i = x_i + v_i$.

Esse algoritmo é realizado sequencialmente, como descrito anteriormente. Pode-se observar, pelos resultados mostrados na Figura 4.10, que o valor de p que mais se aproxima do fluxo máximo obtido nos dados reais é $p = 0,4$. Um dos principais problemas desse modelo é o fato

de que o fluxo máximo ocorre com uma densidade maior do que a encontrada em dados reais, Figura 2.4(a), e não consegue representar a meta-estabilidade. Outro problem é que o cálculo da velocidade e da posição dos veículos não são atualizados simultaneamente.

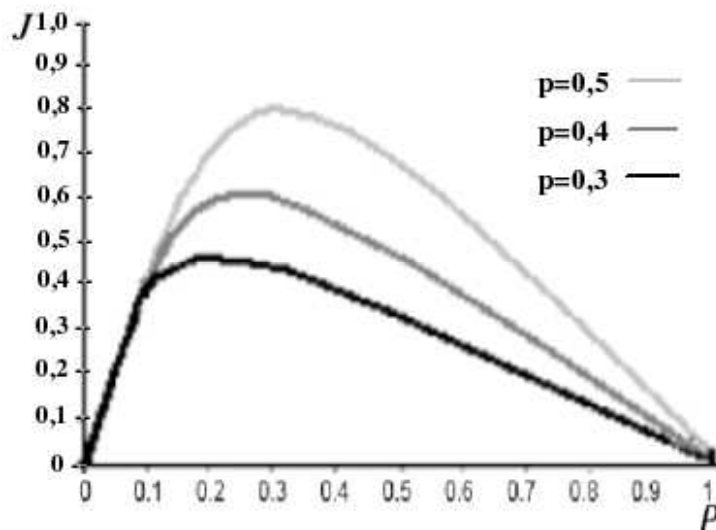


Figura 4.10: Modelo Emmerich e Rank para diferentes valores de p

Capítulo 5

Modelo de antecipação proposto

Para melhor representar tanto a relação fluxo densidade como a meta-estabilidade propõe-se um modelo probabilístico que, para atualizar a velocidade de cada veículo, considera além do espaço disponível para seu deslocamento em um certo instante de tempo, o possível movimento do veículo à sua frente. Um procedimento iterativo é utilizado para tornar o algoritmo explícito no tempo e evitar colisões irreais causadas pelo procedimento antecipatório adotado caso o veículo à frente não ande o esperado. Um fator α é utilizado para simular diferentes comportamentos do motorista ao dirigir.

5.1 Proposta de antecipação

Para melhorar a relação fluxo densidade propõe-se um modelo de antecipação de movimento que tem por base o modelo proposto em [18] que tem como objetivo o ajuste automático da velocidade dos veículos considerando, além das informações descritas no modelo NaSch (Seção 4.4.1), a possível movimentação do veículo à sua frente, estimando o espaço disponível para seu deslocamento. Esta antecipação é feita adicionando-se uma parcela que calcula uma aproximação para distância efetiva que o veículo terá para se deslocar, Figura 5.1, isto é:

$$d_{is}^t = d_i^{t-1} + (1 - \alpha)v_{i+1}^{t-1} \quad (5.1)$$

onde d_{is}^t é a distância efetiva; d_i^{t-1} é o espaço à frente do veículo considerado no tempo $t - 1$; v_{i+1}^{t-1} é a velocidade do veículo à frente do veículo i e α é um valor entre 0 e 1, representando o grau de agressividade dos motoristas ao dirigir em uma determinada via. Assim, valores de α próximos à 1 indicam que o motorista sempre mantém uma distância do veículo que o antecede, considerando neste caso uma pequena movimentação em relação à velocidade no tempo $t - 1$, sendo que no caso $\alpha = 1$ nenhuma política de antecipação está sendo adotada,

como no modelo NaSch. Pequenos valores de α indicam um comportamento agressivo, ou seja, o motorista age como se o veículo à frente não fosse alterar muito sua velocidade no tempo t em relação ao tempo $t - 1$. A parcela aditiva $(1 - \alpha) \cdot v_{i+1}^{t-1}$ representa essa antecipação da distância considerando o possível movimento do veículo $i + 1$ com velocidade v_{i+1}^{t-1} , ponderada pelo fator $(1 - \alpha)$. Como os valores das velocidades são sempre inteiros, escolhe-se o inteiro mais próximo para o valor dessa expressão.

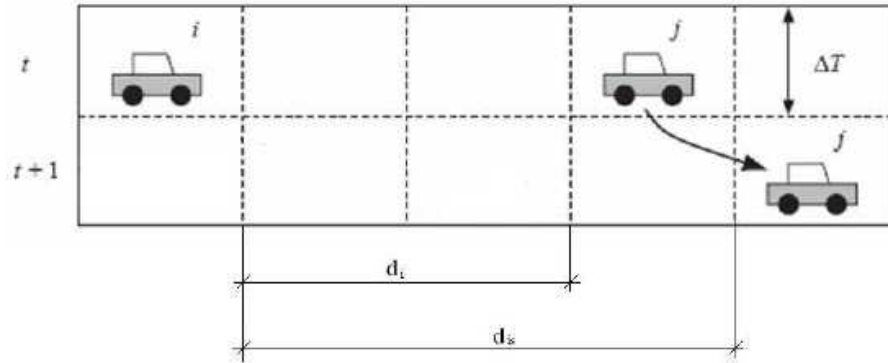


Figura 5.1: Representação da distância efetiva.

O algoritmo para esse procedimento é dado por:

Algoritmo I do modelo de antecipação

Para todos os veículos faça

aceleração:	$v_i^t = \min(v_i^{t-1} + 1, v_{max});$
randomização:	$v_i^t = (v_i^t - 1) > 0$ com probabilidade p ;
comportamento:	α – Monte Carlo;
distância efetiva:	$d_{is}^t = d_{is}^{t-1} + [(1 - \alpha) \cdot v_{i+1}^{t-1}];$
freada:	$v_i^t = d_{is}^t$ se $v_i^t > d_{is}^t$;
atualização:	$x_i^t = x_i^{t-1} + v_i^t.$

No entanto esse procedimento pode provocar colisões irreais caso o veículo à frente não se desloque como esperado.

5.1.1 Escolha de α

Para levar em consideração o movimento do veículo à frente do veículo analisado e comportamentos diferenciados dos motoristas, propõe-se uma política de antecipação diferenciada para cada motorista. A política de antecipação é determinada, como mencionado anteriormente, pela escolha do valor de α . Assim para $\alpha = 1$, não há política de antecipação e nunca ocorrerá uma colisão irreal. Para $\alpha = 0$ o motorista vai ajustar a velocidade de seu veículo supondo que será

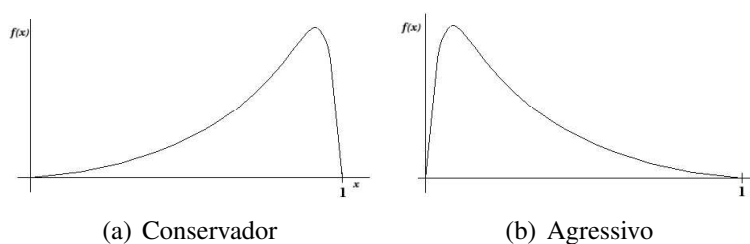


Figura 5.2: Comportamentos

mantido um deslocamento uniforme, ou seja, o veículo deslocará o mesmo número de células que no instante anterior. Isso nem sempre acontece e é esse fato que causa essas possíveis colisões irreais. Assim, propõe-se que o valor de α , que determina o comportamento médio do motorista em determinada via, não seja constante e sim que seja definido por uma função densidade de probabilidade. Se os valores de α estiverem próximos a um, o comportamento do motorista será conservador, Figura 5.2(a). Se os valores de α estiverem próximos de zero, o comportamento será agressivo, Figura 5.2(b). Com isso, para realizar a escolha do α , utiliza-se o método de Monte-Carlo (técnica da rejeição), que consiste em:

Dada a função $f : A \rightarrow \mathbb{R}$, $f(x) = y$ onde

- A é o conjunto de valores que $\alpha \in [0, 1]$ pode assumir;
- $\int_0^1 f(x) dx = 1$;

a técnica da rejeição é dada pelo seguinte procedimento:

1. sorteia-se um par ordenado (x, y) ;
2. se $y > f(x)$ então o par é rejeitado e volta-se ao passo 1 senão vai pra 3;
3. $\alpha \leftarrow x$

Nos experimentos realizados utilizou-se $f(x)$ como uma função composta de três regiões, em que a soma das áreas S_1 , S_2 e S_3 é igual a 1, Figura 5.4.

5.2 Evitando colisões irreais

Colisões irreais podem acontecer quando existem veículos a uma pequena distância relativa do veículo à frente e esse anda menos do que o esperado, ou seja $v_{i+1}^t < (1 - \alpha)v_{i+1}^{t-1}$. Com o objetivo de evitar essas colisões propõe-se um algoritmo que informa quando algum veículo

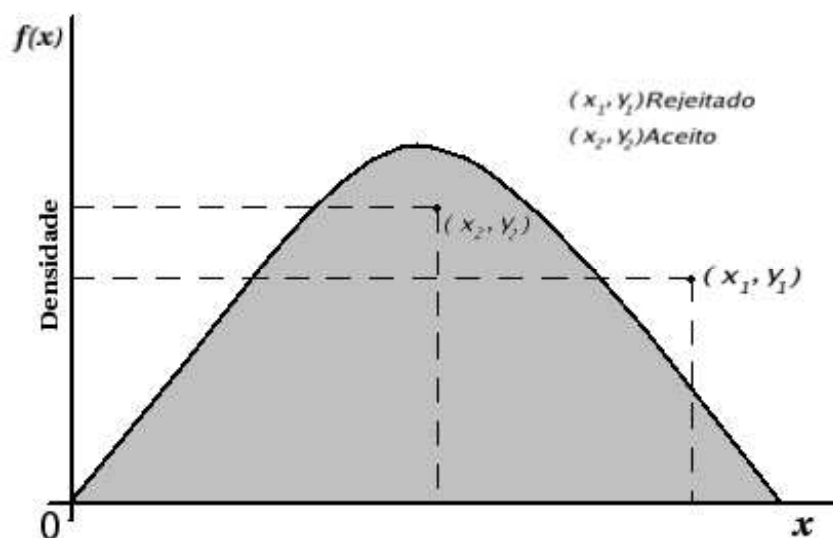


Figura 5.3: Técnica da rejeição.

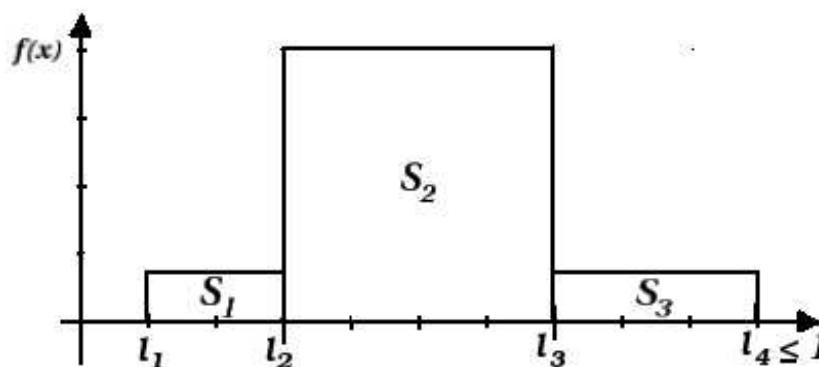


Figura 5.4: Distribuição de comportamentos.

desacelera bruscamente. Nesse caso a velocidade dos veículos que o antecedem é recalculada por um processo iterativo. Para avaliar se, com essa desaceleração brusca, haverá risco de colisões adota-se o seguinte procedimento:

a) o veículo analisado estima o quanto o veículo de trás espera que ele se movimente, ou seja:

$$(1 - \alpha_i)v_i^{t-1} \quad (5.2)$$

sendo considerado neste caso, $\alpha_i = \min(\alpha), \forall f(\alpha) \neq 0$ (Figura 5.4), ou seja, $\alpha_i = l_1$ o menor valor de $\alpha \neq 0$;

b) caso esse veículo considerado não possua a distância necessária para esse deslocamento, medido por:

$$(1 - \alpha)v_i^{t-1} > d_{is}^t$$

ele envia essa informação para que os veículos que o precedem tomem a atitude necessária. A

essa alteração do movimento esperado chamou-se de “freada brusca”. Esse processo se repete iterativamente até que, com essa alteração do movimento esperado não haja nenhum risco de colisão. Para avaliar se, com essa desaceleração, haverá risco de colisão utiliza-se a seguinte expressão:

$$(1 - \alpha).v_i^{t-1} > d_{is} \quad (5.3)$$

Quando há freada brusca, define-se um *cluster* de influência, composto por todos os veículos que serão influenciados por esta freada. Nesse caso as velocidades de todos os veículos de cada *cluster* onde houve uma freada brusca, serão recalculadas por um processo iterativo redefinindo-se, a cada iteração, um novo *cluster* de influência, se necessário. Esse processo se repete até que não haja risco de colisões em nenhum *cluster*. Esse procedimento foi incorporado ao Algoritmo I do modelo de antecipação, fornecendo:

Algoritmo II do modelo de antecipação

Para todos os veículos faça

aceleração: $v_i^t = \min(v_i^{t-1} + 1, v_{max});$
 randomização: $v_i^t = (v_i^t - 1) > 0$ com probabilidade p ;
 comportamento: α – Monte Carlo;
 distância efetiva: $d_{is}^t = d_i^{t-1} + [(1 - \alpha) \cdot v_{i+1}^{t-1}];$
 freada: $v_i^t = d_{is}^t$ se $v_i^t > d_{is}^t$;
 freada brusca Se $[(1 - \alpha) v_i^{t-1}] > d_{is}^t$;

define *clusters*;

Para todo veículo em um *clusters* com freada brusca faça

comportamento: α – Monte Carlo;
 distância efetiva: $d_{is}^t = d_i^{t-1} + [(1 - \alpha) \cdot v_{i+1}^t];$
 freada: $v_i^t = d_{is}^t$ se $v_i^t > d_{is}^t$;
 freada brusca: Se $[(1 - \alpha) v_i^t] > d_{is}^t$;

define novo *cluster*;

Para todos os veículos faça

atualização: $x_i^t = x_i^{t-1} + v_i^t.$

5.2.1 Definição dos *clusters*

Com o objetivo de restringir o número de veículos que necessitam ter suas velocidades recalculadas propõe-se uma divisão da pista, em *clusters*, feita de forma a separar os veículos em

regiões de influência.

Um *cluster* sempre tem seu início determinado pela posição de um veículo que freou bruscamente e prossegue, no sentido contrário ao fluxo, até a posição de um veículo que tenha distância suficiente para não ser influenciado por eventual alteração de movimento do veículo à sua frente, ou seja, $v_i^t \leq d_i^{t-1}$. Em um mesmo *cluster* pode haver mais de um veículo que freou bruscamente.

Para ilustrar o procedimento de definição de um *cluster* apresenta-se, na Figura 5.5, um exemplo, para um determinado tempo t , quando $\alpha = 0$. Os valores dentro de cada célula representam as velocidades dos veículos e V_1 até V_5 são os 5 veículos considerados. Na iteração $k + 1$ o veículo 4 reduziu sua velocidade de $v_4^k = 5$ para $v_4^{k+1} = 3$. Como o veículo 3 esperava que o veículo 4 movimentasse 5 posições, ele incrementou sua velocidade e $v_3^{k+1} = 5$, não dispondo de espaço suficiente para seu deslocamento. Logo este deverá ter sua velocidade recalculada o que influenciará no movimento esperado do veículo V_2 e assim sucessivamente. Como o veículo V_1 poderá se movimentar, independentemente do comportamento dos outros veículos apresentados, está, dessa forma, fora do *cluster* de influência do veículo 4.

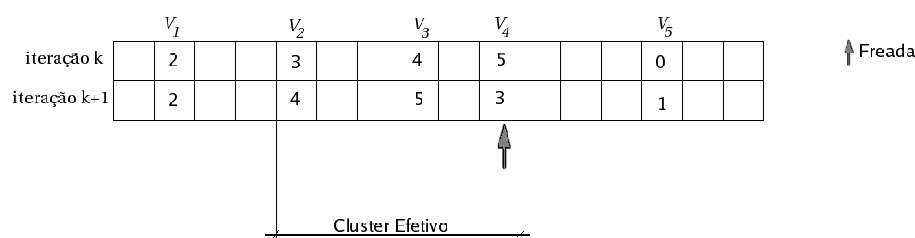


Figura 5.5: Região da pista que deve ser recalculada.

5.2.2 Custo computacional

A análise da complexidade do algoritmo proposto pode ser dividida em quatro partes: cálculo das velocidades dos veículos; definição dos *clusters* de influência; reavaliação iterativa da velocidade dos veículos com risco de colisão e atualização da posição dos veículos.

O cálculo da velocidade e a atualização da posição dos veículos são realizados uma única vez para cada veículo assim, cada uma dessas etapas tem custo computacional $O(n)$.

Para se definir os *clusters* utilizam-se duas listas, uma com a posição dos veículos com espaço suficiente para seu deslocamento e uma com veículos que frearam bruscamente, criando-se uma pilha de *clusters* que deverão ser recalculados a partir dessas listas. Como a soma do tamanho das duas listas que geram os *clusters* é menor ou igual a n então a complexidade para

a definição dos *clusters* de influência é também de $O(n)$.

A situação mais desfavorável no procedimento iterativo, que reavalia a velocidade dos veículos, é quando o *cluster* contém todos os veículos da pista e todos eles precisam ter suas velocidades recalculadas a cada iteração para que não haja riscos de colisões. Nessa pior situação o *cluster* terá, a cada iteração i , $(n - i)$ veículos, sendo n o número de veículos da pista. Nesse caso, o procedimento terá custo computacional de $O(n^2)$.

Para testar o quanto esse procedimento iterativo influencia no custo computacional do algoritmo proposto, apresenta-se na Figura 5.6 um gráfico com o número médio de vezes que a velocidade de cada veículo foi recalculada, para diferentes densidades, no caso mais desfavorável, ou seja, onde há maior possibilidade de colisões irreais ($\alpha = 0$). Pode-se observar que, de forma geral, esse procedimento não altera substancialmente o número de vezes que a velocidade dos veículos é recalculada.

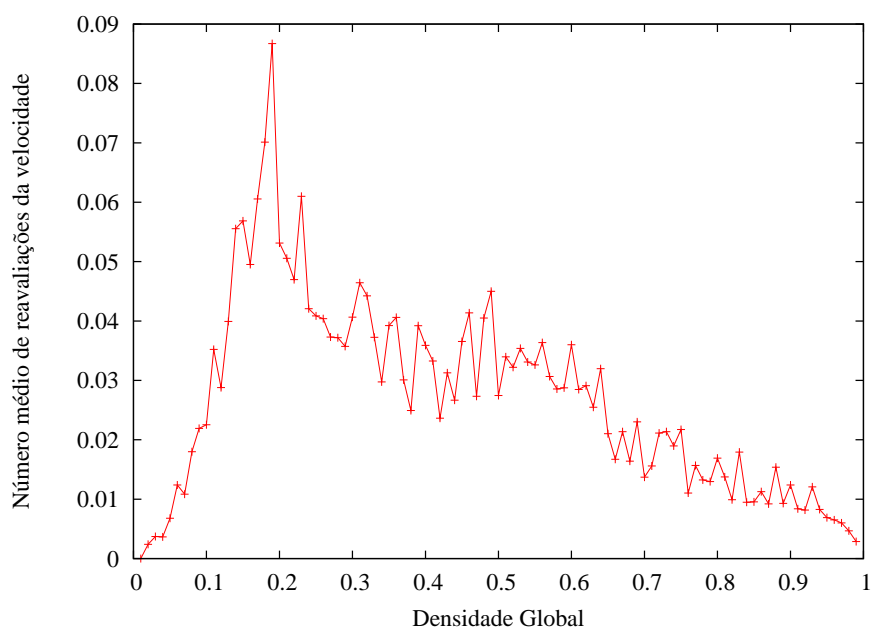


Figura 5.6: Número médio de reavaliações da velocidade por densidade.

Esse modelo, se comparado ao modelo NaSch, além de acrescentar antecipação, possui o passo de randomização antes do passo de frenagem, como proposto em [18].

5.3 Simulações

Para testar o modelo proposto, apresentam-se resultados das simulações para $p = 0,35$, utilizando uma pista circular com 300 células. Foi utilizada a média temporal para simular o detector de movimento. Assim, como descrito na Seção 4.4.1, variou-se a densidade da pista de

$\rho = 0,01$ à $\rho = 0,99$, realizando 10.000 (dez mil) passos, onde os 1000 primeiros foram desprezados para que a configuração inicial adotada não influenciasse nos resultados apresentados.

Para testar diferentes comportamentos do motorista na via definem-se três diferentes funções de densidade de probabilidade. Uma, onde os valores de α vão de 0 a 1, com 80% de probabilidade de estar entre 0,3 e 0,7 (Figura 5.7(a)). A essa distribuição chamou-se de baixa agressividade. Outra onde o valor máximo de $\alpha = 0,7$ e 80% dos valores estão entre 0,2 e 0,4, que foi chamada de média agressividade (Figura 5.8(a)) e finalmente o comportamento de alta agressividade onde $\alpha \leq 0,6$, com 90% dos valores menores que 0,2 (Figura 5.9(a)).

O resultado, apresentado na Figura 5.7(b), mostra que para baixa agressividade embora a relação fluxo-densidade tenha melhora em relação NaSch e outros já citados, ele não representa bem a região de meta-estabilidade. Nos testes realizados observou-se que isso acontece sempre que os valores de α são escolhidos próximos de $\alpha = 1$, mesmo que com uma pequena probabilidade de ocorrência. Pode-se observar que quanto maior for a agressividade melhor representada é a região de meta-estabilidade (Figura 5.9(b) e 5.8(b)).

Os diagramas velocidade-densidade (Figuras 5.7(c), 5.8(c) e 5.9(c)) apresentam a mesma qualidade dos apresentados na bibliografia com outras formulações [37, 38, 39] e não tiveram uma alteração significativa com a variação da função densidade de probabilidade. Esse diagrama só tem seus valores alterados quando se altera a velocidade máxima ou a probabilidade de randomização do modelo.

Os diagramas, apresentados nas Figuras 5.7(d), 5.8(d) e 5.9(d), representam a relação velocidade-fluxo. Nesses diagramas a diferença fica por conta de um aumento no fluxo máximo, que se dá próximo à velocidade máxima, ou seja, quanto mais agressivo é o comportamento adotado mais os veículos mantêm-se na velocidade máxima.

Outro fato que pode ser observado nos diagramas espaço-tempo, é que uma política mais agressiva influencia no aparecimento grandes regiões de engarrafamentos, Figura 5.9(f), enquanto que os comportamentos de média e baixa agressividade geram pequenos engarrafamentos como as mostradas nas Figuras 5.7(f) e 5.8(f).

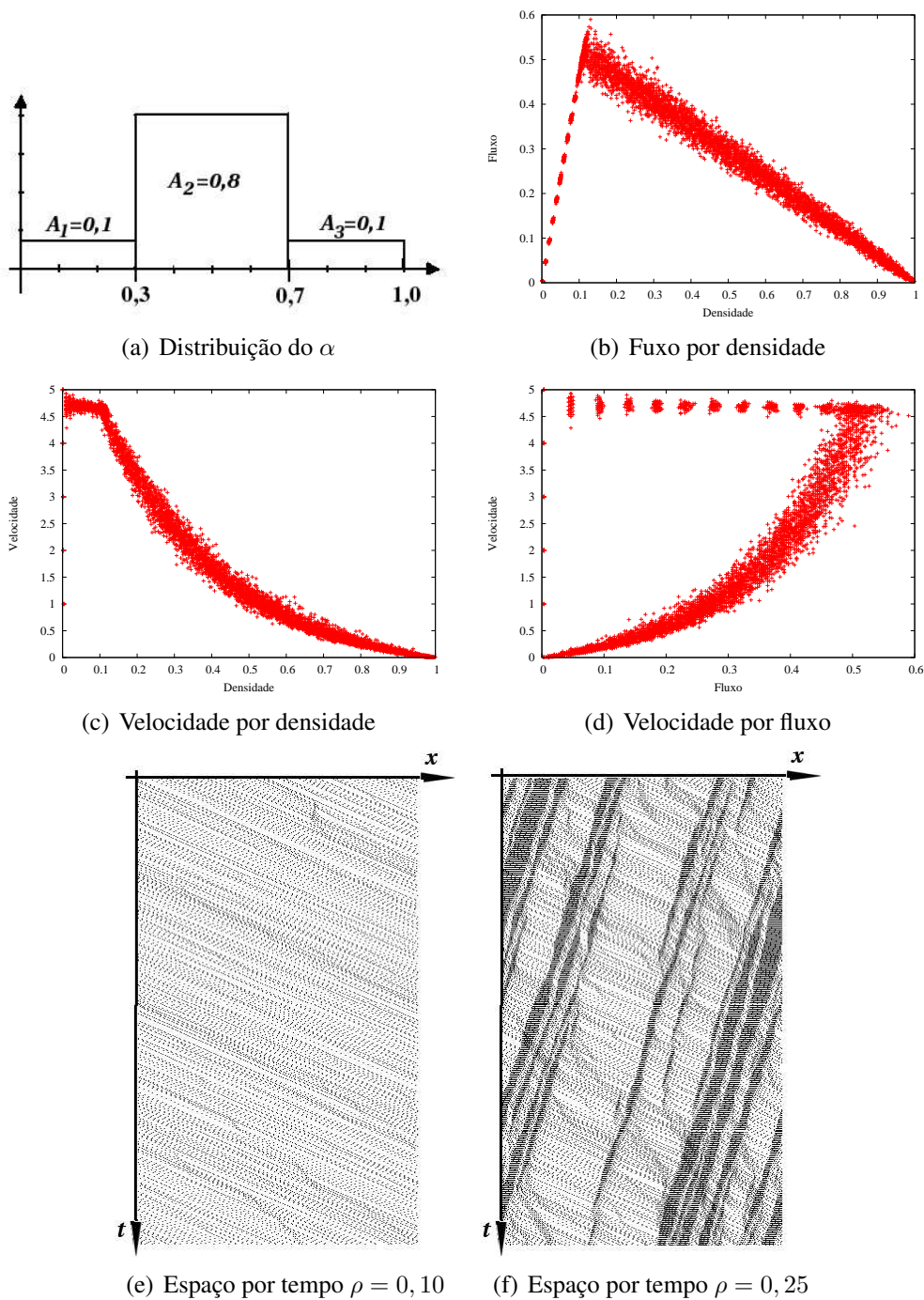


Figura 5.7: Baixa agressividade.

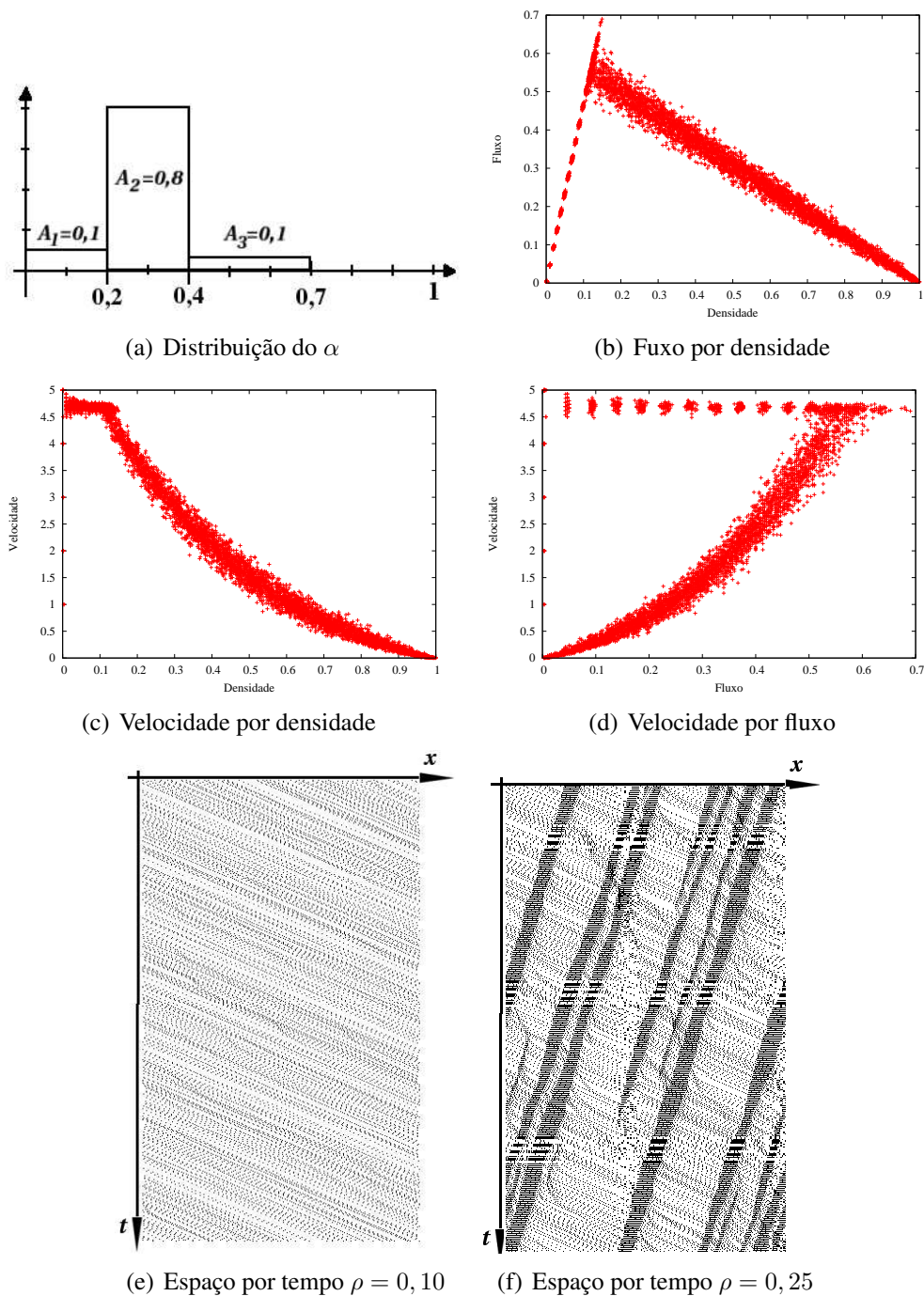


Figura 5.8: Média agressividade.

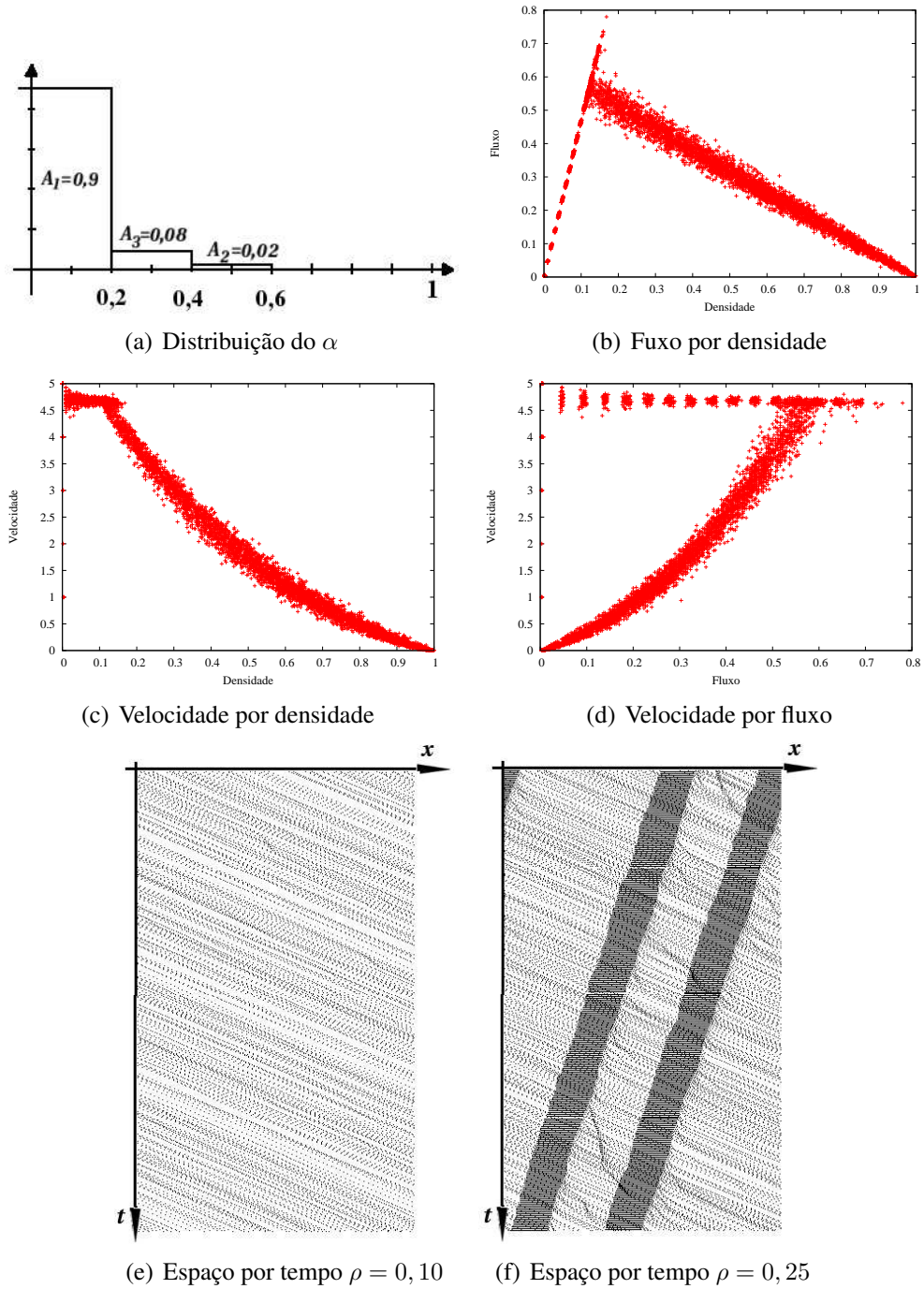


Figura 5.9: Alta agressividade.

Capítulo 6

Modelo multi-célula

O refinamento de malhas é uma conhecida técnica para estabilizar ou melhorar a aproximação de métodos numéricos. O mesmo acontece com os modelos TCA possibilitando, com essa mudança, não só considerar veículos com tamanhos diferentes, como também refinar a representação de diferentes velocidades mantendo o mesmo Δt o que, como já foi dito no Capítulo 4, é o tempo de reação dos motoristas.

Nos modelos TCA apresentados nos capítulos anteriores, cada veículo ocupa somente uma única célula e por isso o estado de cada célula contém somente a informação da velocidade do veículo ocupante. No caso de se refinar a discretização do domínio, um veículo ocupará duas ou mais células (Figura 6.1), possibilitando considerar diferentes tamanhos para esses veículos (l_i).

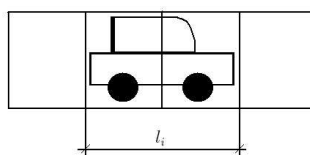


Figura 6.1: Veículo ocupando duas células.

Se cada célula for subdividida em 5 outras células, por exemplo, o veículo que nos capítulos anteriores ocupava 1 célula, ocupará agora, 5 células. O mesmo ocorre com a velocidade: considerando-se a velocidade de $1\text{cel}/\text{seg}$, como nos modelos anteriores, para se manter a mesma velocidade real para essa nova discretização, a velocidade deve ser de 5 células por segundo.

Nesse capítulo são apresentados os modelos de TCA originalmente propostos como multi-células HS [3], luz de freio [4] e KKW [15] que utilizam desse tipo de representação para descrever melhor a dinâmica do tráfego.

d_i	$V(d_i)$	d_i	$V(d_i)$	d_i	$V(d_i)$	d_i	$V(d_i)$
0, 1	0	7	4	11	8	19 – 23	13
2, 3	1	8	5	12	9	24 – 36	14
4, 5	2	9	6	14, 15	10	≥ 37	15
6	3	10	7	16 – 18	11		

Tabela 6.1: Função $V(d)$

Apesar do algoritmo proposto no Capítulo 5 ter sido concebido para células simples, propõe-se nesse capítulo uma modificação que possibilita tanto a representação de diferentes acelerações, quanto sua aplicação para diferentes comprimentos de veículos.

6.1 O modelo de Helbing e Schreckenberg(HS)

Helbing e Schreckenberg em 1999 [3] propuseram um modelo multi-célula, que é baseado nos mesmos princípios dos modelos de velocidade ótima, descrito na Seção 2.4, ou seja, usam uma função que calcula a velocidade ótima a partir de sua velocidade v_i e do seu espaço para o deslocamento d_i .

Nesse modelo a função velocidade ótima é representada pela função discreta $V(d_i)$, descrita na Tabela 6.1. Essa função discreta leva em consideração somente o espaço para determinar a velocidade do veículo, como apresentado na Seção 2.4 e tem seu algoritmo descrito como:

Algoritmo HS:	
Para todos os veículos faça	
aceleração:	$v_i^t = v_i^{t-1} + [\sigma(V(d_i^{t-1})) - v_i^{t-1}];$
randomização:	$v_i^t = (v_i^t - 1) > 0$ com probabilidade p ;
atualização:	$x_i^t = x_i^{t-1} + v_i^t.$

Pode-se observar, na Tabela 6.1, que o uso de uma configuração multi-célula possibilita uma definição mais refinada na variação da velocidade do que a apresentada no modelo Emmmerik e Rank, no Capítulo 4.

Embora os autores [3] afirmem ter encontrado muitos dos comportamentos observados em um tráfego real, Knospe e outros [7] mostram que este modelo não é totalmente livre de colisões e falha na representação de um fluxo sincronizado, não apresentando região de meta-estabilidade.

Na Figura 6.2 pode-se observar que apesar da altura máxima do fluxo ser bem representada, esse modelo não apresenta uma região de meta-estabilidade e torna-se nulo para densidades

pouco maiores que 0,6.

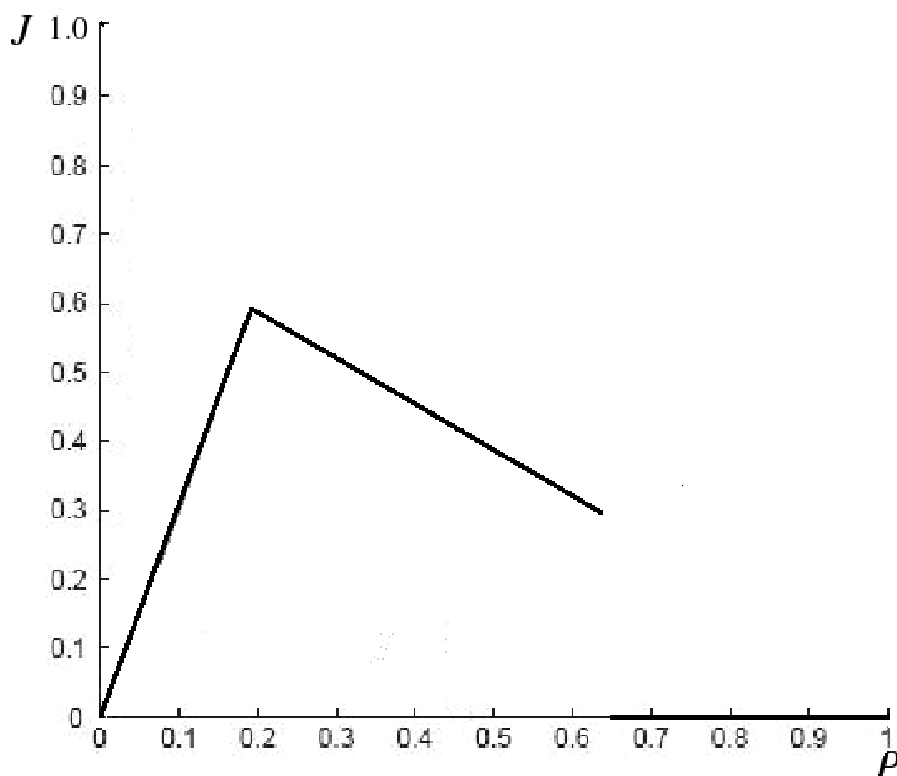


Figura 6.2: Diagrama fundamental do modelo HS [3].

6.2 Modelo luz de freios

Em 2000 Knospe et. al [4] propuseram um modelo que considera a antecipação introduzindo uma variável, que serve como reguladora da velocidade para os veículos anteriores, chamada de “luz de freio”. Nesse modelo um motorista não acelera se o veículo à sua frente está freiando, ou seja, está com luz de freio acesa.

No modelo luz de freio a probabilidade de frenagem aleatória (p) fica em função da luz de freio do veículo à frente (b_{i+1}), de tal forma que um veículo terá uma probabilidade $p = p_b$ de desacelerar, se o veículo à sua frente acabou de frear e se encontra relativamente próximo. Além disso, o modelo também utiliza um efeito *slow-to-start*, como no modelo VDR, com probabilidade $p = p_0$ e em qualquer outro caso a probabilidade considerada é $p_m < p_0 < p_b$.

O modelo usa a noção de “distância temporal”, definida como sendo o número de segundos que um veículo está do outro, descrita como: $t_{h_i}^{t-1} = d_i^{t-1}/v_i^{t-1}$. Nesse modelo o motorista acelera quando o veículo à sua frente não freou no instante anterior ou se existe uma boa “distância temporal” à sua frente. Para isso utiliza-se $t_{h_i}^{t-1} < t_{s_i}^{t-1}$, sendo $t_{s_i}^{t-1} = \min(v_i^{t-1}, h)$, onde h é um

parâmetro que pode ser interpretado como a menor “distância temporal” segura a ser utilizada por todos os veículos da pista.

Nesse modelo o esquema de antecipação utilizado tenta prever se o veículo à frente terá que desacelerar devido a um pequeno espaço vazio à sua frente, evitando assim colisões. Esse cálculo é feito levando-se em consideração o espaço vazio à frente do veículo $i + 1$ no instante de tempo anterior d_{i+1}^{t-1} , sendo esse o menor espaço que o veículo à frente terá para se mover. Por esse motivo o espaço adicional proporcionado pela movimentação do veículo não será menor que $\max\{\min(v_{i+1}^{t-1}, d_{i+1}^{t-1}) - d_{\text{seguro}}, 0\}$, onde d_{seguro} é a menor distância de segurança utilizada quando os veículos estão em movimento.

Apresenta-se a seguir o algoritmo desse modelo.

Algoritmo luz de freios :

Para todos os veículos faça

$$\begin{array}{ll}
 \text{determinação do } p: & p^t = \begin{cases} p_b & \text{se } b_{i+1}^{t-1} = 1 \text{ e } t_{h_i}^{t-1} < t_{s_i}^{t-1}; \\ p_0 & \text{se } v_i^{t-1} = 0; \\ p_m & \text{caso contrário;} \end{cases} \\
 & b_i = 0; \\
 \text{aceleração:} & v_i^t = \min(v_i^{t-1} + 1, v_{\text{max}}) \text{ se } (b_i^{t-1} = b_{i+1}^{t-1} = 0) \\
 & \text{ou } (t_{h_i}^{t-1} \geq t_{s_i}^{t-1}); \\
 \text{espaço efetivo:} & d_{i_s}^t = d_i^{t-1} + \max\{\min(v_{i+1}^{t-1}, d_{i+1}^{t-1}) - d_{\text{seguro}}, 0\}; \\
 \text{frenagem:} & v_i^t = \min(v_i^t, d_{i_s}^t) \\
 & b_i^t = 1 \text{ se } v_i^t < v_i^{t-1}; \\
 \text{randomização:} & v_i^t = \max(v_i^t - 1, 0), \text{ com probabilidade } p^t \\
 & b_i^t = 1 \text{ se } p^t = p_b; \\
 \text{atualização:} & x_i^t = x_i^{t-1} + v_i^t.
 \end{array}$$

O resultado apresentado na Figura 6.3 foi gerado com os seguintes parâmetros: $v_{\text{max}} = 108 \text{ km/h} = 20 \text{ células/seg}$; $p = 0, 1$; $p_0 = 0, 5$; $p_b = 0.94$; $d_{\text{seguro}} = 7$ e $h = 6$.

Como mencionado, este modelo incorpora antecipação e o efeito *slow-to-start*. O resultado é um modelo que consegue ter uma boa relação fluxo-densidade, representa a região de meta-estabilidade e sem risco de acidentes. Isso devido a sua característica multi-célula que faz com que esse tipo de antecipação seja ao mesmo tempo eficiente e segura. Essas alterações possibilitam que os motoristas dificilmente desacelerem, mantendo assim a maioria dos veículos com baixa probabilidade de desacelerar, $p = 0, 1$. O principal problema desse modelo é o número de variáveis de calibração e o fato de somente ter um bom resultado com uma grande

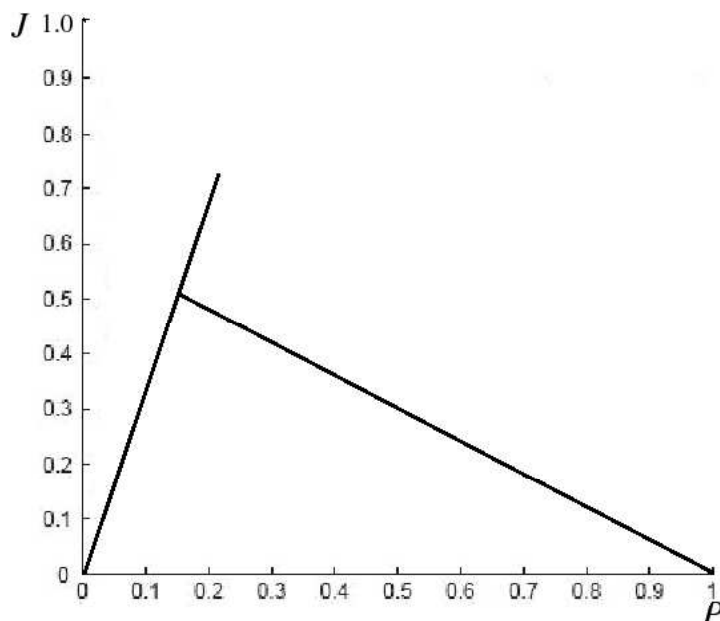


Figura 6.3: Diagrama fundamental do modelo luz de freio [4].

refinamento das células.

6.3 O modelo de Kerner, Klenov e Wolf

Uma extensão do modelo luz de freio foi proposta por Kerner et al [15], que resultou em uma família de modelos que utilizam a noção de distância sincronizada entre cada veículo [5], que é a distância de interação entre os veículos em função da velocidade, ou seja, um veículo rápido é influenciado pela velocidade de um veículos que está mais distante. A partir dessa sincronização é feito o cálculo para se determinar o quanto um veículo irá acelerar e desacelerar, fazendo uso de um efeito *slow-ro-start*.

Nesse modelo o cálculo da distância sincronizada (D_i) é feito, por exemplo, por uma função linear, função de D_0 e D_1 , que têm que ser estimados. A sincronização da velocidade, Δ_{acci} , representa a motivação do motorista a acelerar quando o veículo à sua frente acelera e desacelerar quando este o faz. Os parâmetros a e b representam a capacidade de aceleração e desaceleração do veículo respectivamente. Esses parâmetros são utilizados na estimativa da velocidade desejada (v_{des_i}) que vai depender não somente da sincronização da velocidade como também do espaço, ou seja, a variação da velocidade do veículo líder só afetará o veículo se este estiver próximo o suficiente. Assim, nesse modelo, a velocidade (v_i^t) é determinística e função também da velocidade desejada. Além disso, o uso de multi-célula possibilita uma alteração no valor dessa velocidade incluindo, na randomização, uma parcela de ajuste de aceleração.

Para isso, a partir do passo de determinação da probabilidade de aceleração (p_a^t) é iniciada a aleatoriedade do modelo. Nesse passo é determinada a probabilidade que o veículo tem de acelerar mais uma vez, p_a , baseado na velocidade calculada nos passos anteriores e no parâmetro v_p que define a velocidade mínima para fluxos livres. No passo de determinação de probabilidade de freada aleatória, p_b , calcula-se a probabilidade de um motorista desacelerar o veículo, sem motivo aparente, baseado nos modelos *slow-to-start*. Essas probabilidades são utilizadas no passo de aleatoriedade que definem o valor para a parcela aditiva, η_i , à velocidade. Utilizando a parcela η_i e fazendo uso das probabilidade de aceleração (p_a) e desaceleração (p_b), faz-se a randomização da velocidade considerando que esta não pode passar da velocidade máxima e nem ser negativa.

Algoritmo KKW:

Para todos os veículos faça

distância sincronizada:	$D_i^t = D_0 + D_1 v_i^{t-1};$
sincronização da velocidade:	$\Delta_{acci}^t = \begin{cases} a & \text{se } v_i^{t-1} < v_{i+1}^{t-1}; \\ 0 & \text{se } v_i^{t-1} = v_{i+1}^{t-1}; \\ -b & \text{se } v_i^{t-1} > v_{i+1}^{t-1}; \end{cases}$
velocidade desejada:	$v_{des_i}^t = \begin{cases} v_i^{t-1} + a & \text{se } d_i > (D_i^t - l_i); \\ v_i^{t-1} + \Delta_{acci}^t & \text{se } d_i \leq (D_i^t - l_i); \end{cases}$
velocidade determinística:	$v_i^t = \max\{0, \min(v_{max}, d_i, v_{des_i}^t)\};$
probabilidade de aceleração:	$p_a^t = \begin{cases} p_{a1} & \text{se } v_i^t < v_p; \\ p_{a2} & \text{se } v_i^t \geq v_p; \end{cases}$
probabilidade de desaceleração:	$p_b^t = \begin{cases} p_0 & \text{se } v_i^t = 0; \\ p_m & \text{se } v_i^t > 0; \end{cases}$
aleatoriedade:	$\eta_i^t = \begin{cases} a & \text{com probabilidade } p_a; \\ -b & \text{com probabilidade } p_b; \\ 0 & \text{caso contrário;} \end{cases}$
velocidade aleatória:	$v_i^t = \max\{0, \min(v_{max}, v_i^t + \eta_i^t, v_i^t + a)\};$
atualização:	$x_i^t = x_i^{t-1} + v_i^t.$

A simulação, apresentada na Figura 6.4, foi obtida com os parâmetros: $D_0 = 60$; $D_1 = 2.55$; $a = b = 1$; $v_p = 28$, $p_{a1} = 0.2$, $p_{a2} = 0.052$, $p_0 = 0.425$ $p_d = 0.04$, $v_{max} = 60$ células/segundo e $l = 15$ células.

Apesar do diagrama fundamental do modelo apresentar uma boa altura ele não consegue representar, de forma satisfatória, a região de meta-estabilidade apresentando uma região de

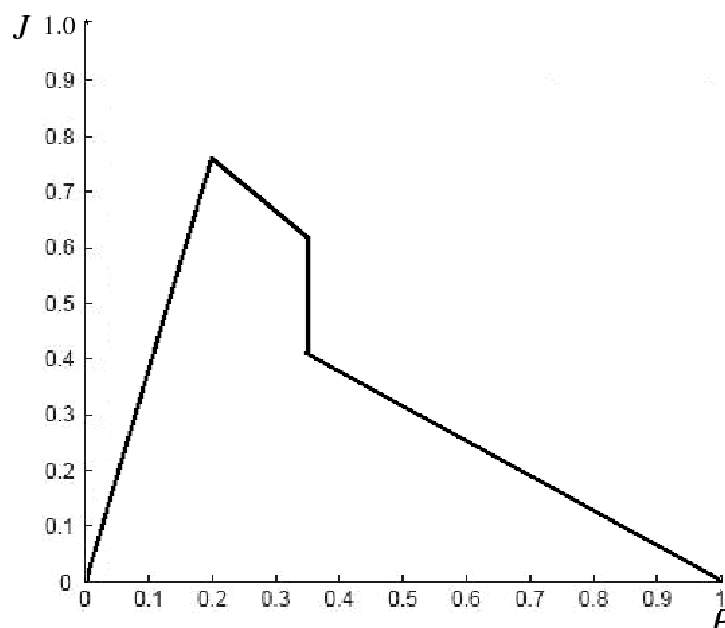


Figura 6.4: Diagrama fundamental do modelo KKW [5].

diminuição de fluxo antes de torna-se engarrafado. Além disso o modelo requer uma grande quantidade de variáveis de calibração.

6.4 Modelo proposto com multi-célula

Como visto nas seções anteriores os modelos multi-células não restringem suas acelerações a uma célula por segundo, como acontece nos modelos de célula única, podendo variar sua velocidade em várias células por passo.

Utilizando dessa possibilidade, de acelerar ou desacelerar mais de uma célula por passo, propõe-se uma alteração no modelo proposto (Capítulo 5), a partir de um modelagem multi-célula, ou seja, que considera a ocupação de mais células por um único veículo. As regras de aceleração e randomização foram modificadas para permitir, ao veículo analisado, ter ao final desses passos uma velocidade incrementada de uma ou duas unidades a mais que a do instante anterior, possibilitando assim dois tipos de comportamentos, distintos: uma, chamada veículos rápidos e outra, chamada de veículos lentos.

Para simular esses dois tipos de comportamentos, define-se a aceleração máxima (a_{max}), variável inteira e divisível por dois, definida como o máximo que um veículo pode acelerar, e divide-se a probabilidade p , do passo de randomização, em duas: uma, para simular veículos lentos (p_{vl}), e outra, para veículos rápidos (p_{vr}). Assim, se um veículo desacelera, pela randomização, ele terá uma probabilidade p_{vl} de desacelerar a_{max} e $p_{vr} = 1 - p_{vl}$ de desacelerar a

metade desse valor, ou seja, um veículo desacelera a_{max} com probabilidade $p \cdot p_{vl}$ e $a_{max}/2$ com probabilidade $p \cdot p_{vr}$

Foram feitas as seguintes modificações nos passos de aceleração e randomização:

Alterações no algoritmo do modelo proposto	
aceleração:	$v_i = \begin{cases} v_{max} & \text{se } v_i + a_{max} > v_{max}; \\ v_i + a_{max} & \text{se } v_i + a_{max} \leq v_{max}; \end{cases}$
randomização:	$v_i = \begin{cases} v_i - a_{max} & \text{com probabilidade } p \cdot p_{vl}; \\ v_i - (a_{max}/2) & \text{com probabilidade } p \cdot p_{vr}; \\ v_i & \text{com probabilidade } 1 - p \end{cases}$

6.4.1 Simulações

Nas simulações apresentadas utilizou-se a função densidade de probabilidade descrita no Capítulo 5, Figura 5.9(a). No entanto, nos testes realizados mostrou-se que a escolha de outros tipos de comportamentos não compromete os resultados, mantendo-os equivalentes ao aqui apresentados.

Foram feitos dois tipos de simulações para testar o esquema proposto: na primeira simulação todos os veículos ocupam 2 células e na segunda simulação utilizou-se diferentes tamanhos de veículos, com 1 célula de 7,5 metros para carros pequenos e médios e 2 células para veículos longos.

- *Simulação 1: veículos ocupando duas células.*

Para as primeiras simulações fez-se todos os veículos ocuparem 2 células cada um, com o tamanho da célula igual a 3,75 metros. Para manter a mesma velocidade máxima utilizou-se $v_{max} = 10$.

Foram utilizadas diferentes probabilidades de desaceleração e diferentes probabilidades de veículos lentos e rápidos.

Pode-se observar pelos resultados mostrados nas Figuras 6.5, 6.6 e 6.7, que o uso de diferentes valores para as probabilidades p , p_{vl} e p_{vr} , não altera a amplitude da meta-estabilidade. Além disso, valores pequenos de p_{vl} geram um aumento significativo no fluxo. Outra característica observada é que o uso de valores maiores de p_{vl} aumenta a variação das velocidades dos

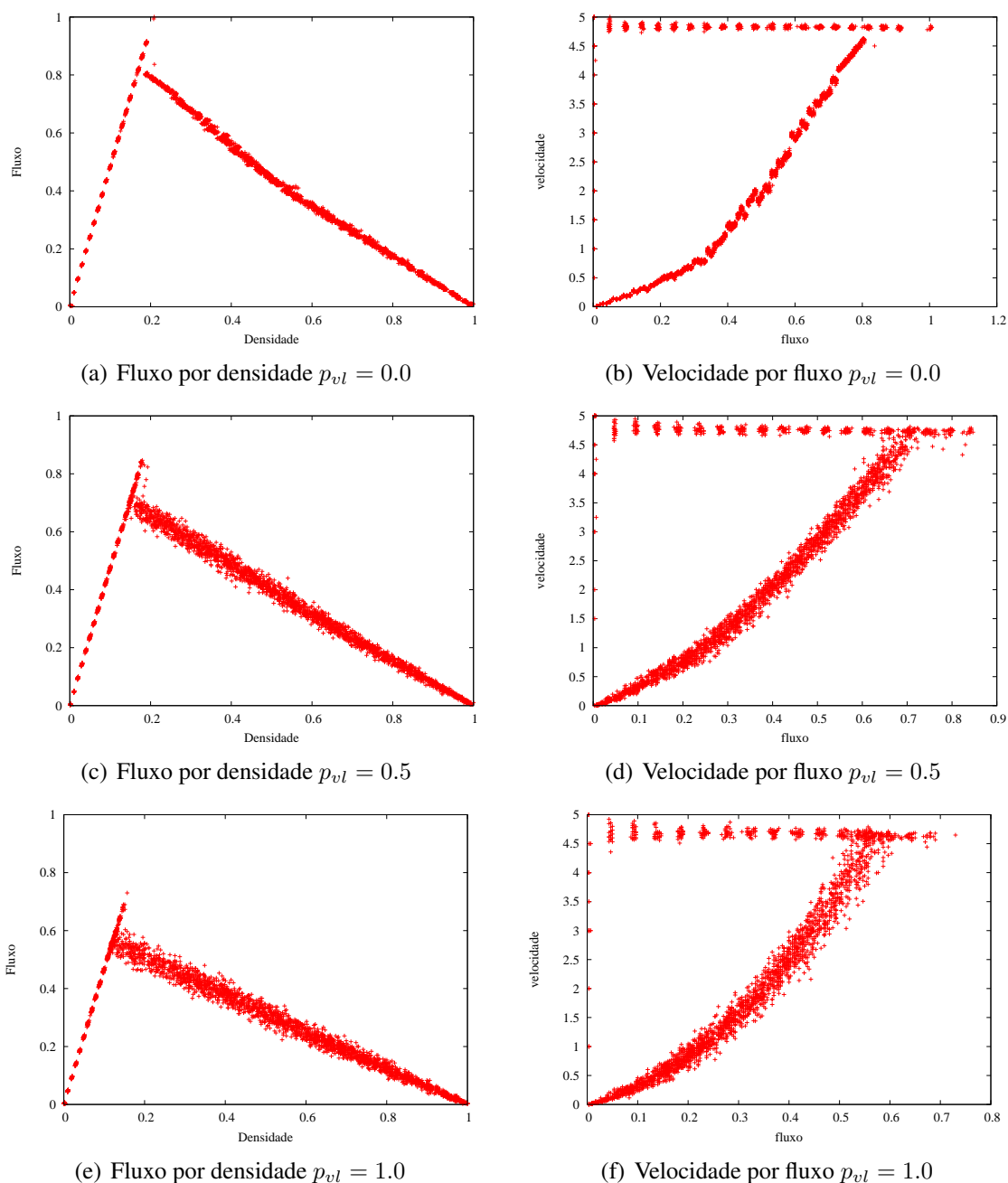


Figura 6.5: Multi-célula: simulação 1, $p = 0,35$.

veículos para um mesmo fluxo, representada pela maior dispersão das velocidades, como pode ser observado nos gráficos (b), (d) e (f) das Figuras 6.5 à 6.7.

- *Simulação 2: veículos de tamanhos diferentes*

Para esta segunda simulação utilizou-se um percentual de veículos longos (P_{longos}), veículos ocupando o dobro do número de células, simulando assim a existência de caminhões. Cada veículo longo ocupa duas células, e cada veículo curto ocupa uma célula. Como critério de

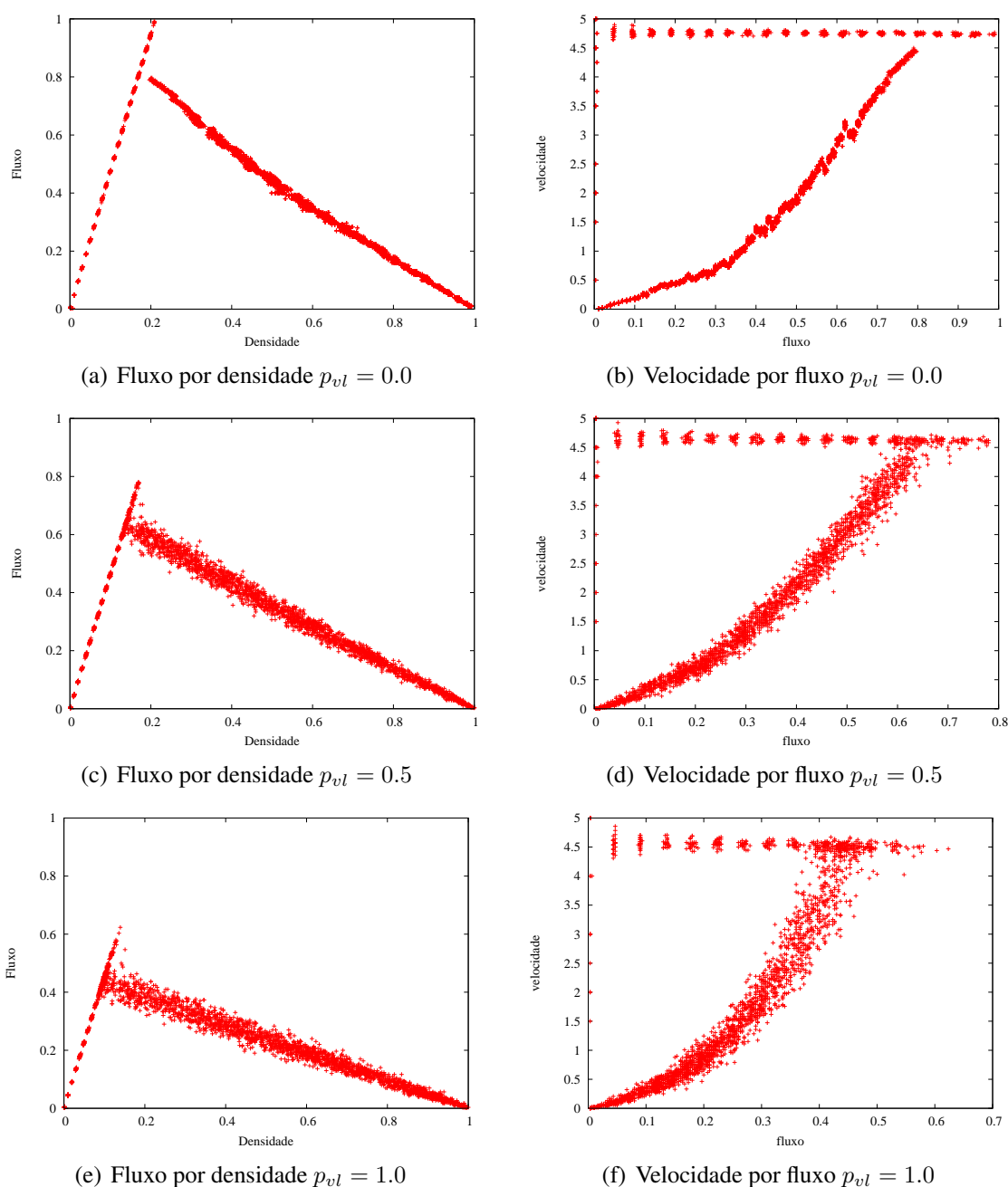


Figura 6.6: Multi-célula: simulação 1, $p = 0, 50$.

comparação com os resultados apresentados no Capítulo 5 utilizou-se $a_{max} = 1$, $v_{max} = 5$, $p = 0, 35$ e $p_{vl} = 1, 0$.

Na Figura 6.8 são mostrados resultados para diferentes percentuais de veículos longos. Com esses resultados pode-se observar que, apesar dos veículos acelerarem e desacelerarem igualmente, a maior ocupação da pista por veículos longos causa uma diminuição significativa do fluxo, uma vez que uma quantidade menor de veículos gera uma ocupação maior da via.

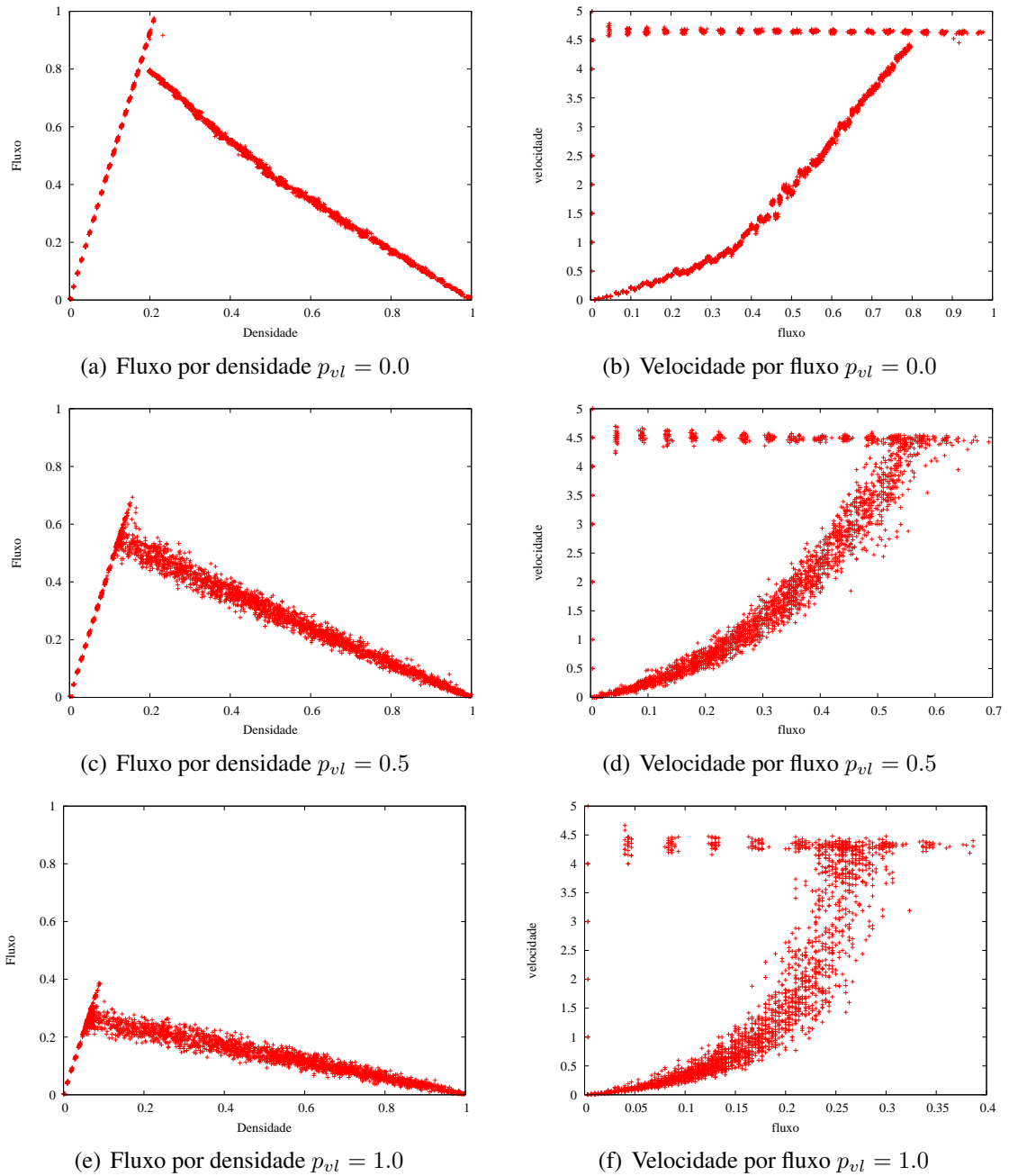


Figura 6.7: Multi-célula: simulação 1, $p = 0, 70$.

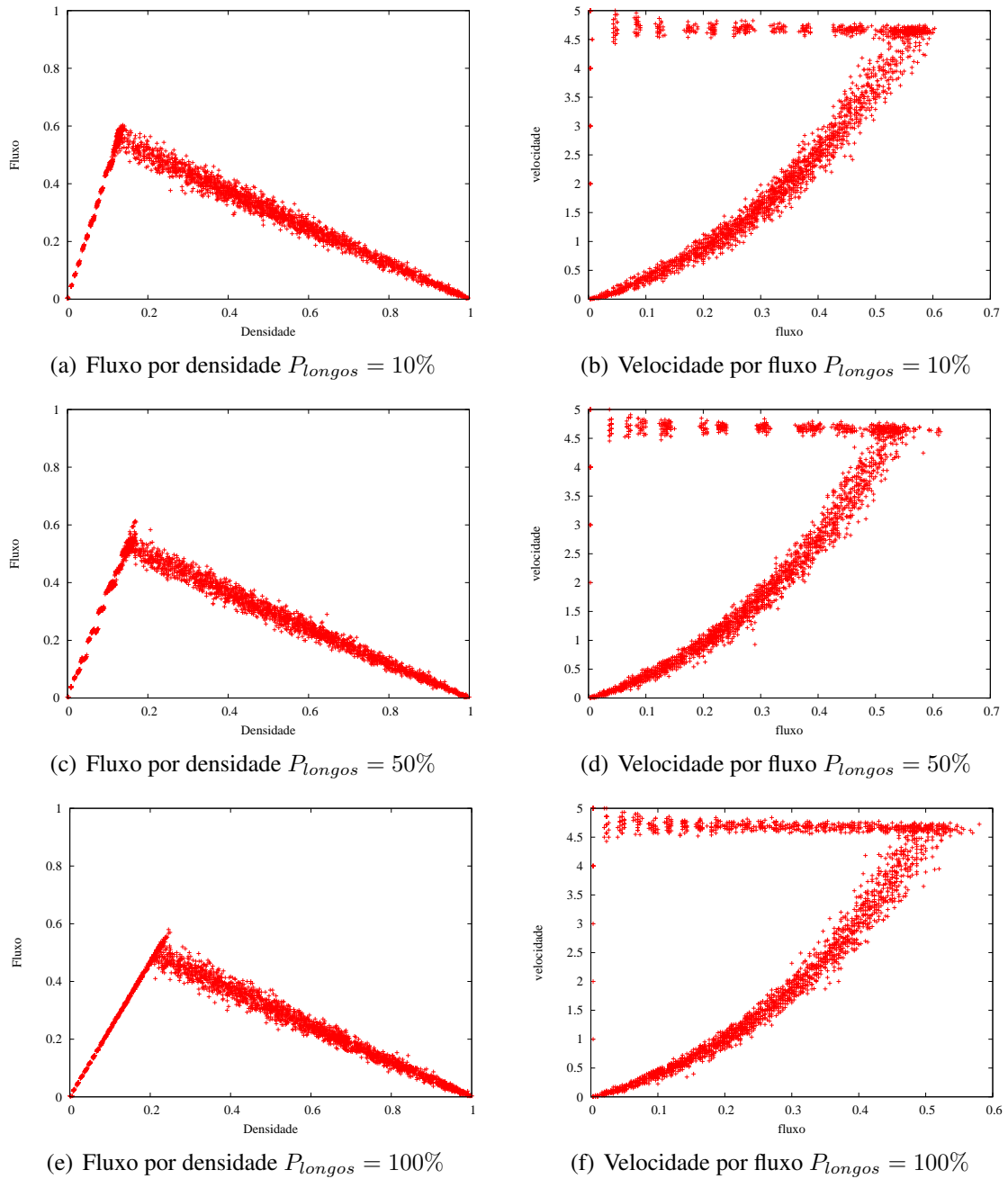


Figura 6.8: Multi-célula: simulação 2, $p = 0,35$.

Capítulo 7

Modelo multi-faixas

Para modelar outras situações, tais como vias com mais de uma faixa, apresenta-se neste Capítulo, uma extensão de modelos de autômatos celulares multi-faixas para o tráfego. A maioria dos modelos de multi-faixas encontrados na bibliografia, é dividida em dois passos independentes [6, 34, 40, 41]:

1. regras de mudança de faixa, que são aplicadas simultaneamente para todos os veículos;
2. um algoritmo de TCA, de faixa única, para todas as faixas.

Para isso, são descritos inicialmente, noções e comportamentos gerais de algoritmos de mudanças de faixas. A seguir apresentam-se dois modelos de mudança de faixa que consideram as regras de mudança para a faixa da esquerda diferente da regra de mudança para a direita, chamados de mudanças assimétricas, o primeiro faz uso de uma compensação do espaço disponível e o segundo modelo faz uso da antecipação de movimento sugerida no modelo de luz de freio, Capítulo 6. Na seção 7.3 propõe-se um conjunto de regras para mudança de faixa a serem utilizadas no modelo de antecipação proposto. Resultados são apresentados para validar o procedimento adotado e finalizando apresenta-se uma simulação considerando um acidente em uma das faixas.

7.1 Mudança de faixas

As regras utilizadas nos modelos de mudanças de faixas, podem ser divididas em dois tipos: imperativas e descritivas [33]. A imperativa é a mudança que o motorista é obrigado a realizar, seja por obrigatoriedade da lei ou porque o motorista precisa chegar à uma saída da pista. A descritiva é a mudança de faixa que o motorista escolhe realizar para manter uma dirigibilidade confortável, por exemplo, por haver um veículo lento à sua frente.

Além disso a mudança de faixa pode ser simétrica ou assimétrica. A mudança simétrica é aquela que utiliza a mesma regra para realizar todas as mudança de faixas, tanto da esquerda para direita quanto da direita para esquerda [33] e quando em um tráfego existe preferência por algumas das faixas, diz-se que essa mudança é assimétrica. De acordo com o código brasileiro de trânsito “quando uma pista de rolamento comportar várias faixas de circulação no mesmo sentido, são as da direita destinadas ao deslocamento dos veículos mais lentos e de maior porte, quando não houver faixa especial a eles destinada, e as da esquerda, destinadas à ultrapassagem e ao deslocamento dos veículos de maior velocidade” [42].

Os procedimentos de mudanças de faixas consistem de um conjunto de regras que, aplicadas sobre cada veículo, determinam a faixa que este deve estar. Essas regras podem ser divididas em: regras de incentivo à mudança, simulando o desejo do motorista de mudar de faixa, e regras de segurança, que verificam a possibilidade da realização dessa mudança. O veículo só mudará de faixa se ambas forem satisfeitas.

Além disso os modelos utilizados para mudar de faixa devem reproduzir o efeito de inversão de uso de faixa em um modelo assimétrico, que é descrito na Seção 7.1.1, e tentar evitar o efeito ping-pong comum nesse tipo de simulação (Seção 7.2).

7.1.1 Inversão do uso de faixa

Um fenômeno característico no tráfego, regido por regras assimétricas de mudança de faixa, é a inversão do uso de faixa, que é visualizado pelo diagrama de uso percentual de cada faixa por fluxo [6] (Figura 7.1).

No tráfego, com mudança assimétrica de faixa, se um motorista não se sente bloqueado por outro à sua frente este permanece na faixa da direita, ou retorna a ela. Por sua vez, sempre que um veículo mais lento está à sua frente, o motorista sente a necessidade de ultrapassado pela faixa da esquerda.

Esse comportamento gera uma inversão do uso das faixas quando varia-se a densidade (Figura 7.1). Pode-se observar que quando a densidade é baixa os veículos estão quase todos na faixa da direita. Quando a densidade é incrementada os veículos da faixa da direita começam a ser bloqueados pelos outros, fazendo com eles mudem para a faixa da esquerda, até que o uso da faixa da esquerda supere o da direita [6].

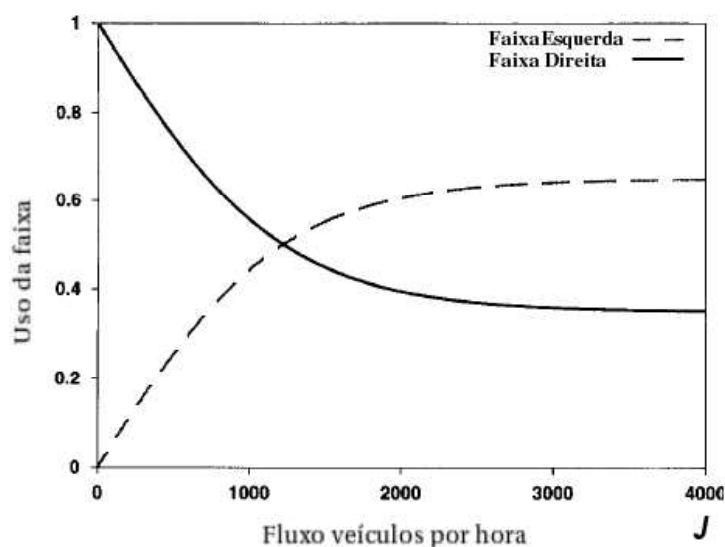


Figura 7.1: Fenômeno de inversão de pista.

7.2 Efeito ping-pong

O efeito ping-pong é um resultado não realista que ocorre em modelos multi-faixa, e acontece quando um determinado veículo muda de faixas durante vários passos consecutivos.

Em muitas regras [43], um veículo que deixa uma faixa, pode estar apto a voltar para a faixa de origem no próximo instante de tempo. Se isso ocorrer ele terá uma grande probabilidade de desejar realizar esta mudança novamente, caso as condições desta faixa não tenham sido alteradas.

Uma maneira de evitar o aparecimento do efeito ping-pong é adicionando-se uma aleatoriedade às regras de mudança de faixa [44], ou seja, mesmo tendo condições para a troca de faixa o motorista realizará tal mudança com probabilidade p_{muda} .

7.3 Modelos de mudanças de faixas

Supondo que os motoristas seguem o código de trânsito, um modelo que possa representar o fluxo de veículos nas estradas brasileiras precisa ser assimétrico. Os efeitos esperados para este modelo são: apresentar o fenômeno de inversão do uso de faixa, baixo efeito ping-pong e o fluxo em cada faixa deve ser semelhante ao fluxo em modelos de faixa única.

Como mencionado anteriormente as regras de mudança de faixa podem ser divididas em regras de motivação e regras de segurança:

- i) As regras de motivação são aquelas que traduzem as razões que motivam um determinado

motorista a mudar de faixa. Geralmente os motivos que determinam o desejo de mudança de faixa estão ligados a condições da faixa que o motorista está, por exemplo a posição e a velocidade dos veículos da mesma faixa (Figura 7.2). Representa-se por d a distância até o veículo à frente e v_F a velocidade desse veículo e d_T a distância para o veículo atrás e v_T a velocidade desse veículo.

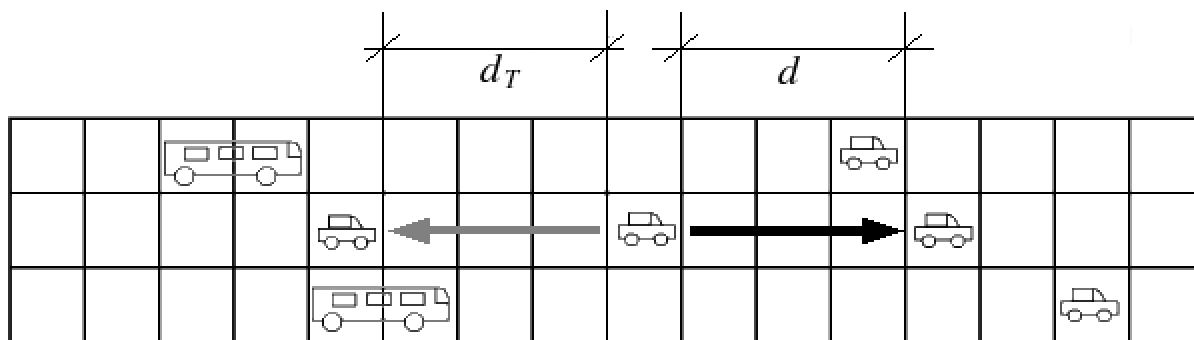


Figura 7.2: Variáveis: veículos da mesma faixa.

Para simular um tráfego com mudança assimétrica de faixa define-se uma regra para mudança da direita para a esquerda e uma da esquerda para a direita.

Apesar das regras serem definidas entre duas faixas essas podem ser facilmente estendidas para pistas com mais faixas [15].

ii) As regras de segurança são utilizadas quando um motorista está motivado a mudar de faixa. Neste caso o seu próximo passo é verificar se esta mudança pode ser efetuada, ou seja, se a faixa destino comporta seu veículo e sua velocidade.

Para esta avaliação cada motorista utiliza as informações dos veículos próximos a ele na faixa para a qual ele deseja realizar tal mudança.

Quando um motorista está motivado em realizar a mudança para a faixa da direita ele deve avaliar a posição e velocidade dos veículos próximos a ele nessa faixa, Figura 7.3, sendo d_{TD} a distância para o veículo atrás e d_{FD} a distância para o veículo da frente e v_{TD} e v_{FD} suas respectivas velocidades. Analogamente se um motorista está motivado em realizar uma mudança para a faixa da esquerda, os veículos próximos a ele na faixa da esquerda devem ser considerados, como exemplificado na Figura 7.4, com as respectivas distâncias d_{TE} e d_{FE} e velocidade v_{TE} e v_{FE} . Esse conjunto de regras garante que um veículo mude de faixa sem riscos de colisões e sem que seja bloqueado por outro veículo na outra faixa.

A seguir, a critério de comparação de resultados, apresentam-se dois modelos assimétricos de mudança de faixas: o modelo com compensação de velocidade proposto em [6] e o modelo luz de freio multi-faixa [7].

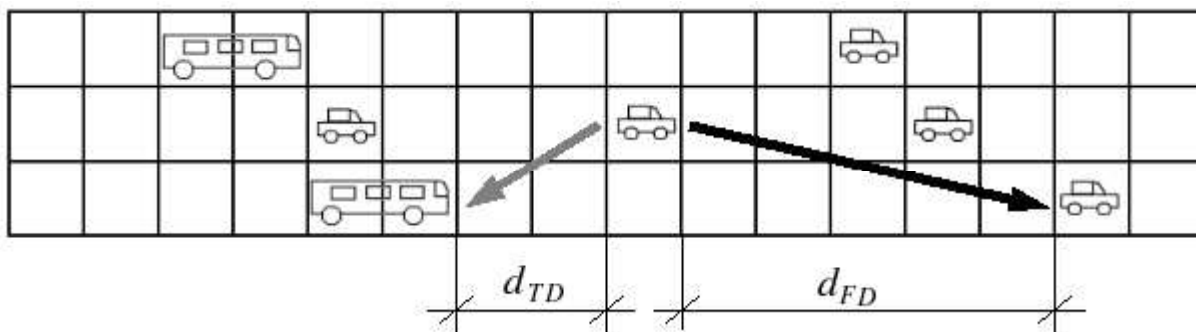


Figura 7.3: Variáveis: veículos da faixa à direita.

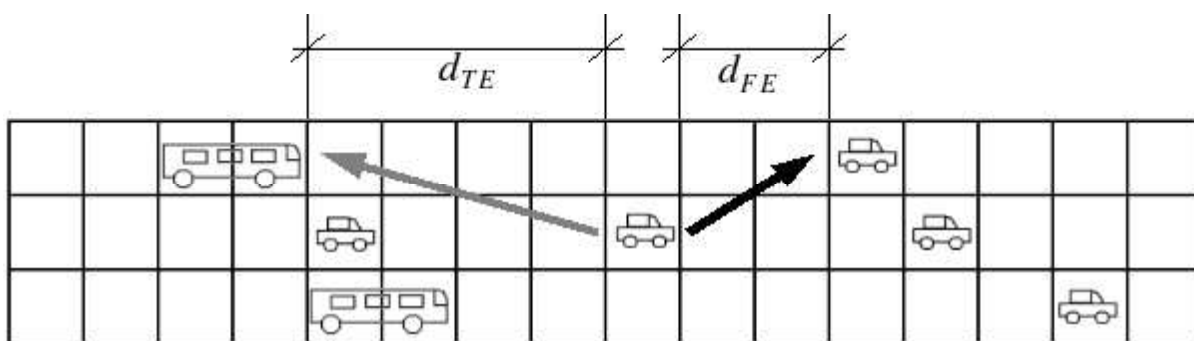


Figura 7.4: Variáveis: veículos da faixa à esquerda.

7.3.1 Modelo multi-faixas com compensação de velocidade

O modelo proposto em [6] utiliza um redutor de espaço para regular as mudanças de faixas. Esse redutor de espaço age como uma compensação da velocidade do veículo analisado, já que nesse modelo somente o espaço para o veículo na outra pista é considerado e não sua velocidade.

i) Mudança da direita para a esquerda

A mudança da faixa da direita para a faixa da esquerda, nesse modelo, é realizada quando um motorista percebe que não poderá andar na velocidade máxima e quando o espaço na faixa da esquerda é maior que o da faixa da direita.

- *Regra 1:*

$$(v_{max} > d - 1) \text{ e } (d_{FE} \geq d)$$

Como na maioria dos modelos de múltiplas faixas, a regra de segurança é definida de forma que o veículo não bloqueie e não seja bloqueado por outro. Nesse modelo não se leva em consideração a velocidade de outros veículos, somente a do veículo analisado. Por isso a velocidade do veículo na outra pista é definida como a máxima. Assim tem-se a regra de segurança definida como:

- *Regra 2:*

$$(v_{max} \leq d_{TE} - 1) \text{ e } (v \leq d_{FE} - 1)$$

ii) Mudança da esquerda para a direita

Como é desejado que um motorista mude para a faixa da direita quando a pista encontra-se com baixa densidade, é natural que essa regra leve em consideração as distâncias nas duas faixas. Nesse modelo essa avaliação é feita comparando-se as distâncias com o máximo deslocamento possível e além disso utiliza-se uma compensação de velocidade que diminui essa distância. Essa compensação torna possível o ajuste do ponto de inversão do uso das faixas, tornando o modelo mais próximo da realidade.

- *Regra 3:*

$$(v_{max} < d - 1 - v_{off}) \text{ e } (v_{max} < d_{FD} - 1 - v_{off})$$

onde v_{off} é a compensação da velocidade.

A regra de segurança é definida da mesma forma que a definida para a mudança da direita para a esquerda, ou seja:

- *Regra 4:*

$$(v_{max} \leq d_{TD} - 1) \text{ e } (v \leq d_{FD} - 1).$$

Pode-se observar na Figura 7.5, o resultado da aplicação dessas regras com o uso do modelo NaSch com $v_{max} = 6$, $p = 0.2$ e $v_{off} = 8$. O uso de $v_{off} = 8$ possibilitou a representação e a calibração do fenômeno de inversão do uso de faixa, Figura 7.5(a). A Figura 7.5(b) mostra que o uso desta configuração multi-faixas não aumenta o fluxo do modelo NaSch em nenhuma das faixas e a faixa da direita tem seu valor diminuído.

7.3.2 Modelo luz de freio multi-faixas.

Em [7] apresentam-se regras assimétricas de mudanças de faixas que fazem uso das regras do modelo luz de freio, apresentado no Capítulo 6. Nas regras propostas nesse trabalho, além do uso da variável luz de freio, que informa se um veículo freou no passo anterior, utiliza-se a antecipação do movimento proposta em [4], que estima o espaço para o deslocamento do veículo analisado baseado no mínimo que o veículo à sua frente irá se deslocar.

i) Mudança da direita para a esquerda

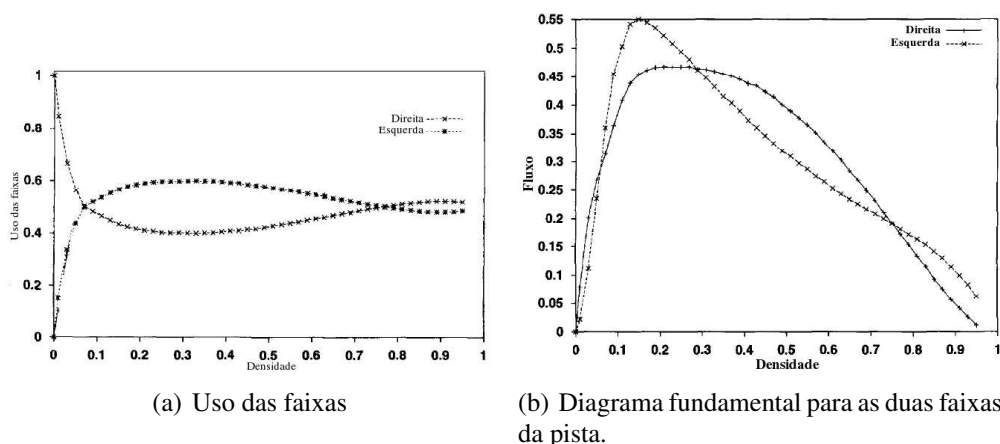


Figura 7.5: Resultados do modelo multi-faixa com compensação da velocidade [6].

A mudança da faixa da esquerda para a direita é realizada quando um motorista sente-se bloqueado por outro em sua faixa, ou seja $v > d$, e quando o motorista em questão não freou. Assim a regra de motivação de mudança da direita para esquerda desse modelo, pode ser escrita com:

- *Regra 1:*

$$(v \geq d) \text{ e } (b = 0)$$

sendo b a variável que informa quando o motorista freou.

Como mencionado anteriormente a mudança de faixa vai depender das regras de segurança que verificam se tal mudança é segura. Nesse modelo utiliza-se a distância efetiva na faixa destino, dada neste modelo por:

$$d_{sFE} = d_{FE} + \max\{\min(v_{FE}, gap_{FE}) - d_{seguro}, 0\}, \quad (7.1)$$

onde gap_{FE} é a distância que o veículo à frente na faixa da esquerda tem para se movimentar, como mostrado na Figura 7.6.

Uma mudança da faixa da direita para a esquerda será realizada se o veículo não bloquear outro que encontra-se nesta faixa, $d_{TE} \geq v_{TE}$, e se ele não for bloqueado por outro veículo na faixa da esquerda, contando com seu possível deslocamento ($d_{sFE} \geq v$). Assim a regra de segurança para esse modelo pode ser escrita como:

- *Regra 2:*

$$d_{TE} \geq v_{TE} \text{ e } d_{sFE} \geq v$$

ii) Mudança da esquerda para a direita

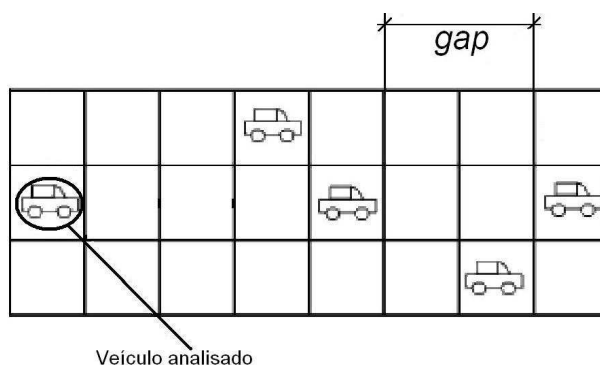


Figura 7.6: Representação da variável *gap*.

Para que um motorista tenha preferência pela faixa da esquerda quando a densidade é baixa, afim de gerar o efeito de inversão do uso de faixa, a regra de motivação da mudança da esquerda para a direita é descrita como:

- *Regra 3:*

$$(b = 0) \text{ e } d_{FD} > t_{h1}v_{FD} \text{ e } (d > t_{h2}v \text{ ou } (v > gap))$$

Nessa regra faz-se uso a noção de “distância temporal”, ou seja, um veículo está longe do outro se, mantidas as velocidades, um está a menos de um determinado tempo do outro (t_{h1} e t_{h2}), Figura 7.7. Assim, com o uso dessa regra, um motorista será incentivado a mudar da faixa da esquerda para a da direita se: não acabou de frear e o veículo à sua frente na faixa da esquerda não está próximo, ou seja, mais longe que t_{h1} segundos e se o veículo à sua frente está longe (mais de t_{h2} segundos) ou tem espaço para o deslocamento inferior à velocidade do veículo analisado.

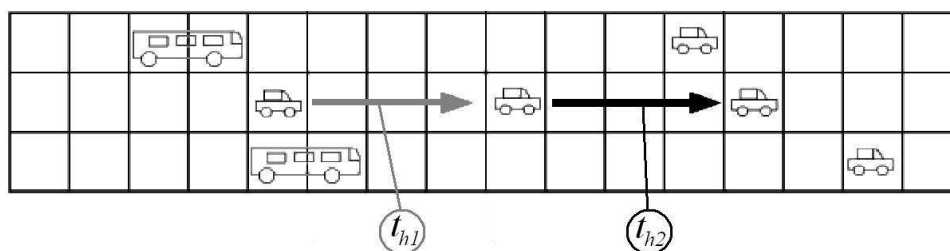


Figura 7.7: “Distância temporal”.

A regra de segurança pode ser definida da mesma forma que para a mudança da direita para a esquerda, ou seja:

- *Regra 4:*

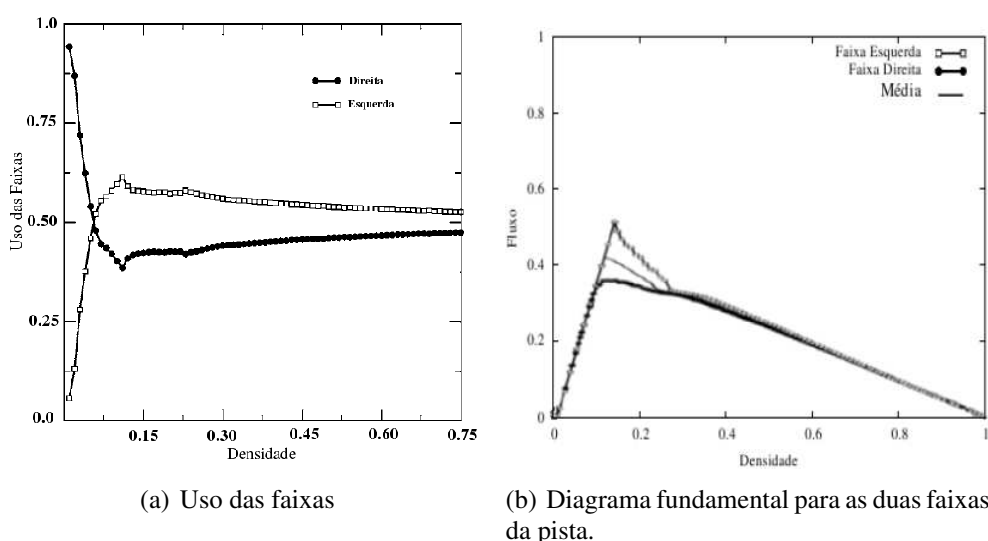


Figura 7.8: Resultados do modelo luz de freio multi-faixa [7].

$$d_{TD} \geq v_{TD} \quad \text{e} \quad d_{sFD} \geq v$$

A Figura 7.8 apresenta resultados deste modelo, uso das faixas e diagrama fundamental, utilizando os parâmetros apresentados na Seção 6.2, $t_{h1} = 3$ e $t_{h2} = 6$.

O fenômeno de inversão do uso das faixas é bem representado por esse modelo, Figura 7.8(a). Pode-se ver na Figura 7.8(b) que o fluxo na faixa da direita é ligeiramente menor que o fluxo da faixa da esquerda, e além disso o diagrama fundamental da faixa mais lenta, da direita, não representa a região de meta-estabilidade.

7.4 Modelo proposto

O modelo proposto nesse trabalho baseia-se nas regras de mudanças de faixas propostas no modelo luz de freio [7], por se tratar também de um modelo antecipatório, como no modelo de faixa única proposto no Capítulo 5, e apresentar bons resultados. No modelo luz de freio utiliza-se somente o espaço à frente do veículo líder para se calcular a antecipação e no modelo proposto, para calcular a antecipação, utiliza-se somente a velocidade do veículo líder.

i) Mudança da direita para esquerda

Para simular a intenção de um motorista mudar para a faixa da esquerda leva-se em consideração o código de trânsito que diz que esta faixa é destinada à ultrapassagens. Logo, se um motorista sentir-se bloqueado por um veículo à sua frente este desejará mudar para a faixa da esquerda para ultrapassá-lo. Logo define-se:

a) Regra de motivação

Para avaliar se um veículo está sendo bloqueado por outro utiliza-se a distância efetiva, descrita no Capítulo 5, ou seja, $d_s = d + [(1 - \alpha) \cdot v_F]$ (Figura 7.9), em que α assume o máximo da função densidade de probabilidade, l_4 na Figura 5.4. Então o critério de motivação à mudança de faixa para esquerda é dado pela seguinte condição:

- *Regra 1:*
 $v \geq d_s$

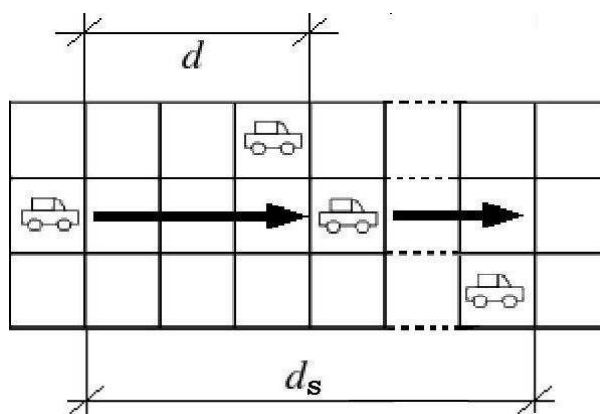


Figura 7.9: Distância efetiva à frente.

b) Regras de segurança

Para que a mudança de faixa possa ser realizada existe a necessidade de haver espaço suficiente para que o veículo possa realizar tal mudança com segurança, ou seja, o espaço efetivo à frente desse veículo na faixa da esquerda (d_{sFE}) deve ser suficiente para seu deslocamento e o veículo não deve bloquear outros que já estão nesta faixa (d_{TE}) (Figura 7.10), ou seja:

- *Regra 2:*
 $v < d_{sFE}$ e $d_{TE} > v_{TE}$

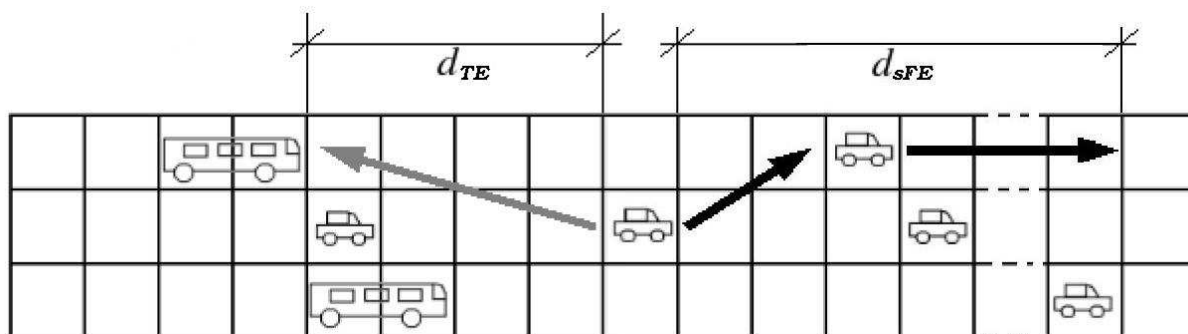


Figura 7.10: Distância efetiva à esquerda.

Essas duas regras garantem que os veículos mudem de faixa sem haver perigo de colisões.

ii) Mudança da esquerda para direita

Como de acordo com o código brasileiro de trânsito as faixas da direita são destinadas a veículos lentos ou de grande porte, propõe-se as seguintes regras de motivação.

a) Regras de motivação

Quando um motorista que está à esquerda percebe a aproximação de um veículo mais rápido, este deve ir para a faixa da direita para dar passagem a este veículo mais rápido. Isso é viabilizado nas seguintes condições:

- *Regra 3:*
 $v < v_T$ e $d_T < t_{h1}v_T$.

Pode-se observar que nessa regra de motivação também utiliza-se a noção de “distância temporal” (Seção 7.3.2), ou seja, verifica se o veículo detrás está a menos de t_{h1} segundos do veículo considerado (Figura 7.7).

No caso do veículo da frente estar muito distante, o motorista também opta por mudar para a pista da direita. Para simular essa inversão de pista utiliza-se a relação:

- *Regra 4:*
 $d > t_{h2}v$

Novamente utiliza-se a noção de “distancia temporal”. Quando o veículo à frente está a mais de t_{h2} segundos de distância, Figura 7.7, então a pista tem sua densidade baixa naquele trecho e não existe a necessidade de permanecer na faixa da esquerda.

Para que um motorista fique motivado a mudar para a direita basta que uma das regras seja satisfeita. A *Regra 3* garante que os veículos mais lentos estarão na faixa da direita e a *Regra 4* garante a preferência dos motoristas pela pista da direita quando em trânsito livre, simulando assim o fenômeno de inversão do uso da faixa.

b) Regras de segurança

Para segurança da mudança de faixa utilizou-se regras análogas às utilizadas para mudança da direita para esquerda, ou seja:

- *Regra 5*
 $v < d_{sFD}$ e $d_{TD} > v_{TD}$

como mostrado na Figura 7.11.

Há também a necessidade de se prever situações onde a faixa da esquerda está bloqueada, por exemplo, por algum acidente, necessitando de um desvio pela direita. Para se resolver essa

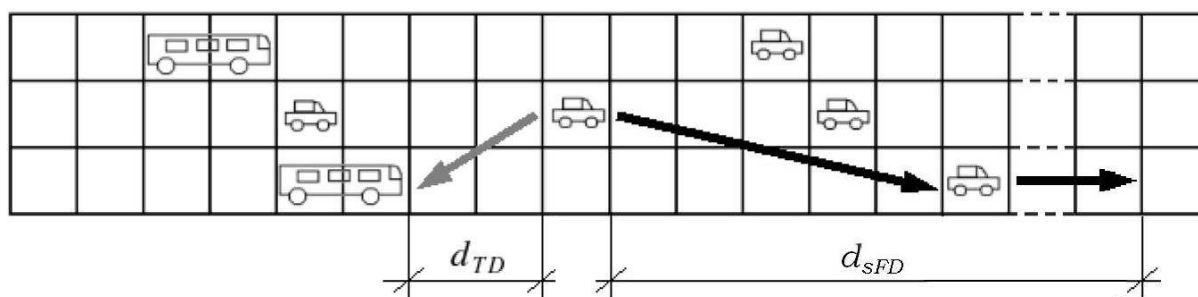


Figura 7.11: Distância efetiva à direita.

situação adicionou-se, nesse caso, uma regra de mudança simétrica entre as faixas, que consiste em: caso o motorista, mesmo ao longe, perceba durante um determinado intervalo de tempo (b_{lk}) que o veículo à sua frente está parado, ele fica motivado a mudar de faixa.

- *Regra 6*

$$t_{blk} > b_{lk}.$$

7.5 Simulações realizadas

Com o objetivo de validar o modelo de mudanças de faixas proposto realizou-se, inicialmente, uma simulação para avaliar as características do diagrama fundamental, da inversão de uso de faixa e da frequência dessas mudanças.

Na segunda simulação apresentam-se os resultados da movimentação dos veículos quando surge algum tipo de impedimento na pista da esquerda e nesse caso os motoristas fazem ultrapassagens pela direita.

As simulações foram feitas usando condições de contorno periódicas, com duas faixas de 300 células cada. e foi utilizado comportamento de alta agressividade como descrito no Capítulo 5.

7.5.1 Simulação 1

Esta primeira simulação tem por objetivo comparar os resultados do modelo multi-faixa aqui proposto, com outros encontrados na bibliografia, em particular os modelos assimétricos de compensação da velocidade [6] e o modelo luz de freio [7], descritos anteriormente. Para isso apresenta-se inicialmente o gráfico com a frequência de mudanças de faixa, para definir a probabilidade a ser utilizada para evitar o efeito ping-pong. A seguir os gráficos de uso de faixa e o diagrama fundamental do tráfego, são apresentados.

i) Para avaliar o comportamento do modelo de mudança de faixa em relação a outros modelos propostos [6, 7, 34, 40, 45], foi utilizado na *Regra 3*, $t_{h1} = 3$ e $t_{h2} = 6$ na *Regra 4* como sugerido em [7]. Apresenta-se na Figura 7.12 a relação entre a frequência de mudanças de faixa e a densidade total da pista, para diferentes probabilidades de um motorista não realizar mudanças. Essa frequência de mudanças de faixas deve ter seu máximo em uma densidade $\rho = 0,5$, atingindo nesse ponto, o valor aproximado de 800 mudanças de faixas em um quilômetro por uma hora e deve tender a zero para altas e baixas densidades, como descrito em [7]. Como pode ser visto na Figura 7.12 a inclusão da probabilidade de não mudar de faixa gera, como esperado, uma variação da frequência de mudanças alcançando o valor esperado com a probabilidade de 50%.

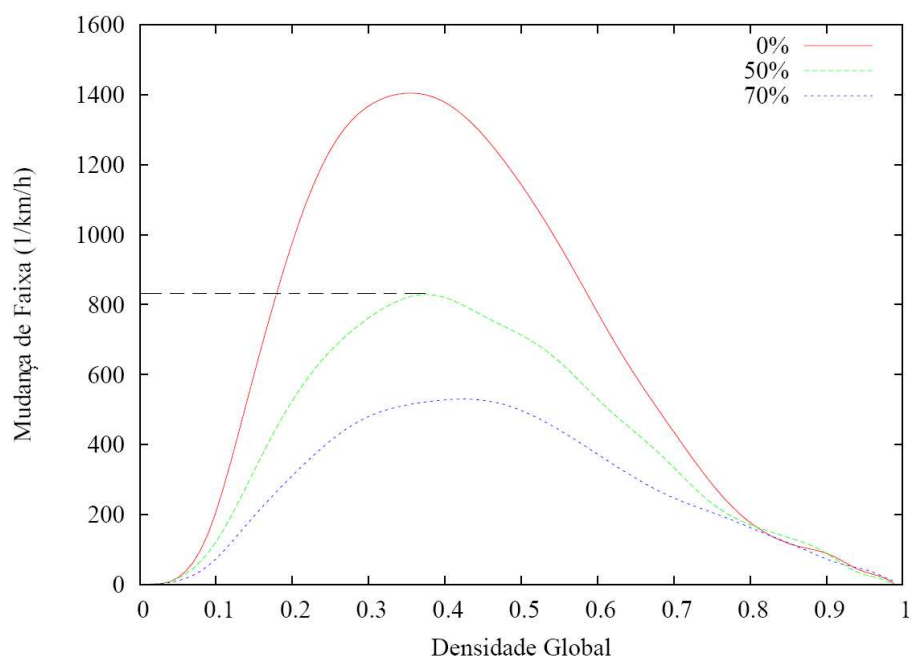


Figura 7.12: Frequência de mudanças de faixa.

ii) A seguir apresenta-se na Figura 7.13, o gráfico da inversão do uso de faixa, para o caso de probabilidade de 50% do motorista optar por não trocar de faixa, mesmo podendo. Pode-se observar que o algoritmo reproduz o efeito de inversão desejado, tendendo para uma utilização homogênea das faixas quando $\rho \rightarrow 1$. A escolha de t_{h1} e t_{h2} tem efeito neste gráfico, da seguinte forma: (i) alterando a densidade em que ocorre a inversão do uso, quando se altera t_{h1} , e (ii) o ponto máximo de uso da faixa da esquerda e o mínimo de uso da faixa da direita quando altera-se o valor de t_{h2} .

iii) A Figura 7.14 apresenta o diagrama fundamental para cada faixa da via. Pode-se observar que o fluxo na faixa da esquerda não sofre alteração quando comparado com o modelo de faixa simples, enquanto que na faixa da direita tem-se uma pequena diminuição no fluxo.

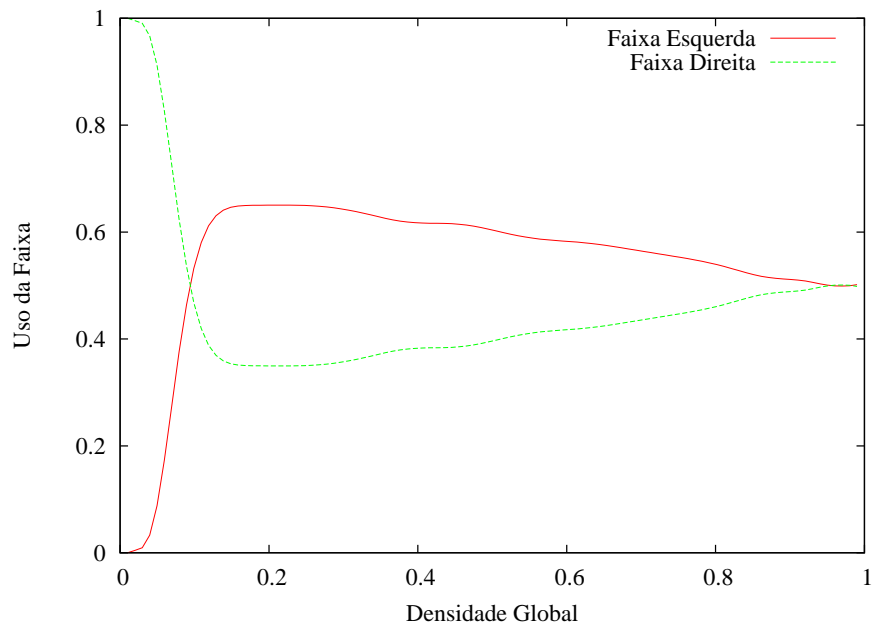


Figura 7.13: Distribuição do uso das faixas.

Esse efeito é esperado já que os veículos mais lentos mudam para a faixa da direita, contudo a capacidade da faixa da esquerda não tem sua capacidade aumentada pela presença de veículos rápidos, uma vez que sua capacidade continua a mesma.

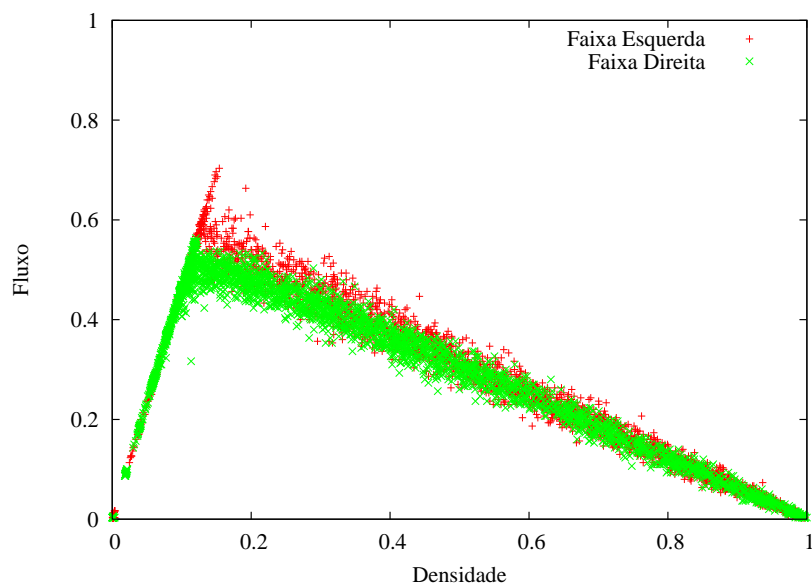


Figura 7.14: Diagrama fundamental para as duas faixas da pista.

7.5.2 Simulação 2

Como dito anteriormente, o objetivo dessa simulação é verificar o comportamento do algoritmo quando a faixa da esquerda encontra-se bloqueada. Para isso, utilizou-se $b_{lk} = 3$ segundos na

Regra 6, $t_{h1} = 3$, $t_{h2} = 6$ e 50% de probabilidade de trocar de faixa.

Para simular esse comportamento escolheu-se um veículo ao acaso e gerou-se um acidente, ou seja, o veículo tem sua velocidade alterada para zero ($v = 0$) e não pode mudar nem de faixa nem de velocidade.

Na Figura 7.15 cada veículo está identificado pela sua cor e os números dentro das células representam as velocidades desses veículos. O veículo preto é o que terá sua velocidade alterada para zero e identificado pela letra **P** nesse momento.

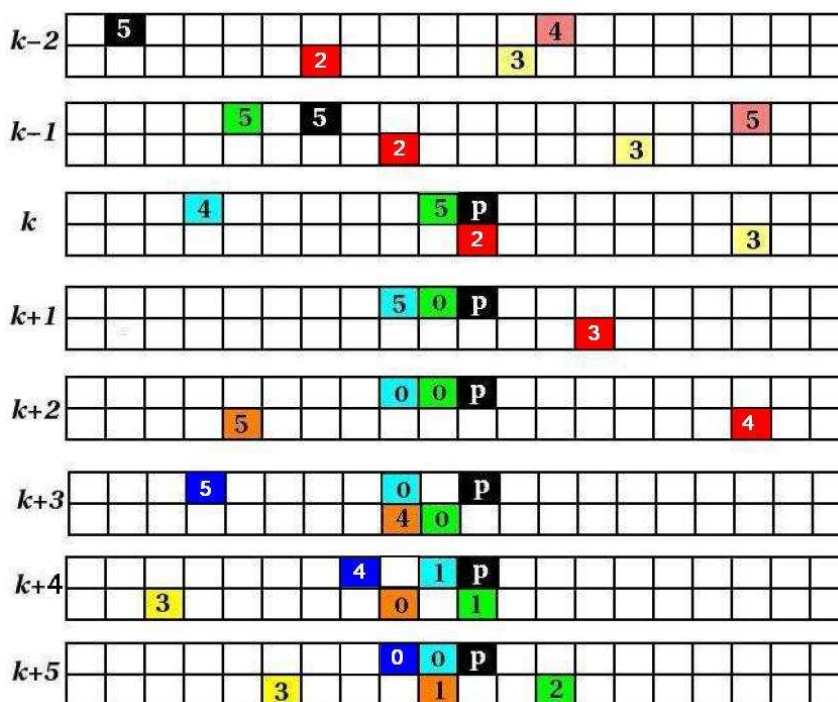


Figura 7.15: Exemplo de acidente em uma das pistas.

Pode-se observar que o acidente em uma das faixas provoca um engarrafamento no sentido contrário ao tráfego, nas duas faixas como esperado, e um fluxo livre após o acidente. Além disso, nos testes realizados observou-se que, mesmo com alta densidade, os veículos sempre escapam do acidente, pois nesse caso a velocidade dos veículos é baixa gerando assim vários espaços para mudança de faixa.

Capítulo 8

Conclusões

Neste trabalho apresenta-se um modelo microscópico de simulação do tráfego viário, baseado em autômatos celulares. O modelo proposto é um procedimento explícito e que, apesar disso, considera uma expectativa de movimento dos veículos no tempo considerado. Para evitar colisões irreais, caso algum veículo não se movimenta conforme esperado, propõe-se um processo iterativo que recalcula a velocidade de todos os veículos que podem ser afetados por essa mudança de expectativa. Além disso, possibilita-se a diferenciação do comportamento dos motoristas na via considerada, através da definição de uma função densidade de probabilidade.

Pode-se observar que a antecipação do movimento é indispensável para aumentar o fluxo de veículos em um modelo TCA, utilizando-se a probabilidades de randomização da velocidade sugeridos na bibliografia. Um dos grandes problemas da antecipação, o risco de acidentes, pode ser solucionado com a adição de uma rotina explícita no tempo. Essa rotina recalcula a velocidade dos veículos por um processo iterativo e apesar disso não gera um grande aumento no número de cálculos e não torna o algoritmo implícito no tempo.

O modelo proposto tem a vantagem de apresentar um bom resultado com poucas variáveis de calibração e ser estável, ou seja, tendo seus resultados não dependentes da configuração inicial. Mostrou-se ainda que uma configuração multi-célula não afeta as principais características desse modelo, possibilitando a consideração tanto de diferentes possibilidades de aceleração e desaceleração, quanto a diferença entre veículos mais lentos e veículos rápidos. O esquema multi-célula também permite a consideração de diferentes tamanhos de veículos.

O esquema de múlti-faixas permite a simulação de situações mais reais e com as regras propostas para mudança de faixa, o modelo possibilitou uma boa frequência de mudanças de faixa, apresentando resultados desejados, para uma probabilidade de não realizar tais mudanças em 50% dos casos. Nesse caso o diagrama fundamental apresentou bons resultados se comparado

a outros encontrados na bibliografia e com um número menor de parâmetros a serem calibrados. Apesar do modelo ter sido proposto para regras de tráfego assimétricas, o procedimento sugerido possibilitou que motoristas desviassem para a direita quando a pista da esquerda está bloqueada. Observou-se, nestes casos, uma lenta resposta do motorista para troca de faixa, que não foi alterada para diferentes valores de t_{h1} , t_{h2} e blk . Esse procedimento pode ser melhorado considerando-se também para esse caso, a distância efetiva.

Sugere-se, como continuidade do trabalho, estudar melhorias no modelo multi-faixa para, por exemplo, melhorar a percepção de um motorista sobre a necessidade de se mudar de faixa. Além disso, no modelo como descrito, um motorista com velocidade 0 (Zero) pode trocar de faixa. Sugere-se que nesse caso, as trocas de faixas só sejam permitidas com velocidades maiores que zero. Uma outra sugestão para a extensão desse trabalho é considerar-se outros tipos de pistas, como pistas abertas, simulando entradas e saídas de veículos, bem como a paralelização do algoritmo proposto.

Os resultados encontrados abrem caminho para um estudo mais aprofundado da interação entre veículos, gerando assim a possibilidade de se desenvolver ferramentas de controle ou informação sobre a situação do tráfego que possam resultar em uma melhora na qualidade de vida nos grandes centros urbanos.

Referências

- [1] SILVA, P. C. M. da. *apostila: Teoria do fluxo de tráfego*. Material didático do curso de engenharia de tráfego da Universidade de Brasília, 2001.
- [2] EMMERICH, H.; RANK, E. An improved cellular automaton model for traffic flow simulation. *Physica A*, v. 234, p. 676–686, 1997.
- [3] HELBING, D.; SCHRECKENBERG, M. Cellular automata simulating experimental properties of traffic flow. *Physical Review E*, v. 59, n. 3, p. R2505–R2508, 1999.
- [4] KNOSPE, W. et al. Towards a realistic microscopic description of highway traffic. *J. Phys. A: Math. Gen.*, v. 33, p. L477–L485, 2000.
- [5] KERNER, B. S.; KLENOV, S. L. Microscopic theory of spatial-temporal congested traffic patterns at highway bottlenecks. *Physical Review E*, v. 68, n. 036130, 2003.
- [6] WAGNER, P.; NAGEL, K.; WOLF, D. E. Realistic multi-lane traffic rules for cellular automata. *Physica A*, v. 234, p. 687–698, 1997.
- [7] KNOSPE, W. et al. A realistic tow-lane traffic model for highway traffic. *J. Phys. A: Math. Gen.*, v. 35, p. 3369–3388, 2002.
- [8] JIANG, R.; WU, Q. A stopped time dependent randomization cellular automata model for traffic flow controlled by traffic light. *Physica A*, v. 364, p. 493–496, 2006.
- [9] LIA, X.-G. et al. A realistic two-lane cellular automata traffic model considering aggressive lane-changing behavior of fast vehicle. *Physica A*, v. 367, p. 479–486, 2006.
- [10] LÁRRAGA, M.; RÍO, J.; SCHADSCHNEIDE, A. Platoon formation in a traffic model with variable anticipation. *ArXiv Condensed Matter e-prints*, 2003.
- [11] NASSABA, K. et al. Effect of the lane reduction in the cellular automata models applied to the two-lane traffic. *Physica A*, v. 369, p. 841–852, 2006.
- [12] WU, Q.-S.; LI X.-B. AND HU, M.-B.; JIANG, R. Study of traffic flow at an unsignalized t-shaped intersection by cellular automata model. *The European Physical Journal B*, v. 48, p. 265269, 2005.
- [13] NRW, T. S. in. *Expansion for the motorway network*. Referência on-line em <http://www.autobahn.nrw.de/>, visitado pela última vez em 16 de agosto de 2007.
- [14] NAGEL, K.; SCHRECKENBERG, M. A cellular automaton model for freeway traffic. *J. Phys. I France*, v. 2, p. 2221–2229, 1992.
- [15] KERNER, B.; KLENOV, S.; WOLF, D. Cellular automata approach to three-phase traffic theory. *J. Phys. A: Math. Gen.*, v. 35, p. 9971–10013, 2002.

- [16] NAGEL, K.; PACZUSKI, M. Emergent traffic jams. *Phys. Rev. E*, v. 51, n. 4, p. 2909–2918, 1995.
- [17] NISHINARI, K.; FUKUI, M.; SCHADSCHNEIDER, A. A stochastic cellular automaton model for traffic flow with multiple metastable states. *J. Phys. A: Math. Gen.*, v. 37, p. 3101–3110, 2004.
- [18] LÁRRAGA, M.; RÍO, J. A.; SCHADSCHNEIDER, A. New kind of phase separation in a ca traffic model with anticipation. *J. Phys. A: Math. Gen.*, v. 37, p. 3769–3781, 2004.
- [19] MAY, A. D. *Traffic Flow Fundamentals*. [S.l.]: Prentice Hall, 1997.
- [20] DAGANZO, C. F. *Fundamentals of Transportation and Traffic Operations*. [S.l.]: Elsevier Science, 1997.
- [21] CHOWDHURY, D.; SANTEN, L.; SCHADSCHNEIDER, A. Statistical physics of vehicular traffic and some related systems. *Physics Reports*, v. 329, p. 199–329, 2000.
- [22] LIGHTHILL, M.; WHITHAM, G. On kinematic waves. *Proc. Roy. Soc. A*, v. 229, p. 281–345, 1955.
- [23] ROTHERY, R. et al. Traffic flow theory. *Transportation Research Board (TRB) Special Report*, v. 165, 1998.
- [24] PIPES, L. A. An operational analysis of traffic dynamics. *J. Appl. Phys.*, v. 24, p. 274–281, 1953.
- [25] CHANDLER, R. E.; HERMAN, R.; MONTROLL, E. W. Traffic dynamics: studies in car following. *Operations Research*, v. 6, p. 165–184, 1958.
- [26] GAZIS, D.; HERMAN, R.; POTTS, R. Car-following theory of steady-state traffic flow. *Operations Research*, v. 7, n. 4, p. 499–505, 1959.
- [27] BANDO, M. et al. Dynamical model of traffic congestion and numerical simulation. *Phys. Rev. E*, v. 51, n. 2, p. 1035–1042, 1995.
- [28] NAKANISHI, K. et al. Solvable optimal velocity models and asymptotic trajectory. *Phys. Rev. E*, v. 55, n. 6, p. 6519–6532, 1997.
- [29] BEYER, W. A.; SELLERS, P. H.; WATERMAN, M. S. Stanislaw M. Ulam’s contributions to theoretical theory. *Springer Netherlands*, v. 10, p. 231–242, 1985.
- [30] PESAVENTO, U. An implementation of Von Neumann’s self-reproducing machine. *Princeton University*, v. 2, n. 4, p. 337–354, 1995.
- [31] CONWAY, J. H. Mathematical games. *Scientific American*, v. 223, p. 120–123, 1970.
- [32] WOLFRAM, S. *A New Kind of Science*. [S.l.]: Wolfram Media, 2001.
- [33] MAERIVOET, S.; MOOR, B. D. Transportation planning and traffic flow models. *Katholieke Universiteit Leuven, Department of Electrical Engineering ESAT-SCD (SISTA)*, p. 05–155, 2005.

- [34] LI, F.; ZHANG, X.-Y.; GAO, Z.-Y. The effect of restricted velocity in two-lane on-ramp system. *Physica A*, v. 374, p. 827–834, 2007.
- [35] WAHLE, J. et al. A cellular automaton traffic flow model for online simulation of traffic. *Parallel Computing*, v. 27, p. 719–735, 2001.
- [36] FUKUI, M.; ISHIBASHI, Y. Traffic flow in 1d cellular automata model including cars moving high speed. *Journal of the Physical Society of Japan*, v. 65, n. 6, p. 1868–1870, 1996.
- [37] BARLOVIC, R. et al. Metastable states in cellular automata for traffic flow. *Eur. Phys. J. B.*, v. 5, n. 3, p. 793–800, 1998.
- [38] BENJAMIN, S.; JOHNSON, N.; HUI, P. Cellular automata models of traffic flow along a highway containing a junction. *J. Phys. A: Math. Gen.*, v. 29, p. 3119–3127, 1996.
- [39] TAKAYASU, M.; TAKAYASU, H. $1/f$ noise in a traffic model. *Fractals 1*, p. 860–866, 1993.
- [40] CHOWDHURY, D.; WOLF, D. E.; SCHRECKENBERG, M. Particle hopping models for two-lane traffic with two kinds of vehicles: Effects of lane-changing rules. *Physica A*, v. 235, p. 417–439, 1997.
- [41] HAFSTEIN, S. et al. A high-resolution cellular automata traffic simulation model with application in a freeway traffic information system. *Computer Aided Civil and Infrastructure Engineering*, v. 19, n. 5, p. 338–350, 2004.
- [42] PAULA, A. de. *Código Brasileiro de Trânsito Lei 9.503/97*. [S.l.]: DPeA, 2002.
- [43] NAGATANI, T. Dynamical jamming transition induced by a car accident in traffic-flow model of a two-lane roadway. *Physica A: Statistical and Theoretical Physics*, v. 202, p. 449–458, 1994.
- [44] RICKERT, M. et al. Two lane traffic simulations using cellular automata. *Physica A*, v. 231, p. 534–550, 1996.
- [45] NAGEL, K. et al. Two-lane traffic rules for cellular automata: A systematic approach. *Phys. Rev. E*, v. 58, n. 2, p. 1425–1437, 1998.

APÊNDICE A - Código computacional.

Para efetuar as simulações apresentadas neste trabalho, desenvolveu-se um código computacional na linguagem “C”. Para isto apresenta-se inicialmente a descrição do arquivo de entrada de dados a ser utilizado nas simulações. A seguir é apresentada uma descrição do código implementado contendo um fluxograma com os principais procedimentos desenvolvidos, as descrições desses procedimentos, bem como os procedimentos de suporte utilizados.

a)Entrada de dados: a entrada de dados é feita através de um arquivo texto. Para facilitar a reutilização do arquivo de dados, este deve conter em cada linha, um nome identificando a variável, sem espaços em branco, seguido de um espaço e do valor da variável, ou este valor pode ser colocado na linha seguinte como apresentado na Figura A.1. A primeira informação contém tipo de simulação desejada, ou seja:

- 1) Escolha_do_tipo_de_simulação_1-antecipação_2-multi-célula_3-multi-faixa: o usuário deve fornecer 1 (um) para simular o modelo de antecipação proposto (Capítulo 5); 2 (dois) se desejar realizar a simulação com multi-célula descrita no Capítulo 6; e 3 (três) se desejar fornecer informações para simulação com multi-faixas, Capítulo 7.
- i) Para simulação do algoritmo proposto, no Capítulo 5, o arquivo deve também conter as seguintes informações:
 - 2) Determina_como_vai_ser_a_distribuição_inicial_dos_veículos_0-homogêneo_e_1-engarrafado: se o usuário desejar uma configuração inicial da pista homogênea deve colocar 0 e 1 para uma configuração inicial descrevendo um grande engarrafamento.
 - 3) Número_de_células_da_pista_tamanho_da_pista: número total de células da pista.
 - 4) Velocidade_máxima_da_pista: velocidade máxima dos veículos.
 - 5) Número_de_passos_totais_a_serem_realizadas: número de passos a serem realizados para cada densidade.

```

Escolha_do_tipo_de_simulação_1-antecipação_2-multi-célula_3-multi-faixa
1
Determina_como_vai_ser_a_distribuição_inicial_dos_veiculos_0-homogêneo_e_1-engarrafado
0
Número_de_células_da_pista_tamanho_da_pista
300
Velocidade_máxima_da_pista
5
Número_de_passos_totais_a_serem_realizadas
10000
Número_de_passos_desprezados_do_total_de_passos
1000
Limite_inferior_da_região_1_da_FDP
0.0
Limite_superior_da_região_1_e_inferior_da_região_2_da_FDP
0.2
Limite_superior_da_região_2_e_inferior_da_região_3_da_FDP
0.4
Limite_superior_da_região_3_da_FDP
0.7
Probabilidade_de_alfa_ser_escolhida_na_região_1
0.8
Probabilidade_de_alfa_ser_escolhida_na_região_2
0.15
Probabilidade_de_alfa_ser_escolhida_na_região_3
0.05
Probabilidade_de_randomização_da_velocidade:
35
Quantidade_de_células_que_um_veiculo_ocupa
1
Aceleração_máxima_que_um_veiculo_pode_ter
1
Percentual_de_veiculos_rapidos
0
Percentual_de_veiculos_longos
0
Número_de_faixas_da_pista
1
Tolerância_a_bloqueio
3
Probabilidade_de_mudar_de_faixa
0.50
Distancia_temporal_1
6
Distancia_temporal_2
3

```

Figura A.1: Exemplo de arquivo de entrada de dados.

- 6) Número_de_passos_desprezados_do_total_de_passos: número de passos iniciais a serem desprezados para que não haja influência direta da configuração inicial da pista nos resultados.
- 7) Limite_inferior_da_região_1_da_FDP: valor de $\alpha = l_1$ (Figura 5.4).
- 8) Limite_superior_da_região_1_e_inferior_da_região_2_da_FDP: valor de $\alpha = l_2$ (Figura 5.4).
- 9) Limite_superior_da_região_2_e_inferior_da_região_3_da_FDP: valor de $\alpha = l_3$ (Figura 5.4).
- 10) Limite_superior_da_região_3_da_FDP: valor de $\alpha = l_4$ (Figura 5.4).
- 11) Probabilidade_de_alfa_ser_escolhida_na_região_1: de probabilidade do α ser escolhido no intervalo $[l_1, l_2)$.
- 12) Probabilidade_de_alfa_ser_escolhida_na_região_2: probabilidade do α ser escolhido no intervalo $[l_2, l_3)$.
- 13) Probabilidade_de_alfa_ser_escolhida_na_região_3: probabilidade do α ser es-

colhido no intervalo $[l_3, l_4]$.

- 14) Probabilidade_de_randomização_da_velocidade: probabilidade de randomização da velocidade de um veículo p_m .
- ii) Para simulação do algoritmo de antecipação com um esquema multi-célula, Capítulo 6, além dos dados anteriores, devem ser fornecidos:
- 15) Quantidade_de_células_que_um_veiculo_ocupa: número de células que cada veículo ocupa.
 - 16) Aceleração_máxima_que_um_veículo_pode_ter: valor inteiro e divisível por dois que determina o valor de aceleração e desaceleração maiores que uma célula por segundo (a_{max}).
 - 17) Percentual_de_veículos_rapidos: probabilidade de um veículo desacelerar a metade do valor da aceleração máxima (p_{vr}).
 - 18) Percentual_de_veículos_longos: percentual inteiro de veículos longos presentes na pista, ou seja, ocupando o dobro de células que os outros.
- iii) Se o usuário desejar simular o algoritmo de antecipação com multi-célula e multi-faixa (Capítulo 7), além dos dados fornecidas no item (ii), será necessário fornecer as seguintes informações:
- 19) Número_de_faixas_da_pista: número de faixas da pista a ser simulada.
 - 20) Tolerância_a_bloqueio: número de instantes de tempo que um motorista espera, atrás de outro parado, até tomar a decisão de mudar de pista.
 - 21) Probabilidade_de_mudar_de_faixa: probabilidade de um determinado veículo motivado a mudar de faixa, realizar tal mudança.
 - 22) Distancia_temporal_1: distância temporal em segundos t_{h1} , inteiro, que determina se um veículo que se aproxima, está próximo.
 - 23) Distancia_temporal_2: distância temporal em segundos, t_{h2} , inteiro, que determina se o veículo analisado, está longe do veículo à sua frente.

Caso deseje-se simular o modelo de antecipação proposto com multi-faixa usando um esquema de células simples, como nos exemplos do Capítulo 7, apresentado no presente trabalho, os dados do item (ii) devem ser fornecidos adequadamente.

b)Descrição do código: o código foi desenvolvido variando-se a densidade de 0,01 até 0,99 e como o objetivo deste trabalho é testar o comportamento dos algoritmos apresentados, o código foi elaborado para o caso mais geral, algoritmo de antecipação multi-célula e

multi-faixa. Apresenta-se, na Figura A.2, um fluxograma com os principais módulos implementados, onde:

i) Configuração inicial: neste módulo define-se a configuração inicial da pista, ou seja:

- *int engarrafado(veiculo *pista, int n, int num_carros, int pcaminhoes)*:
preenche o vetor *pista*, de tamanho *n*, com todos os veículos (*num_carros*) em um mesmo engarrafamento sendo uma porcentagem desses, veículos lentos (*pcaminhoes*).
- *int homoganeo(veiculo *pista, int n, int num_carros, int pcaminhoes)*:
preenche o vetor *pista*, de tamanho *n*, com todos os veículos (*num_carros*) igualmente distribuídos sendo uma porcentagem desses, veículos lentos (*pcaminhoes*).

ii) *int muda_de_faixa_BR(veiculo **pista, int n, int nfaixas, int v_maxima)*:

conjunto de regras de mudança de faixas, onde *n* é o tamanho de cada uma das faixas (*nfaixas*) do vetor *pista* de velocidade máxima *v_max*. Este módulo faz uso dos seguintes procedimentos:

- *int distancia_anterior(veiculo *pista, int n, int posi_car)*:
calcula a distância de um veículo, na posição *posi_car*, para o veículo de trás ambos no vetor *pista* de tamanho *n*.
- *int velocidade_anterior(veiculo *pista, int n, int posi_car)*:
retorna a velocidade do veículo atrás da posição *posi_car* do vetor *pista* de tamanho *n*.
- *int L_anterior(veiculo *pista, int n, int posi_car)*:
informa o tamanho do veículo atrás da posição *posi_car* do vetor *pista* de tamanho *n*.

iii) *int antec_clusters(veiculo *pista, int n, int vel_max)*:

atualiza a velocidade dos veículos, com as regras do modelo de antecipação proposto, para os veículos do vetor *pista* de tamanho *n* e com limite superior para a velocidade, *vel_max*. Além disso esse módulo utiliza os seguintes procedimentos:

- *int distancia_proximo(veiculo *pista, int n, int posi_car)*:
calcula a distância de um veículo, na posição *posi_car*, para o veículo da frente, ambos no vetor *pista* de tamanho *n*.
- *int velocidade_proximo(veiculo *pista, int n, int posi_car)*:
retorna a velocidade do veículo à frente da posição *posi_car* do vetor *pista* de tamanho *n*.

- *int cria_clusters(int *freados,int NF,int *espacos,int NE):*
 define os clusters de influência a serem recalculados. O vetor *freados* é um vetor contendo todas as posições dos veículos que frearam bruscamente; *NF* é o tamanho do vetor *freados*; *espacos* é um vetor contendo as posições dos veículos que não precisam de antecipação para manterem suas velocidades; e *NE* é o tamanho desse vetor.
- *int recalcula_freada(veiculo *pista,int n,int vel_max, int ini, int fim):*
 recalcula a velocidade dos veículos, do vetor *pista* de tamanho *n* compreendidos entre a posição *ini* e a posição *fim*, realizando somente as rotinas referentes a desaceleração do veículo, devido ao pouco espaço à sua frente, descritas no algoritmo proposto.

iv) *int atualiza_pista_antecipacao(veiculo *pista,int n,int k):*
 atualiza a posição dos veículos do vetor *pista* de tamanho *n*.

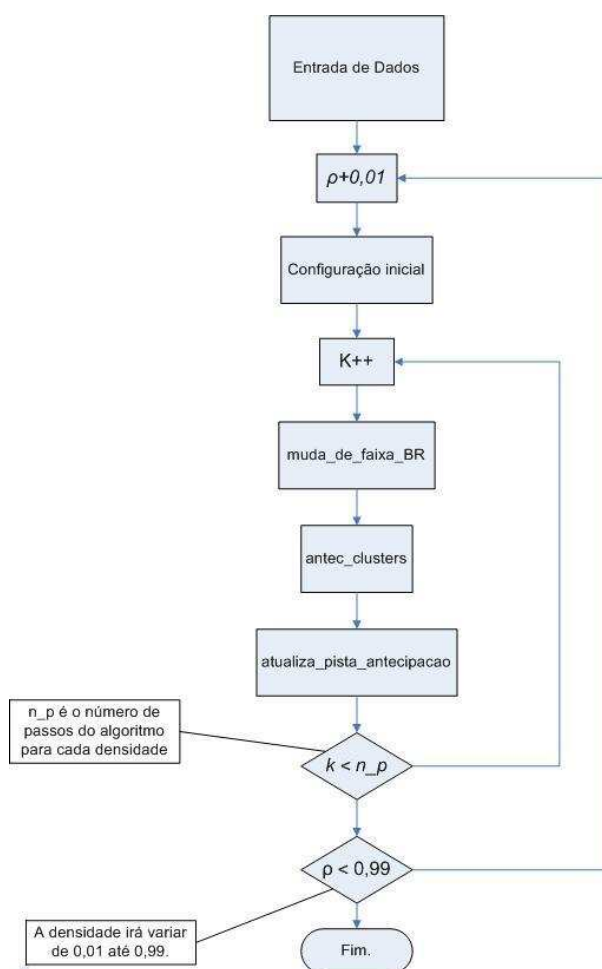


Figura A.2: Fluxograma.

A saída de dados é feita em cinco tipos de arquivos com os dados do diagrama fundamental, da relação velocidade pelo fluxo, da relação velocidade pela densidade, do uso das faixas por densidade e do número médio de reavaliações da velocidade por densidade. Para os três primeiros tipos de arquivos são gerados dados para cada faixa variando de 1 (um), para a faixa mais a esquerda, até número de faixas, para a faixa mais a direita. Os nomes desses arquivos são iniciados pelo nome do arquivo de entrada seguidos pelo sufixo identificador:

- *_FxD_Nf(número de faixas)_F (número da faixa).dat* para o diagrama fundamental;
- *_VxF_Nf(número de faixas)_F(número da faixa).dat* para o diagrama velocidade por fluxo;
- *_VxD_Nf(número de faixas)_F(número da faixa).dat* para o diagrama velocidade por densidade;
- *_USO_Nf(número de faixas)_F (número da faixa).dat* para o uso das diferentes faixas da pista;
- *_recalculados.dat* para o número médio de reavaliações.

Apresenta-se a seguir o código computacional:

```

/*decriacao do automata celulare. Estados das celulas
{-1,0,1,2,3,4,5,...} onde -1 significa celula vazia e maior que -1
significa celula ocupada por um carro com velociade igual ao valor
multiplicado por 7,5 metros por segundo.*/

#include <stdio.h> #include <stdlib.h> #include <math.h> #include
<string.h>

typedef struct Veiculo{
    int velocidade; /*velocidade do veiculo*/
    int L; /*tamanho do veiculo*/
    int v_max; /*velociade maxima*/
    int Amuda; /*pode ou nao mudar de faixa*/
    int blq; /*segundos bloqueados*/

```

```

    int id; /*identificação individual*/
}veiculo;

int l=1; /*quantidade de ceulas que um veiculo ocupa.*/ int
faixas=1; /*numero de faixas da pista */ int ac_max=1; /*aceleracao
maxima que um veiculo pode ter*/ int tamanho_da_pista=300; /*numero
de celulas da pista.*/ int numero_de_carros=1; /*numero de carros
que estao na pista.*/ int velocidade_maxima=5; /*veocidade maxima da
pista.*/ int numero_iteracoes=10000; /*numero de iteracoes totais a
serem relizadas.*/ int passos_despresados=1000; /*numro de passos
despresados do ,total de passos.*/ long int *n_carros; /*variavel
global para guardar o numero de carros que passam por um detector.*/

long int *n_car_faixa; /*variavel global para guardar o numero de
carros que passam por um detector.*/

int inicial=0; /*determina como vai ser a distriuiacao inicial dos veiculos
0-hogenio e 1-engarrafado*/

    float *svelocidade; /*variável global para guardar o somatorio das velocidades.*/
float *velocidade_media; /*variavel global para guardar a velocidade media
dos veiculos na pista.*/ float *Ns; /*numero de vaiculos aprados sobre o
detector*/ int ifb=0; /*conta o numero de vezes que um temos que recalculer
as velocidades dos veiculos*/ int ncarros; /*guarda o número de veiculos*/

float l1=0.0; /*limite inferior da regioa 1 da FDP*/ float l2=0.2; /*limite
superior da regioa 1 e inferior da regioa 2 da FDP*/ float l3=0.4; /*limite
superior da regioa 2 e inferior da regioa 3 da FDP*/ float l4=0.7; /*limite
superior da regioa 3 da FDP*/ int p1=90; /*probabilidade de alfa ser
escolhida na regioa 1*/ int p2=8; /*probabilidade de alfa ser escolhida na
regioa 2*/ int p3=2; /*probabilidade de alfa ser escolhida na regioa 3*/

int Pm=35; /*probabilidade de randomização da velocidade*/ int

```

```

Ps=35; /*probabilidade de randomização da velocidade para efeito
slow-to-start*/ int VL=100; /*percentual de veiculos lentos*/ int
caminhoes=0; /*percentual de veiculos longos*/ int tblq=2; // tolerancia a
bloqueio int pmuda=100; // probabilidade de mudar de faixa int
th1=3; /*distancia tempora 1*/ int th2=6; /*distancia tempora 2*/ int
fmuda=0; // frequencia de mudancas de faixa int np1=0; int np2=0; int np3=0;
int npn=0; int nid=0;

void le_arquivo(char *arquivo); int engarrafado (veiculo *pista,int n,int
num_carros,int pcaminhoes); int homoganeo (veiculo *pista,int n,int
num_carros,int pcaminhoes); int distancia_proximo(veiculo *pista,int n,int
posi_car); int distancia_anterior(veiculo *pista,int n,int posi_car); int
velocidade_proximo(veiculo *pista,int n,int posi_car); int
velocidade_anterior(veiculo *pista,int n,int posi_car); int
L_anterior(veiculo *pista,int n,int posi_car); int cria_clusters(int
*freados,int NF,int *espacos,int NE); int recalcular_freada(veiculo
*pista,int n,int vel_max, int ini, int fim); void antec_clusters(veiculo
*pista,int n,int vel_max); int atualiza_pista_antecipacao(veiculo
*pista,int n,int k); int muda_de_faixa_BR(veiculo **pista,int n,int
nfaixas,int v_maxima);

/*-----
Função principal.
-----*/

main(int argc, char *argv[]){
    FILE *arq,*arq2;
    char narq[100],arqv[50];
    int flag[tamanho_da_pista],i,j,k;
    float fluxo,densidade;

```

```
veiculo **pista;

long int s_car_faixa;

if(argc>=2){
    le_arquivo(argv[1]);
    sprintf(arqv, "%s", argv[1]);
}

else{
    sprintf(arqv, "teste");
}

n_carros = (long int*) malloc(faixas*(sizeof(long int)));
n_car_faixa = (long int*) malloc(faixas*(sizeof(long int)));
svelocidade = (float*) malloc(faixas*(sizeof(float)));
velocidade_media = (float*) malloc(faixas*(sizeof(float)));
Ns = (float*) malloc(faixas*(sizeof(float)));

for (i=0; i<faixas; i++){
    n_carros[i]=0;
    svelocidade[i]=0.0;
    velocidade_media[i]=0.0;
    Ns[i]=0.0;
    n_car_faixa[i]=0;
}

pista= (veiculo**) malloc(faixas*(sizeof(veiculo*)));

for(i=0; i<faixas; i++){
    pista[i]= (veiculo*) malloc(tamanho_da_pista*(sizeof(veiculo)));
}

for (k=0; k<faixas; k++){
    sprintf(narq, "%s_FxD_Nf%d_F%d.dat", arqv, faixas, k);
    arq=fopen(narq, "w");
    fclose(arq);
    sprintf(narq, "%s_VxF_Nf%d_F%d.dat", arqv, faixas, k);
```

```
    arq=fopen(narq, "w");
    fclose(arq);
    sprintf(narq, "%s_VxD_Nf%d_F%d.dat", arqv, faixas, k);
    arq=fopen(narq, "w");
    fclose(arq);
    sprintf(narq, "%s_USO_Nf%d_F%d.dat", arqv, faixas, k);
    arq=fopen(narq, "w");
    fclose(arq);
}
sprintf(narq, "%s_recalculados.dat", arqv);
arq2=fopen(narq, "w");
for (i=1; i<100; i++){
    for (j=0; j<faixas; j++){
        for (k=0; k<tamanho_da_pista; k++){
            pista[j][k].velocidade=-1;
            pista[j][k].L=0;
        }
    }
    numero_de_carros=(tamanho_da_pista*i)/(100);
    for (j=0; j<faixas; j++){
        if (j<faixas-1){
            if (inicial==0){
                homogeneo(pista[j], tamanho_da_pista, numero_de_carros, 0);
            }
            if (inicial==1){
                engarrafado(pista[j], tamanho_da_pista, numero_de_carros, 0);
            }
        }
    }
    else{
```

```
        if (inicial==0){
            homogeneo (pista[j], tamanho_da_pista, numero_de_carros
                , (caminhoes*faixas));
        }
        if(inicial==1){
            engarrafado (pista[j], tamanho_da_pista, numero_de_carros
                , (caminhoes*faixas));
        }
    }
};

for (k=0;k<faixas;k++){
    svelocidade[k]=0.0;
    n_carros[k]=0;
}

printf("%d por cento \n", i);

for (j=0; j<numero_iteracoes; j++){
    if (j<passos_despresados){
        for (k=0;k<faixas;k++){
            svelocidade[k]=0.0;
            n_carros[k]=0;
            Ns[k]=0.0;
            n_car_faixa[k]=0;
        }
        ifb=0.0;
    }
    else{
        if ((j-passos_despresados)%300==0){
            for (k=0;k<faixas;k++){
                if (svelocidade[k]>0){
                    fluxo=n_carros[k];
                }
            }
        }
    }
}
```

```
fluxo=fluxo/300.0;

densidade=(float)n_carros[k];

densidade=((densidade*fluxo)/(svelocidade[k]))
+(Ns[k]/300.0);

svelocidade[k]=svelocidade[k]/(n_carros[k]+Ns[k]);

sprintf(narq,"%s_FxD_Nf%d_F%d.dat",arqv,faixas,k);

arq=fopen(narq,"a");
fprintf(arq,"%f %f \n",densidade,fluxo);
fclose(arq);

sprintf(narq,"%s_VxF_Nf%d_F%d.dat",arqv,faixas,k);

arq=fopen(narq,"a");
fprintf(arq,"%f %f \n",fluxo,svelocidade[k]);
fclose(arq);

sprintf(narq,"%s_VxD_Nf%d_F%d.dat",arqv,faixas,k);

arq=fopen(narq,"a");
fprintf(arq,"%f %f \n",densidade,svelocidade[k]);

fclose(arq);

};

n_carros[k]=0;

svelocidade[k]=0.0;

Ns[k]=0.0;

};

};

muda_de_faixa_BR(pista,tamanho_da_pista,faixas,velocidade_maxima);

ncarros=0;
```

```

    for (k=0;k<faixas;k++){
        antec_clusters (pista[k], tamanho_da_pista, velocidade_maxima);

        atualiza_pista_antecipacao (pista[k], tamanho_da_pista, k);
    }
}

fprintf(arq2, "%f %f \n", (((float) i)/100.0), ((float) ifb/
((float) ((numero_iteracoes-passos_despresados)*numero_de_carros))));

ifb=0;

s_car_faixa=0;

for (k=0;k<faixas;k++){
    s_car_faixa=s_car_faixa+n_car_faixa[k];
}

for (k=0;k<faixas;k++){
    sprintf(narq, "%s_USO_Nf%d_F%d.dat", arqv, faixas, k);
    arq=fopen(narq, "a");
    fprintf(arq, "%f %f \n", (((float) i)/100.0),
((float) (n_car_faixa[k])/((float) s_car_faixa)));
    fclose(arq);
};
}

fclose(arq2);
}

/*-----
Le o arquivo de entrada de dados.
-----*/

void le_arquivo(char *arquivo){
    FILE *arq;

```



```
char ns[100];

int tipo;

float aux;

arq=fopen(arquivo, "r");

fscanf(arq, "%s", &ns);

fscanf(arq, "%d", &tipo);

fscanf(arq, "%s", &ns);

fscanf(arq, "%d", &inicial); /*determina como vai ser a distriucio inicial dos
veiculos 0-hogenio, 1-engarrafado e 2-aleatorio */

fscanf(arq, "%s", &ns);

fscanf(arq, "%d", &tamanho_da_pista); /*numero de celulas da
pista.*/

fscanf(arq, "%s", &ns);

fscanf(arq, "%d", &velocidade_maxima); /*veocidade maxima da
pista.*/printf("\n%d", velocidade_maxima);

fscanf(arq, "%s", &ns);

fscanf(arq, "%d", &numero_iteracoes); /*numero de passos totais a serem
relizadas.*/

fscanf(arq, "%s", &ns);

fscanf(arq, "%d", &passos_despresados); /*numero de passos despresados do ,total de passos.*/

fscanf(arq, "%s", &ns);

fscanf(arq, "%f", &l1); /*limite inferior da regioao 1 da FDP*/

fscanf(arq, "%s", &ns);

fscanf(arq, "%f", &l2); /*limite superior da regioao 1 e inferior da regioao 2 da FDP*/

fscanf(arq, "%s", &ns);

fscanf(arq, "%f", &l3); /*limite superior da regioao 2 e inferior da regioao 3 da FDP*/

fscanf(arq, "%s", &ns);

fscanf(arq, "%f", &l4); /*limite superior da regioao 3 da FDP*/
```

```
fscanf(arq, "%s", &ns);

fscanf(arq, "%f", &aux); /*probabilidade de alfa ser escolhida na regioao 1*/

p1=(int)(aux*100.0);

fscanf(arq, "%s", &ns);

fscanf(arq, "%f", &aux); /*probabilidade de alfa ser escolhida na regioao 1*/

p2=(int)(aux*100.0);

fscanf(arq, "%s", &ns);

fscanf(arq, "%f", &aux); /*probabilidade de alfa ser escolhida na regioao 1*/

p3=(int)(aux*100.0);

fscanf(arq, "%s", &ns);

fscanf(arq, "%f", &aux); /*probabilidade de randomiza?o da velocidade*/

Pm=(int)(aux*100.0);

Ps=Pm;

if (tipo>1){

    fscanf(arq, "%s", &ns);

    fscanf(arq, "%d", &l); /*quantidade de ceulas que um veiculo ocupa.*/

    fscanf(arq, "%s", &ns);

    fscanf(arq, "%d", &ac_max); /*aceleracao maxima que um veiculo pode ter*/

    fscanf(arq, "%s", &ns);

    fscanf(arq, "%d", &VL); /*percentual de veiculos rapidos*/

    VL=100-VL;

};

if (tipo>2){

    fscanf(arq, "%s", &ns);

    fscanf(arq, "%d", &caminhoes); /*percentual de veiculos longos*/

    fscanf(arq, "%s", &ns);

    fscanf(arq, "%d", &faixas); /*numero de faixas da pista */

    fscanf(arq, "%s", &ns);

    fscanf(arq, "%d", &tblq); // tolerancia a bloqueio

    fscanf(arq, "%s", &ns);
```

```

    fscanf(arq, "%d", &aux); // probabilidade de mudar de faixa
    pmuda=(int) (aux*100.0);
    fscanf(arq, "%s", &ns);
    fscanf(arq, "%d", &th1); /*distancia tempora 1*/
    fscanf(arq, "%s", &ns);
    fscanf(arq, "%d", &th2); /*distancia tempora 2*/
};

}

/*-----
Preenche a pista com todos os carros em um mesmo engarrafamento,
onde o vetor pista é o vetor que se esta querendo preencher, n é o
tamanho dessa pista, num_carros é o número de células que pretende-se
preencher com veículos e pcaminhoes é o percentual de veículos lentos
na pista.
-----*/

int engarrafado (veiculo *pista, int n, int num_carros, int pcaminhoes) {
    int i,k,ncar,sl;
    int tveiculos[n];
    ncar=0;
    sl=0;
    while (num_carros>0) {
        i=1;
        tveiculos[ncar]=i;
        ncar++;
        num_carros=num_carros-i;
        sl=sl+i;
    }
    k=0;

```

```

for (i=0;i<ncar;i++){
    pista[k].velocidade=(rand()%(velocidade_maxima+1));
    pista[k].blq=0;
    nid++;
    pista[k].id=nid;
    pista[k].L=tveiculos[i];
    if (rand()%100<pcaminhoes){
        pista[k].v_max=(velocidade_maxima+1)/2;
        pista[k].Amuda=0;
    }
    else{
        pista[k].v_max=velocidade_maxima;
        pista[k].Amuda=1;
    };
    k=k+1;
}

}

/*-----
preenche a pista com Veículos igualmente espaçados, onde o vetor
pista é o vetor que se esta querendo preencher, n é o tamanho dessa
pista, num_carros é o número de células que pretende-se peencer com
veículos e pcaminhoes é o percentual de veículos lentos na pista.
-----*/

int homogeneo (veiculo *pista,int n,int num_carros,int pcaminhoes){
    int i,d,r,k,ncar,sl;
    int tveiculos[n];
    ncar=0;
    sl=0;

```

```
while (num_carros>0) {  
    i=1;  
    tveiculos[ncar]=i;  
    ncar++;  
    num_carros=num_carros-i;  
    sl=sl+i;  
}  
d=(n-sl)/ncar;  
r=(n-sl)%ncar;  
k=0;  
for (i=0;i<ncar;i++){  
    pista[k].velocidade=(rand()%(velocidade_maxima+1));  
    pista[k].L=tveiculos[i];  
    if (rand()%100<pcaminhoes) {  
        pista[k].v_max=(velocidade_maxima+1)/2;  
        pista[k].Amuda=0;  
    }  
    else{  
        pista[k].v_max=velocidade_maxima;  
        pista[k].Amuda=1;  
    };  
    if (i<r) {k=k+(d+1)+tveiculos[i];}  
    else {k=k+d+tveiculos[i];}  
}  
}
```

/*-----
Calcula a distância de um veículo para o veículo da frente. O vetor

pista é a pista a ser simulada, n é o tamanho dessa pista e posi_car é a posição do veículo considerado.

```

-----*/
int distancia_proximo(veiculo *pista,int n,int posi_car) {
    int i;
    if (posi_car==n-1) {i=0;}
    else{i=posi_car+1;};
    while (pista[i].velocidade== -1 && i!=posi_car) {
        if (i==n-1) {i=0;}
        else {i++;};
    }
    if ((i==posi_car) && pista[i].velocidade<0) {return (n);}
    else {
        if (i<=posi_car) {i=(i-1)+(n-posi_car);}
        else{i=(i-1)-posi_car;};
        return (i);
    };
}

```

```

/*-----*/
Calcula a distância de um veículo para o veículo de tras. O vetor
pista é a pista a ser simulada, n é o tamanho dessa pista e posi_car
é a posição do veículo considerado.
-----*/
int distancia_anterior(veiculo *pista,int n,int posi_car) {
    int i;
    if (posi_car==0) {i=n-1;}
    else{i=posi_car-1;};
    while ((pista[i].velocidade>-1) && (i!=posi_car)){

```

```

        if (i==0){i=n-1;}

        else {i--;};

    }

if ((i==posi_car) && (pista[i].velocidade<0)){return (n);}

else {

    if (i>posi_car){i=(posi_car-1)+(n-i);}

    else{i=(posi_car-1)-i;};

    return (i);

};

}

```

/*-----

Retorna a velocidade do veículo a frente do analisado. O vetor pista é a pista a ser simulada, n é o tamanho dessa pista e posi_car é a posição do veículo considerado.

-----*/

```

int velocidade_proximo(veiculo *pista,int n,int posi_car){

    int i;

    if (posi_car==n-1){i=0;}

    else{i=posi_car+1;};

    while (pista[i].velocidade== -1 && i!=posi_car){

        if (i==n-1){i=0;}

        else {i++;};

    }

    if ((i==posi_car) && pista[i].velocidade<0){return (velocidade_maxima);}

    else {

        return (pista[i].velocidade);

    };

}

```

```

/*-----
Retorna a velocidade do veículo a tras do analisado. O vetor pista é
a pista a ser simulada, n é o tamanho dessa pista e posi_car é a
posição do veículo considerado.
-----*/

```

```

int velocidade_anterior(veiculo *pista,int n,int posi_car){
    int i;
    if (posi_car==0){i=n-1;}
    else{i=posi_car-1;};
    while ((pista[i].velocidade>-1) && (i!=posi_car)){
        if (i==0){i=n-1;}
        else {i--;};
    }
    if ((i==posi_car) && (pista[i].velocidade<0)){return(0);}
    else {
        return(pista[i].velocidade);
    };
}

```

```

/*-----
Retorna o tamanho do veículo a frente do analisado. O vetor pista é
a pista a ser simulada, n é o tamanho dessa pista e posi_car é a
posição do veículo considerado.
-----*/

```

```

int L_anterior(veiculo *pista,int n,int posi_car){
    int i;
    if (posi_car==0){i=n-1;}
    else{i=posi_car-1;};

```



```

while ((pista[i].velocidade>-1) && (i!=posi_car)){
    if (i==0){i=n-1;}
    else {i--};
}
if ((i==posi_car) && (pista[i].velocidade<0)){return (1);}
else {
    return(pista[i].L);
};
}

```

/*-----*/

Define os clusters de influencia que serão recalculados. O vetor freados e uma vetor contendo todas as posições dos veículos que frearam bruscamente, NF é o tamanho do vetor freados, espacos é um vetor contendo as posições dos veículos que não precisam de antecipação para manterem suas velocidades e NE é o tamanho desse vetor.

-----*/

```

int cria_clusters(int *freados,int NF,int *espacos,int NE) {
    int p,i,j,k;
    int Naux=0;
    int aux[NE][2],aux2[NF][2];
    if (NE>0) {
        if (espacos[NE-1]<espacos[NE-2]) {NE--;}
        p=-1;
        aux[0][0]=espacos[NE-1];
        aux[0][1]=espacos[0];
        for (i=1;i<NE;i++){
            aux[i][0]=espacos[i-1];
            aux[i][1]=espacos[i];
        }
    }
}

```

```
i=0;

while ((freados[i]<aux[0][1]) && (i<NF)) {

    p=freados[i];

    i++;

}

if ((p==-1) && (freados[NF-1]>aux[0][0])) {

    aux2[0][0]=aux[0][0];

    aux2[0][1]=freados[NF-1];

    Naux++;

}

else {

    if (p>-1) {

        aux2[0][0]=aux[0][0];

        aux2[0][1]=p;

        Naux++;

    }

    else {i=0;};

}

p=-1;

k=1;

while ((i<NF) ? (freados[i]<aux[NE-1][1]) : 0) {

    while (((freados[i]<aux[k][0]) || (freados[i]>aux[k][1])) {

        if (p>-1) {

            aux2[Naux][0]=aux[k][0];

            aux2[Naux][1]=p;

            Naux++;

            p=-1;

        };

        k++;

    };

};
```

```

        p=freados[i];

        i++;

    };

    if (p>-1) {

        aux2[Naux][0]=aux[k][0];

        aux2[Naux][1]=p;

        Naux++;

        p=-1;

    };

    for (i=0;i<Naux;i++){

        freados[i]=aux2[i][1];

        if (aux2[i][0]>=0) {

            espacos[i]=aux2[i][0];

        }

        else{

            espacos[i]=0;

        }

    };

};

return (Naux);

};

/*-----

Recalcula a velocidades dos veículos que estão da posição ini até a
posição fim do vetor pista de tamanho n, retornando a maior posição
do veículo que freou bruscamente durante esse recalclo.

-----*/

int recalcular_freada(veiculo *pista,int n,int vel_max, int ini, int fim){

    int i,j,def,k,d,vproximo,vant;

    float alfa;

```

```
int Rnd;

int NP, NE, NC;

NP=0;

NE=0;

NC=0;

int T=0;

int p=-1;

for (i=ini; i<=fim; i++) {

    if (pista[i].velocidade>=0) {

        ifb++;

        vant=pista[i].velocidade;

        d=distancia_proximo (pista, n, i);

        vproximo=velocidade_proximo (pista, n, i);

        def=d- (pista[i].L-1);

        Rnd=rand() %100;

        if (Rnd<p1) {

            np1++;

            alfa=(l2-l1) * ((float) (rand() %100) /100) +l1;

        }

        else{

            if((Rnd>=p1) && (Rnd<(p1+p2))) {

                np2++;

                alfa=(l3-l2) * ((float) (rand() %100) /100) +l2;

            }

            else{

                if((Rnd>=(p1+p2)) && (Rnd<(p1+p2+p3))) {

                    np3++;

                    alfa=(l4-l3) * ((float) (rand() %100) /100) +l3;

                }

            }

        }

    }

}
```

```

};

if ((vel_max-vproximo)<ac_max) {
    k=ac_max-(vel_max-vproximo);
    if ((vproximo-k)>0) {
        vproximo=vproximo-k;
    }
    else{
        vproximo=0;
    }
}

def=def+((int) ((1.0-alfa) * ((float) vproximo)+0.5));

if (pista[i].velocidade>def) {
    if(((int) (((1.0-11) * (float) vant)+0.5))>def) {
        p=i;
    }
    pista[i].velocidade=def;
}

if(pista[i].velocidade<0){printf("\nvelocidade negativa 2\n");}
}
}

return(p);
}

```

/*-----

Função de atualização das velocidades do modelo de antecipação proposto, onde o vetor pista é a pista consideranda, n é o tamanho dessa pista e vel_max é a velocidade maxima da psita.

-----*/

```

void antec_clusters(veiculo *pista,int n,int vel_max) {
    int i,j,def,k,d,vproximo,vant;

```

```
int p,paux,a;

int ps=Ps;

int pm=Pm;

float alfa;

int li,Rnd;

int parados[n],espacos[n];

int NP,NE,NC;

int pvl,pvr;

int nv;

NP=0;

NE=0;

NC=0;

pvl=(pm*VL)/100;

pvr=pm-pvl;

nv=0;

int T=0;

for (i=0;i<n;i++){

    if (pista[i].velocidade>-1){

        vant=pista[i].velocidade;

        nv++;

        if((pista[i].velocidade+ac_max)>pista[i].v_max){

            pista[i].velocidade=pista[i].v_max;

        }

        else {pista[i].velocidade=pista[i].velocidade+ac_max;};

        p=pm;

        Rnd=rand()%100;

        if (Rnd<pvl){

            if ((pista[i].velocidade-ac_max)<0){pista[i].velocidade=0;}

            else{

                pista[i].velocidade=pista[i].velocidade-ac_max;
```

```
    }
}
else{
    if ((Rnd>=pvl) && (Rnd<pvl+pvr) && (pista[i].velocidade>0)){

        pista[i].velocidade=pista[i].velocidade-(ac_max/2);

    }
}
d=distancia_proximo(pista,n,i);
if (d<(vel_max+pista[i].L)){
    if (i+d+1<n){
        vproximo=pista[i+d+1].velocidade;
    }
    else{
        vproximo=pista[d-(n-i)+1].velocidade;
    }
    def=d-(pista[i].L-1);
    if (pista[i].velocidade<=def){
        if(i+d>=n){
            espacos[NE]=d-(n-i);
            NE++;
            if ((d-(n-i))<0){printf("espaco errado %d",d-(n-i));}
        }
        else{
            espacos[NE]=i+d;
            NE++;
            if ((d+i)<0){printf("espaco errrado %d",d+i);}
        }
    }
}
```

```
Rnd=rand()%100;

if (Rnd<p1){

    np1++;

    alfa=(l2-l1)*((float) (rand()%100)/100)+l1;

}

else{

    if((Rnd>=p1) && (Rnd<(p1+p2))){

        np2++;

        alfa=(l3-l2)*((float) (rand()%100)/100)+l2;

    }

    else{

        if((Rnd>=(p1+p2)) && (Rnd<(p1+p2+p3))){

            np3++;

            alfa=(l4-l3)*((float) (rand()%100)/100)+l3;

        }

    }

};

if ((alfa>l1) && (alfa<l2)){np1++;}

else{

    if ((alfa>l2) && (alfa<l3)){np2++;}

    else{

        if ((alfa>l3) && (alfa<l4)){np3++;}

        else {nbn;}

    }

}

if ((vel_max-vproximo)<ac_max){

    k=ac_max-(vel_max-vproximo);

    if ((vproximo-k)>0){

        vproximo=vproximo-k;

    }

}
```



```
        else{
            vproximo=0;
        }
    }
    def=def+((int)((1.0-alfa)*(float)vproximo)+0.5);
    if (pista[i].velocidade>def){
        if(((int)((1.0-l1)*(float)vant)+0.5)>def){
            parados[NP]=i;
            NP++;
        };
        pista[i].velocidade=def;
    };
};

if(pista[i].velocidade<0){printf("\nvelocidade negativa 1\n");};
}
}

if (nv>0){
    if ((NP>0) && (NE>0)){
        NC=cria_clusters(parados, NP, espacos, NE);
    }
    else{
        if(NE==0){
            NC=1;
            parados[0]=n-1;
            espacos[0]=0;
            T=1;
        };
    };
};

if (NC>0 && nv>0){
```

```

    for (j=0; j<NC; j++) {
        p=-1;
        if (espacos[j]>parados[j]) {
            p=recalcular_freada (pista, n, vel_max, espacos[j], n-1);
            paux=recalcular_freada (pista, n, vel_max, 0, parados[j]);
            if (paux>-1) {p=paux;};
        }
        else{
            p=recalcular_freada (pista, n, vel_max, espacos[j], parados[j]);
        }
        if (p>-1) {
            parados[NC]=p;
            espacos[NC]=espacos[j];
            NC++;
        }
    }
}

```

/*-----*/

Atualiza as posições dos veículos no vetor pista a partir das
 velocidade calculadas. n é o tamanho da pista, k é o número da faixa
 que esta sendo atualizada.

-----*/

```

int atualiza_pista_antecipacao (veiculo *pista, int n, int k) {
    int i, v, soma;
    veiculo pista_aux[n];
    for (i=0; i<n; i++) {
        pista_aux[i].velocidade=-1;
    }
}

```

```

    pista_aux[i].L=0;

    pista_aux[i].Amuda=0;

    pista_aux[i].v_max=0;

    pista_aux[i].blq=0;
}
for (i=0;i<n;i++){
    v=pista[i].velocidade;

    if (v>=0){

        if(v==0 && (i+pista[i].L)>=(n)) {Ns[k]=Ns[k]+1.0;};

        ncarros++;

        if ((v+i)>=n){

            pista_aux[v-(n-i)]=pista[i];

            n_carros[k]++;

            svelocidade[k]=svelocidade[k]+((float)v)/((float)pista[i].L);

        }

        else {

            pista_aux[i+v]=pista[i];

        }

    };

};

for (i=0;i<n;i++){pista[i]=pista_aux[i];};
};

```

/*-----*/

Rotina de mudança de faixas para as diferentes faixas da matriz

pista. n é o tamanho de cada faixa nfaixas é o número de faixas da

pista e v_max é a velocidade máxima da pista.

-----*/

```

int muda_de_faixa_BR(veiculo **pista,int n,int nfaixas,int v_maxima){
    int i, j, d, da, pos_ant, nv, def;

    int mudou;

    veiculo pistaux[nfaixas][n];

    nv=0;

    for (i=0; i<nfaixas; i++){
        for (j=0; j<n; j++){
            pistaux[i][j].velocidade=-1;
        }
    }

    for (i=0; i<nfaixas; i++){
        for (j=0; j<n; j++){
            if ((pista[i][j].velocidade>-1) && (pista[i][j].Amuda>0)){
                nv++;

                d=distancia_proximo(pista[i], n, j);

                mudou=0;

                if (i>0){
                    def=distancia_proximo(pista[i], n, j)-
                    (pista[i][j].L-1) + (int) ((1-14) *
                    ((float) velocidade_proximo(pista[i], n, j)) +0.5);

                    if ((pista[i][j].velocidade>def) ||
                    (pista[i][j].blq>tblq)){
                        d=distancia_proximo(pista[i-1], n, j);
                        da=distancia_anterior(pista[i-1], n, j);

                        if((d<v_maxima) || (da<v_maxima) ){
                            def=distancia_proximo(pista[i-1], n, j)-
                            (pista[i][j].L-1) + (int) ((1-14) *
                            ((float) velocidade_proximo(pista[i-1], n, j))
                            +0.5);

                            if ((pista[i][j].velocidade<def) &&

```

```

        ((da-(l-1))>
        (velocidade_anterior(pista[i-1],n,j)+1))
        && (pista[i-1][j].velocidade<0) &&
        (rand()%100<pmuda)) {
            pistaux[i-1][j]=pista[i][j];
            pistaux[i-1][j].blq=0;
            mudou=1;
        };
    }
    else {
        if ((pista[i-1][j].velocidade<0) &&
        (rand()%100<pmuda)) {
            pistaux[i-1][j]=pista[i][j];
            pistaux[i-1][j].blq=0;
            mudou=1;
        }
    };
};
}
else{
    da=distancia_anterior(pista[i],n,j);
    if (((mudou=0) &&
    (pista[i][j].velocidade<
    velocidade_anterior(pista[i],n,j)-1)
    && (da<(th1*velocidade_anterior(pista[i],n,j)))) ||
    (d>th2*pista[i][j].velocidade) ||
    (pista[i][j].blq>tblq)) {
        if (i<(nfaixas-1)) {
            d=distancia_proximo(pista[i+1],n,j);
            da=distancia_anterior(pista[i+1],n,j);

```

```

def=distancia_proximo(pista[i+1],n,j)
-(pista[i][j].L-1) + (int)((1-l4)*
(float)velocidade_proximo(pista[i+1],n,j))
+0.5);
if((def<v_maxima) || (da<v_maxima) ){
    if ((pista[i][j].velocidade<def) &&
        ((da-(l-1))>
        (velocidade_anterior(pista[i+1],n,j)+1))
        && (pista[i+1][j].velocidade<0) &&
        (rand()%100<pmuda) ){
        pistaux[i+1][j]=pista[i][j];
        pistaux[i+1][j].blq=0;
        mudou=1;
    };
}
else{
    if ((pista[i+1][j].velocidade<0) ){
        pistaux[i+1][j]=pista[i][j];
        pistaux[i+1][j].blq=0;
        mudou=1;
    }
};
};
};
if (mudou==0) {
    pistaux[i][j]=pista[i][j];
    if (velocidade_proximo(pista[i],n,j)==0) {
        pistaux[i][j].blq++;
    };
};

```

```
        }
        else {fmuda++;};
    }
    else{
        if (pista[i][j].velocidade>-1) {
            pistaux[i][j]=pista[i][j];
            pistaux[i][j].blq=0;
        }
    };
};

};

for (i=0;i<nfaixas;i++){
    for (j=0;j<n;j++){
        pista[i][j]=pistaux[i][j];
        if (pista[i][j].velocidade>-1) {
            nv--;
            n_car_faixa[i]++;
        }
    }
}

};
```