

**UNIVERSIDADE DE BRASÍLIA  
FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA CIVIL E  
AMBIENTAL**

**PROCEDIMENTO PARA A CONCEPÇÃO DE  
CONTROLADORES SEMAFÓRICOS *FUZZY* COM  
O USO DE ALGORITMO GENÉTICO**

**PAULO SÉRGIO FRANÇA DE SOUSA JÚNIOR**

**ORIENTADOR: MARIA ALICE P. JACQUES**

**DISSERTAÇÃO DE MESTRADO EM TRANSPORTES  
BRASÍLIA / DF: JANEIRO / 2010**

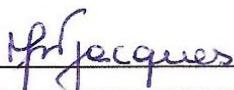
**UNIVERSIDADE DE BRASÍLIA  
FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA CIVIL E AMBIENTAL**

**PROCEDIMENTO PARA A CONCEPÇÃO DE  
CONTROLADORES SEMAFÓRICOS *FUZZY* COM  
O USO DE ALGORITMO GENÉTICO**

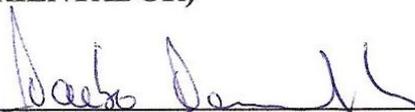
**PAULO SÉRGIO FRANÇA DE SOUSA JÚNIOR**

**DISSERTAÇÃO DE MESTRADO SUBMETIDA AO DEPARTAMENTO DE  
ENGENHARIA CIVIL E AMBIENTAL DA FACULDADE DE TECNOLOGIA  
DA UNIVERSIDADE DE BRASÍLIA COMO PARTE DOS REQUISITOS  
NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE.**

**APROVADA POR:**

  
\_\_\_\_\_

**MARIA ALICE PRUDÊNCIO JACQUES, Ph.D (UnB)  
(ORIENTADOR)**

  
\_\_\_\_\_

**YAEKO YAMASHITA, Ph.D (UnB)  
(EXAMINADOR INTERNO)**

  
\_\_\_\_\_

**CARLOS FELIPE GRANGEIRO LOUREIRO, Ph.D (UFC)  
(EXAMINADOR EXTERNO)**

**Data: Brasília/DF, 29 de janeiro de 2009.**

## FICHA CATALOGRÁFICA

SOUSA JR., PAULO SÉRGIO FRANÇA DE

Procedimento para a concepção de controladores semafóricos *fuzzy* com o uso de algoritmo genético. Brasília, 2010.

xiv, 143p., 210x297mm (ENC/FT/UnB, Mestre, Transporte, 2010).

Dissertação de Mestrado – Universidade de Brasília. Faculdade de Tecnologia. Departamento de Engenharia Civil e Ambiental.

1 – Controle de Tráfego

2 – Controle Semafórico em Interseção Isolada

3 – Controladores *Fuzzy*

4 – Algoritmo Genético

I – ENC/FT/UnB

II – Título (série)

## REFERÊNCIA BIBLIOGRÁFICA

SOUSA JR., P.S.F. (2010). Procedimento para a concepção de controladores semafóricos *fuzzy* com o uso de algoritmo genético. Dissertação de Mestrado, Publicação T.DM-002 A/2010, Departamento de Engenharia Civil e Ambiental, Faculdade de Tecnologia, Universidade de Brasília, DF, 143p.

## CESSÃO DE DIREITOS

AUTOR: Paulo Sérgio França de Sousa Júnior

TÍTULO: Procedimento para a concepção de controladores semafóricos *fuzzy* com o uso de algoritmo genético.

GRAU: Mestre ANO: 2010.

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desta dissertação de mestrado pode ser reproduzida sem a autorização por escrito do autor.



Paulo Sérgio França de Sousa Júnior

SQSW 302, Bloco H, Apto 303, Sudoeste – (61) 8129-2211

CEP: 70.673-208 – Brasília/DF – Brasil

## **AGRADECIMENTOS**

A Deus, por tudo o que é e tem feito por mim, pela saúde e sabedoria para vencer mais esta etapa de minha vida.

Aos meus pais, meus grandes mestres, pelo exemplo a ser seguido em tudo, pelo amor e suporte para que eu chegasse aqui, pelo estímulo de sempre crescer na vida e pela amizade verdadeira.

À minha irmã, discípula dos mesmos mestres, pela grande amizade ao longo dos anos.

A todos meus familiares e amigos, pela alegria compartilhada.

À professora e orientadora Maria Alice, pela constante disposição de ensinar e pelo exemplo de dedicação e zelo profissional.

Aos professores, colegas e funcionários da Universidade de Brasília com os quais convivi, em especial aos do Programa de Pós-Graduação em Transportes, por tornarem aquele lugar um centro de excelência e de agradável convivência.

## RESUMO

### PROCEDIMENTO PARA A CONCEPÇÃO DE CONTROLADORES SEMAFÓRICOS *FUZZY* COM O USO DE ALGORITMO GENÉTICO

Estudos desenvolvidos com controladores semafóricos *fuzzy* têm apresentado resultados promissores para o seu uso no controle de tráfego em interseções. Este tipo de controlador realiza um controle do tipo atuado pelo tráfego, fazendo uso do princípio da extensão adaptado. Neste princípio, as extensões do tempo de verde são fornecidas levando-se em conta a situação de todas as aproximações da interseção. No entanto, a concepção desse controlador, isto é, a definição do modo de implementação de seus componentes, é de difícil realização por parte de seu projetista, pois existem inúmeras soluções possíveis que podem ser adotadas no projeto. Além disso, deve-se levar em conta que controladores *fuzzy* nem sempre produzem superfícies de controle com as características desejáveis para o caso em estudo. Sabendo-se que as diferentes formas de implementação afetam significativamente no resultado do controlador, e que, para o controle de tráfego, a superfície deve ser contínua, suave e coerente, considera-se de grande valia o uso de uma ferramenta que trabalhe bem em espaços de busca muito grandes, de maneira que possa auxiliar o projetista na definição destes controladores. Neste sentido, o presente trabalho propõe um procedimento para a concepção de controladores semafóricos *fuzzy* que faz uso de algoritmo genético (AG) para a definição do modo de implementação de seus componentes, e que considera a avaliação da qualidade das superfícies de controle geradas, com o objetivo de garantir um bom desempenho do tráfego na interseção controlada. Os controladores resultantes desse procedimento apresentaram resultados satisfatórios tanto para volumes altos quanto para volumes baixos, sendo que seus atrasos médios veiculares e superfícies de controle foram melhores do que aqueles fornecidos pelo controlador utilizado como referência para comparação. Estes resultados comprovaram a hipótese do estudo de que um procedimento que faz uso de AG permite a concepção adequada desses controladores, de modo a permitir que a atuação do controlador reflita as necessidades específicas do tráfego a ser controlado, em termos de variável de desempenho de tráfego, e forneça uma superfície de controle contínua, suave e coerente. Acredita-se que os resultados desse procedimento podem ser melhorados ainda mais com o desenvolvimento de outros estudos que considerem a calibração dos parâmetros do AG e incorporem mais variáveis de desempenho de tráfego no processo de busca do melhor controlador.

## **ABSTRACT**

### **PROCEDURE FOR THE CONCEPTION OF FUZZY SIGNAL CONTROLLERS WITH THE USE OF GENETIC ALGORITHM**

Studies developed with fuzzy signal controllers have presented promising results for its use in traffic control at intersections. This type of controller is traffic-actuated, making use of the principle of green extension. In this principle, the extensions of the green time are provided taking in account the situation of all the approaches of the intersection. However, the conception of this controller, that is, the definition of its components implementation mode, is of difficult execution for its planner, because there are a great number of possible solutions which can be adopted in a project. Moreover, it must be considered that fuzzy controllers not always provide control surfaces with the desirable characteristics for the studied case. Knowing that different forms of implementation affect significantly in the controller's result, and that, in traffic control, the surface must be continuous, smooth and coherent, it is considered of great value the use of a tool that works well in search spaces which are very big, so that it can assist the planner in the definition of these controllers. In this sense, the present work proposes a procedure for the conception of fuzzy signal controllers which uses a genetic algorithm (GA) to define the implementation mode of its components, and that considers the evaluation of the quality of the generated control surfaces, with the objective to guarantee a good performance of the traffic in the controlled intersection. The resultant controllers of this procedure had presented satisfactory results for high and low volumes, and its average delays and control surfaces had been better evaluated than those provided by the controller used as reference for comparisons. These results had proven the hypothesis of the study that a procedure that makes use of GA allows the adequate conception of these controllers, in order to allow that the performance of the controller reflects the specific necessities of the traffic to be controlled, in terms of traffic performance variable, and supply a continuous, smooth and coherent control surface. It is believed that the results of this procedure can still be more improved with the development of other studies that consider the calibration of the parameters of the GA and incorporate more traffic performance variables in the searching process of the best controller.

# SUMÁRIO

<b>1. INTRODUÇÃO.....</b>	<b>1</b>
1.1 APRESENTAÇÃO .....	1
1.2 DEFINIÇÃO DO PROBLEMA.....	2
1.3 JUSTIFICATIVA.....	2
1.4 HIPÓTESE.....	3
1.5 OBJETIVO.....	3
1.6 ESTRUTURA DA DISSERTAÇÃO.....	4
<b>2. CONTROLE SEMAFÓRICO.....</b>	<b>5</b>
2.1 PLANO SEMAFÓRICO .....	5
2.2 TIPOS DE CONTROLE .....	7
2.3 PRINCÍPIO DA EXTENSÃO.....	8
<b>3. LÓGICA FUZZY .....</b>	<b>10</b>
3.1 CONJUNTOS FUZZY .....	10
3.1.1 Função de Pertinência .....	12
3.1.2 Partição do Universo de Discurso .....	13
3.2 SISTEMAS FUZZY .....	15
3.2.1 Sistemas de Controle <i>Fuzzy</i> (Controladores <i>Fuzzy</i> ) .....	15
3.2.2 Sistemas <i>Fuzzy</i> e a Engenharia de Transportes.....	17
3.2.3 Sistemas <i>Fuzzy</i> e Algoritmos Genéticos.....	19
<b>4. CONTROLADORES SEMAFÓRICOS FUZZY .....</b>	<b>20</b>
4.1 COMPONENTES DOS CONTROLADORES SEMAFÓRICOS FUZZY .....	20
4.1.1 Interface de Fuzificação.....	21
4.1.2 Base de Conhecimento.....	21
4.1.3 Lógica para Tomada de Decisões.....	22
4.1.4 Interface de Defuzificação.....	24
4.2 CLASSIFICAÇÃO DOS CONTROLADORES - MAMDANI E SUGENO.....	26
4.3 SUPERFÍCIE DE CONTROLE.....	27
4.4 CONTROLADORES SEMAFÓRICOS FUZZY PARA INTERSEÇÕES ISOLADAS.....	29
4.4.1 Controlador de Pappis e Mamdani.....	29
4.4.2 Controlador de Favilla, Machion, e Gomide.....	31
4.4.3 Controlador de Niittymäki e Pursula.....	33
4.5 CONSIDERAÇÕES RELACIONADAS À CONCEPÇÃO DE UM CONTROLADOR SEMAFÓRICO FUZZY .....	34
<b>5. ALGORITMOS GENÉTICOS .....</b>	<b>36</b>
5.1 COMPUTAÇÃO INSPIRADA NA BIOLOGIA.....	36
5.1.1 Computação Evolutiva.....	36
5.1.2 Redes Neurais.....	38
5.2 APLICAÇÕES DOS AGS.....	38
5.3 TERMOS BIOLÓGICOS IMPORTANTES.....	40
5.4 ALGORITMOS GENÉTICOS: O QUE SÃO E COMO FUNCIONAM .....	40
5.4.1 População Inicial.....	42
5.4.2 Função Objetivo e Função Aptidão (Fitness).....	43

5.4.3 Operadores.....	43
5.5 ALGORITMOS GENÉTICOS E OUTROS MÉTODOS DE BUSCA E OTIMIZAÇÃO .....	46
<b>6. PROCEDIMENTO PARA A CONCEPÇÃO DE CONTROLADORES SEMAFÓRICOS FUZZY COM O USO DE ALGORITMO GENÉTICO.....</b>	<b>48</b>
6.1 CONSIDERAÇÕES INICIAIS.....	48
6.2 PROCEDIMENTO PROPOSTO.....	49
6.2.1 Algoritmo para Execução do Procedimento Proposto.....	51
6.2.2 Número de Vezes que o Algoritmo é Rodado.....	52
6.3 MÓDULO INICIAL - DEFINIÇÕES BÁSICAS.....	52
6.3.1 Variáveis Fuzzy, Conjuntos Fuzzy e Base de Regras.....	52
6.3.2 Características da Interseção e do Tráfego a Ser Controlado.....	54
6.3.3 Variáveis que Caracterizam Cada Indivíduo.....	55
6.3.4 Espaço de Busca para Cada Variável.....	56
6.3.5 Codificação.....	66
6.3.6 Definição do Cromossomo.....	68
6.3.7 Definição da População Inicial.....	70
6.4 MÓDULO CONTROLADOR <i>FUZZY</i> .....	70
6.4.1 Matlab.....	70
6.4.2 Tabela de Extensões.....	83
6.4.3 Superfície de Controle.....	84
6.5 MÓDULO SIMULADOR.....	84
6.5.1 Simulador.....	84
6.5.2 Medida de Desempenho.....	90
6.6 MÓDULO SUPERFÍCIE DE CONTROLE.....	90
6.6.1 Avaliação da Superfície de Controle.....	90
6.6.2 Penalização “P”.....	93
6.7 MÓDULO GENÉTICO.....	93
6.7.1 Detecção de Indivíduos Problemáticos.....	93
6.7.2 Substituição de Indivíduos Problemáticos.....	94
6.7.3 Geração “Y”.....	94
6.7.4 Função Aptidão.....	94
6.7.5 Verificação da Condição de Controle.....	95
6.7.6 Operadores do AG e Elitismo.....	95
6.7.7 Nova População.....	98
6.9 COMPARAÇÃO DOS MELHORES RESULTADOS.....	98
<b>7. APLICAÇÃO DA METODOLOGIA E ANÁLISE DE RESULTADOS.....</b>	<b>99</b>
7.1 RESULTADOS OBTIDOS.....	99
7.1.1 Volume Alto.....	99
7.1.2 Volume Baixo.....	108
7.2 ANÁLISE DOS RESULTADOS.....	116
7.2.1 Volume Alto.....	116
7.2.2 Volume Baixo.....	120
7.3 COMPARAÇÃO ENTRE OS MELHORES CONTROLADORES DE CADA NÍVEL DE VOLUME.....	124
<b>8. CONCLUSÕES E RECOMENDAÇÕES.....</b>	<b>126</b>

8.1 CUMPRIMENTO DO OBJETIVO E VERIFICAÇÃO DA HIPÓTESE.....	126
8.2 PRINCIPAIS CONCLUSÕES.....	127
8.3 LIMITAÇÕES DO ESTUDO.....	128
8.4 SUGESTÕES E RECOMENDAÇÕES PARA FUTUROS TRABALHOS.....	129
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>130</b>
<b>APÊNDICE A.....</b>	<b>137</b>

## LISTA DE TABELAS

<b>Tabela 3.1</b>	Tipos de Funções de Pertinência mais Utilizados.....	13
<b>Tabela 3.2</b>	Aplicações da Lógica Fuzzy na Engenharia de Transportes.....	18
<b>Tabela 4.1</b>	Grau de Pertinência dos Elementos e Divisão dos Conjuntos Fuzzy para as Variáveis Tempo (T) e Extensão (E).....	30
<b>Tabela 4.2</b>	Grau de Pertinência dos Elementos e Divisão dos Conjuntos Fuzzy para a Variável Chegada (A).....	30
<b>Tabela 4.3</b>	Grau de Pertinência dos Elementos e Divisão dos Conjuntos Fuzzy para a Variável Fila (Q).....	31
<b>Tabela 5.1</b>	Exemplos de aplicações de algoritmos genéticos em diversas áreas do conhecimento.....	39
<b>Tabela 5.2</b>	Exemplos de aplicações de algoritmos genéticos na Engenharia de Tráfego.....	39
<b>Tabela 6.1</b>	Passos para o Projeto de um Controlador Fuzzy Adotados na Dissertação.....	49
<b>Tabela 6.2</b>	Conjuntos fuzzy e seus limites de referência.....	53
<b>Tabela 6.3</b>	Base de Regras Utilizada Neste Trabalho.....	53
<b>Tabela 6.4</b>	Distribuição de Velocidades para a Geração de Veículos.....	54
<b>Tabela 6.5</b>	Volumes de Tráfego Simulados (veículos/hora).....	55
<b>Tabela 6.6</b>	Número de Possibilidades Para Cada Variável Para o Controlador de Mamdani (Concepção Original).....	60
<b>Tabela 6.7</b>	Número de Possibilidades Para Cada Variável Para o Controlador de Sugeno (Concepção Original).....	61
<b>Tabela 6.8</b>	Tamanho do Espaço de Busca Para a Concepção Original do Procedimento do Trabalho.....	61
<b>Tabela 6.9</b>	Número de Possibilidades Para Cada Variável Para o Controlador de Mamdani (Segunda Concepção).....	64
<b>Tabela 6.10</b>	Número de Possibilidades Para Cada Variável Para o Controlador de Sugeno (Segunda Concepção).....	65
<b>Tabela 6.11</b>	Tamanho do Espaço de Busca Para a Segunda Concepção do Procedimento do Trabalho.....	65
<b>Tabela 6.12</b>	Tamanho do Espaço de Busca para as Etapas 1 e 2.....	66
<b>Tabela 6.13</b>	Classificação do Atraso Segundo HCM (TRB, 2000).....	90
<b>Tabela 6.14</b>	Probabilidades Adotadas Relativas aos Operadores.....	97
<b>Tabela 7.1</b>	Cromossomos que Apresentaram os Melhores Resultados para Volume Alto.....	99
<b>Tabela 7.2</b>	Outros Cromossomos que Apresentaram o Mesmo Resultado da Melhor Resposta do Algoritmo para Volume Alto.....	107
<b>Tabela 7.3</b>	Cromossomos que Apresentaram os Melhores Resultados para Volume Baixo.....	108
<b>Tabela 7.4</b>	Outros Cromossomos que Apresentaram o Mesmo Resultado da Melhor Resposta do Algoritmo para Volume Baixo.....	116
<b>Tabela 7.5</b>	Características do melhor controlador encontrado para volume alto e do controlador utilizado por Andrade (2009).....	118
<b>Tabela 7.6</b>	Redução Média da Soma, em porcentagem, para Intervalos de Gerações (Volume Alto).....	119
<b>Tabela 7.7</b>	Características do melhor controlador encontrado para volume baixo e do controlador utilizado por Andrade (2009).....	122
<b>Tabela 7.8</b>	Redução Média da Soma, em porcentagem, para Intervalos de Gerações (Volume Baixo).....	123
<b>Tabela 7.9</b>	Características do melhor controlador encontrado para volume baixo e do melhor controlador encontrado para volume alto.....	125
<b>Tabela A.1</b>	Resultados para Volume Alto (sem desconsiderar nenhum valor).....	135

<b>Tabela A.2</b>	Resultados para Volume Alto (desconsiderando valores fora do intervalo $X \pm 2S$ ).....	136
<b>Tabela A.3</b>	Resultados para Volume Baixo (sem desconsiderar nenhum valor).....	138
<b>Tabela A.4</b>	Resultados para volume baixo (desconsiderando valores fora do intervalo $X \pm 2S$ ).....	139

## LISTA DE FIGURAS

<b>Figura 2.1</b>	Exemplo de Diagrama de Fases.....	6
<b>Figura 2.2</b>	Exemplo de Diagrama de Intervalos Luminosos.....	6
<b>Figura 2.3</b>	Características dos tipos de controle semafórico.....	7
<b>Figura 2.4</b>	Esquema de intervalos de fase atuados.....	9
<b>Figura 3.1</b>	Exemplo de Conjuntos <i>Fuzzy</i> semafórico.....	11
<b>Figura 3.2</b>	Conjuntos <i>fuzzy</i> para a idade.....	14
<b>Figura 4.1</b>	Diagrama de um Controlador <i>Fuzzy</i> .....	21
<b>Figura 4.2</b>	Conjuntos <i>Fuzzy</i> das Variáveis de Entrada Fila e Chegada.....	23
<b>Figura 4.3</b>	Conjunto <i>Fuzzy</i> da variável de saída Extensão.....	24
<b>Figura 4.4</b>	Resultado da Função de Implicação (Interseção).....	24
<b>Figura 4.5</b>	Exemplo de Agregação de Conjunto <i>Fuzzy</i> .....	25
<b>Figura 4.6</b>	Superfícies de Controle de um Controlador.....	28
<b>Figura 4.7</b>	Conjuntos <i>Fuzzy</i> e funções de pertinência do Controlador de Favilla, Machion, e Gomide.....	32
<b>Figura 4.8</b>	Conjuntos <i>Fuzzy</i> e funções de pertinência do Controlador de Nittymäki e Pursula.....	34
<b>Figura 5.1</b>	Ramificação da Computação Inspirada na Biologia.....	37
<b>Figura 5.2</b>	Diagrama Básico de Funcionamento de um Algoritmo Genético com Elitismo.....	41
<b>Figura 5.3</b>	Recombinação ocorrida no terceiro locus dos cromossomos A e B.....	45
<b>Figura 5.4</b>	Mutação no primeiro locus do cromossomo A.....	45
<b>Figura 6.1</b>	Procedimento Proposto para a Concepção de um Controlador Semafórico <i>Fuzzy</i> com uso de AG.....	50
<b>Figura 6.2</b>	Dados de Entrada do Algoritmo.....	52
<b>Figura 6.3</b>	Interseção objeto de estudo da dissertação.....	54
<b>Figura 6.4</b>	Genes do cromossomo utilizado no estudo, com suas alternativas possíveis.....	68
<b>Figura 6.5</b>	Interface apresentada ao usuário na FLT-M para o controlador de Mamdani.....	71
<b>Figura 6.6</b>	Interface apresentada ao usuário da FLT-M para a definição das funções de pertinência dos conjuntos <i>fuzzy</i> .....	72
<b>Figura 6.7</b>	Exemplos de Conjuntos <i>Fuzzy</i> “Zero”, “Pouca”, “Média” e “Muita”, da Variável Chegada, Definidos Por Funções de Pertinência Triangulares.....	74
<b>Figura 6.8</b>	Exemplos de Conjuntos <i>Fuzzy</i> “Zero”, “Pouca”, “Média” e “Muita”, da Variável Chegada, Definidos Por Funções de Pertinência Trapezoidais.....	76
<b>Figura 6.9</b>	Exemplo de Conjunto <i>Fuzzy</i> “Pequena”, da variável Fila, Definido Por uma Função de Pertinência Z.....	77
<b>Figura 6.10</b>	Exemplo de Conjunto <i>Fuzzy</i> “Longa”, da variável Extensão, Definido Por uma Função de Pertinência S.....	78
<b>Figura 6.11</b>	Exemplo um conjunto <i>fuzzy</i> “Média”, da variável Fila, definido por uma função de pertinência Gaussiana.....	80
<b>Figura 6.12</b>	Exemplo de Conjunto <i>Fuzzy</i> “Muita”, da Variável Chegada, Definido Por uma Função de Pertinência Sigmoidal.....	82
<b>Figura 6.13</b>	Interface apresentada ao usuário na FLT-M para Inserir a Base de Regras.....	83
<b>Figura 6.14</b>	Tabela de Extensões (em décimos de segundo).....	84
<b>Figura 6.15</b>	Exemplo de interseção desenhada no UnB-Sitacs.....	86
<b>Figura 6.16</b>	Janela apresentada ao se clicar duas vezes sobre um gerador de veículos.....	87
<b>Figura 6.17</b>	Janela apresentada para definição dos parâmetros gerais dos veículos.....	87
<b>Figura 6.18</b>	Janela apresentada para definição de um controlador semafórico <i>fuzzy</i> .....	87
<b>Figura 6.19</b>	Superfícies de Controle com suas Respective Penalizações.....	92

<b>Figura 7.1</b>	Superfícies de Controle dos Cromossomos com Melhores Resultados para Volume Alto com suas Respectivas Penalizações.....	101
<b>Figura 7.2</b>	Evolução do AG para Cada Semente da Etapa 1 (Volume Alto).....	104
<b>Figura 7.3</b>	Evolução do AG em Todas as Sementes da Etapa 1 (Volume Alto).....	105
<b>Figura 7.4</b>	Evolução do AG para Cada Semente da Etapa 2 (Volume Alto).....	106
<b>Figura 7.5</b>	Evolução do AG em Todas as Sementes da Etapa 2 (Volume Alto).....	107
<b>Figura 7.6</b>	Superfícies de Controle dos Cromossomos com Melhores Resultados para Volume Baixo com suas Respectivas Penalizações.....	110
<b>Figura 7.7</b>	Evolução do AG para Cada Semente da Etapa 1 (Volume Baixo).....	113
<b>Figura 7.8</b>	Evolução do AG em Todas as Sementes da Etapa 1 (Volume Baixo).....	113
<b>Figura 7.9</b>	Evolução do AG para Cada Semente da Etapa 2 (Volume Baixo).....	115
<b>Figura 7.10</b>	Evolução do AG em Todas as Sementes da Etapa 2 (Volume Baixo).....	115
<b>Figura 7.11</b>	Superfície do Controlador Utilizado por Andrade (2009).....	117

## LISTA DE SÍMBOLOS, ABREVIATURAS E ACRÔNIMOS

ADN	- Ácido desoxirribonucleico
AE	- Algoritmo Evolutivo
AG	- Algoritmo Genético
AM	- Atraso médio veicular
COG	- <i>Center of Gravity</i>
DENATRAN	- Departamento Nacional de Trânsito
Eq.	- Equação
FLT-M	- <i>Fuzzy Logic Toolbox</i>
GMP	- <i>Generalized Modus Ponens</i>
HCM	- Highway Capacity Manual
m	- Metros
MATLAB	- <i>MATrix LABoratory (Software)</i>
MAX	- Máximo
MIN	- Mínimo
MOM	- <i>Mean Of Maximum</i>
PPGT	- Programa de Pós-Graduação em Transportes
Prod	- Produto
s	- Segundos
SOM	- <i>Smallest Of Maximum</i>
Td	- Tempo ideal para que o veículo percorra uma dada distância
Tr	- Tempo real gasto pelo veículo para percorrer uma dada distância
UnB	- Universidade de Brasília
UnB-Sitracs	- Simulador da operação do tráfego em interseções controladas por semáforos
Wtaver	- Média ponderada
Wtsum	- Soma ponderada

# 1. INTRODUÇÃO

## 1.1. Apresentação

Os semáforos são dispositivos de controle de tráfego cujo uso está consolidado nas regiões urbanas. Eles podem realizar, basicamente, dois tipos de controle sobre as interseções: por tempo fixo ou atuado pelo tráfego. A vantagem deste último em relação ao primeiro é que ele se adapta às condições reais do tráfego, alteradas ao longo do tempo pela variação da demanda, sendo assim muito útil para o controle de interseções onde a demanda sofre variações freqüentes, difíceis de representar em múltiplos planos por tempo fixo. Além disso, elimina a necessidade de atualizações freqüentes nos planos semafóricos, mediante extensiva coleta e tratamento de dados de campo, requeridas para assegurar um controle por tempo fixo eficaz.

Um controlador semafórico *fuzzy* é do tipo atuado pelo tráfego e faz uso da lógica *fuzzy* para sua programação interna. Embora estudos desenvolvidos, em relação à sua aplicação a interseções isoladas, tenham apresentado resultados promissores, a concepção deste tipo de controlador é dificultada na definição da forma de implementação de seus componentes integrantes. Cada um desses componentes (função de pertinência, operadores, método de defuzificação, etc.) pode ser implementado de várias formas, possibilitando, assim, grande número de combinações possíveis para a composição do controlador.

Estudos comprovam que o desempenho de um controlador é influenciado por essas diferentes combinações. Santos (2003) constatou que diferentes conjuntos *fuzzy* de entrada, especificamente no que se refere à partição dos respectivos universos de discurso, impactam significativamente na operação dos controladores semafóricos *fuzzy*. Vaz (2006) conclui que não se pode rejeitar a hipótese de que o tipo de função de pertinência utilizada na definição dos conjuntos *fuzzy* afeta significativamente a resposta do controlador e, portanto, o desempenho do tráfego por ele controlado.

Além disso, estudos de Jacques *et al.* (2002c) indicam que uma importante fonte de variações na resposta do controlador são os diferentes métodos de defuzificação. Mostram também (Jacques *et al.*, 2002b) que os resultados são influenciados pelos diferentes tipos de conectores.

Todos os estudos anteriormente referidos mostram, ainda, que o desempenho dos controladores semafóricos *fuzzy* é afetado pelo volume médio a ser controlado. Isto é, um controlador com bom desempenho para um determinado volume médio pode não ser o mais eficaz para um volume médio diferente.

A definição do modo de implantação dos componentes de um controlador *fuzzy*, entretanto, traz grande dose de subjetividade. Isto porque os controladores são desenvolvidos com base no conhecimento e na experiência de especialistas, não existindo uma regra definida para a escolha do modo de implementação dos seus componentes. Além disso, essa implementação deve considerar que cada interseção tem suas especificidades de tráfego e que o controle do tráfego em cada local pode ter diferentes objetivos.

Ademais, cabe aqui destacar que as respostas do controlador semafórico *fuzzy* devem ser corretamente interpretadas pelos técnicos (com o auxílio de uma superfície de controle) e facilmente assimiladas pelos motoristas. Ou seja, é desejável que em um dado local, situações análogas de tráfego produzam ações de controle similares, o que somente é obtido se a superfície de controle que reflete os diferentes componentes do controlador for contínua e variar de modo suave.

## **1.2. Definição do Problema**

Como definir o modo de implementação mais adequado dos componentes de um controlador semafórico *fuzzy*, em termos de variável de desempenho de tráfego e das características da superfície de controle do controlador, levando em conta as condições de operação do tráfego a ser controlado?

## **1.3. Justificativa**

Sabendo-se que o modo de implementação dos componentes de um controlador semafórico *fuzzy* exerce impacto significativo na resposta deste e, portanto, na fluidez e segurança do tráfego na interseção controlada, e tendo ciência de que tal implementação não possui uma regra geral, sendo feita de maneira subjetiva, entende-se ser de grande valia o empreendimento de esforços que busquem encontrar métodos que auxiliem o projetista nesse processo.

Devido ao fato de que cada componente pode ser implementado de várias formas, o espaço de busca do melhor modo de implementação é muito grande, impossibilitando uma análise extensiva. Considerações iniciais realizadas para o presente estudo forneceram um espaço de busca de  $1,08562 \times 10^{21}$  possíveis soluções. Uma análise manual de todas essas possibilidades é inviável devido ao tempo necessário para sua execução. Caso seja realizada uma análise sem contemplar todas as possibilidades, o processo fica exposto ao risco de ser finalizado em ótimos locais, e não na melhor solução (ótimo global).

Neste contexto, a literatura indica que o uso de Algoritmos Genéticos é promissor para se encontrar uma solução em espaços de busca muito grandes. Embora ainda exista o risco de que o processo de busca seja finalizado em uma solução ótima local, este é minimizado pelo fato de que o AG começa a busca a partir de vários pontos iniciais e não de um único ponto. Assim, a análise contempla diferentes regiões do espaço de busca à procura da melhor solução.

#### **1.4. Hipótese**

Um procedimento para a concepção de controladores semafóricos *fuzzy* que use algoritmo genético permite a definição adequada de um modo de implementação dos componentes desses controladores, de modo a permitir que a atuação do controlador reflita as necessidades específicas do tráfego a ser controlado, em termos de variável de desempenho de tráfego, e forneça uma superfície de controle contínua, suave e coerente.

#### **1.5. Objetivo**

Elaborar um procedimento para a concepção de controladores semafóricos *fuzzy* que use algoritmo genético para definir um modo de implementação dos componentes desses controladores – tipo de função de pertinência, operadores, função de implicação, agregação, método de defuzzificação e partição dos conjuntos *fuzzy* – que:

- a) atenda da melhor forma possível, em termos de variável de desempenho de tráfego, o controle deste em uma determinada interseção, levando em conta suas características físicas e de volume de tráfego;

- b) assegure uma superfície de controle contínua, suave e coerente, que possa ser corretamente interpretada pelo técnico e facilmente assimilada pelo motorista.

## **1.6. Estrutura da Dissertação**

A dissertação está estruturada em oito capítulos. No Capítulo 1 é realizada a apresentação do assunto, a definição do problema, a justificativa do trabalho, sua hipótese e objetivo. O Capítulo 2 aborda o assunto controle semafórico, explicando um plano semafórico, os tipos de controle e o princípio da extensão. O terceiro capítulo apresenta a lógica *fuzzy*, seus conceitos e elementos, e explica o que é um sistema *fuzzy*, dando ênfase aos sistemas de controle *fuzzy*.

O Capítulo 4 versa sobre os controladores semafóricos *fuzzy*, descrevendo seus elementos integrantes. É apresentada a classificação usual dos controladores, bem como alguns exemplos de controladores semafóricos *fuzzy* desenvolvidos para interseções isoladas encontrados na literatura. O tema do Capítulo 5 é algoritmos genéticos (AGs). Inicialmente, é relatado um breve histórico dos AGs. Em seguida, são descritos sua forma de funcionamento, seus operadores e demais elementos que o compõem. São apresentadas também situações em que um AG pode ser considerado ferramenta promissora para solução do problema e o motivo da escolha de utilizá-lo no presente trabalho.

O Capítulo 6 apresenta e descreve a metodologia utilizada na dissertação, o algoritmo desenvolvido para executar o procedimento proposto, as considerações e simplificações necessárias e cada um dos programas computacionais utilizados neste trabalho. No Capítulo 7, a metodologia é aplicada e os resultados obtidos são apresentados e analisados. O Capítulo 8, por fim, contém as conclusões obtidas nesta pesquisa, suas limitações e recomendações para trabalhos futuros.

## **2. CONTROLE SEMAFÓRICO**

A disputa de fluxos de veículos conflitantes pelo espaço físico é característica marcante da operação de tráfego em interseções. Não bastasse o fato de ser ponto de convergência de fluxos veiculares, soma-se a isto a necessidade de travessia de pedestres e ciclistas no mesmo local. Sendo assim, o uso de dispositivos de controle de tráfego, mais especificamente o do semáforo, visa solucionar, ou pelo menos mitigar, tal problema.

Ao implementar a estratégia de separação dos conflitos no tempo, alternando o direito de uso da área da interseção, o semáforo pode contribuir para a redução do tempo perdido em cruzamentos, bem como do número de determinados tipos de acidentes. Logo, com sua devida implantação, espera-se proporcionar maior segurança e bem-estar à população.

Os semáforos visam assegurar, principalmente, dois atributos operacionais: fluidez e segurança. A fluidez, entendida como facilidade de se realizar o escoamento das correntes de tráfego, pode ser avaliada por alguns indicadores, sendo que o atraso médio veicular é bastante empregado para esse propósito. Neste contexto, entende-se por atraso a diferença entre o tempo de percurso sob condições de fluxo livre e o tempo efetivamente gasto no percurso sob as condições existentes. Isto é, o atraso reflete tanto as condições gerais de operação da corrente de tráfego em fluxo ininterrupto quanto o efeito da operação do semáforo sobre o deslocamento dos veículos por ele controlados.

### **2.1. Plano Semafórico**

O plano semafórico é o conjunto de elementos que caracteriza a programação da sinalização semafórica para certo período do dia. Seu papel é fundamental para o bom desempenho do tráfego na interseção. Integram o plano, o diagrama de fases (estágios) e o diagrama de intervalos luminosos.

- a) Diagrama de fases: representa graficamente a alocação dos movimentos em fases distintas. Os movimentos pertencentes a uma mesma fase têm de ser compatíveis (não-conflitantes). Neste trabalho, fase é sinônimo de estágio, isto é, o intervalo de tempo em que um ou mais movimentos recebem simultaneamente o direito de passagem.

A fase compreende o tempo de verde e o tempo de entreverdes que o segue. Este último é o intervalo de tempo compreendido entre o final do verde de uma fase e o início do verde da fase subsequente. A Figura 2.1 ilustra o exemplo de um diagrama de fases em um cruzamento.

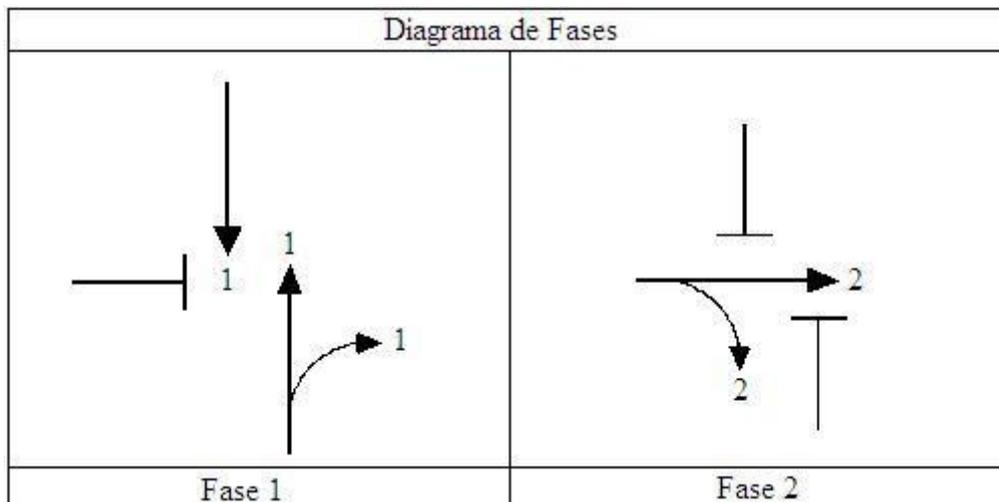


Figura 2.1 – Exemplo de Diagrama de Fases

b) Diagrama de intervalos luminosos: representa a duração e sequência dos intervalos luminosos e fases com o auxílio de barras horizontais, associando-os aos grupos semafóricos correspondentes. A Figura 2.2 ilustra o exemplo de um diagrama de intervalos luminosos.

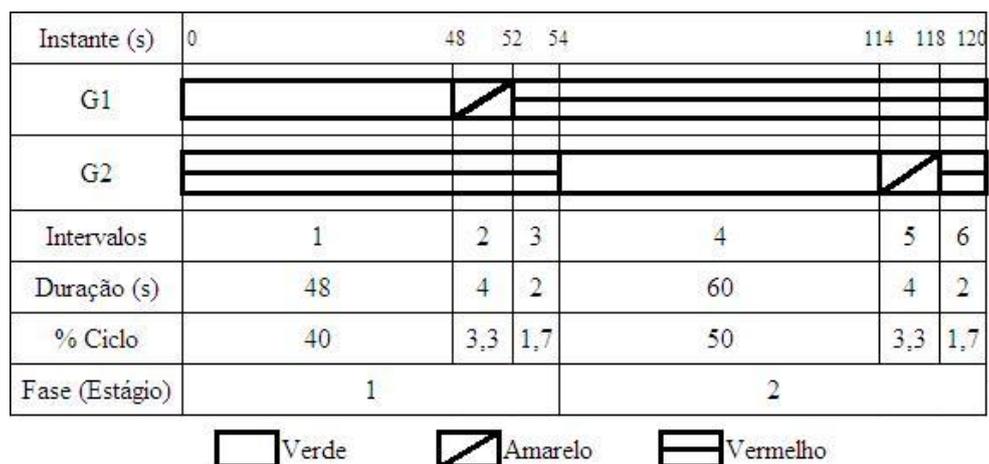


Figura 2.2 – Exemplo de Diagrama de Intervalos Luminosos

Para o caso de semáforos operando em rede, há ainda mais um elemento integrante do plano, a defasagem (*offset*). Segundo Denatran (1984), ela é a diferença de tempo entre

os instantes de início de uma fase predeterminada numa interseção (geralmente, a fase verde para a via principal) e a correspondente fase numa interseção de referência.

## 2.2. Tipos de Controle

No que diz respeito aos **tipos de controle**, os mais usuais são: por tempo fixo e por demanda de tráfego (atuado). O primeiro faz uso de planos previamente calculados, determinados por meio de dados históricos do tráfego, para diferentes períodos do dia. Diferentemente disso, o controle atuado ajusta-se às variações da demanda de tráfego, alterando a duração do tempo de verde para cada corrente de tráfego, conforme a necessidade. Estas variações de demanda são obtidas com o auxílio de detectores de veículos, usualmente instalados no pavimento da via, e de botoeiras a serem acionados por pedestres.

Para o caso de controle atuado, existem vários modelos com programações semaforicas distintas. Estes promovem o controle do tráfego usando desde simples heurísticas (como é o caso dos sistemas que adotam o princípio da extensão), até técnicas mais elaboradas como as de otimização e de inteligência artificial (Andrade, 2004). A Figura 2.3 resume as principais características dos dois tipos de controle semaforico.

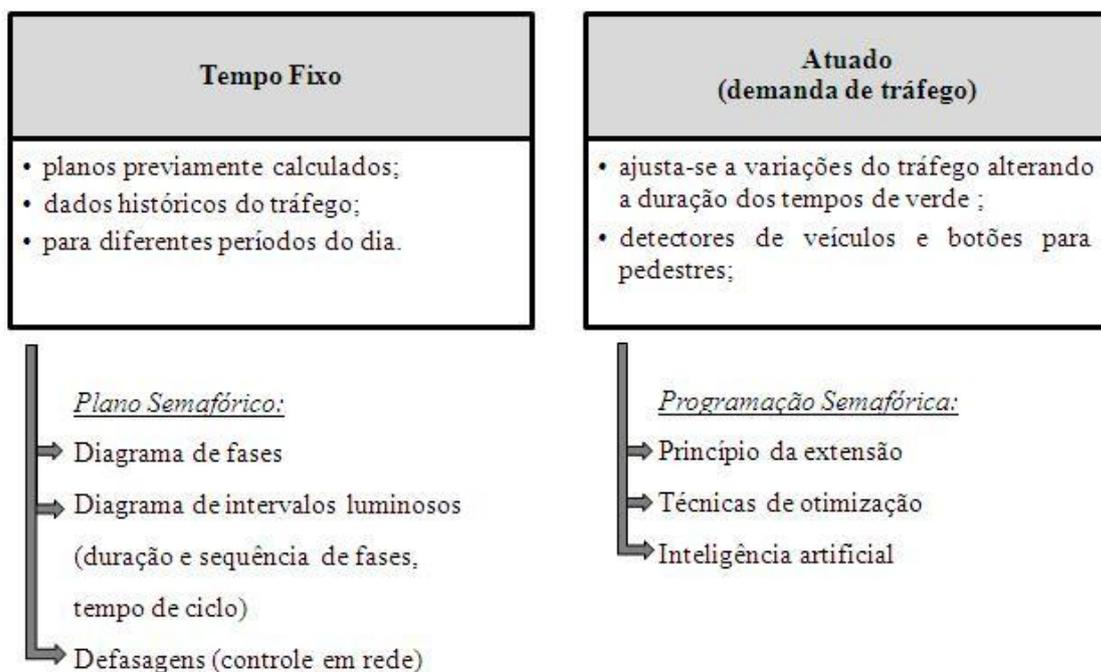


Figura 2.3 – Características dos tipos de controle semaforico

O controle por demanda de tráfego é usualmente subdividido em duas categorias, semi-atuado e totalmente atuado. A diferença básica entre as duas é que, na primeira, uma das aproximações não é atuada (geralmente, a via principal), enquanto que no segundo todas as aproximações são atuadas. Assim, no caso do controle semi-atuado, a aproximação principal recebe indicação verde por tempo indefinido, até que haja uma atuação na outra aproximação.

### **2.3. Princípio da Extensão**

A idéia básica deste princípio é de minimizar o tempo perdido na interseção. Para tanto, busca-se evitar esperas desnecessárias pelo direito de uso do cruzamento, e descarregar mais rapidamente a fila formada. A espera desnecessária ocorre quando uma aproximação muito movimentada recebe indicação vermelha por muito tempo, ao passo que a outra, pouco ou não-movimentada, recebe verde excessivo. Neste caso, haverá tempo desperdiçado, pois a indicação verde permanecerá para a aproximação menos movimentada mesmo depois de findada sua demanda, enquanto que na aproximação movimentada existem veículos na espera.

No princípio da extensão, fornece-se, inicialmente, um tempo mínimo de verde para uma aproximação e, caso seja necessário, este tempo é estendido. O período mínimo de verde corresponde ao tempo mínimo requerido para atender, com segurança, a pelo menos um veículo. Esse período é formado pela soma dos tempos do intervalo inicial (definido por meio de estudos) e do intervalo veicular (ver Figura 2.4).

O intervalo veicular é o intervalo máximo de tempo de verde permitido entre atuações sucessivas. Ele é o tempo necessário para que o veículo que entrou na interseção saia da mesma em segurança. O intervalo veicular é reiniciado após cada detecção. Para evitar espera excessiva dos veículos da outra aproximação, que recebe vermelho, fixa-se um tempo de verde máximo, cuja contagem é feita a partir do início do primeiro intervalo (intervalo inicial). Além disso, se durante um intervalo veicular concedido nenhum veículo for detectado, o verde da aproximação é encerrado ao final desse intervalo.

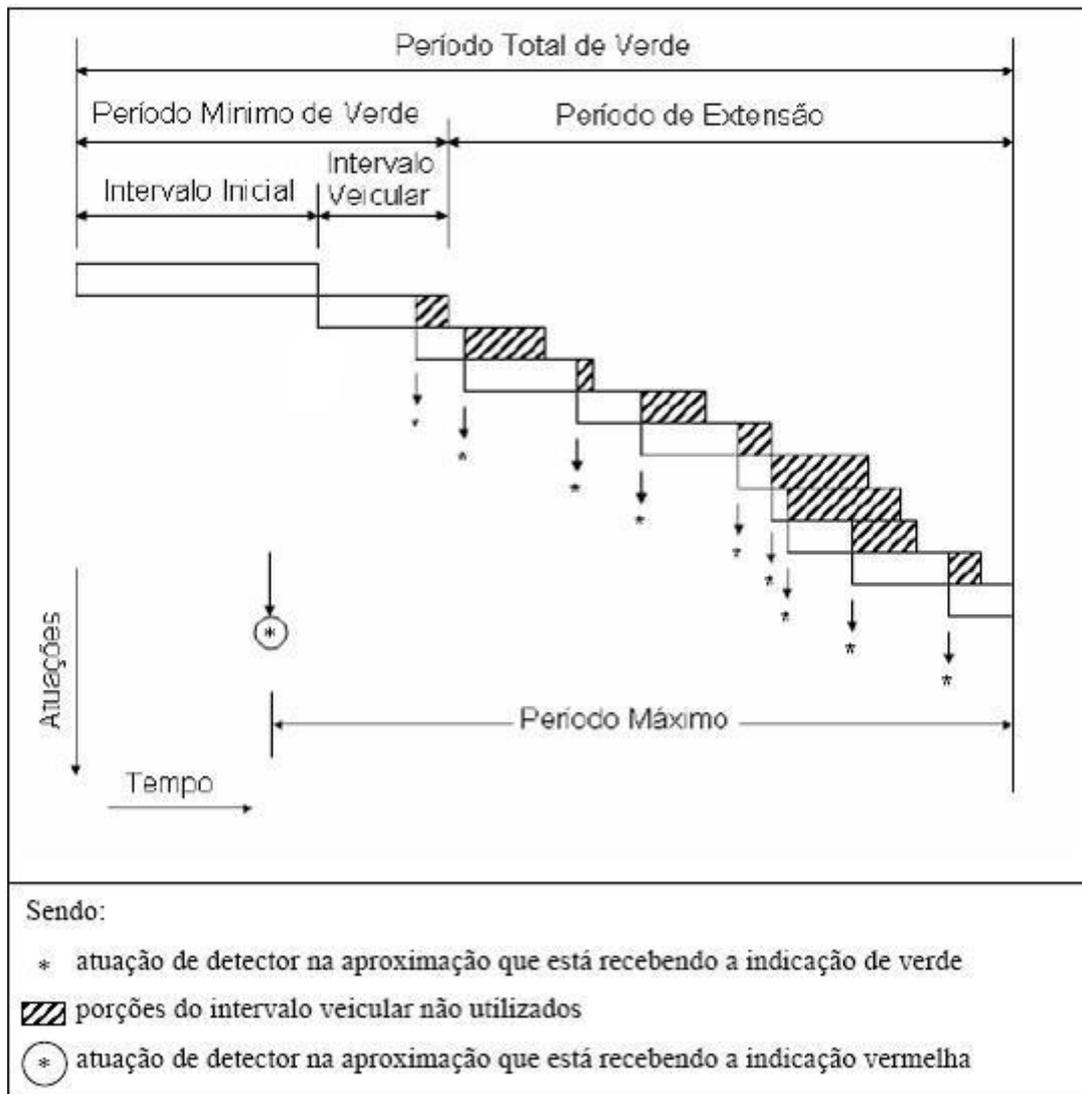


Figura 2.4 - Esquema de intervalos de fase atuados (adaptado de McShane e Roess, 1990)

Uma das características do princípio clássico da extensão aqui descrito é que a decisão do controlador de conceder ou não a extensão é baseada somente na observação da corrente de tráfego que está recebendo a indicação verde. Além disso, a priori, as extensões têm duração constante.

Esse princípio, entretanto, pode ser adaptado de modo a permitir que, na definição de extensões para a aproximação que está recebendo o verde, sejam consideradas, além do tráfego nessa aproximação, as condições da fila que está aguardando na aproximação que recebe indicação vermelha. O controlador semafórico *fuzzy* objeto do estudo desta dissertação é um exemplo do uso do princípio clássico da extensão adaptado. Esse controlador é detalhado no Capítulo 4.

### 3. LÓGICA FUZZY

O termo *fuzzy* foi utilizado originalmente pelo professor Lofti A. Zadeh da Universidade da Califórnia, Berkeley, em 1965, como uma forma de modelar a incerteza da linguagem natural observada no pensamento e julgamento humano (Santos, 2003). Diferentemente da precisão matemática, que muitas vezes se pauta unicamente no sim ou no não, a mente humana, diante de situações diárias, se rende ao uso do talvez. Assim, a lógica *fuzzy* busca fazer com que as decisões computacionais sejam mais humanizadas, refletindo escolhas que as pessoas fariam. O desafio, então, é obter dados computacionalmente manipuláveis a partir de variáveis linguísticas não muito precisas, isto é, transformar a linguagem natural em números.

Na lógica bivalente, a constatação de um evento pode ser dada por meio de dois valores absolutos, 0 (“falso”) e 1 (“verdadeiro”). Na lógica multivalorada, a constatação de um evento pode ser dada por meio de mais valores, por exemplo, 0,  $\frac{1}{2}$  e 1. Na lógica *fuzzy*, o valor “verdade” pode ser qualquer valor real no intervalo [0, 1]. Sendo assim a lógica *fuzzy* é uma extensão da lógica multivalorada, a qual utiliza diferentes metodologias para modelar conceitos vagos (Andrade, 2004).

#### 3.1. Conjuntos Fuzzy

Os conjuntos *fuzzy* representam valores que podem ser assumidos pelas variáveis linguísticas. Considere-se, por exemplo, que *fila* seja uma variável a ser analisada. Assim, pode-se definir os seguintes conjuntos *fuzzy* para sua classificação: pequena, média e grande (ver Figura 3.1).

Na teoria clássica dos conjuntos, um elemento pertence integralmente a um conjunto ou não. Ele pode até pertencer a mais de um conjunto, desde que pertença por inteiro a cada um deles. Consideremos, por exemplo, os conjuntos dos números naturais (N) e dos números inteiros (Z), conforme representado abaixo:

$$N = \{1, 2, 3, 4, 5, \dots\}$$

$$Z = \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$$

O elemento -2 pertence integralmente ao conjunto dos números inteiros, mas não pertence ao conjunto dos números naturais. Já o elemento 3, está totalmente inserido no conjunto dos inteiros e, além disso, também pertence por completo ao conjunto dos naturais. Assim, pode-se dizer que, para um determinado universo W, um elemento qualquer w de um conjunto A tem sua pertinência a este definida pela função característica  $\mu_A$ :

$$\mu_A(w) = \begin{cases} 1, & \text{se e somente se } w \in A \\ 0, & \text{se e somente se } w \notin A \end{cases} \quad (\text{Eq. 3.1})$$

No exemplo dado acima, temos então:

$$\mu_N(-2) = 0;$$

$$\mu_Z(-2) = 1;$$

$$\mu_N(3) = 1; \text{ e}$$

$$\mu_Z(3) = 1.$$

Na lógica *fuzzy*, além da possibilidade de um elemento pertencer integralmente ou não a um conjunto, introduz-se a possibilidade de ele pertencer *parcialmente* a um conjunto. Assim como nos conjuntos clássicos, em lógica *fuzzy* um elemento também pode pertencer a mais de um conjunto, no entanto, caso isso ocorra, não pertencerá integralmente a cada um deles. A Figura 3.1 exemplifica essa situação.

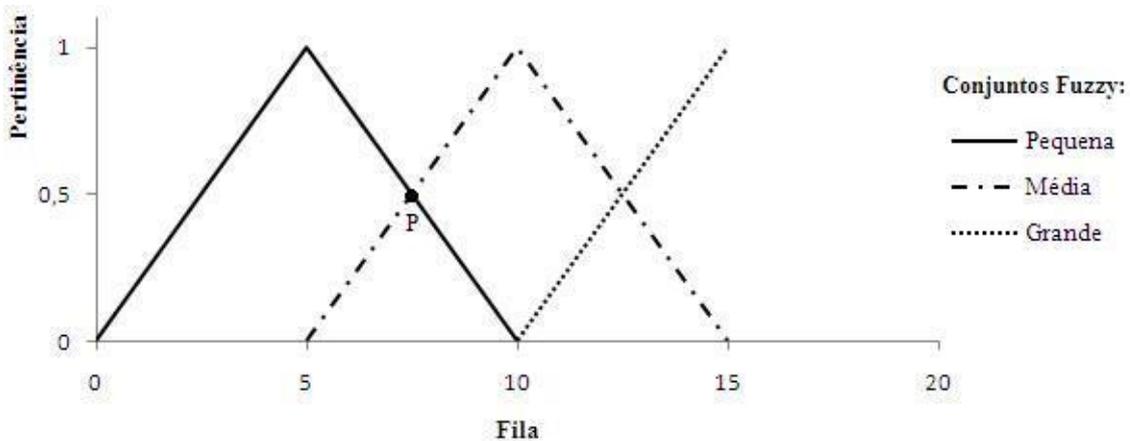


Figura 3.1 – Exemplo de Conjuntos *Fuzzy*

Nesse caso, o ponto  $P$  pertence a dois grupos, no entanto, sua pertinência não é 1 para nenhum um deles, mas sim:

$$\mu_{pequena}(p) = 0,5; \text{ e}$$

$$\mu_{média}(p) = 0,5.$$

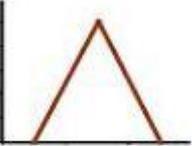
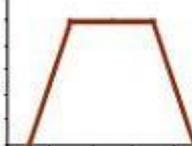
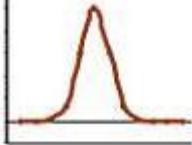
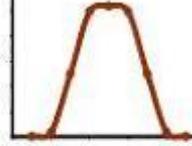
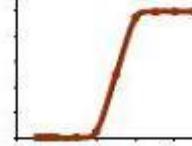
Os conjuntos *fuzzy* são formados pelos elementos propriamente ditos, associados ao grau de pertinência de cada elemento ao conjunto. Dessa forma, um conjunto *fuzzy* deve ser representado através do grau de relação existente entre cada elemento e o conjunto (Santos, 2003). Para se quantificar o grau de pertinência de um elemento a um conjunto, faz-se uso da função de pertinência.

### 3.1.1. Função de Pertinência

Uma função de pertinência associa a cada elemento do conjunto um grau de pertinência ao mesmo, isto é, indica o quanto um elemento pertence a ele. Nos exemplos anteriores, a considerada *função característica*  $\mu$  representava a função pertinência. Como se pode perceber, para os conjuntos clássicos,  $\mu$  pode assumir somente dois valores, 0 ou 1, enquanto que no caso dos conjuntos *fuzzy*, além de 0 e 1, a função pertinência comporta também os valores compreendidos entre os mesmos, isto é, assume valores do intervalo  $[0, 1]$ .

A função pertinência delinea os limites de um conjunto *fuzzy*, podendo assumir várias formas gráficas, sendo que o contexto de aplicação é que determinará se uma forma específica é adequada ou não àquela situação. A caixa de ferramenta que opera a lógica *fuzzy* no *software* Matlab<sup>®</sup> (Mathworks, 2008), chamada de *Fuzzy Logic Toolbox* – FLT-M (Jang e Gulley, 1995), apresenta onze tipos de funções de pertinência: triangular, trapezoidal, Z, S,  $\pi$ , gaussiana, sino, sigmoidal, diferença entre duas sigmoidais, combinação de duas gaussianas e produto entre duas sigmoidais (para o detalhamento de cada função, ver Vaz (2006)). Segundo Andrade (2004), os tipos mais frequentes de função pertinência são os que estão representados na Tabela 3.1.

Tabela 3.1 – Tipos de Funções de Pertinência mais Utilizados (adaptado de Andrade, 2004)

Tipo	Representação Gráfica
Triangular	
Trapezoidal	
Gaussiana	
Sino generalizado	
Sigmoidal ou Logística	

### 3.1.2 Partição do Universo de Discurso

Cada variável linguística é associada a um universo de discurso numérico, sendo que a partição deste universo é feita de acordo com os valores que a variável pode assumir. Estes valores correspondem aos conjuntos *fuzzy* (Vaz, 2006). Caso a variável a ser analisada seja a cor, por exemplo, seu universo de discurso será a frequência de ondas. A partição desse universo se dará com a divisão das frequências de onda em faixas. Cada faixa representa um conjunto *fuzzy*, e será rotulada com o nome de uma cor.

Para melhor esclarecimento dos conceitos que acabaram de ser definidos, propõe-se classificar uma pessoa de 32 anos segundo a sua idade. Ela é considerada jovem, de meia-idade ou idosa? Não há uma regra absoluta para responder a este questionamento.

No entanto, pode-se respondê-la segundo a lógica *fuzzy*, com o auxílio do seguinte exemplo, elaborado por Klir e Yuan (1995).

São considerados três conjuntos *fuzzy* para a representação de três conceitos atribuídos a uma pessoa: jovem, meia-idade e idosa. A variável linguística a ser analisada é a idade e o universo de discurso é o intervalo  $[0, 80]$ . As funções de pertinência de cada conjunto são  $A_1(x)$ ,  $A_2(x)$ , e  $A_3(x)$ , para jovem, meia-idade e idoso, respectivamente. Em que:

$$A_1(x) = \begin{cases} 1 & \text{para } x \leq 20 \\ (35 - x) / 15 & \text{para } 20 < x < 35 \\ 0 & \text{para } x \geq 35 \end{cases}$$

$$A_2(x) = \begin{cases} 0 & \text{para } x \leq 20 \text{ ou } \geq 60 \\ (x - 20) / 15 & \text{para } 20 < x < 35 \\ (60 - x) / 15 & \text{para } 45 < x < 60 \\ 1 & \text{para } 35 \leq x \leq 45 \end{cases}$$

$$A_3(x) = \begin{cases} 0 & \text{para } x \leq 45 \\ (x - 45) / 15 & \text{para } 45 < x < 60 \\ 1 & \text{para } x \geq 60 \end{cases}$$

Na representação gráfica das funções de pertinência, percebe-se que estas são do tipo trapezoidal, conforme ilustra a Figura 3.2.

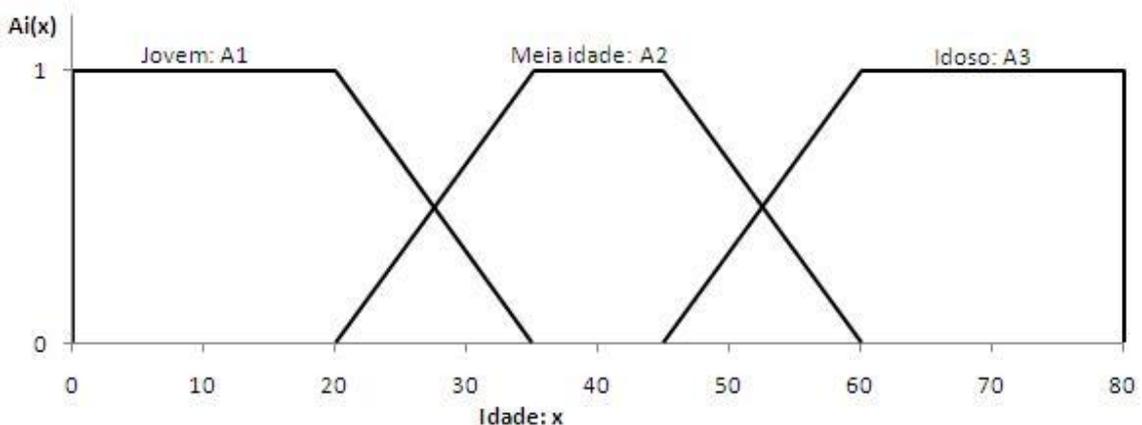


Figura 3.2 – Conjuntos *fuzzy* para a idade

Uma pessoa de 32 anos, portanto, tem grau de pertinência de 0,2 ao conjunto jovem e de 0,8 ao conjunto meia-idade. Cabe aqui ressaltar que o grau de pertinência não deve ser interpretado como um valor probabilístico. No exemplo dado, não é correto dizer que

uma pessoa de 32 anos tem 20% (0,2) de probabilidade de pertencer ao conjunto jovem e 80% (0,8) de pertencer ao conjunto meia-idade; o que ocorre é que essa pessoa possui características que permitem classificá-la nos dois conjuntos.

### **3.2. Sistemas *Fuzzy***

Segundo Klir e Yuan (1995), em geral, um sistema *fuzzy* é qualquer sistema cujas variáveis (ou, pelo menos, algumas delas) possam variar entre diferentes estados (*states*) que sejam representados por conjuntos *fuzzy*. Além disso, Kasabov (1998) afirma que os sistemas *fuzzy* são sistemas especialistas baseados em regras *fuzzy* e inferência *fuzzy*.

Um sistema especialista é aquele baseado no conhecimento de um assunto específico, e se propõe a resolver um determinado problema da mesma maneira que especialistas o fariam. Tal conhecimento é fruto de longo tempo de experiência acumulada de especialistas. No caso do sistema *fuzzy*, o conhecimento especializado está contido nas regras *fuzzy* e na definição dos conjuntos *fuzzy* das variáveis linguísticas envolvidas nessas regras.

Kasabov (1998) explica que um sistema *fuzzy* é definido por três componentes principais: variáveis *fuzzy* de entrada e de saída, definidas por seus valores *fuzzy*; conjunto de regras *fuzzy*; e mecanismo de inferência *fuzzy*.

#### *3.2.1. Sistemas de Controle Fuzzy (Controladores Fuzzy)*

Segundo Ross (2008), sistemas que podem ser controlados possuem três itens essenciais: entradas, saídas e parâmetros de controle (ou ações de controle) que são utilizados para alterar (perturbar) o sistema para um estado desejado. O sistema é monitorado e, enquanto ele permanece no estado desejado, não sofre intervenções. Caso não esteja no estado desejado, são realizadas ações de controle sobre o mesmo para levá-lo ao estado que se quer.

O controle *fuzzy* se difere do controle comum principalmente em três aspectos: (i) o uso de conceitos descritos linguisticamente, em vez de fórmulas; (ii) o uso de conhecimento de senso comum, em vez de conhecimento matemático; e (iii) uso de métodos de lógica *fuzzy* (Kasabov, 1998).

Ross (2008) destaca ainda que seis suposições básicas são comumente feitas quando se adota uma política de controle baseada em regras *fuzzy*:

1. aquilo que se propõe controlar é observável e controlável: as variáveis de entrada e saída, e as variáveis de estado (valores *fuzzy*) geralmente estão disponíveis para observação e medição;
2. existe uma base de conhecimento relacionada com um conjunto de regras linguísticas, senso comum da engenharia, intuição ou um conjunto de dados de medição de entrada e saída dos quais as regras podem ser extraídas;
3. existe uma solução;
4. o controle está buscando uma solução “boa o suficiente”, não necessariamente a solução ótima;
5. o controlador será projetado com uma margem aceitável de precisão;
6. as questões de estabilidade e otimalidade da operação do sistema não são explicitamente consideradas; estes aspectos ainda são problemas em aberto nos projetos de controladores *fuzzy*.

A seguir, são apresentados os passos para se projetar um controlador *fuzzy* (Kasabov, 1998 e Ross, 2008):

1. identificar as variáveis de entrada, estado (rótulos dos conjuntos *fuzzy*), e saída;
2. fazer a partição do universo de discurso em um número de conjuntos *fuzzy*, designando para cada conjunto um rótulo linguístico;
3. assumir ou determinar uma função de pertinência para cada conjunto *fuzzy*;
4. definir a base de regras, isto é, definir as relações entre entradas, conjuntos *fuzzy* e saída;
5. escolher um fator de escala apropriado para as variáveis de entrada e saída, de modo a normalizar as variáveis, em geral, no intervalo [0,1];

6. escolher os métodos de inferência, fuzificação e defuzificação (caso este último seja necessário);
7. testar o protótipo e fazer as alterações necessárias, tais como troca de função de pertinência e modificações na base de regras.

Os pontos críticos na construção de um sistema *fuzzy*, segundo Kasabov (1998), são a articulação das regras fuzzy e as funções de pertinência, haja vista que existem infinitas possibilidades para a definição destes.

### 3.2.2. *Sistemas Fuzzy e a Engenharia de Transportes*

De acordo com Teodorović (1998), a grande maioria dos problemas nos campos do planejamento de transportes e do controle de tráfego ainda está mal definida, ambígua e vaga. Isto porque em muitos dos problemas de transportes e tráfego, fenômenos e parâmetros são caracterizados pela subjetividade. O julgamento subjetivo está presente na escolha de rotas, estabelecimento dos níveis de serviço, percepção e reação dos motoristas, e definição de padrões de segurança, entre outros.

Neste sentido, sabendo-se que a lógica *fuzzy* permite um trato mais adequado com a subjetividade e a incerteza, ela tem sido utilizada para auxiliar a resolução desses problemas de transporte e tráfego. A Tabela 3.2 apresenta algumas dessas aplicações da lógica *fuzzy* encontradas na literatura.

Tabela 3.2 – Aplicações da Lógica *Fuzzy* na Engenharia de Transportes

<b>Aplicação da Lógica <i>Fuzzy</i></b>	<b>Pesquisadores</b>
Controle de Tráfego em Interseções	Andrade (2009, 2004)
	Jacques <i>et al.</i> (2002a,b,c)
	Niittymäki e Pursula (2000)
	Gadêlha <i>et al.</i> (1999)
	Trabia <i>et al.</i> (1999)
	Czogalla (1997)
	Kim (1994)
	Favilla <i>et al.</i> (1993)
	Chiu (1992)
	Pappis e Mamdani (1977)
Modelagem de Transportes	Abreu (1999)
	Kalić e Teodorović (1996, 1997a,b)
	Akiyama e Tsuboi (1996)
	Teodorović e Kalić (1995)
	Vythoulkas e Koutsopoulos (1994)
	Lotan e Koutsopoulos (1993a,b)
	Xu e Chan (1993a,b)
	Akiyama <i>et al.</i> (1993)
	Teodorović e Kikuchi (1990)
Análise e Prevenção de Acidentes	Schretter e Hollatz (1996)
	Busch <i>et al.</i> (1995)
	Sayed <i>et al.</i> (1995)
	Akiyama e Shao (1993)
Determinação de Níveis de Serviço	Pattnaik e Ramesh Kumar (1996)
	Ndoh e Ashford (1994)
	Chakroborthy e Kikuchi (1990)
Transporte Aéreo	Teodorović e Kalić (1995)
	Teodorović <i>et al.</i> (1994)
	Teodorović e Babić (1993)
	Larkin (1985)
Transporte Hidroviário	Vukadinović e Teodorović (1994)

Fonte: Teodorović (1998), Niittymäki (1998), Jacques *et al.* (2002a,b,c), Torres (2003), Andrade (2004), Sampaio (2004).

### 3.2.3. *Sistemas Fuzzy e Algoritmos Genéticos*

A relação entre sistemas *fuzzy* e algoritmos genéticos é bidirecional. Em uma direção, os algoritmos genéticos são utilizados para lidar com diversos problemas de otimização envolvendo sistemas *fuzzy*. Um problema importante para o qual os algoritmos genéticos se mostraram bem úteis é o problema de otimizar regras de inferência *fuzzy* em controladores *fuzzy*. Na outra direção, os algoritmos genéticos clássicos podem ser fuzificados. Os resultantes *algoritmos genéticos fuzzy* tendem a ser mais eficientes e mais apropriados para algumas aplicações (Klir e Yuan, 1995).

Nesta dissertação, os algoritmos genéticos serão utilizados no contexto da primeira direção citada, isto é, promover a otimização de um problema que envolve um sistema *fuzzy*, mais especificamente, um controlador semafórico *fuzzy*.

## 4. CONTROLADORES SEMAFÓRICOS FUZZY

O controlador semafórico *fuzzy* é um controlador *fuzzy* que realiza o controle semafórico do tipo atuado pelo tráfego. Sua programação interna, além de ser baseada na lógica *fuzzy*, faz uso do princípio da extensão adaptado. Como este último possui algumas modificações em relação à sua forma clássica, pode ser chamado também de princípio da extensão *fuzzy*.

No princípio da extensão clássico, cada extensão dada após uma detecção possui um tempo fixo e a análise é feita somente com os dados de uma aproximação. Diferentemente disso, no princípio da extensão *fuzzy* a extensão dada após o final da extensão corrente não é fixa, mas sim de duração variável. Além disso, a análise é feita na interseção como um todo, isto é, em todas as aproximações.

De acordo com Jacques *et al.* (2002a), a lógica *fuzzy* apresenta-se como adequada para aplicação em controle de tráfego, pois o conhecimento especialista acerca do sistema de tráfego pode ser extraído e organizado em conceitos e regras lingüísticas para a estruturação do banco de regras. Assim, os controladores baseados em lógica *fuzzy* são projetados com o propósito de capturar os fatores chave para promover o controle do processo, sem a necessidade de conhecimento de modelos matemáticos relativos a este processo.

### 4.1. Componentes dos Controladores Semafóricos *Fuzzy*

Compõem o controlador: variáveis lingüísticas, conjuntos *fuzzy*, base de regras, função de pertinência, operadores e método de defuzificação. Estes componentes estão inseridos nos módulos ilustrados na Figura 4.1 e serão descritos nas seções a seguir.

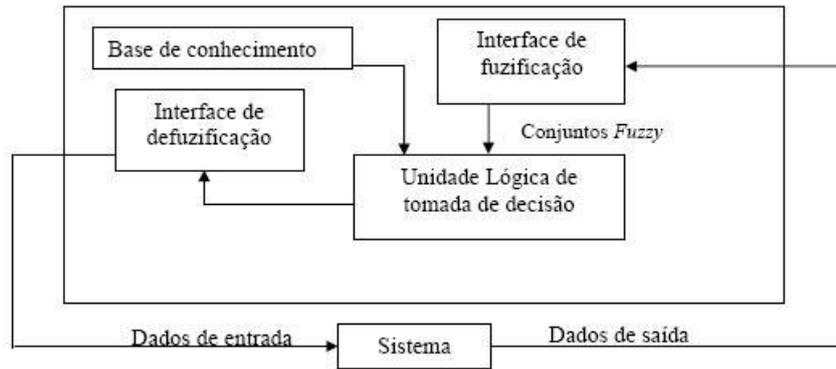


Figura 4.1 – Diagrama de um Controlador *Fuzzy* (Fonte: Niittymäki,1998)

#### 4.1.1. Interface de Fuzificação

A fuzificação consiste em transformar uma quantidade precisa em quantidade *fuzzy*. Nesta interface, os valores observados das variáveis de entrada são associados ao respectivo universo de discurso, permitindo uma avaliação do grau de pertinência aos conjuntos *fuzzy* associados a cada variável (Zadeh, 1973). Em outras palavras, os dados extraídos da situação real são traduzidos para a lógica *fuzzy*.

#### 4.1.2. Base de Conhecimento

Zadeh (1973) define a base de conhecimento como um conjunto ordenado de instruções *fuzzy* sobre cuja execução reside uma solução aproximada para um dado problema. Esta base é formada por um conjunto de regras de inferência *fuzzy* do tipo “Se – Então”. Andrade (2009) explica que essas regras são do tipo GMP (*Generalized Modus Ponens*), e uma dada regra ( $R_i$ ) pode ser apresentada como:

Antecedente:  $x$  é  $A'$  e  $y$  é  $B'$

Regra ( $R_i$ ): se  $x$  é  $A_i$  e  $y$  é  $B_i$  então  $z$  é  $C_i$

Conseqüente:  $z$  é  $C'_i$

em que  $x$  e  $y$  são variáveis lingüísticas relacionadas ao estado do processo e  $z$  é a variável lingüística de controle;  $A'$ ,  $A_i$ ,  $B'$ ,  $B_i$ ,  $C'_i$  e  $C_i$  são conjuntos *fuzzy* de  $x$ ,  $y$  e  $z$  nos universos de discurso  $U$ ,  $V$  e  $W$ , respectivamente.

Um método muito comum para a obtenção das regras *fuzzy* é a consulta a especialistas. Nele, as regras são baseadas na experiência e conhecimento destes. As funções de

pertinência e o número de rótulos devem ser definidos pelos especialistas. Contudo, além da subjetividade presente neste método (até mesmo especialistas têm opiniões divergentes), outra limitação que pode ser encontrada é a dificuldade de comunicação entre o consultado e quem realiza a entrevista. O entrevistador nem sempre tem total domínio do assunto específico e o especialista pode não estar habituado à lógica *fuzzy*. Tal fato pode levar a informações mal interpretadas.

Kasabov (1998) apresenta duas outras maneiras de se obter as regras *fuzzy*. A primeira delas é a formulada por Yamakawa (1993), em que o projetista imagina o comportamento físico do sistema real e pensa em seu significado físico nas linguagens natural e técnica. Não são necessárias equações matemáticas, no entanto, requer-se certa familiaridade do especialista com o sistema para imaginar seu comportamento. Como se pode perceber, este método é fortemente baseado na intuição e, portanto, muito subjetivo.

A segunda maneira apresentada por Kasabov (1998) para a obtenção de regras *fuzzy* é a utilização de métodos de aprendizado de máquinas, redes neurais e algoritmos genéticos para o aprendizado dessas regras e das funções de pertinência, caso estas não tenham sido escolhidas. O presente trabalho utilizará um algoritmo genético para a definição das funções de pertinência e de vários outros componentes.

#### 4.1.3. Lógica para Tomada de Decisões

Os controladores semafóricos *fuzzy* têm sido desenvolvidos no contexto de inferência *fuzzy* e de acordo com a regra de implicação *modus ponens* generalizada. A ação de controle é determinada pelas entradas observadas e pela base de conhecimento (Jacques *et al.*, 2002c).

Quando a parte inicial da regra – “Se” – é composta por mais de uma condição, aplicam-se os operadores *fuzzy* “E” e “OU”. A função de implicação é responsável por executar a regra “Então”. Por exemplo:

*SE* Fila é pequena *E* Chegada é pouca, *ENTÃO* extensão é curta.

De acordo com Jacques *et al.* (2002b), existem vários métodos para se realizar a implementação dos operadores e da função de implicação. Na *Fuzzy Logic Toolbox* –

FLT-M (Jang e Gulley, 1995) do *software* Matlab<sup>®</sup> (Mathworks, 2008), usada como referência para o desenvolvimento da presente pesquisa, estão disponibilizadas para uso direto do usuário duas operações para se implementar tanto o operador “E” quanto a função de implicação. São elas: interseção (ou mínimo, *min*) e produto algébrico (*prod*). Na operação interseção, a função de pertinência resultante da interseção de dois conjuntos *fuzzy* de entrada A e B, com funções de pertinência  $\mu_A(w)$  e  $\mu_B(w)$ , respectivamente, é definida como o valor mínimo, para cada ponto do domínio, entre as funções de pertinência  $\mu_A(w)$  e  $\mu_B(w)$ . Já o produto algébrico corresponde ao produto das funções de pertinência dos conjuntos *fuzzy* de entrada.

Para se implementar o “OU”, o FLT-M permite o uso direto de outras duas operações: união (ou máximo, *max*) e soma algébrica (*probor*). Na operação união, a função de pertinência resultante da união de dois conjuntos *fuzzy* de entrada A e B, com funções de pertinência  $\mu_A(w)$  e  $\mu_B(w)$ , respectivamente, é definida como o valor máximo, para cada ponto do domínio, entre as funções de pertinência  $\mu_A(w)$  e  $\mu_B(w)$ . A soma algébrica, por sua vez, corresponde à soma das funções de pertinência dos conjuntos *fuzzy* entrada menos o produto destas funções.

Considere-se, por exemplo, uma determinada situação do tráfego em que a fila seja 12 e a chegada 9. Quando estes dados são inseridos nos conjuntos *fuzzy* apresentados na Figura 4.2, tem-se que o grau de pertinência de 12 ao conjunto “Longa”, da variável Fila, é de 0,4, e o grau de pertinência de 9 ao conjunto “Média”, da variável Chegada, é de 0,8.

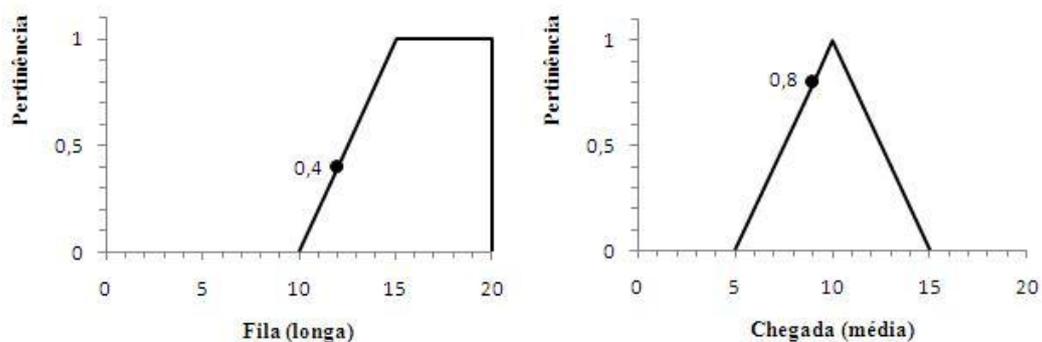


Figura 4.2 – Conjuntos *Fuzzy* das Variáveis de Entrada Fila e Chegada

Suponha-se que na base de conhecimento existe uma regra que afirma que se a fila é longa e a chegada é média, então a extensão deve ser caracterizada pelo conjunto *fuzzy*

“Média”. Para interpretar essa situação, o controlador semafórico *fuzzy* necessita implementar o operador “E” e uma função de implicação que representa o “Então” da regra. Caso a forma de implementação do “E” seja a operação interseção, o controlador interpretará como resultado dessa etapa o número 0,4 (isto é, o menor entre os dois valores).

Este valor é o resultado da primeira parte de uma regra do tipo “Se-Então”. Agora, é necessário que seja aplicada a segunda parte da regra (“Então”), isto é, utilizar a função de implicação e o conjunto *fuzzy* “Média”, para a variável Extensão. Para tanto, suponha-se que a forma de implementação da função de implicação seja também a operação interseção. Os dados de entrada dessa operação agora são o valor 0,4 e o conjunto *fuzzy* da variável de saída (Figura 4.3), o qual será “cortado” pelo primeiro dado de entrada. O resultado é o conjunto ilustrado na figura 4.4.

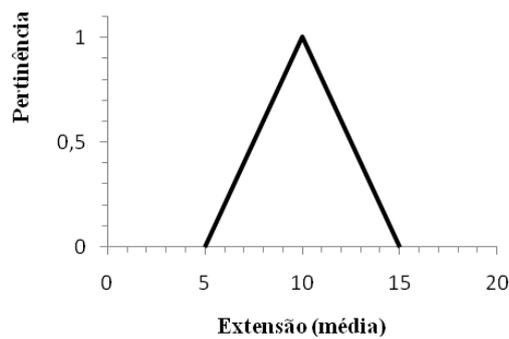


Figura 4.3 – Conjunto *Fuzzy* da variável de saída Extensão

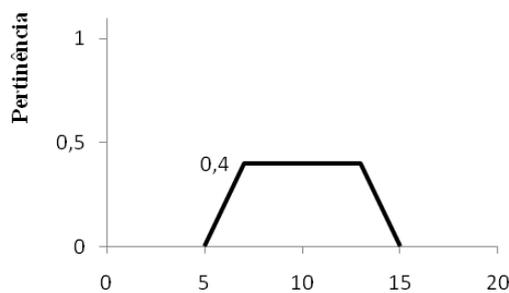


Figura 4.4 – Resultado da Função de Implicação (Interseção)

#### 4.1.4. Interface de Defuzificação

Após ter passado pela interface de fuzificação e pela unidade lógica de tomada de decisão, os produtos obtidos ainda são conjuntos *fuzzy*. É necessário então que estes sejam agregados de forma que a saída corresponda a um único conjunto *fuzzy* para cada

variável de saída, formado por todos os possíveis valores de controle e seus respectivos graus de pertinência. A FLT-M apresenta os seguintes operadores para implementar a agregação dos conjuntos: união (max), soma algébrica (probor) e a soma de cada regra de saída (sum).

A Figura 4.4, por exemplo, corresponde ao conjunto *fuzzy* de saída de uma única regra. No entanto, como a base de conhecimento é constituída de várias regras, o que acontece na realidade é que são gerados vários conjuntos *fuzzy* de saída e estes devem ser agregados para formar um único conjunto *fuzzy* final. A Figura 4.5 ilustra a agregação, implementada pela operação união (max), de conjuntos *fuzzy* de saída para um outro exemplo em que a base de conhecimento tenha 14 regras. Neste caso, utilizou-se o operador E, implementado pela operação interseção (min), e a função de implicação também foi implementada pela interseção. Na situação apresentada, Fila é igual a 12 e Chegada é igual a 8. O conjunto *fuzzy* final é o do canto inferior direito da Figura 4.5.

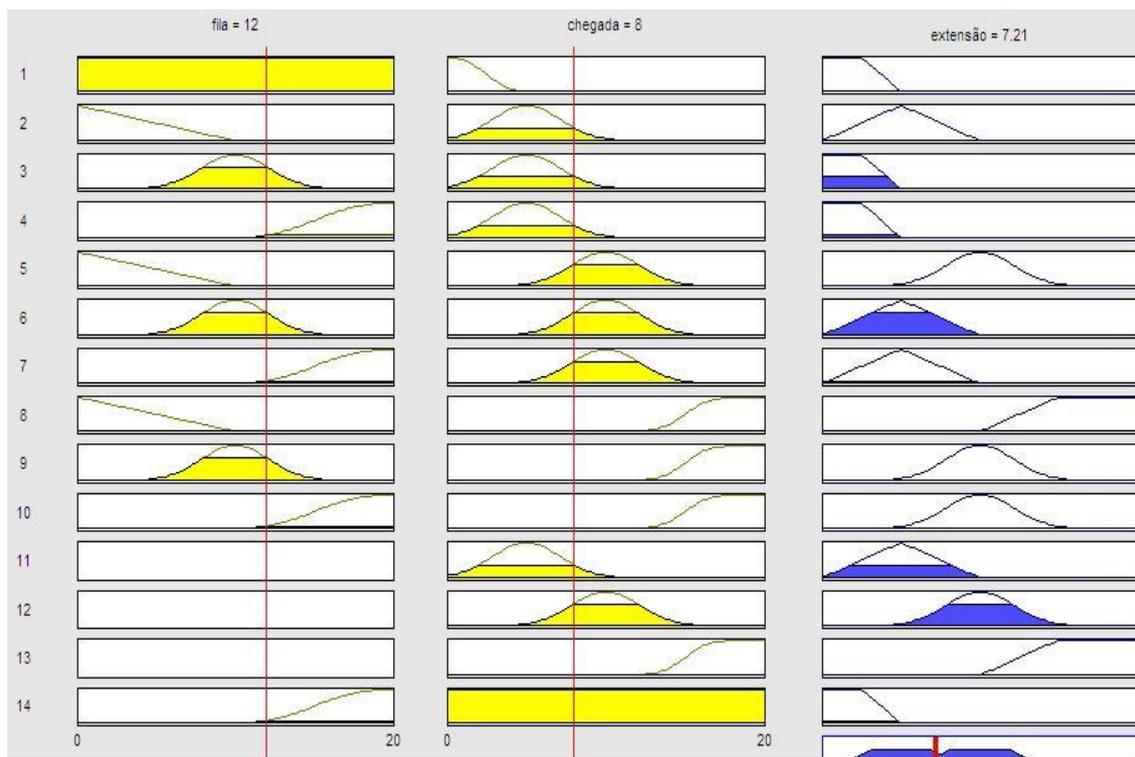


Figura 4.5 – Exemplo de Agregação de Conjunto *Fuzzy*

Em seguida, o conjunto final é convertido em um número que desencadeará a ação de controle. A defuzificação consiste em transformar o resultado *fuzzy* da variável de saída em um valor numérico. Zadeh (1973) afirma ser razoável supor que o resultado desse processo seja o elemento do conjunto *fuzzy* que possua o maior grau de pertinência a

este. No entanto, esse elemento nem sempre é único e, nesses casos, pode-se utilizar critérios para que se obtenha um único elemento. Na FLT-M são disponibilizados os seguintes métodos de defuzificação:

- **Média dos Máximos (MOM)** – corresponde à média de todos os valores de saída cujos graus de pertinência atingiram o máximo;
- **Maior dos Máximos (LOM)** – seleciona o maior entre os valores de saída que têm grau de pertinência máximo;
- **Método do Centro de Área (COA) ou Método do Centro de Gravidade (COG)** – obtém o centro de gravidade da distribuição de possibilidades da ação de controle;
- **Mínimo do Máximo (SOM)** – seleciona a menor saída com o valor máximo de grau de pertinência;
- **Bissetor da Área** – seleciona o valor que divide o conjunto *fuzzy* de saída, já agregado, em duas regiões de mesma área.

No exemplo apresentado na Figura 4.5, o método de defuzificação utilizado foi o do centro de gravidade. O resultado é representado pela barra vertical vermelha, isto é, a qual indica uma extensão de 7,21 segundos. Os cinco métodos de defuzificação descritos acima fornecem um único número como resultado. No entanto, também é possível que a defuzificação tenha como produto um conjunto clássico. Para mais informações a respeito, ver Ross (2008).

#### **4.2. Classificação dos Controladores – Mamdani e Sugeno**

De acordo com Sugeno (1985), os controladores *fuzzy* podem ser divididos em dois grandes grupos. O primeiro grupo contempla as estruturas baseadas nas funções de implicação *fuzzy* e em operadores de composição para a definição da saída *fuzzy* do controlador. Já os do segundo grupo dispensam a definição de funções de implicação e operadores para a inferência (Andrade, 2009).

O controlador descrito na Seção 4.1, é conhecido como controlador de Mamdani e integra o primeiro grupo. Os chamados controladores de Sugeno, por sua vez,

integrantes do segundo grupo, consistem numa simplificação do controlador de Mamdani. A parte conseqüente de cada regra é definida como uma função das variáveis lingüísticas de entrada, assim, não se tem mais um conjunto *fuzzy* como resultado de cada regra, mas sim um valor numérico. Com isso, não existe a necessidade de se utilizar uma função de implicação. No controlador de Sugeno, uma regra  $R_i$  pode ser escrita da seguinte maneira:

$$R_i: \text{ se } x \text{ é } A_i \text{ e } y \text{ é } B_i \text{ então } z = f_i(x, y)$$

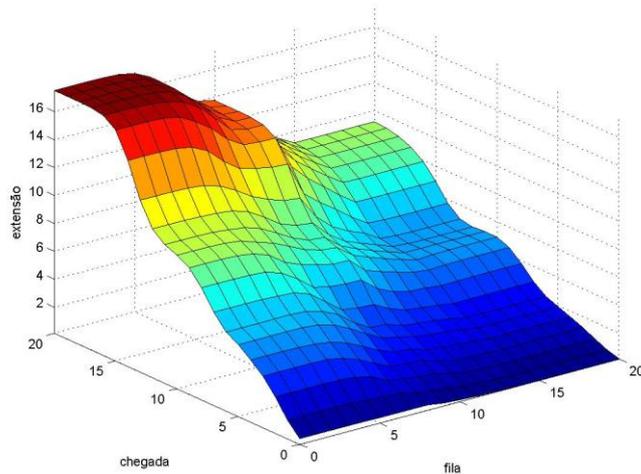
No controlador de Sugeno, a agregação e a defuzificação acontecem da seguinte forma: os resultados das regras são agregados como somas ponderadas das ações de controle correspondentes a cada regra, em que o peso de cada regra é o grau de pertinência calculado para o antecedente da regra, e, com isso, obtém-se um valor numérico final.

O valor de  $z$  pode também ser definido como um valor constante, que pode ser interpretado como um conjunto *fuzzy* com a característica especial de apresentar um único valor com pertinência igual a um e todos os demais com pertinência zero. Este tipo de conjunto *fuzzy* é denominado *singleton*, e o seu emprego permite a definição de regras com valores de saída que representam uma classificação da resposta do controlador, sem alterar a forma simplificada da determinação da resposta final do controlador (Andrade, 2009).

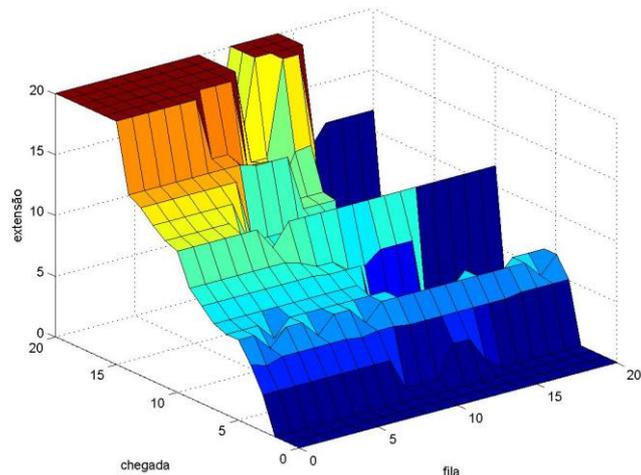
### 4.3. Superfície de Controle

As respostas do controlador semafórico *fuzzy* podem ser analisadas por meio de uma superfície de controle. Esta descreve a dinâmica de funcionamento do controlador, expressando graficamente os valores de entrada associados aos seus respectivos valores de saída.

No caso de análise de uma interseção semaforizada, por exemplo, fila e chegada podem ser as variáveis de entrada, e extensão a variável de saída. Nesse contexto, fila é o número de veículos aguardando na aproximação que recebe o indicador vermelho, enquanto que chegada representa o número de veículos na aproximação que recebe a indicação verde. A Figura 4.6 ilustra duas superfícies de controle para esta situação, geradas pelo *software* Matlab®.



(a) superfície com características desejáveis



(b) superfície com características não desejáveis

Figura 4.6 – Superfícies de Controle de um Controlador

Devido a incoerências que podem aparecer na superfície de qualquer controlador, Ross (2008) sugere que o projeto do controlador contemple técnicas que realizem ajustes e alterações na forma da superfície de controle quando necessário. A superfície ilustrada na Figura 4.6(b) apresenta incoerências como, por exemplo, o fato de, algumas vezes, o valor da extensão diminuir com o aumento da chegada para uma fila fixa.

Além da coerência, são características desejáveis de uma superfície de controle semafórico: continuidade e suavidade. A continuidade garante que cada situação real terá uma ação de controle associada a ela. E a suavidade é desejável porque, caso

ocorram mudanças bruscas ao longo da superfície, a interpretação por parte do técnico (e do usuário final, o motorista, por exemplo) fica dificultada. Alterações bruscas e não-padrionizadas no tempo de verde, por exemplo, poderiam confundir motoristas e comprometer a segurança da interseção. A superfície ilustrada na Figura 4.2(a) é um exemplo de uma superfície de controle contínua e suave.

#### **4.4. Controladores Semafóricos *Fuzzy* para Interseções Isoladas**

Na literatura, são encontradas descrições de diferentes tipos de controladores semafóricos *fuzzy* (Tabela 3.2, células referentes ao Controle de Tráfego em Interseções). Como a presente dissertação trabalha com o controle de tráfego em uma interseção isolada, considerou-se oportuno apresentar os controladores constantes das seções a seguir.

##### *4.4.1 Controlador de Pappis e Mamdani*

Pappis e Mamdani (1977) foram os primeiros a aplicar princípios de lógica *fuzzy* ao controle semafórico de uma interseção. Nesta aplicação, assumiu-se uma distribuição uniforme de veículos chegando a uma interseção isolada com vias de mão única. Supôs-se que os detectores de veículos estavam a uma distância suficiente à montante da interseção, de maneira que fosse possível informar o controlador sobre chegada de veículos na interseção nos próximos 11,5 segundos. Com essas informações, o controlador definia o valor da extensão a ser dada.

O ciclo foi dividido em dois períodos, verde efetivo e vermelho efetivo, para cada fase. O controlador é ativado a cada 10 segundos durante o período de verde de cada fase. A primeira intervenção ocorre após os primeiros 7 segundos de verde (período mínimo de verde). O número máximo de intervenções fornecidas pelo sistema é 5, sendo que cada uma pode variar de 1 a 10 segundos. Caso o tempo calculado para uma intervenção seja menor que 10 segundos, não será realizada outra intervenção e, portanto, a fase será finalizada ao final deste tempo. Assim, o tempo de verde máximo possível para uma aproximação é de 57 segundos.

As variáveis de entrada utilizadas foram tempo (T), transcorrido desde a mudança de indicação luminosa na interseção, número de chegadas (A) e fila (Q), enquanto que a

variável de saída era extensão (E). Para controlar o tráfego na interseção, os autores desenvolveram um algoritmo com várias regras do tipo “Se – Então”, como a seguinte:

SE  $T$  é muito curto e  $A$  é maior que nenhum e  $Q$  é qualquer  
 ENTÃO  $E$  é muito curto

Os resultados encontrados no controlador de Pappis e Mamdani, quando comparados aos obtidos por meio de outro controlador atuado convencional, foram melhores em todas as combinações de fluxo veicular propostas (redução do atraso médio veicular de 11-21%). As Tabelas 4.1, 4.2 e 4.3 apresentam a divisão dos conjuntos *fuzzy* e o grau de pertinência de cada elemento para as variáveis de entrada e saída desse controlador;

Tabela 4.1 - Grau de Pertinência dos Elementos e Divisão dos Conjuntos *Fuzzy* para as Variáveis Tempo (T) e Extensão (E)

Conjuntos <i>Fuzzy</i>	Tempo (s)	1	2	3	4	5	6	7	8	9	10
	Muito Curto		1	0,5	0	0	0	0	0	0	0
Curto		0	0,5	1	0,5	0	0	0	0	0	0
Médio		0	0	0	0,5	1	0,5	0	0	0	0
Longo		0	0	0	0	0	0,5	1	0,5	0	0
Muito Longo		0	0	0	0	0	0	0	0,5	1	1

Fonte: Pappis e Mamdani (1977)

Tabela 4.2 - Grau de Pertinência dos Elementos e Divisão dos Conjuntos *Fuzzy* para a Variável Chegada (A)

Conjuntos <i>Fuzzy</i>	Chegada (veíc.)	1	2	3	4	5	6	7	8	9	10
	Nenhum		0,5	0,2	0,1	0	0	0	0	0	0
Um Pouco		1	0,5	0,2	0,1	0	0	0	0	0	0
Pouco		0,5	1	0,5	0,2	0,1	0	0	0	0	0
Médio		0,2	0,5	1	0,5	0,2	0,1	0	0	0	0
Muito		0,1	0,2	0,5	1	0,5	0,2	0,1	0	0	0
Excessivo		0	0,1	0,2	0,5	1	0,5	0,2	0,1	0	0

Fonte: Pappis e Mamdani (1977)

Tabela 4.3 - Grau de Pertinência dos Elementos e Divisão dos Conjuntos *Fuzzy* para a Variável Fila (Q)

Conjuntos <i>Fuzzy</i> Fila (veíc.)	Muito Pequena	Pequena	Pequena Maior	Média	Grande	Muito Grande
4	0	0	0	0	0	0
5	0,5	0	0	0	0	0
6	0,7	0	0	0	0	0
7	0,9	0	0	0	0	0
8	0,1	0	0	0	0	0
9	0,9	0,5	0	0	0	0
10	0,7	0,7	0	0	0	0
11	0,5	0,9	0	0	0	0
12	0	1	0	0	0	0
13	0	0,9	0,5	0	0	0
14	0	0,7	0,7	0	0	0
15	0	0,5	0,9	0	0	0
16	0	0	1	0	0	0
17	0	0	0,9	0,5	0	0
18	0	0	0,7	0,7	0	0
19	0	0	0,5	0,9	0	0
20	0	0	0	1	0	0
21	0	0	0	0,9	0,5	0
22	0	0	0	0,7	0,7	0
23	0	0	0	0,5	0,9	0
24	0	0	0	0	1	0
25	0	0	0	0	0,9	0,5
26	0	0	0	0	0,7	0,7
27	0	0	0	0	0,5	0,9
28	0	0	0	0	0	1
29	0	0	0	0	0	0,9
30	0	0	0	0	0	0,7
31	0	0	0	0	0	0,5
32	0	0	0	0	0	0

Fonte: adaptado de Pappis e Mamdani (1977)

#### 4.4.2 Controlador de Favilla, Machion, e Gomide

O controlador de Favilla *et al.* (1993), basicamente, compara as chegadas veiculares da aproximação que está recebendo a indicação de verde com a fila em espera na aproximação que recebe vermelho, para então decidir se fornece ou não a extensão. Seu diferencial é a capacidade de ajustar as funções de pertinência da variável de saída (extensão) de forma *on line*, isto é, de acordo com as condições do tráfego. Tal

modificação é baseada na percepção do desempenho do controlador, medido em termos de fila residual ao final da fase de verde e na variação da fila durante a fase de verde.

São utilizadas seis variáveis linguísticas, sendo que três delas são iguais às de Pappis e Mamdani, Chegada, Fila e Extensão. Contudo, a definição dos conjuntos *fuzzy* para estas três é diferente (ver Figura 4.7). As outras variáveis são *Fila residual no final da fase verde*, *Variação da fila durante a fase verde* e *Variação do limite máximo das funções de pertinência para Extensão*.

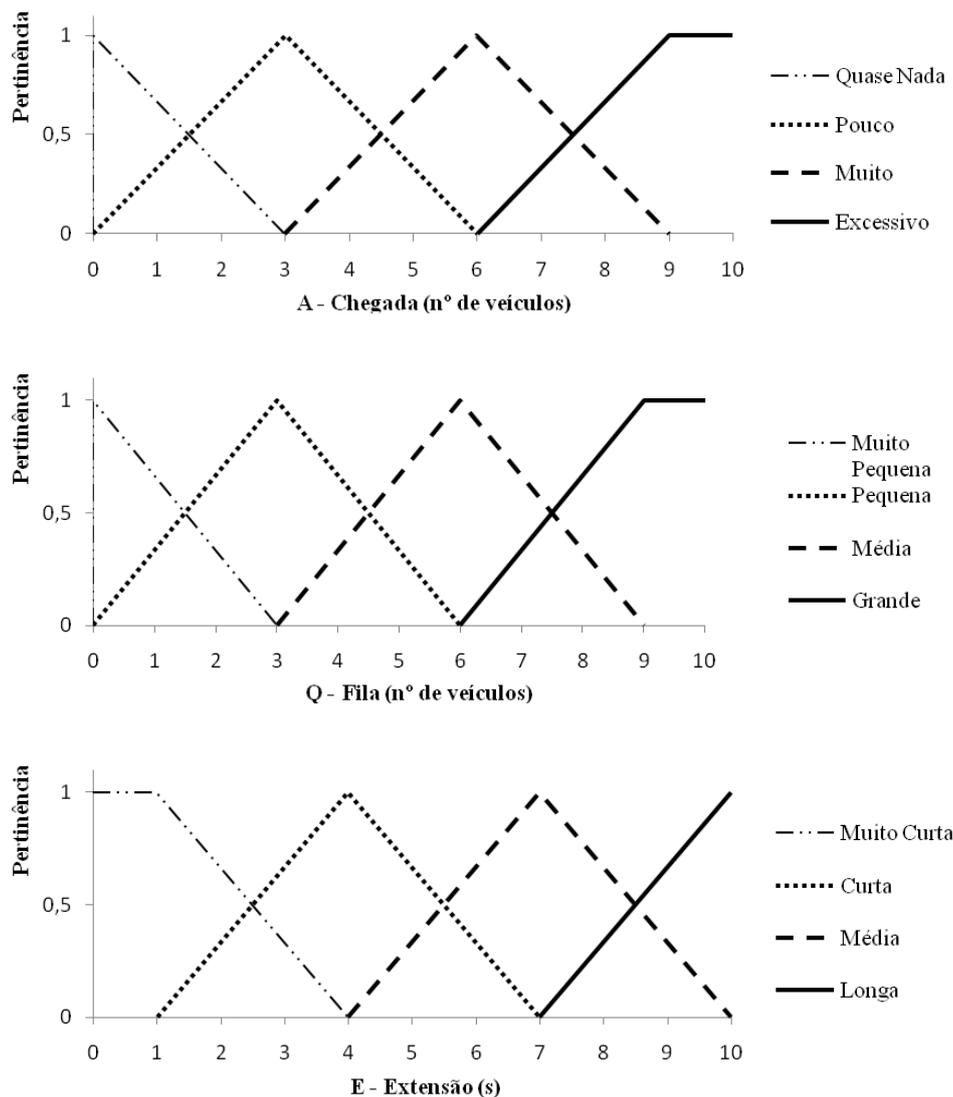


Figura 4.7 - Conjuntos *Fuzzy* e funções de pertinência do Controlador de Favilla, Machion, e Gomide (Fonte: adaptado de Favilla *et al.* (1993))

Assim como no estudo de Pappis e Mamdani, a interseção analisada neste caso era isolada. Sua geometria era formada por duas aproximações, igualmente solicitadas,

compostas por duas e três faixas de trânsito. Quanto aos resultados do estudo, conseguiu-se uma redução de 8,9% do atraso médio veicular quando comparado aos resultados do controlador de Pappis e Mamdani.

#### 4.4.3 Controlador de Niittymäki e Pursula

O modelo proposto por Niittymäki e Pursula (2000) aplica a lógica *fuzzy* aos seguintes componentes de seu controlador: seletor de fases *fuzzy* e extensor *fuzzy*. O primeiro componente tem como objetivo determinar a sequência de fases mais apropriada. Findada uma fase, o controlador deve decidir qual é a próxima. Isto é feito atribuindo-se pesos a cada fase.

O principal objetivo do extensor *fuzzy* é maximizar a capacidade de uma extensão minimizando os períodos de entreverdes dos grupos semaforicos (Könönen e Niittymäki, 2000). São instalados dois detectores em cada aproximação, um a 100 metros a montante da faixa de retenção e o segundo na própria faixa, delimitando assim a chamada zona de aproximação. O tempo mínimo de verde é de 5 segundos e o sistema permite um máximo de cinco intervenções, sendo que a primeira ocorre logo após o tempo mínimo de verde.

Niittymäki e Nevala (2001) afirmam que o controlador, quando comparado ao de Pappis e Mamdani, apresentou uma redução de 10-20% no atraso médio veicular, na simulação da operação do tráfego controlado. Já na análise em campo, o controlador reduziu os atrasos de 3-8s e diminuiu o número de veículos que paravam de 2-12%. As variáveis de entrada foram *chegada* e *fila*, e a de saída foi *extensão*. A Figura 4.8 ilustra os conjuntos *fuzzy* e as funções de pertinência do controlador.

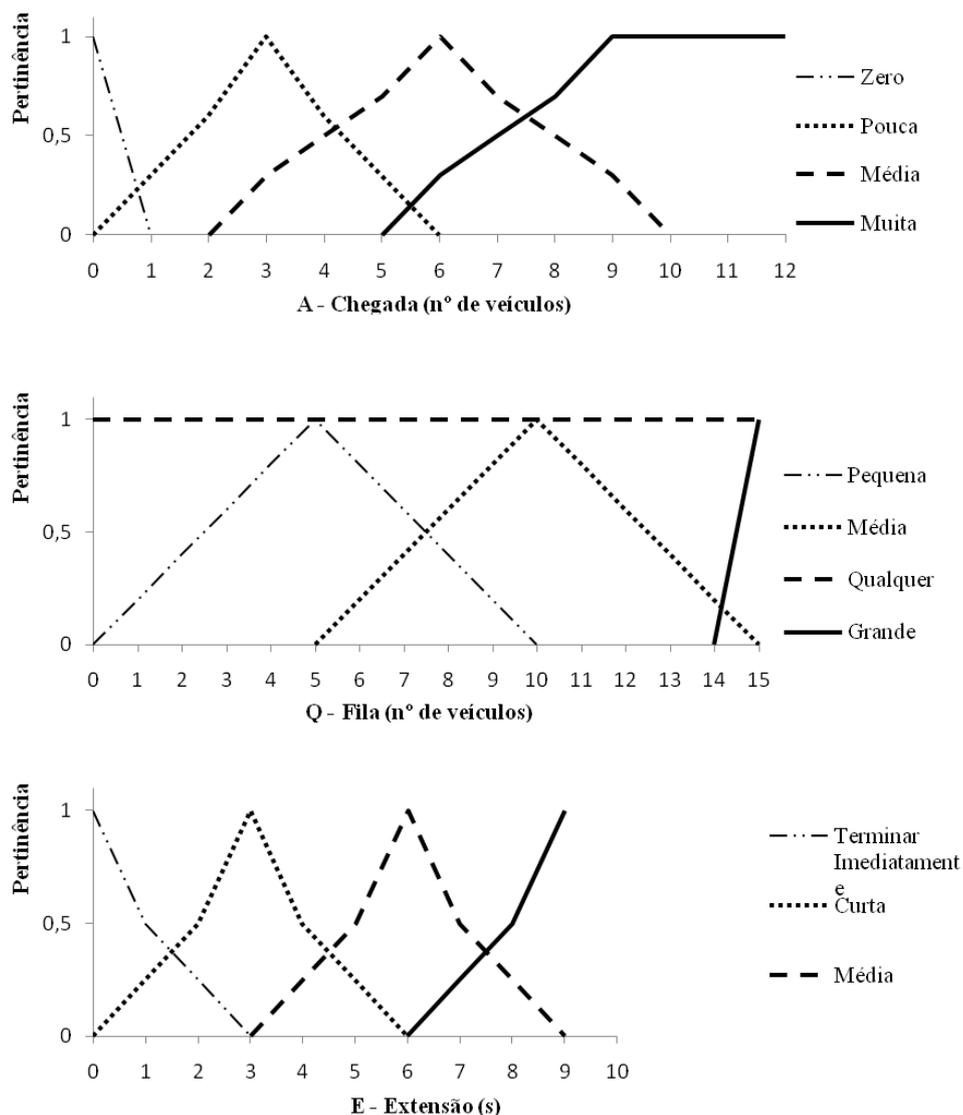


Figura 4.8 – Conjuntos *Fuzzy* e funções de pertinência do Controlador de Niittymäki e Pursula (Fonte: adaptado de Niittymäki e Pursula. (2000), e Niittymäki (1998))

#### 4.5. Considerações Relacionadas à Concepção de um Controlador Semafórico *Fuzzy*

O grande desafio da concepção de um controlador *fuzzy* em geral, e dos controladores semafóricos *fuzzy* em particular, é a definição do modo de implementação dos componentes do controlador de modo a permitir que os objetivos do controle sejam alcançados, isto é, a escolha do tipo de função de pertinência, método de defuzificação, dentre outros. No caso dos controladores semafóricos *fuzzy*, as características geométricas da interseção, o volume e a composição do tráfego também devem ser levadas em conta na fase de projeto do controlador.

A dificuldade em realizar todas as escolhas exclusivamente com base na experiência do especialista sugere o benefício potencial de se usar recursos de inteligência artificial para auxiliá-lo nesta tarefa. Dentre esses recursos, o presente trabalho explorará a utilização de algoritmos genéticos para a definição do modo de implementação de alguns componentes do controlador semafórico *fuzzy*. Buscar-se-á, ainda, implantar recursos que garantam as características desejáveis da superfície de controle (continuidade, suavidade e coerência).

## 5. ALGORITMOS GENÉTICOS (AGs)

### 5.1. Computação Inspirada na Biologia

O desejo de se criar a inteligência artificial vem desde o início da era da computação. Os primeiros estudiosos do assunto eram motivados, em grande parte, pela visão de dotar os programas computacionais de inteligência, habilidade para replicar e capacidade adaptativa para aprender e controlar ambientes. Eram interessados nos campos da biologia, psicologia e eletrônica (Mitchell, 1998).

Com o passar do tempo, a computação inspirada na biologia foi se ramificando, dividindo-se em três campos, a saber: computação evolutiva, redes neurais e aprendizado de máquinas (ver Figura 5.1). Os algoritmos genéticos (AGs), ramo da computação evolutiva, e as redes neurais são apontados por Torres (2003) como os exemplos mais utilizados da computação inspirada na biologia. Os AGs serão detalhados na seção 5.4 e as redes neurais, devido à sua importância citada, serão brevemente apresentadas na seção 5.1.2.

#### 5.1.1. Computação Evolutiva

De acordo com Mitchell (1998), nas décadas de 1950 e 1960 muitos pesquisadores passaram a estudar, de forma independente, sistemas evolutivos com a idéia de que a evolução poderia ser utilizada como uma ferramenta de otimização para problemas de engenharia. A idéia era promover a evolução de uma população de possíveis soluções usando operadores inspirados na variação genética natural e na seleção natural.

Segundo Holland (1975, *apud* Silva, 2003), a computação evolutiva sustenta-se nos algoritmos evolutivos (AEs), os quais têm como finalidade promover a “sobrevivência do mais apto”, que é conseguida por meio de “adaptação ao ambiente”, isto é, maior chance de sobrevivência dos mais aptos. Estes, conseqüentemente, possuem maior possibilidade de gerar descendentes, transmitindo as suas características genéticas.

Silva (2003) destaca que para a criação de um algoritmo evolutivo são necessários os seguintes itens:

- uma população de indivíduos (conjunto de soluções candidatas);

- um processo de seleção que avalie a aptidão de cada indivíduo da população;
- um processo de transformação, o qual gerará novas soluções a partir das existentes.

Neste contexto, os algoritmos genéticos foram desenvolvidos por John Holland, seus colegas e seus alunos na Universidade de Michigan, EUA, na década de 1960. Goldberg (1989) explica que a pesquisa de Holland e seus parceiros tinha como objetivos: (i) abstrair e explicar rigorosamente os processos adaptativos dos sistemas naturais, e (ii) projetar um *software* de sistemas artificiais que contivesse mecanismos importantes dos sistemas naturais.

A literatura aponta os algoritmos genéticos (AGs) como o exemplo de maior proeminência do ramo da computação evolutiva. Estes não apresentam certas limitações, tais como população inicial pequena e impossibilidade de usar a recombinação (*crossover*) para variação, encontradas nas Estratégias Evolutivas, de Rechenberg (1973), e na Programação Evolutiva, de Fogel *et al.* (1966).

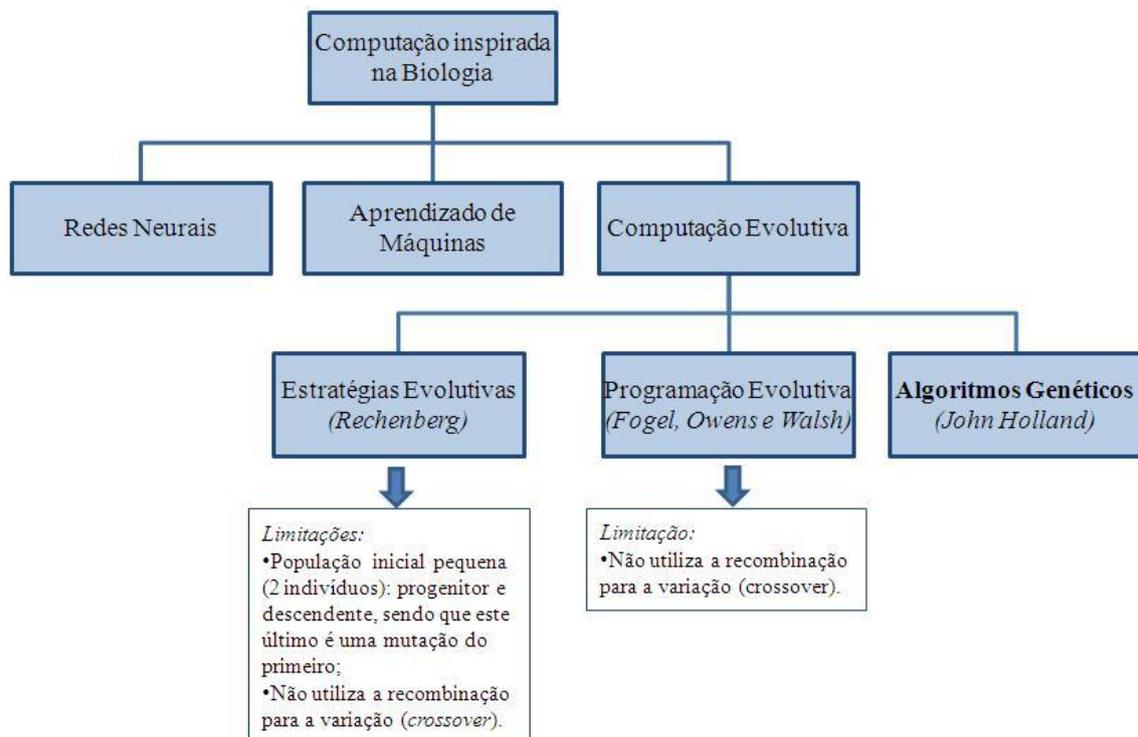


Figura 5.1 – Ramificação da Computação Inspirada na Biologia

### 5.1.2. Redes Neurais

Inspiradas no funcionamento do cérebro humano, as Redes Neurais Artificiais, ou somente Redes Neurais, são definidas como um poderoso processador compacto, composto de diversas unidades de processamento simples (neurônios ou nós) distribuídas em paralelo (PDP, *parallel distributed processing*) e que adquirem conhecimento de forma experimental por meio do processo de aprendizagem, criando regras próprias para o ambiente no qual está inserido (Haykin, 1999).

Os neurônios são dispostos em camadas e interligados por conexões. O conhecimento adquirido é armazenado nos pesos das conexões dos neurônios (pesos sinápticos). Segundo Haykin (2001), no modelo geral de neurônio artificial os sinais de entrada são multiplicados pelos seus pesos sinápticos e em seguida são somados. Esta soma passa por uma função de ativação, a qual deve restringir a amplitude do sinal de saída a um valor finito.

De acordo com Pedroso (2006), a vantagem mais significativa para a aplicação da técnica de redes neurais para estudos em transportes é sua elevada capacidade de associação de padrões com elevado grau de não-linearidade. Por outro lado, dentre suas desvantagens, destacam-se a inexistência de regras bem definidas que apontem para a escolha da melhor arquitetura a ser utilizada e a dificuldade de escolha dos parâmetros relacionados à definição da rede (número de camadas, número de neurônios em cada camada, função de ativação dos neurônios, etc.).

Diante das dificuldades citadas acima, e tendo ciência de que a literatura aponta os algoritmos genéticos como sendo uma ferramenta de fácil elaboração e manipulação (também possuem parâmetros a ser definidos, todavia, em menor quantidade) e com bons resultados, optou-se pelo uso dos AGs neste trabalho. Outros fatores que contribuíram para esta escolha estão na seção 5.5.

## 5.2. Aplicações dos AGs

Na literatura são encontradas várias aplicações dos Algoritmos Genéticos em diversas áreas do conhecimento, conforme apresentam as Tabelas 5.1 e 5.2.

Tabela 5.1 - Exemplos de aplicações de algoritmos genéticos em diversas áreas do conhecimento

<b>Área de estudo</b>	<b>Aplicação do Algoritmo Genético</b>
Ecologia	Modelagem de fenômenos ecológicos.
Sistemas Sociais	Estudar aspectos evolutivos dos sistemas sociais, como a evolução do comportamento social em colônias de insetos.
Programação Automática	Desenvolver programas de computador para tarefas específicas e para projetar outras estruturas computacionais.
Aprendizado Artificial	Várias aplicações, incluindo tarefas de classificação e predição, como predição do tempo ou de estruturas protéicas.
Economia	Desenvolver estratégias de venda e modelagem de processos de inovação.
Sistemas Imunológicos	Modelagem de vários aspectos do sistema imunológico natural, incluindo mutação somática durante o período de vida de um indivíduo.
Evolução e Aprendizado	Estudo de como o aprendizado individual e a evolução de espécies afetam um ao outro.
Otimização	Ampla variedade de tarefas de otimização, incluindo otimização numérica e problemas de otimização combinatória como o desenho de circuitos.

Fonte: Goldberg (1989) e Mitchell (1998)

Tabela 5.2 – Exemplos de aplicações de algoritmos genéticos na Engenharia de Tráfego

<b>Ano</b>	<b>Descrição</b>	<b>Pesquisadores</b>
2009	Uso de algoritmo genético para controle de tráfego aéreo	Hu, X.-B.; Di Paolo, E.
2006	Estudo sobre controle inteligente de interseções semaforizadas baseado em algoritmo genético	Yuanli, G.; Shanmei, L.; Lei, Y.; Chunfu, S.
2005	Otimização em tempo real para controle semafórico ajustável ao tráfego utilizando algoritmo genético	Lee, J.; Abdulhai, B.; Shalaby, A.; Chung, E.-H.
2004	Algoritmo genético para calibração automática de um simulador de tráfego em rodovias simples	Egami, C. Y. e Setti, J. R.
2003	Modelagem de algoritmo genético para otimização, previsão e orientação de tráfego viário	Silva, O. Q.
2003	Determinação de Rotas Ótimas de Ônibus Urbanos Utilizando Algoritmo Genético	Torres, A. C. S.
2002	Método para calcular valores da função aptidão em algoritmos genéticos para a otimização das redes de ônibus	Bielli, M., Caramia, M. e Carotenuto, P.
1999	Otimização multiobjetivo de um sistema de controle de tráfego	Sayers, T., Anderson, J. e Clement, S.

### 5.3. Termos Biológicos Importantes

Para o bom entendimento do funcionamento de um algoritmo genético, será feita em seguida uma breve definição de alguns componentes da terminologia biológica a serem utilizados.

- Ácido desoxirribonucléico – ADN (DNA): composto orgânico cujas moléculas contêm as instruções genéticas que coordenam o desenvolvimento e funcionamento de todos os seres vivos e alguns vírus;
- Genes: segmentos de ADN que são responsáveis por carregar a informação genética;
- Alelo: cada uma das várias formas alternativas do mesmo gene, ocupando um dado *locus* (posição) num cromossomo;
- Cromossomo: longa sequência de ADN que contém vários genes. Neste trabalho, o termo cromossomo, muitas das vezes, será referido como indivíduo, sendo que este é o portador do cromossomo.

Para melhor compreensão, considere-se que parte de um dado cromossomo seja a sequência de ADN (gene) responsável pela cor dos olhos de um ser humano. Cada variação de coloração (azul, por exemplo), então, representa um alelo.

### 5.4. Algoritmos Genéticos: O Que São e Como Funcionam

Na ciência da computação, pode-se entender o espaço de busca como sendo um conjunto de soluções candidatas a resolução de um determinado problema. Um algoritmo que realiza uma busca nesse espaço é um método que escolhe qual solução candidata será testada em cada etapa da busca. Na maioria dos casos, as próximas soluções candidatas a serem testadas dependerão dos resultados dos testes anteriores; os algoritmos mais utilizados assumem que haverá alguma correlação da qualidade nas proximidades (“vizinhança”) de soluções candidatas (Mitchell, 1998).

Mitchell (1998) destaca três significados, dentre outros existentes, para o termo “busca” em ciência da computação:

- i. busca de dados armazenados: deseja-se obter uma informação previamente armazenada na memória do computador;
- ii. busca de caminhos para os objetivos: a idéia é a de se encontrar, eficientemente, um conjunto de ações que possibilitarão a condução de um estado inicial até um objetivo dado;
- iii. busca de soluções: o que se quer aqui é encontrar, eficientemente, uma solução para um determinado problema em um grande espaço de busca (isto é, em meio a várias outras soluções candidatas).

Algoritmos genéticos são algoritmos de busca de soluções baseados nos mecanismos da seleção natural e na genética. Cada elemento (indivíduo) do espaço de busca é representado por uma estrutura *string*, que contém todas as características do indivíduo que estão sendo estudadas.

Em um AG, define-se primeiramente, de forma aleatória ou não, uma população inicial dentro do espaço de busca. Depois, cada cromossomo tem seu nível de aptidão avaliado pela função aptidão (*fitness*). Em seguida, o algoritmo genético seleciona aleatoriamente os cromossomos para a reprodução. A probabilidade de um cromossomo ser selecionado está relacionada com o nível de aptidão que este tem. Com isso, pode-se escolher os mais aptos sem se desprezar a diversidade provinda dos menos aptos.

Uma vez selecionados os cromossomos para a reprodução, as operações de mutação e recombinação são aplicadas. Estas, no entanto, não ocorrem necessariamente em todos os indivíduos da população; são atribuídos valores de probabilidade de ocorrência das mesmas. Tais valores são definidos a critério de quem cria o AG, de acordo com os objetivos a serem alcançados. Após as operações, a nova geração produzida é avaliada com a função aptidão.

Antes de se iniciar novamente as operações, pode-se utilizar um recurso denominado **elitismo**. Este força o AG a inserir um determinado número de melhores indivíduos da última geração na geração subsequente. Além disso, também é possível que indivíduos indesejados (“problemáticos”), conforme critérios estipulados pelo usuário, sejam eliminados da população. Retomadas as operações, o processo iterativo será, então, repetido até que se chegue ao resultado desejado, isto é, até que se alcancem os níveis

de aptidão desejados, ou até que se chegue a um limite máximo de iterações. A Figura 5.2 ilustra o diagrama básico de funcionamento de um AG com elitismo.

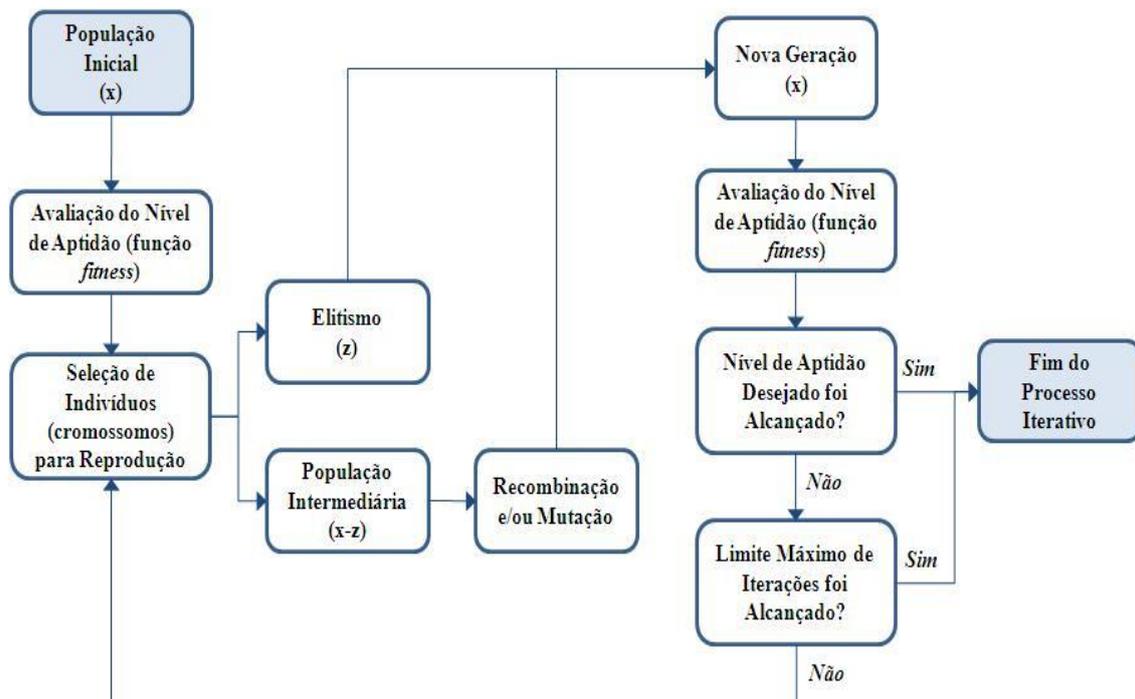


Figura 5.2 – Diagrama Básico de Funcionamento de um Algoritmo Genético com Elitismo

Os elementos básicos de um Algoritmo Genético são explicados nas seções a seguir.

#### 5.4.1. População Inicial

São as primeiras possíveis soluções para o problema, escolhidas, geralmente, de forma aleatória. Silva (2003) afirma que o tamanho da população afeta diretamente o desempenho global e a eficiência de um AG. Uma grande população geralmente fornece cobertura representativa do domínio do problema e previne de convergências prematuras para soluções locais, e não globais. Para se trabalhar com grandes populações, todavia, necessita-se de maiores recursos computacionais ou que o algoritmo trabalhe por período de tempo maior.

Os indivíduos que compõem a população inicial passam previamente pelo processo de codificação, isto é, são transformados para sua forma de representação no AG. Segundo Holland (1975, *apud* Silva, 2003), existem três tipos de codificação possíveis para os cromossomos: binária, inteira ou real.

Pacheco (1999, *apud* Sampaio, 2004), Ochi e Rocha (2000, *apud* Sampaio, 2004) afirmam que, independente da representação escolhida, o importante é que esta deve ser capaz de representar todo espaço de busca que se deseja investigar. Ou seja, toda solução deve estar associada a um cromossomo e, reciprocamente, todo cromossomo gerado pelo AG deve estar associado a uma solução válida do problema analisado.

#### 5.4.2. Função Objetivo e Função Aptidão (Fitness)

Segundo Novaes (1978), a função objetivo, utilizada em técnicas de otimização, exprime a medida de mérito do sistema analisado, por meio de uma escala única. A função aptidão apresenta idéia semelhante. Ela mede o nível de aptidão de um indivíduo. A aptidão de um indivíduo depende de quão bem ele resolve o problema dado. O diferencial da função aptidão é que ela permite que seja aplicada uma penalização quando alguma restrição do problema for violada. A função aptidão  $F(x)$  é dada por (Lemonge, 1999, *apud* Sampaio, 2004):

$$F(x) = f(x) + penal(x) \quad (\text{Eq. 5.1})$$

Em que  $f(x)$  é a função objetivo e  $penal(x)$  é a função de penalização.

Caso o problema não tenha restrições, a função de penalização não é utilizada e a função aptidão passa a ser a própria função objetivo. Sampaio (2004) explica que problemas que tenham restrições, geralmente, não são de fácil resolução devido à dificuldade de se definir a ponderação da função de penalização.

WRIGHT (1931, *apud* Mitchell, 1998) criou o conceito de superfície *fitness*. Esta é uma representação de todos os elementos possíveis do espaço associados ao seu nível de aptidão. Essa superfície, definida pela função aptidão, possui “picos” e “vales”. O Algoritmo Genético, por meio de seus operadores de recombinação (*crossover*) e mutação, desloca a população ao longo dessa superfície.

#### 5.4.3. Operadores

a) **Seleção:** seleciona indivíduos (cromossomos) da população para reprodução. Analogamente ao processo da Seleção Natural proposto por Charles Darwin, quanto maior for o nível de aptidão de um cromossomo, maior será sua probabilidade de ser

selecionado mais vezes para reproduzir. Os indivíduos que foram selecionados são copiados e posteriormente passam pela recombinação.

Um método muito utilizado para se implementar a seleção é o da roleta (*roulette wheel*). A idéia é de que cada indivíduo seja representado em uma roleta por meio de um pedaço proporcional ao seu índice de aptidão. A roleta é rodada  $N$  vezes, sendo  $N$  o número de indivíduos da população. Para cada rodada, é selecionado um indivíduo para compor uma população intermediária. Os indivíduos desta população intermediária reproduzirão (operadores recombinação e mutação) para compor a nova geração.

A probabilidade de um indivíduo ser selecionado corresponde à sua aptidão dividida pelo somatório das aptidões da geração. Uma maneira de se implementar o método da roleta para uma população de  $N$  indivíduos é: (1) calcular a aptidão de cada indivíduo; (2) ordenar os indivíduos em ordem decrescente de aptidão; (3) criar uma coluna de aptidão acumulada (a soma de todas as aptidões de uma geração será denominada de  $T$ ); (4) escolher aleatoriamente um número inteiro  $x$  entre 0 e  $T$ ; (5) selecionar o primeiro indivíduo (seguindo a coluna de cima para baixo) que possui a aptidão acumulada maior que  $x$ ; e (6) repetir os passos 4 e 5  $N$  vezes.

Muitas vezes, indivíduos com aptidão baixa têm probabilidade quase nula de serem escolhidos. Para esses casos, há um recurso utilizado nos algoritmos genéticos, denominado **janela**, o qual visa garantir a variabilidade genética. Todos os indivíduos que possuam aptidão de valor menor do que aquele estipulado pelo desenvolvedor do algoritmo (a janela) passam a ter aptidão igual ao valor da janela.

b) **Recombinação (*crossover*)**: aqui, os cromossomos escolhidos na seleção são agrupados aleatoriamente em duplas. Em seguida, defini-se aleatoriamente um *locus* e trocam-se as subsequências de antes e depois deste em cada dupla, originando descendentes distintos (quase sempre) de seus progenitores. A Figura 5.3 ilustra um exemplo de recombinação no terceiro *locus* entre os cromossomos A e B, previamente escolhidos na etapa de seleção, produzindo os descendentes P e Q.



Figura 5.3 – Recombinação ocorrida no terceiro *locus* dos cromossomos A e B.

c) **Mutação:** nesta operação, o valor de um alelo (ou mais) é trocado aleatoriamente em um cromossomo. Note-se que não ocorre uma permuta como na recombinação, mas sim uma alteração de valor. De acordo com a literatura, a probabilidade atribuída à ocorrência de mutação, geralmente, é muito pequena. Isto porque seu papel principal não é o de promover muitas variações na população, como é o caso da recombinação, mas sim de evitar que a evolução tome um rumo muito direcionado sem considerar outros pontos do espaço de busca.

Tomando novamente a Figura 5.3 como exemplo, ambos os cromossomos A e B começam com “1”, assim, ainda que ocorram inúmeras recombinações, os descendentes de A e B sempre terão “1” como início. A ocorrência de mutação no cromossomo A permite que este “1” seja alterado para “0”, com isso, o AG poderá explorar novos pontos do espaço de busca que antes estavam sendo desconsiderados. A Figura 5.4 exemplifica a mutação ocorrida no primeiro *locus* do cromossomo A, em que o valor 1 é modificado para 0, transformando-se no cromossomo A’.

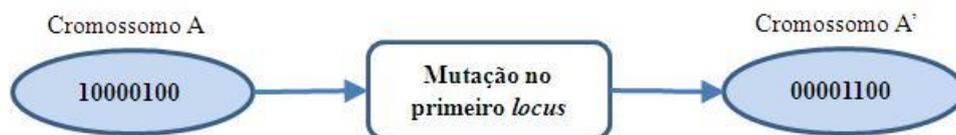


Figura 5.4 – Mutação no primeiro *locus* do cromossomo A.

Segundo Mitchell (1998), não há estudos conclusivos a respeito de qual é a melhor definição dos parâmetros de um AG, isto é, tamanho da população, probabilidade de recombinação e probabilidade de mutação, sendo que tais escolhas, geralmente, são baseadas em estudos anteriores que foram bem sucedidos. Afirma ainda que esses

parâmetros interagem entre eles de forma não linear, o que impossibilita de que sejam otimizados um de cada vez. Na literatura (Chambers (2001), Goldberg (1989), Mitchell (1998)) encontram-se vários parâmetros que forneceram bons resultados, sendo que o tamanho da população, usualmente, fica entre 30 e 100 indivíduos e as probabilidades de mutação e recombinação são no máximo igual a 0,05 e 0,08, respectivamente.

Goldberg (1989) explica que para formar um algoritmo de busca os AG's combinam dois fatores: (1) indivíduos sobreviventes (entre as estruturas de *strings* mais aptas) e (2) troca de informação estruturada, porém, aleatória. Em cada geração, novos indivíduos (*strings*) são criados utilizando-se partes dos mais aptos da geração anterior. Segundo Mitchell (1998), os AG's assumem que "pais" com bons níveis de aptidão, de diferentes regiões do espaço de busca, podem se reproduzir e gerar "descendência" de soluções candidatas com bons níveis de aptidão.

### **5.5. Algoritmos Genéticos e Outros Métodos de Busca e Otimização**

Goldberg (1989) afirma que já foi provado teórica e empiricamente que os algoritmos genéticos proporcionam uma busca robusta em espaços de busca complexos. Segundo ele, artigos e dissertações indicam a validação da técnica dos AGs na otimização de funções e aplicações de desenvolvimento de sistemas de controle. Cabe aqui ressaltar que a otimização de uma função ou de um processo é bem sucedida não só quando se encontra um valor ótimo, mas sim quando se obtém sucessivos resultados satisfatórios, melhores que os primeiros.

Goldberg (1989) cita quatro características dos AGs que os diferenciam dos tradicionais métodos de busca e otimização, que são:

- 1. Trabalham com uma codificação dos parâmetros escolhidos, e não com os próprios parâmetros;*
- 2. Realizam a busca a partir de uma população de pontos, e não de um único ponto – ao invés de se partir de um único ponto para buscar o próximo com o auxílio de regras de transição, o AG trabalha simultaneamente com vários pontos iniciais. A vantagem desse procedimento do AG é que se diminui o risco de se parar em um falso pico (ótimo local, e não global), isto é, um resultado aquém do que se poderia obter;*

3. *Trabalham com uma função objetivo, com informações de custo ou recompensa, e não derivadas ou outro conhecimento complementar;*
4. *Utilizam regras de transição probabilísticas, em vez de regras determinísticas.*

Não há uma resposta precisa para quais são os casos em que o AG é um bom método para ser utilizado para se alcançar resultados desejados. Todavia, Mitchell (1998) afirma que muitos pesquisadores compartilham as intuições de que se o espaço de busca é grande, e é sabido que este não é perfeitamente regular e unimodal (ou seja, não consiste em um único pico suave), ou não é bem compreendido, e se encontrar uma boa solução (não necessariamente a ótima) é o suficiente, então o uso do AG pode apresentar resultados satisfatórios neste problema.

Como o presente trabalho apresenta as características acima descritas, isto é, espaço de busca grande, de comportamento desconhecido, e não se conhece precisamente qual é a solução ótima para o controle de tráfego em uma interseção isolada, aceitando-se, portanto, como soluções aquelas consideradas como boas (melhores que outras estudadas), optou-se pela aplicação do AG neste estudo. Suas características serão detalhadas no capítulo seguinte.

## 6. PROCEDIMENTO PARA A CONCEPÇÃO DE CONTROLADORES SEMAFÓRICOS *FUZZY* COM O USO DE ALGORITMO GENÉTICO

### 6.1. Considerações Iniciais

O presente trabalho parte dos pressupostos (baseados em estudos anteriores) de que:

- i. o controlador *fuzzy* é aplicável ao controle semafórico com sucesso;
- ii. a resposta do controlador às solicitações do tráfego, assim como o desempenho do tráfego controlado, são significativamente afetados por características dos diferentes modos de implementação dos componentes do controlador;
- iii. usualmente, a definição dos componentes associados às variáveis de entrada e saída, e definição do conjunto de regras a ser utilizado, é feita com base no conhecimento dos especialistas em tráfego, num processo usualmente demorado;
- iv. a definição do modo de implementação dos componentes do controlador, tais como operadores, função de implicação das regras *fuzzy* e métodos de defuzzificação, é mais difícil de ser feita a partir da experiência do especialista em tráfego, requerendo, em geral, um processo de busca sobre todas as opções possíveis de combinações entre eles. Como esse número de combinações é bastante elevado, a utilização de uma técnica de busca eficiente é recomendada.
- v. os AGs podem ser empregados para o tipo de busca pretendido.

Neste trabalho, os valores lingüísticos que as variáveis de entrada e saída podem assumir (representados por rótulos de conjuntos *fuzzy*), bem como as regras que relacionam estes valores entre si (base de regras do controlador), são considerados como definidos *a priori* e não serão objeto do procedimento proposto. Com o auxílio da Tabela 6.1, adotou-se os passos propostos por Kasabov (1998) e Ross (2008) para se projetar um controlador *fuzzy*.

Tabela 6.1 – Passos para o Projeto de um Controlador *Fuzzy* Adotados na Dissertação

Passos para se Projetar um Controlador <i>Fuzzy</i> (Kasabov, 1998 e Ross, 2008)		Passo Adotado?	Comentários
1	Identificar as variáveis de entrada, estado (rótulos dos conjuntos <i>fuzzy</i> ) e saída;	Sim	Ver item 6.3.1.
2	Fazer a partição do universo de discurso em um número de conjuntos <i>fuzzy</i> , designando para cada conjunto um rótulo linguístico	Sim	Ver item 6.3.1.
3	Assumir ou determinar uma função de pertinência para cada conjunto <i>fuzzy</i>	Sim	Ver item 6.3.4.
4	Definir a base de regras	Sim	Ver item 6.3.1.
5	Escolher um fator de escala apropriado para as variáveis de entrada e saída, de modo a normalizar as variáveis, em geral, no intervalo [0,1]	Sim	Ver item 6.3.1.
6	Escolher os métodos de inferência, fuzificação e defuzificação (caso este último seja necessário)	Sim	Ver item 6.3.4.
7	Testar o protótipo e fazer as alterações necessárias, tais como troca de função de pertinência e modificações na base de regras	Sim	O AG realizará esse passo, com exceção das modificações na base de regras, que é fixa.

## 6.2. Procedimento Proposto

Para efeitos didáticos, o procedimento proposto nesta pesquisa, ilustrado na Figura 6.1, foi dividido em seis módulos: Inicial – Definições Básicas, Controlador *Fuzzy*, Simulador, Superfície de Controle, Genético e Final - Resultados. Cada módulo será detalhado nos itens que serão apresentados mais adiante.

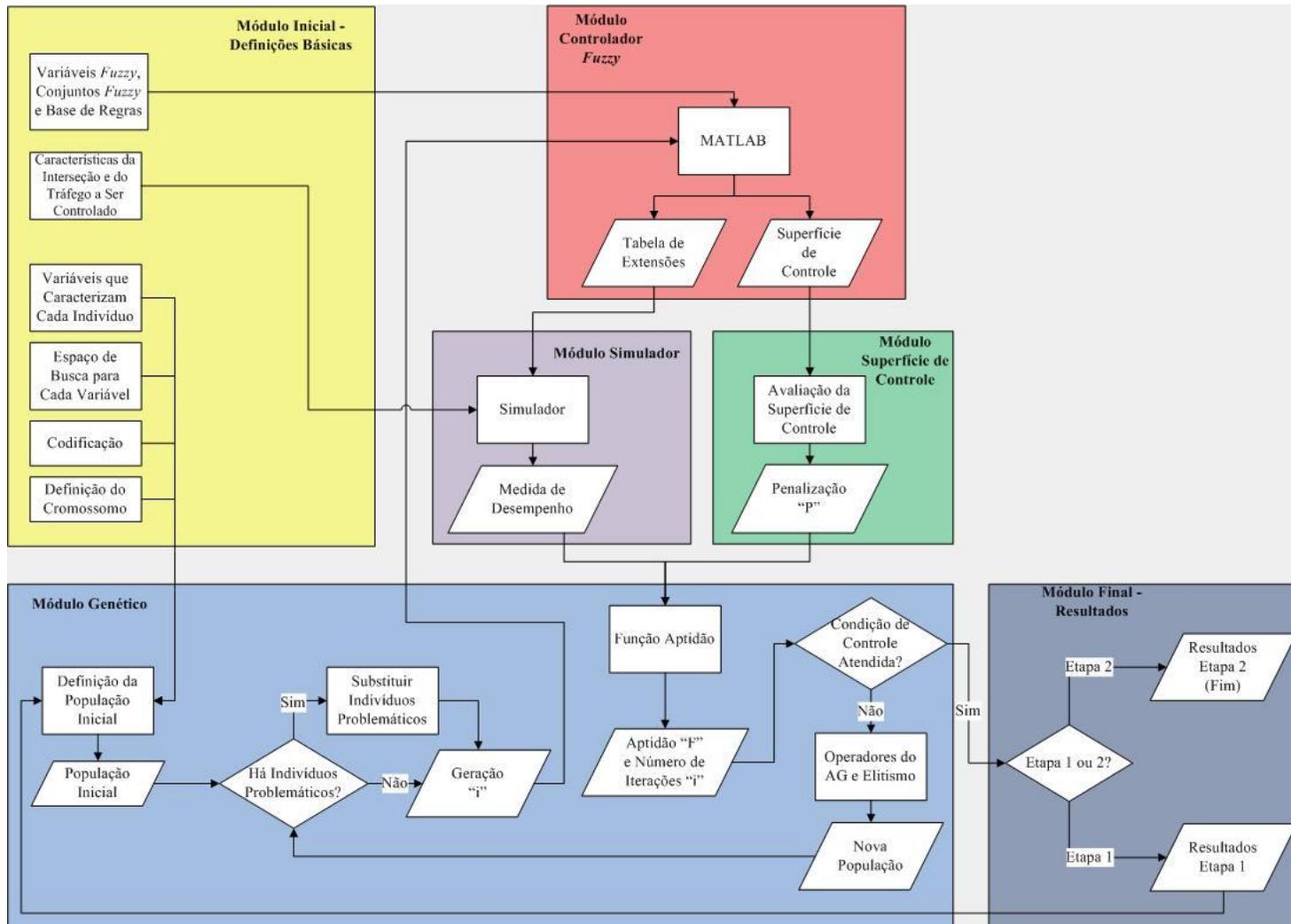


Figura 6.1 – Procedimento Proposto para a Concepção de um Controlador Semafórico *Fuzzy* com uso de AG

### 6.2.1. Algoritmo para Execução do Procedimento Proposto

Paralelamente a este estudo, Labanca (2009) testou a eficácia da utilização dos algoritmos genéticos para a definição de alguns componentes de controlador semafórico *fuzzy*. Em seu algoritmo, foram consideradas 4 variáveis (modo de implementação do operador “E”, modo de implementação da função de implicação, modo de implementação da agregação e método de defuzzificação), perfazendo um espaço de busca de 60 possibilidades. A função aptidão correspondia ao atraso médio veicular. Para as condições definidas naquele trabalho (com espaço de busca muito reduzido), os resultados obtidos apontaram para a eficácia do uso do algoritmo genético.

Para a execução do procedimento apresentado na Figura 6.1, foi desenvolvido neste trabalho um algoritmo que realizasse todas as etapas. O algoritmo em questão é uma expansão do proposto por Labanca (2009). O novo algoritmo trabalha com 27 variáveis, ao invés de 4, e percorre um espaço de busca maior, com mais de 10 milhões de possibilidades (ver Tabela 6.12). Permite também a avaliação da superfície de controle, aplicando um critério de penalização à função aptidão, e faz uso do recurso de eliminação de indivíduos problemáticos.

Neste algoritmo, o elitismo foi implementado e introduziu-se o recurso de definição de sementes para a definição da população inicial (ver item 6.3.7). Além disso, o tamanho da população inicial passou a ser um dado inserido pelo usuário, diferentemente do trabalho anterior, no qual o tamanho da população inicial era fixo dentro do próprio código de programação do AG.

A Figura 6.2 apresenta os campos de entrada de dados a serem inseridos pelo usuário. O campo “Diretório” deve conter local da pasta que armazenará os resultados. Nesta pasta também deve estar inserido o arquivo do simulador. “Nome do Simulador” deve ser preenchido com o nome do arquivo do simulador. “Tamanho da População” e “Número Máximo de Gerações” são campos com nomes auto-explicativos. O campo “Simulações para Cada Membro” refere-se ao número de vezes que um mesmo cromossomo será simulado (maiores detalhes no item 6.5.1). E o último campo, “Semente para Gerar População Inicial”, deve conter um número qualquer escolhido pelo usuário.

Diretório:

Nome do Simulador:

Tamanho da População:

Número máximo de gerações:

Simulações para cada membro:

Tempo de cada simulação:

Semente para gerar População Inicial:

OK Cancel

Figura 6.2 – Dados de Entrada do Algoritmo

### 6.2.2. Número de Vezes que o Algoritmo é Rodado

Para cada nível de volume, o algoritmo foi rodado 5 vezes, sendo que cada uma delas possuía populações iniciais distintas, geradas pelas sementes 1, 2, 3, 4 e 5 no Matlab<sup>®</sup>, conforme descrito em 6.3.7. O primeiro objetivo de se utilizar o algoritmo mais de uma vez para cada volume é de verificar se o algoritmo converge para o mesmo resultado em todas as rodagens, fato improvável devido ao tamanho do espaço de busca. O segundo objetivo é, caso os resultados não converjam, identificar quais as características comuns aos melhores controladores de cada rodagem.

## 6.3. Módulo Inicial – Definições Básicas

Neste módulo são feitas algumas definições básicas, descritas a seguir.

### 6.3.1. Variáveis Fuzzy, Conjuntos Fuzzy e Base de Regras

As variáveis *fuzzy*, os conjuntos *fuzzy* e a base de regras adotadas neste trabalho são as mesmas que vêm sendo utilizadas em outros estudos de controladores semafóricos *fuzzy* no Programa de Pós-Graduação em Transportes da Universidade de Brasília. Estes controladores, por sua vez, foram desenvolvidos tendo como referência o controlador de Niittymäki e Pursula (2000) As variáveis *fuzzy* de entrada são *Fila* e *Chegada*, e a

variável *fuzzy* de saída é *Extensão*. Os conjuntos *fuzzy* e seus respectivos limites referenciais (valores inicial e final da partição do universo de discurso correspondente) são definidos de acordo com a Tabela 6.2 (o termo “referenciais” é utilizado porque os limites poderão ser variados ao longo do processo). A base de regras, que será fixa neste estudo, é apresentada na Tabela 6.3.

Tabela 6.2 – Conjuntos *fuzzy* e seus limites de referência

<b>Fila</b>					
<b>Conjuntos Fuzzy</b>	Pequena	Média	Longa	Qualquer	-
<b>Limites</b>	[0 10]	[5 15]	[10 20]	[0 20]	-
<b>Chegada</b>					
<b>Conjuntos Fuzzy</b>	Zero	Pouca	Média	Muita	Qualquer
<b>Limites</b>	[0 5]	[0 10]	[5 15]	[10 20]	[0 20]
<b>Extensão</b>					
<b>Conjuntos Fuzzy</b>	Zero	Curta	Média	Longa	-
<b>Limites</b>	[0 5]	[0 10]	[5 15]	[10 20]	-

Fonte: adaptado de Andrade (2004)

Tabela 6.3 – Base de Regras Utilizada Neste Trabalho

<b>Regra</b>	<b>SE Fila</b>	<b>E Chegada</b>	<b>ENTÃO Extensão</b>
<b>1</b>	Qualquer	Zero	Zero
<b>2</b>	Pequena	Pouca	Curta
<b>3</b>	Média	Pouca	Zero
<b>4</b>	Longa	Pouca	Zero
<b>5</b>	Pequena	Média	Média
<b>6</b>	Média	Média	Curta
<b>7</b>	Longa	Média	Curta
<b>8</b>	Pequena	Muita	Longa
<b>9</b>	Média	Muita	Média
<b>10</b>	Longa	Muita	Média
<b>11</b>	<i>None</i>	Pouca	Curta
<b>12</b>	<i>None</i>	Média	Média
<b>13</b>	<i>None</i>	Muita	Longa
<b>14</b>	Longa	Qualquer	Zero

Fonte: Andrade (2004)

### 6.3.2. Características da Interseção e do Tráfego a Ser Controlado

A interseção objeto deste estudo, apresentada na Figura 6.3, é composta por duas aproximações, com duas faixas cada. Cada faixa possui 250 metros de extensão e 3,5 metros de largura. Os semáforos são posicionados após os primeiros 250 metros dos trechos horizontal e vertical e os laços detectores são instalados 100 metros à montante de cada semáforo. O tráfego é composto exclusivamente por carros. Considerou-se que todos os veículos seguem em direção reta, sem a possibilidade de que sejam realizadas conversões. Os percentuais de velocidade adotados para a geração de veículos no simulador são os mesmos utilizados por Andrade (2009), conforme apresentado na Tabela 6.4.

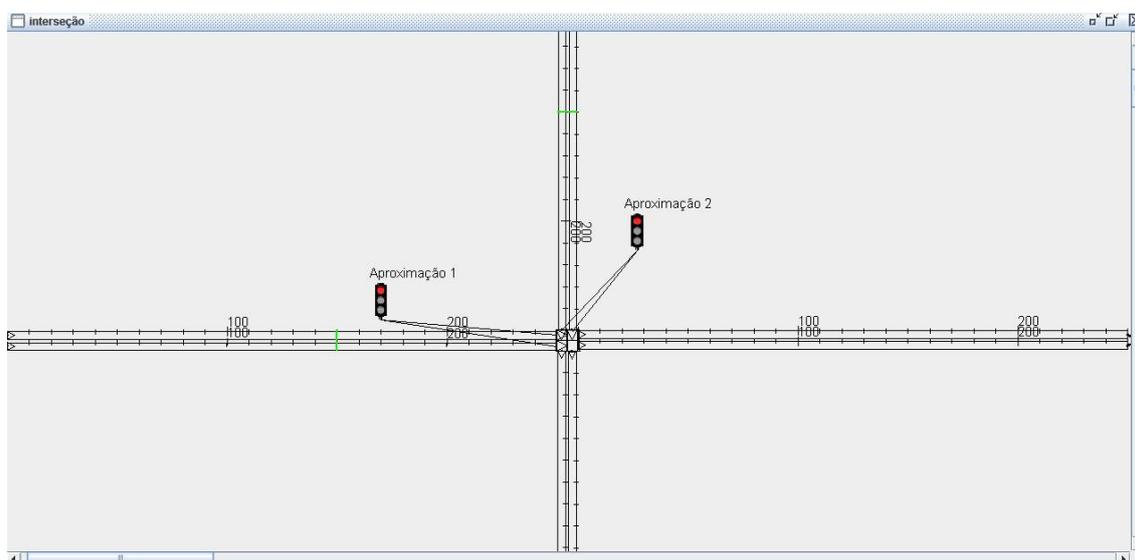


Figura 6.3 – Interseção objeto de estudo da dissertação

Tabela 6.4 – Distribuição de Velocidades para a Geração de Veículos

Velocidade (km/h)	Percentual
30	2%
40	15%
50	70%
60	10%
70	3%

Foram realizadas simulações para 2 tipos de volume, alto e baixo, sendo que as quantidades destes são as mesmas adotadas no trabalho de Andrade (2009). A Tabela 6.5 apresenta os volumes alto e baixo. Todos os dados dessa etapa devem ser inseridos

uma única vez pelo usuário no próprio simulador e salvos em um arquivo do tipo XML, o qual será acessado posteriormente pelo algoritmo para realizar as simulações.

Tabela 6.5 – Volumes de Tráfego Simulados (veículos/hora)

<b>Volume</b>	<b>Via Principal</b>	<b>Via Secundária</b>	<b>Volume Total</b>
Baixo	600	200	800
Alto	1800	600	2400

### 6.3.3. Variáveis que Caracterizam Cada Indivíduo

Nesta etapa, são definidos os genes que compõem o cromossomo, isto é, as variáveis utilizadas para a caracterização de cada indivíduo. Foram utilizadas 27 variáveis, conforme apresentado a seguir:

- 1) tipo do controlador;
- 2) modo de implementação do operador “E” (o “OU” não foi utilizado pois na base de regras só há “E”);
- 3) modo de implementação da função de implicação (somente para o controlador Mamdani);
- 4) modo de implementação da agregação (somente para o controlador Mamdani);
- 5) método de defuzzificação;
- 6) tipo de função de pertinência para o conjunto *fuzzy* “Pequena” da variável “Fila”;
- 7) variação dos limites do conjunto *fuzzy* “Pequena” da variável “Fila”;
- 8) tipo de função de pertinência para o conjunto *fuzzy* “Média” da variável “Fila”;
- 9) variação dos limites do conjunto *fuzzy* “Média” da variável “Fila”;
- 10) tipo de função de pertinência para o conjunto *fuzzy* “Longa” da variável “Fila”;
- 11) variação dos limites do conjunto *fuzzy* “Longa” da variável “Fila”;
- 12) tipo de função de pertinência para o conjunto *fuzzy* “Zero” da variável “Chegada”;
- 13) variação dos limites do conjunto *fuzzy* “Zero” da variável “Chegada”;
- 14) tipo de função de pertinência para o conjunto *fuzzy* “Pouca” da variável “Chegada”;
- 15) variação dos limites do conjunto *fuzzy* “Pouca” da variável “Chegada”;
- 16) tipo de função de pertinência para o conjunto *fuzzy* “Média” da variável “Chegada”;

- 17) variação dos limites do conjunto *fuzzy* “Média” da variável “Chegada”;
- 18) tipo de função de pertinência para o conjunto *fuzzy* “Muita” da variável “Chegada”;
- 19) variação dos limites do conjunto *fuzzy* “Muita” da variável “Chegada”;
- 20) tipo de função de pertinência para o conjunto *fuzzy* “Zero” da variável “Extensão”;
- 21) variação dos limites do conjunto *fuzzy* “Zero” da variável “Extensão”;
- 22) tipo de função de pertinência para o conjunto *fuzzy* “Curta” da variável “Extensão”;
- 23) variação dos limites do conjunto *fuzzy* “Curta” da variável “Extensão”;
- 24) tipo de função de pertinência para o conjunto *fuzzy* “Média” da variável “Extensão”;
- 25) variação dos limites do conjunto *fuzzy* “Média” da variável “Extensão”;
- 26) tipo de função de pertinência para o conjunto *fuzzy* “Longa” da variável “Extensão”;
- 27) variação dos limites do conjunto *fuzzy* “Longa” da variável “Extensão”.

As variáveis do tipo “variação dos limites do conjunto *fuzzy*” expressarão o aumento ou redução do tamanho dos conjuntos *fuzzy* apresentados na Tabela 6.2, variando os seus limites. Maiores detalhes sobre as mesmas serão fornecidos no item a seguir.

#### 6.3.4. Espaço de Busca para Cada Variável

O espaço de busca de cada variável representa as alternativas possíveis que cada variável pode assumir; pode-se dizer que é o domínio de cada variável. Os itens a seguir apresentam o espaço de busca de cada variável que foi proposto na concepção original do trabalho.

- 1) tipo do controlador;
  - Mamdani;
  - Sugeno;
- 2) modo de implementação do operador “E” (o “OU” não foi utilizado pois na base de regras só há “E”);

- Min;
  - Prod;
- 3) modo de implementação da função de implicação;
- Min;
  - Prod;
- 4) modo de implementação da agregação;
- Max;
  - Sum;
  - Probor;
- 5) método de defuzificação;
- a) Mamdani:
- Média dos Máximos (MOM);
  - Maior dos Máximos (LOM);
  - Método do Centro de Área (COA) ou Método do Centro de Gravidade (COG);
  - Mínimo do Máximo (SOM);
  - Bissetor da Área;
- b) Sugeno:
- Soma Ponderada (*Wtaver*);
- 6) Tipo de função de pertinência para cada um dos seguintes conjuntos *fuzzy*: “Pequena” da variável “Fila”, “Zero” da variável “Chegada” e “Zero” da variável “Extensão”;
- Triangular;
  - Trapezoidal;
  - Gaussiana;
  - $\pi$ ;

- Sino;
- Produto entre duas sigmoidais;
- Z;
- Sigmoidal;

7) tipo de função de pertinência para cada um dos seguintes conjuntos *fuzzy*: “Média” da variável “Fila”, “Pouca” da variável “Chegada”, “Média” da variável “Chegada”, “Curta” da variável “Extensão” e “Média” da variável “Extensão”;

- Triangular;
- Trapezoidal;
- Gaussiana;
- $\pi$ ;
- Sino;
- Produto entre duas sigmoidais;

8) tipo de função de pertinência para cada um dos seguintes conjuntos *fuzzy*: “Longa” da variável “Fila”, “Muita” da variável “Chegada”, “Longa” da variável “Extensão”;

- Triangular;
- Trapezoidal;
- Gaussiana;
- $\pi$ ;
- Sino;
- Produto entre duas sigmoidais;
- Sigmoidal;
- S;

9) variação dos limites de cada um dos conjuntos *fuzzy* de cada variável;

- -3;
- -2;
- -1;

- 0;
- +1;
- +2;
- +3;
- +4;

Os valores para a variação dos limites dos conjuntos referem-se à variação que será aplicada em cada um dos extremos de um dos conjuntos básicos apresentados na Tabela 6.2. Por exemplo, se o conjunto em questão for “Média” da variável Fila, e for gerado o número -3, os seus limites originais [5,15] passarão a ser [8,12], isto é, houve uma redução do tamanho do conjunto de -3 em cada extremidade. Caso as operações forneçam um número menor que zero ou maior que vinte para o limites de um conjunto, este será considerado como sendo zero ou vinte, respectivamente.

As Tabelas 6.6 e 6.7 resumem o número de possibilidades para cada variável para os controladores de Mamdani e Sugeno, respectivamente, e a Tabela 6.8 apresenta o tamanho do espaço de busca para a situação proposta acima.

Tabela 6.6 – Número de Possibilidades Para Cada Variável Para o Controlador de Mamdani (Concepção Original)

Variável		Número de Alternativas
Tipo do Controlador (Mamdani)		1
Modo de Implementação do Operador “E”		2
Modo de Implementação da Função de Implicação		2
Modo de Implementação da Agregação		3
Método de Defuzificação		5
Tipo de Função de Pertinência -Fila	Pequena	8
	Média	6
	Longa	8
Tipo de Função de Pertinência -Chegada	Zero	8
	Pouca	6
	Média	6
	Muita	8
Tipo de Função de Pertinência -Extensão	Zero	8
	Curta	6
	Média	6
	Longa	8
Variação dos Limites dos Conjuntos <i>Fuzzy</i> de <i>Fila</i>	Pequena	8
	Média	8
	Longa	8
Variação dos Limites dos Conjuntos <i>Fuzzy</i> de <i>Chegada</i>	Zero	8
	Pouca	8
	Média	8
	Muita	8
Variação dos Limites dos Conjuntos <i>Fuzzy</i> de <i>Extensão</i>	Zero	8
	Curta	8
	Média	8
	Longa	8

Tabela 6.7 – Número de Possibilidades Para Cada Variável Para o Controlador de Sugeno (Concepção Original)

Variável		Número de Alternativas
Tipo do Controlador (Sugeno)		1
Modo de Implementação do Operador “E”		2
Modo de Implementação da Função de Implicação		-
Modo de Implementação da Agregação		-
Método de Defuzificação		1
Tipo de Função de Pertinência -Fila	Pequena	8
	Média	6
	Longa	8
Tipo de Função de Pertinência -Chegada	Zero	8
	Pouca	6
	Média	6
	Muita	8
Tipo de Função de Pertinência -Extensão	Zero	8
	Curta	6
	Média	6
	Longa	8
Variação dos Limites dos Conjuntos <i>Fuzzy</i> de <i>Fila</i>	Pequena	8
	Média	8
	Longa	8
Variação dos Limites dos Conjuntos <i>Fuzzy</i> de <i>Chegada</i>	Zero	8
	Pouca	8
	Média	8
	Muita	8
Variação dos Limites dos Conjuntos <i>Fuzzy</i> de <i>Extensão</i>	Zero	8
	Curta	8
	Média	8
	Longa	8

Tabela 6.8 – Tamanho do Espaço de Busca Para a Concepção Original do Procedimento do Trabalho

Tamanho do Espaço de Busca	
Controlador de Mamdani	$1,0506 \times 10^{21}$
Controlador de Sugeno	$3,502 \times 10^{19}$
Total	$1,08562 \times 10^{21}$

Da Tabela 6.8, percebe-se que o espaço de busca total é de ordem bastante elevada. Conforme será descrito adiante, o presente estudo utilizará 100 gerações com 100 indivíduos cada, tal situação não contemplaria nem  $1 \times 10^{-15}\%$  do espaço de busca. Assim, no intuito de se reduzir o espaço de busca, foi necessário que se realizasse duas simplificações. Na primeira, reduziu-se o número de alternativas dos tipos de funções de pertinência e da variação dos limites dos conjuntos *fuzzy*. O número de alternativas das outras variáveis foi mantido. A seguir, as variáveis que tiveram suas alternativas reduzidas:

1) Tipo de função de pertinência para cada um dos seguintes conjuntos *fuzzy*: “Pequena” da variável “Fila”, “Zero” da variável “Chegada” e “Zero” da variável “Extensão”:

- Triangular;
- Trapezoidal;
- Z;

2) tipo de função de pertinência para cada um dos seguintes conjuntos *fuzzy*: “Média” da variável “Fila”, “Pouca” da variável “Chegada”, “Média” da variável “Chegada”, “Curta” da variável “Extensão” e “Média” da variável “Extensão”:

- Triangular;
- Trapezoidal;
- Gaussiana;

3) tipo de função de pertinência para cada um dos seguintes conjuntos *fuzzy*: “Longa” da variável “Fila”, “Muita” da variável “Chegada”, “Longa” da variável “Extensão”:

- Trapezoidal;
- Sigmoidal;
- S;

4) variação dos limites de cada um dos conjuntos *fuzzy* de cada variável:

- -2;
- 0;
- +2;

Os tipos de função de pertinência triangular e trapezoidal, muito utilizados em outros trabalhos, podem representar qualquer conjunto *fuzzy*. A única exceção é para a função de pertinência triangular, no caso dos conjuntos “Muita” e “Longa”. Estudos anteriores indicam que este tipo de função não é muito adequado para a representação destes dois conjuntos, por isso optou-se por substituí-la, nesses dois casos, pela função sigmoideal. Os demais tipos de função de pertinência foram escolhidos porque se entendeu que seus formatos gráficos eram apropriados para as situações propostas.

Em relação à redução do número de alternativas para a variável “variação dos limites de cada um dos conjuntos *fuzzy* de cada variável”, optou-se por escolher uma situação que reduzisse o intervalo do conjunto *fuzzy*, uma que o aumentasse e outra que mantivesse sua amplitude original. Escolheu-se o valor 2, para a redução e dilatação do conjunto, porque este era um valor intermediário dentre aqueles propostos originalmente.

As Tabelas 6.9 e 6.10 resumem o novo número de possibilidades para cada variável para os controladores de Mamdani e Sugeno, respectivamente, e a Tabela 6.11 apresenta o tamanho do espaço de busca para a nova concepção do trabalho.

Tabela 6.9 – Número de Possibilidades Para Cada Variável Para o Controlador de Mamdani (Segunda Concepção)

Variável		Número de Alternativas
Tipo do Controlador (Mamdani)		1
Modo de Implementação do Operador “E”		2
Modo de Implementação da Função de Implicação		2
Modo de Implementação da Agregação		3
Método de Defuzificação		5
Tipo de Função de Pertinência -Fila	Pequena	3
	Média	3
	Longa	3
Tipo de Função de Pertinência -Chegada	Zero	3
	Pouca	3
	Média	3
	Muita	3
Tipo de Função de Pertinência -Extensão	Zero	3
	Curta	3
	Média	3
	Longa	3
Variação dos Limites dos Conjuntos <i>Fuzzy</i> de <i>Fila</i>	Pequena	3
	Média	3
	Longa	3
Variação dos Limites dos Conjuntos <i>Fuzzy</i> de <i>Chegada</i>	Zero	3
	Pouca	3
	Média	3
	Muita	3
Variação dos Limites dos Conjuntos <i>Fuzzy</i> de <i>Extensão</i>	Zero	3
	Curta	3
	Média	3
	Longa	3

Tabela 6.10 – Número de Possibilidades Para Cada Variável Para o Controlador de Sugeno (Segunda Concepção)

Variável		Número de Alternativas
Tipo do Controlador (Sugeno)		1
Modo de Implementação do Operador “E”		2
Modo de Implementação da Função de Implicação		-
Modo de Implementação da Agregação		-
Método de Defuzificação		1
Tipo de Função de Pertinência -Fila	Pequena	3
	Média	3
	Longa	3
Tipo de Função de Pertinência -Chegada	Zero	3
	Pouca	3
	Média	3
	Muita	3
Tipo de Função de Pertinência -Extensão	Zero	3
	Curta	3
	Média	3
	Longa	3
Partição do Universo de Discurso - Variação dos Limites dos Conjuntos <i>Fuzzy</i> de <i>Fila</i>	Pequena	3
	Média	3
	Longa	3
Partição do Universo de Discurso - Variação dos Limites dos Conjuntos <i>Fuzzy</i> de <i>Chegada</i>	Zero	3
	Pouca	3
	Média	3
	Muita	3
Partição do Universo de Discurso - Variação dos Limites dos Conjuntos <i>Fuzzy</i> de <i>Extensão</i>	Zero	3
	Curta	3
	Média	3
	Longa	3

Tabela 6.11 – Tamanho do Espaço de Busca Para a Segunda Concepção do Procedimento do Trabalho

Tamanho do Espaço de Busca	
Controlador de Mamdani	$1,88286 \times 10^{12}$
Controlador de Sugeno	62.762.119.218
Total	$1,94563 \times 10^{12}$

Da Tabela 6.11, nota-se que, embora o espaço de busca ainda seja grande, com as simplificações realizadas, houve uma redução significativa em sua ordem de grandeza.

A segunda simplificação adotada no trabalho foi sua divisão em 2 etapas. A Etapa 1 é iniciada no Módulo Inicial – Definições Básicas e tem seu fim no Módulo Final – Resultados, da Figura 6.1. Nesta etapa, as variáveis do tipo variação dos limites dos conjuntos *fuzzy* são todas definidas como zero. Isto significa que a partição dos conjuntos é exatamente aquela apresentada na Tabela 6.2. Assim, o AG só poderá alterar as outras variáveis (ver Figura 6.4). Ao final desta etapa, é obtido um controlador que servirá de base para a segunda etapa.

A Etapa 2 tem seu início no resultado final da Etapa 1 e é terminada no Módulo Final – Resultados, conforme ilustra a Figura 6.1. Nela, as variáveis do tipo variação dos conjuntos *fuzzy* são as únicas a serem trabalhadas pelo AG. Todas as outras variáveis são fixas, com valores iguais aos encontrados para o melhor controlador resultante da Etapa 1 (ver Figura 6.4). A divisão do processo em duas etapas segmenta o espaço de busca em dois, conforme apresenta a Tabela 6.12.

Tabela 6.12 – Tamanho do Espaço de Busca para as Etapas 1 e 2

<b>Tamanho do Espaço de Busca para Cada Etapa</b>		
<b>Etapa 1</b>	Controlador de Mamdani	10.628.820
	Controlador de Sugeno	354.294
	<b>Total Etapa 1</b>	<b>10.983.114</b>
<b>Etapa 2</b>	Melhor Controlador Etapa 1	177.147
	<b>Total Etapa 2</b>	<b>177.147</b>

### 6.3.5. Codificação

Na idéia original deste estudo, pretendia-se adotar a codificação binária pelo fato de ser a mais utilizada na literatura. Contudo, deparou-se com uma dificuldade no processo. Esta forma de representação gerava muitos cromossomos que não faziam sentido para o estudo. Por exemplo, a variável método de defuzificação para o controlador de Mamdani requer 5 formas de representação; no entanto, na codificação binária essa representação só seria possível com 3 bits, gerando um excesso de 3 formas de representação, conforme mostrado a seguir:

000 = Média dos Máximos (MOM);

- 001 = Maior dos Máximos (LOM);
- 010 = Método do Centro de Gravidade (COG);
- 011 = Mínimo do Máximo (SOM);
- 100 = Bissetor da Área;
- 101 = Sem Significado Para o Estudo;
- 110 = Sem Significado Para o Estudo;
- 111 = Sem Significado Para o Estudo.

As representações 101, 110 e 111 não correspondem a nenhum método de defuzificação. Numa primeira tentativa de solucionar essa dificuldade, criou-se uma função no AG que ao detectar um cromossomo com essas características (problemático) o substituísse por outro qualquer que não tivesse o mesmo tipo de problemas. Embora tenha funcionado, havia desperdício de tempo nessas substituições, fator indesejado devido ao grande número de simulações que seriam realizadas no estudo.

Pelo exposto, e respaldado na afirmação de Mitchell (1998) de que muitas comparações empíricas entre codificação binária e outras formas de codificação têm apresentado melhores resultados para estas últimas, optou-se pela codificação decimal. Nela, os excessos foram eliminados, conforme demonstra o exemplo a seguir para a representação dos 5 métodos de defuzificação:

- 0 = Média dos Máximos (MOM);
- 1 = Maior dos Máximos (LOM);
- 2 = Método do Centro de Gravidade (COG);
- 3 = Mínimo do Máximo (SOM);
- 4 = Bissetor da Área.

Não existe a possibilidade de que seja gerado o número 5 para a variável método de defuzificação nem um número que não faça sentido para as outras variáveis. Tal restrição foi imposta no AG na função que gera os indivíduos da população inicial e na função que realiza a mutação. Assim, garante-se que todo indivíduo gerado ou que tenha sofrido mutação não será problemático. Não foi necessário colocar essa restrição na função que realiza a recombinação (*crossover*) porque este só ocorre entre dois indivíduos não problemáticos, isto é, que não possuam genes indesejados.

#### 6.3.6. Definição do Cromossomo

Cada cromossomo é formado por 27 genes (pois existem 27 variáveis). Como foi adotada a representação decimal, e todas as variáveis possuem menos que 10 formas de representação, cada gene é formado por um único *bit*. Assim, a estrutura *string* é formada por 27 *bits*. A disposição dos genes (*bits*) no cromossomo é ilustrada na Figura 6.4.

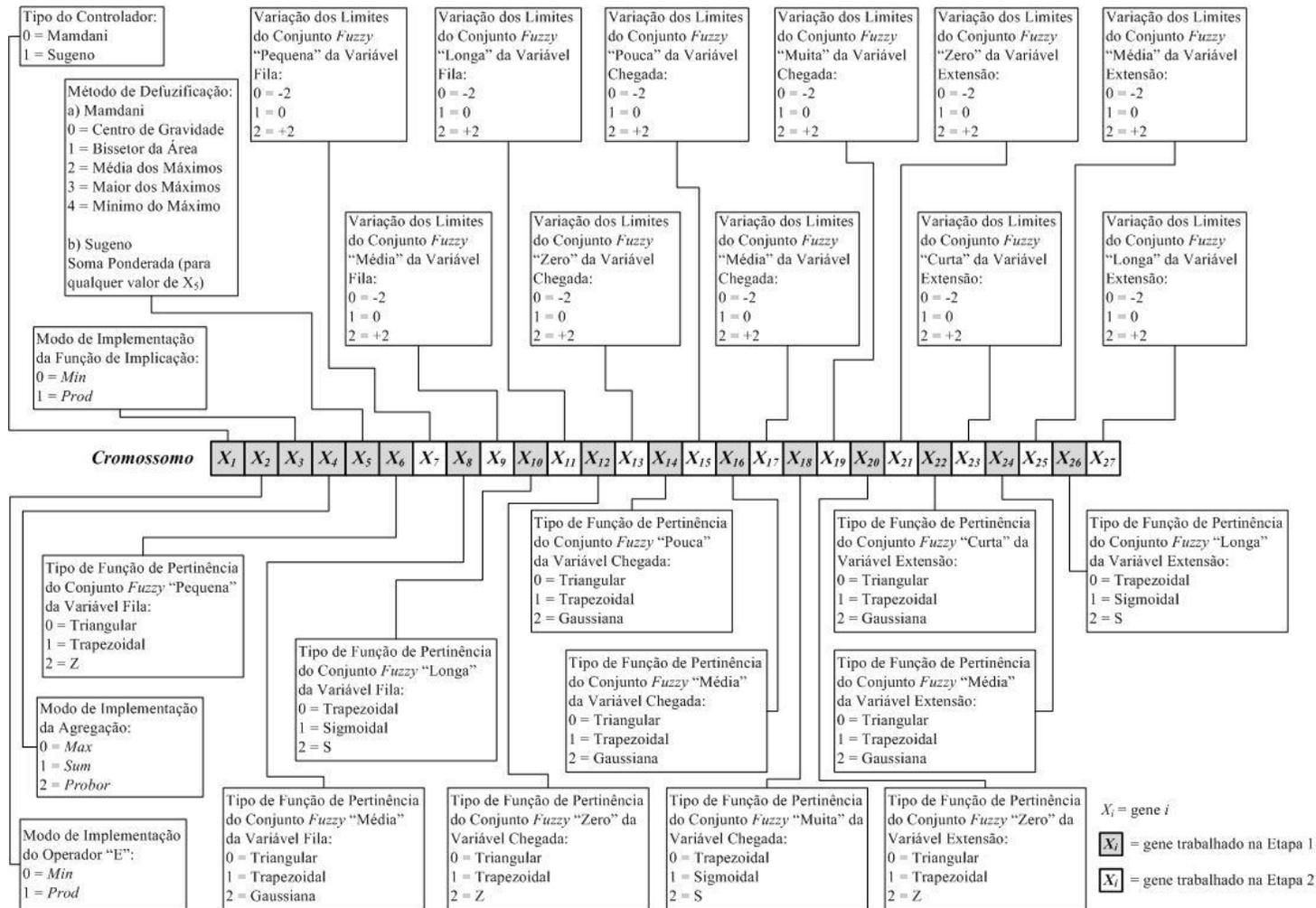


Figura 6.4 – Genes do cromossomo utilizado no estudo, com suas alternativas possíveis.

### 6.3.7. Definição da População Inicial

Cada cromossomo da população inicial foi gerado com o auxílio da função *random()* do Matlab®. A forma de implementação da função é a seguinte: *random('unif',limite\_inferior,limite\_superior)*. O primeiro argumento, 'unif', significa que a distribuição é uniforme. O segundo e terceiro argumentos correspondem aos limites inferior e superior da distribuição, respectivamente. O algoritmo percorre cada *bit* de um cromossomo e gera um número aleatório para o mesmo. O produto dessa etapa é a população inicial, com o número de cromossomos definido inicialmente pelo usuário.

Para se ter um maior controle na geração da população inicial, incluiu-se no algoritmo o recurso de sementes. Nele, o usuário escolhe um número qualquer (a semente) para gerar uma população inicial. Para cada semente, uma população inicial diferente é gerada. A utilização deste recurso permite que, caso se deseje realizar novamente uma simulação com uma determinada população inicial, basta inserir como dado de entrada o valor de sua semente.

## 6.4. Módulo Controlador Fuzzy

### 6.4.1. Matlab

Definidos os cromossomos da geração "i" e a base de regras, já é possível utilizar a FLT-M do Matlab®. Um cromossomo carrega em si todos os dados de entrada requeridos pela FLT-M, com exceção da base de regras. Assim, para cada cromossomo da população, o algoritmo define um controlador específico na FLT-M. Destaca-se, portanto, que a seta que liga a caixa "Geração 'i'" à caixa "Matlab", indica que este processo será realizado individualmente, com cada cromossomo da geração. A Figura 5 apresenta a interface que é apresentada na FLT-M ao usuário, caso este fosse inserir os dados de entrada manualmente, para o controlador de Mamdani.

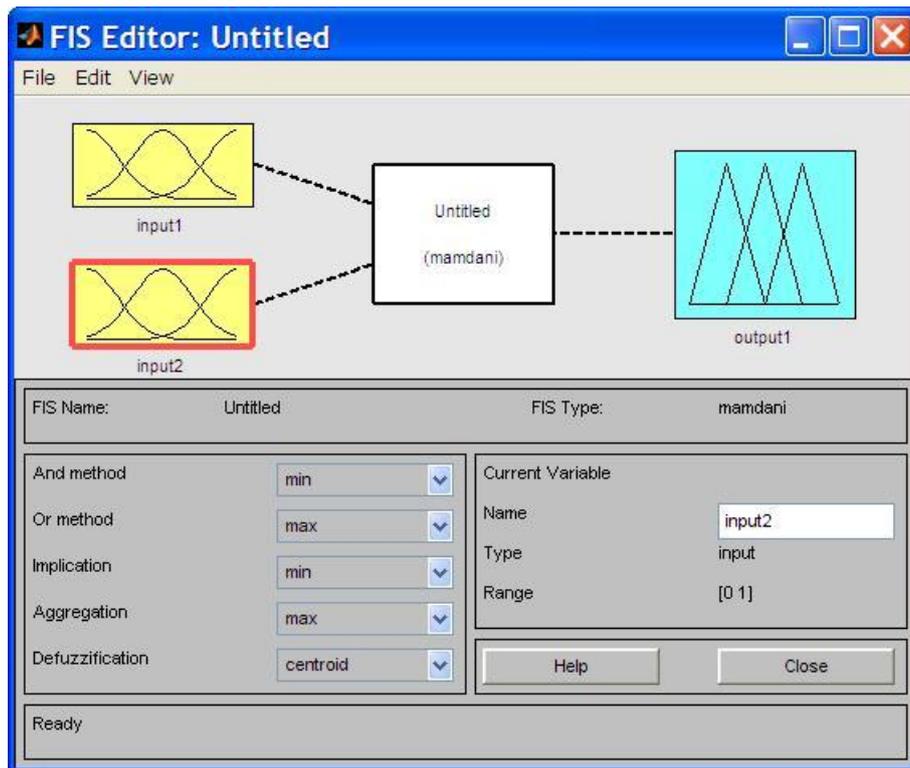


Figura 6.5 – Interface apresentada ao usuário na FLT-M para o controlador de Mamdani

Conforme ilustrado, devem ser inseridas as variáveis: modo de implementação do operador “E”, modo de implementação do operador “OU”, modo de implementação da função de implicação, modo implementação da agregação e método de defuzzificação, além das variáveis *fuzzy* de entrada e saída. Embora o presente trabalho não utilize o operador “OU”, conforme já exposto anteriormente, a FLT-M exige seu campo seja preenchido. Sendo assim, mesmo não interferindo no resultado, o algoritmo sempre preenche este campo com “*max*”.

Para um controlador de Sugeno, a interface apresentada é a mesma, entretanto, os campos que definem o modo de implementação da função de implicação e o modo implementação da agregação ficam desativados, e o campo do método de defuzzificação só pode ser preenchido de duas maneiras, *wtaver* e *wtsum*, sendo média ponderada e soma ponderada, respectivamente. Este trabalho utiliza somente o primeiro, conforme já descrito no item 6.3.4.

Para cada variável *fuzzy* de entrada e saída apresentadas na Figura 6.5 é necessário que sejam definidos o tipo de função de pertinência e os parâmetros da função de

pertinência. A interface apresentada a um usuário que queira inserir tais dados manualmente é ilustrada na Figura 6.6.

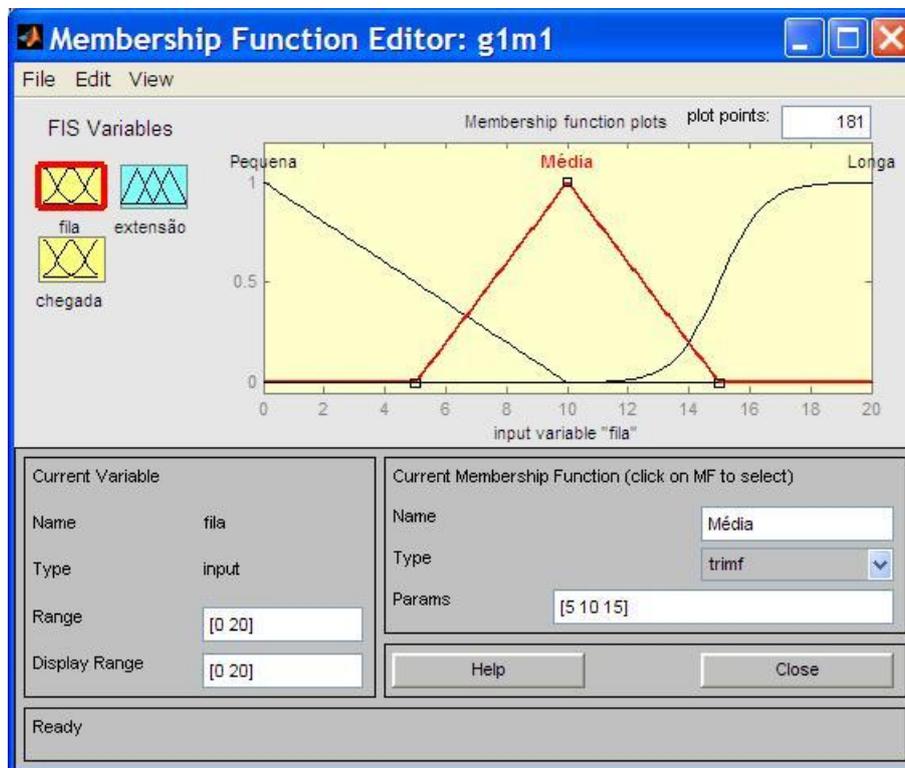


Figura 6.6 – Interface apresentada ao usuário da FLT-M para a definição das funções de pertinência dos conjuntos *fuzzy*

Cada cromossomo possui as informações que definem o tipo de função de pertinência e os limites dos conjuntos *fuzzy*. Os parâmetros de uma função de pertinência, requisitados pela FLT-M no campo “*Params*”, foram definidos no algoritmo como sendo uma função dos limites dos conjuntos *fuzzy* apresentados na Tabela 6.2.

Por exemplo, na Figura 6.6, o conjunto *fuzzy* “Média” da variável Fila está sendo inserido no controlador com uma função de pertinência triangular (“*trimf*” no campo “*Type*”). A FLT-M exige três parâmetros de entrada para a representação de uma função triangular: “a”, ponto inicial com pertinência zero, “b”, ponto com pertinência 1 e “c”, ponto final com pertinência zero. Um cromossomo, contudo, não traz os três parâmetros em si, somente os limites dos conjuntos *fuzzy*, definidos por especialistas, que para este caso é [5 15]. Assim, definiu-se que para uma função triangular, “a” seria o limite inferior do conjunto, “c” o limite superior e “b” o ponto médio entre eles. Por isso, na Figura 6.6, o campo “*Params*” está preenchido com [5 10 15].

A seguir, são apresentadas as funções e simplificações adotadas neste estudo para transformar os conjuntos *fuzzy* em parâmetros de cada tipo de função de pertinência utilizada. Para maiores detalhes a respeito das funções de pertinência utilizadas na FLT-M, ver Vaz (2006).

#### a) Função de Pertinência Triangular

Conforme mencionado, a função de pertinência triangular da FLT-M depende de três parâmetros: [a b c]. A curva triangular é dada pela Equação 6.1:

$$f(x : a, b, c) = \begin{cases} 0, & x \leq a \\ \frac{x - a}{b - a}, & a < x \leq b \\ \frac{c - x}{c - b}, & b < x \leq c \\ 0, & x > c \end{cases} \quad (\text{Eq. 6.1})$$

Os três parâmetros, “a”, “b” e “c”, correspondem aos valores do eixo x dos vértices do triângulo, sendo que “a” e “c” estão relacionados à base do triângulo (pertinência igual a zero) e “b” ao ponto de pertinência igual a 1. Assim, para os conjuntos “Média”, “Curta” e “Pouca”, definiu-se que “a” é o valor do limite inferior do conjunto, “c” é o valor do limite superior, e “b” é o valor médio entre eles. A atribuição de “a”, “b”, e “c” para os outros conjuntos é descrita a seguir.

Na definição dos cromossomos existia a possibilidade de que o intervalo em estudo, [0 20], ficasse com alguns pontos descobertos. Um exemplo, caso um cromossomo tivesse genes que atribuíssem uma variação de -2 aos limites do conjunto *fuzzy* “Zero” da variável Chegada, seus limites passariam de [0 5] para [2 3]. Assim, o intervalo [0 2) ficaria descoberto. Para evitar situações como essa, definiu-se que para os conjuntos *fuzzy* “Zero” e “Pequena”, “a” e “b” serão sempre zero, independente dos genes do cromossomo. Já para os conjuntos *fuzzy* “Muita” e “Longa”, “b” e “c” serão sempre 20. A Figura 6.7 ilustra exemplos de conjuntos *fuzzy* “Zero”, “Pouca”, “Média” e “Muita”, da variável Chegada, definidos por funções de pertinência triangulares.

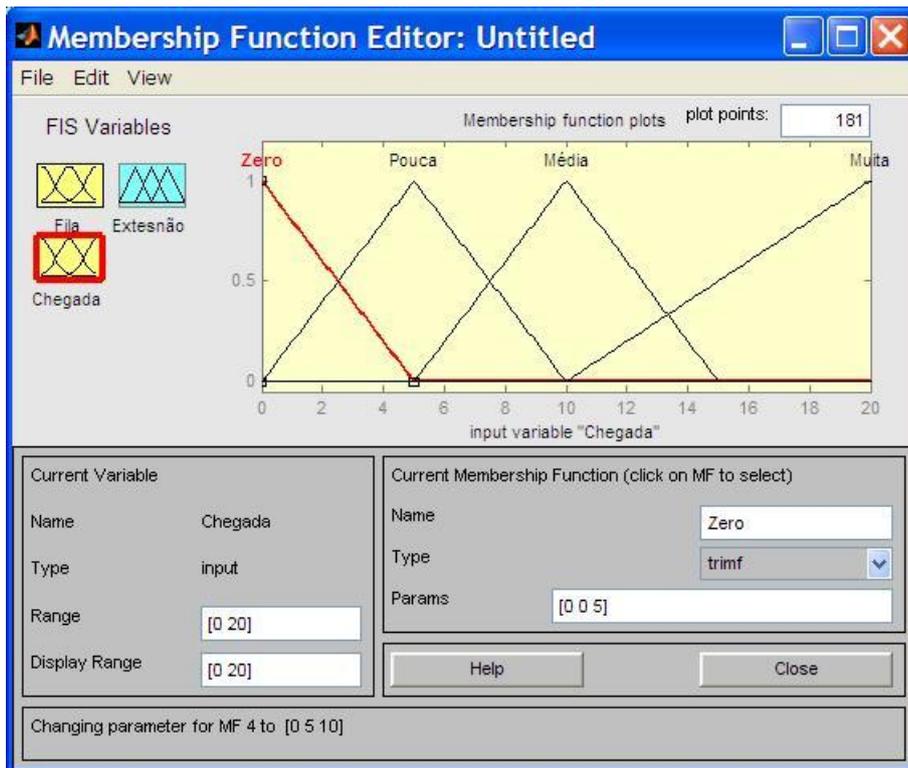


Figura 6.7 – Exemplos de Conjuntos *Fuzzy* “Zero”, “Pouca”, “Média” e “Muita”, da Variável Chegada, Definidos Por Funções de Pertinência Triangulares.

#### b) Função de Pertinência Trapezoidal

A função de pertinência trapezoidal da FLT-M depende de quatro parâmetros: [a b c d].

A curva trapezoidal é dada pela Equação 6.2:

$$f(x : a, b, c, d) = \begin{cases} 0, & x \leq a \\ \frac{x - a}{b - a}, & a < x \leq b \\ 1, & b < x \leq c \\ 0, & x > d \\ \frac{d - x}{d - c}, & c < x \leq d \end{cases} \quad (\text{Eq. 6.2})$$

Os parâmetros “a” e “d” correspondem aos valores de x dos vértices da base do trapézio, isto é, daqueles cuja pertinência é igual a zero. Já os parâmetros “b” e “c” correspondem a valores de x cuja pertinência é igual a 1. No algoritmo, os quatro parâmetros são calculados da seguinte maneira:

- Conjuntos *fuzzy* “Pequena”, “Zero”

$$\begin{aligned}
 a &= 0; \\
 b &= 0; \\
 c &= \frac{\text{limsup}}{2} && \text{(Eq. 6.3)} \\
 d &= \text{lim sup}
 \end{aligned}$$

Em que “lim sup” é o limite superior do conjunto *fuzzy*.

- Conjuntos *fuzzy* “Média”, “Curta” e “Pouca”

$$\begin{aligned}
 a &= \text{lim inf} \\
 d &= \text{lim sup} \\
 b &= a + \left( \frac{d - a}{3} \right) && \text{(Eq. 6.4)}
 \end{aligned}$$

$$c = a + 2 \cdot \left( \frac{d - a}{3} \right) \quad \text{(Eq. 6.5)}$$

Em que “lim inf” é o limite inferior do conjunto *fuzzy* e “lim sup” é o limite superior do conjunto *fuzzy*.

- Conjuntos *fuzzy* “Longa”, “Muita”

$$\begin{aligned}
 a &= \text{lim inf} \\
 b &= \frac{\text{liminf} + 20}{2} && \text{(Eq. 6.6)}
 \end{aligned}$$

$$c = 20$$

$$d = 20$$

Em que “lim inf” é o limite inferior do conjunto *fuzzy*.

A Figura 6.8 apresenta exemplos de conjuntos *fuzzy* “Zero”, “Pouca”, “Média” e “Muita”, da variável Chegada, definidos por funções de pertinência trapezoidais.

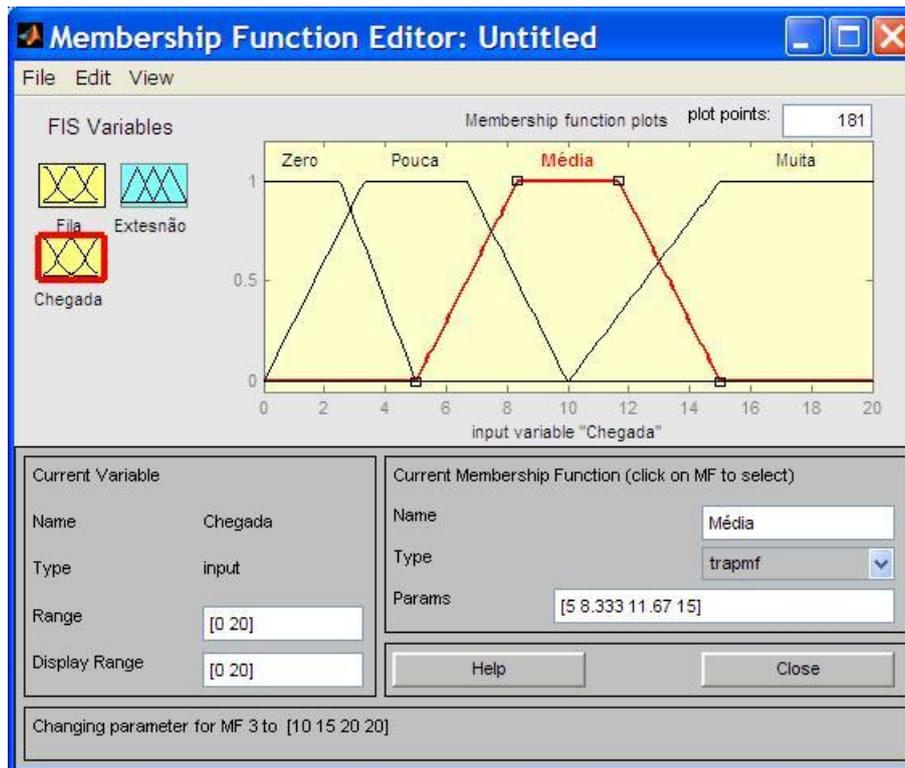


Figura 6.8 – Exemplos de Conjuntos *Fuzzy* “Zero”, “Pouca”, “Média” e “Muita”, da Variável Chegada, Definidos Por Funções de Pertinência Trapezoidais.

### c) Função de Pertinência Z

Por ter a forma parecida com a letra z, esta função recebe este nome. A FLT-M requer dois parâmetros de entrada para a definição deste tipo de função de pertinência, [a b]. O primeiro, “a”, é o valor de x para o qual todo número menor ou igual a ele tenha pertinência igual a 1. Já “b” é o valor de x para o qual todo número maior ou igual a ele tenha pertinência zero. Devido ao seu formato, esta função de pertinência só foi utilizada para os conjuntos *fuzzy* “Zero” e “Pequena”. A curva Z é dada pela Equação 6.4:

$$f(x) = \begin{cases} 1, & x \leq a \\ 1 - 2 \cdot \left( \frac{x-a}{b-a} \right)^2, & a \leq x \leq \frac{a+b}{2} \\ \frac{b-x}{b-a}, & \frac{a+b}{2} \leq x \leq b \\ 0, & x > b \end{cases} \quad (\text{Eq. 6.7})$$

No algoritmo, “a” e “b” são interpretados como sendo, respectivamente, os limites inferior e superior do conjunto definido no cromossomo. A Figura 6.9 apresenta o

exemplo de um conjunto *fuzzy* “Pequena”, da variável Fila, definido por uma função de pertinência Z.

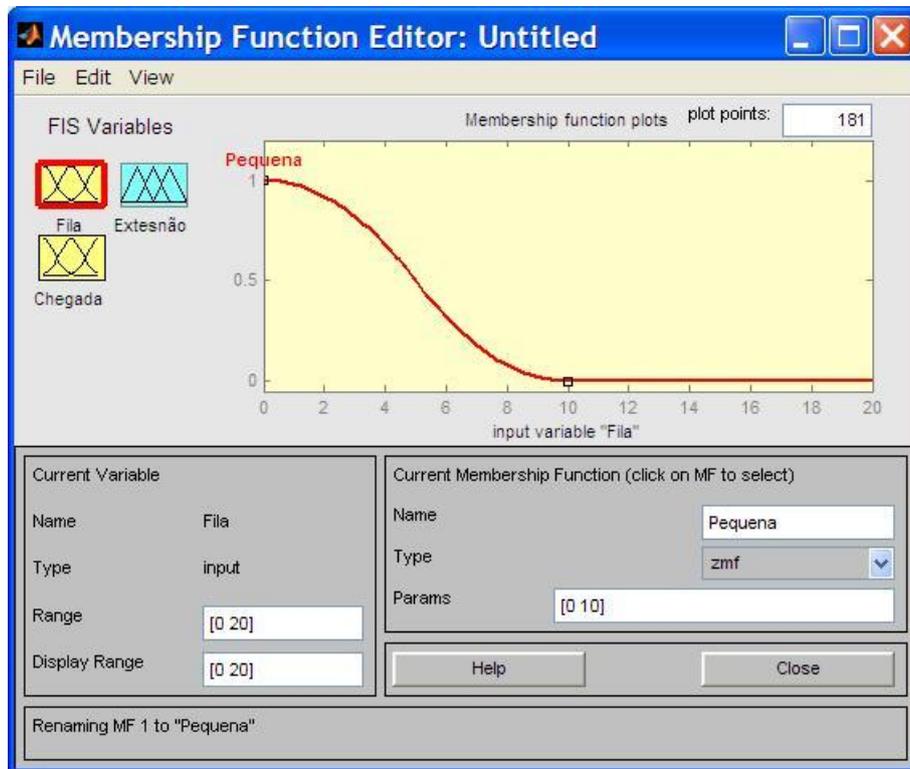


Figura 6.9 – Exemplo de Conjunto *Fuzzy* “Pequena”, da variável Fila, Definido Por uma Função de Pertinência Z.

#### d) Função de Pertinência S

Este nome é dado a esta função devido à semelhança existente entre sua forma gráfica e a letra S. Assim como na função de pertinência Z, a FLT-M requer dois parâmetros de entrada para a utilização da função de pertinência S: [a b]. A diferença, contudo, é que neste caso “a” é o valor de x para o qual todo número maior ou igual a ele tenha pertinência zero, e “b” é o valor de x para o qual todo número menor ou igual a ele tenha pertinência igual a 1. Este tipo de função de pertinência só foi utilizado para os conjuntos *fuzzy* “Muita” e “Longa”, devido à forma gráfica de sua curva. A equação da curva S é:

$$f(x) = \begin{cases} 0, & x < a \\ 2x^2, & a \leq x \leq \frac{a+b}{2} \\ 1 - 2(1-x)^2, & \frac{a+b}{2} \leq x \leq b \\ 1, & x \geq b \end{cases} \quad (\text{Eq. 6.8})$$

Se, ao ler os genes de um cromossomo, o algoritmo identificar que um conjunto *fuzzy* deve ser delimitado pela função de pertinência S, ele atribuirá a “a” e “b”, respectivamente, os valores dos limites inferior e superior daquele conjunto *fuzzy*. Vale lembrar que esses limites também estão representados em um gene do cromossomo. A Figura 6.10 apresenta o exemplo de um conjunto *fuzzy* “Longa”, da variável Extensão, definido por uma função de pertinência S.

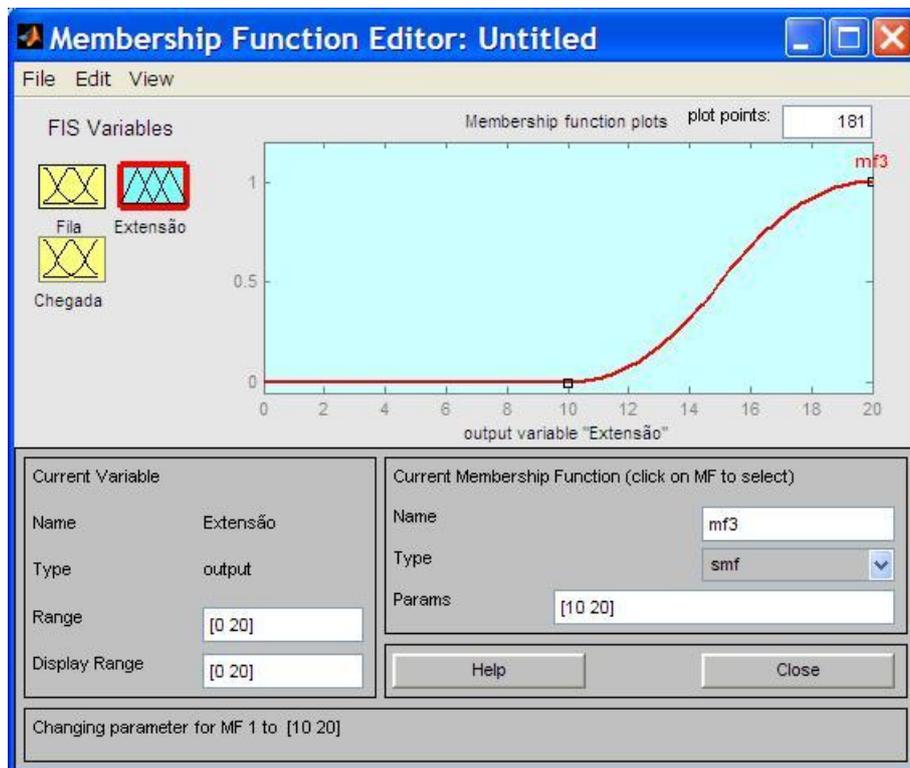


Figura 6.10 – Exemplo de Conjunto *Fuzzy* “Longa”, da variável Extensão, Definido Por uma Função de Pertinência S.

#### e) Função de Pertinência Gaussiana

A FLT-M requer dois parâmetros de entrada para a função de pertinência Gaussiana, o desvio padrão “ $\sigma$ ” e a média “c”. A curva gaussiana é dada pela Equação 6.9:

$$f(x, \sigma, c) = e^{-\frac{(x-c)^2}{2\sigma^2}} \quad (\text{Eq. 6.9})$$

O valor de “c” será a média entre os limites do conjunto *fuzzy*, conforme apresentado na Equação 6.10.

$$c = \frac{\text{liminf} + \text{limsup}}{2} \quad (\text{Eq. 6.10})$$

“Lim inf” é o limite inferior do conjunto *fuzzy* e “lim sup” é o limite superior do conjunto *fuzzy*. Vaz (2006) deduziu que “ $\sigma$ ” pode ser calculado da seguinte maneira:

$$\sigma = \sqrt{\frac{(A - c)^2}{1,38}} \quad (\text{Eq. 6.11})$$

em que “A” é o valor de x do ponto que possui pertinência igual a 0,5.

O cromossomo não traz em si a informação de qual ponto possui pertinência igual a 0,5, sendo assim, fez-se a seguinte consideração no algoritmo para a obtenção de “A”:

$$A = \frac{\text{liminf} + c}{2} \quad (\text{Eq. 6.12})$$

A função de pertinência gaussiana poderá ser utilizada para representar os conjuntos “Média”, “Curta” e “Pouca”. A Figura 6.11 traz como exemplo um conjunto *fuzzy* “Média”, da variável Fila, definido por uma função de pertinência Gaussiana. O intervalo do conjunto do exemplo em questão é [5 15], assim, “c” é 10, “A” é 7,5 (ou 12,5, ambos os valores resultam no mesmo resultado, pois a fórmula de “ $\sigma$ ” eleva a diferença ao quadrado) e “ $\sigma$ ” é 2,128 (a FLT-M trabalha com três casas decimais no campo “*Params*”).

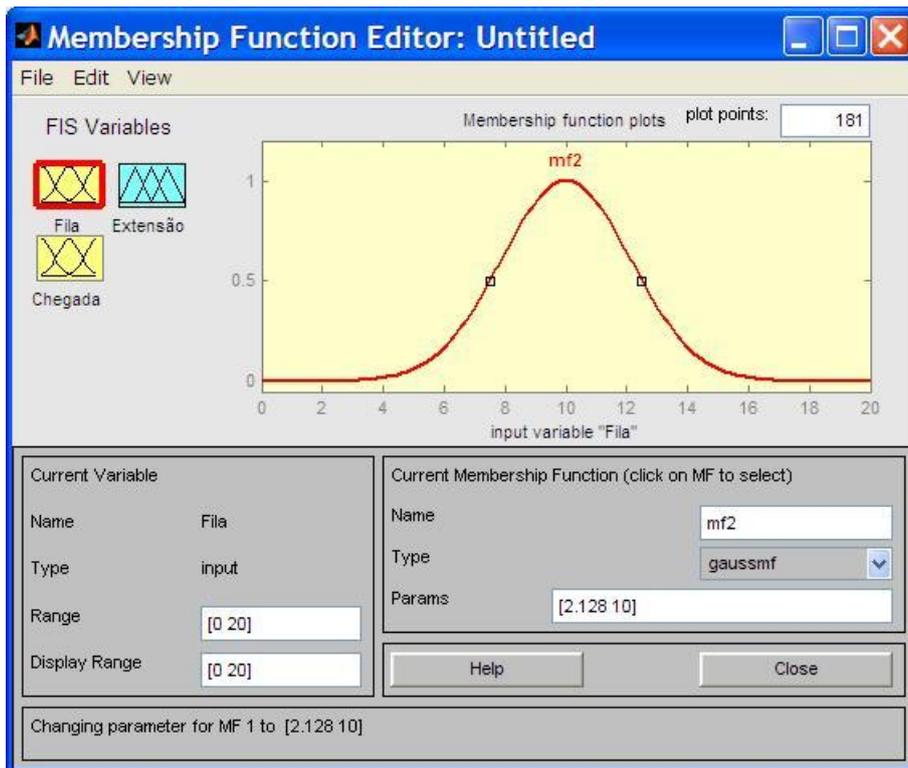


Figura 6.11 – Exemplo um conjunto *fuzzy* “Média”, da variável Fila, definido por uma função de pertinência Gaussiana

#### f) Função de Pertinência Sigmoidal

A forma gráfica desta função se aplicaria tanto aos conjuntos “Zero” e “Pequena” quanto aos conjuntos “Muita” e “Longa”. No entanto, como os dois primeiros já possuíam três tipos de funções de pertinência para sua representação (triangular, trapezoidal e Z), a função de pertinência sigmoidal é utilizada somente para a representação dos conjuntos “Muita” e “Longa”. A FLT-M requer dois parâmetros de entrada para este tipo de função: [a c]. Vaz (2006) deduziu que a função de pertinência sigmoidal é dada por:

$$f(x : a, c) = \frac{1}{1 + e^{-a(x-c)}} \quad (\text{Eq. 6.13})$$

O parâmetro “c” corresponde ao valor de x do ponto que possui pertinência igual a 0,5. Como o cromossomo não traz essa informação em si, adotou-se que “c” é o valor médio entre os limites inferior e superior do conjunto *fuzzy*.

$$c = \frac{\text{liminf} + \text{limsup}}{2} \quad (\text{Eq. 6.14})$$

Novamente, “lim inf” é o limite inferior do conjunto *fuzzy* e “lim sup” é o limite superior do conjunto *fuzzy*. Conhecido o valor de “c”, para se descobrir “a” basta que seja substituído o valor de um ponto (x, f(x)) qualquer na Equação 6.13. Por exemplo, escolhendo-se substituir o ponto que possua pertinência igual a zero, ou seja, o par ordenado (limite inferior do conjunto, 0), chega-se a:

$$0 = \frac{1}{1 + e^{-a(\text{liminf}-c)}} \quad (6.15)$$

A Equação 6.15, contudo, só é satisfeita quando o termo  $1 + e^{-a(\text{liminf}-c)}$  for infinito, fato que impossibilita que se descubra o valor de “a”. Assim, foi necessário que se fizesse uma simplificação. Considerou-se que a pertinência do limite inferior não seria zero, mas sim um número muito próximo dele, 0,001 (a ordem de grandeza deste número é coerente com o número de casas decimais aceito no campo “Params” da FLT-M). Assim, tem-se que:

$$0,001 = \frac{1}{1 + e^{-a(\text{liminf}-c)}} \rightarrow \frac{1}{1000} = \frac{1}{1 + e^{-a(\text{liminf}-c)}} \rightarrow 1000 = 1 + e^{-a(\text{liminf}-c)} \rightarrow$$

$$e^{-a(\text{liminf}-c)} = 999 \rightarrow -a(\text{liminf}-c) = \ln(999) \rightarrow -a = \frac{\ln(999)}{\text{liminf}-c}$$

E, portanto:

$$a = \frac{-\ln(999)}{\text{liminf} - \frac{\text{liminf} + \text{limsup}}{2}} \quad (\text{Eq. 6.16})$$

A Figura 6.12 traz como exemplo um conjunto *fuzzy* “Muita”, com intervalo [10 20], da variável Chegada, definido por uma função de pertinência sigmoideal. Neste caso, aplicando-se as Equações 6.14 e 6.16, descobre-se que o parâmetro “c” é 15, isto é, o valor médio entre os limites do conjunto [10 20], e “a” é igual a 1,381.

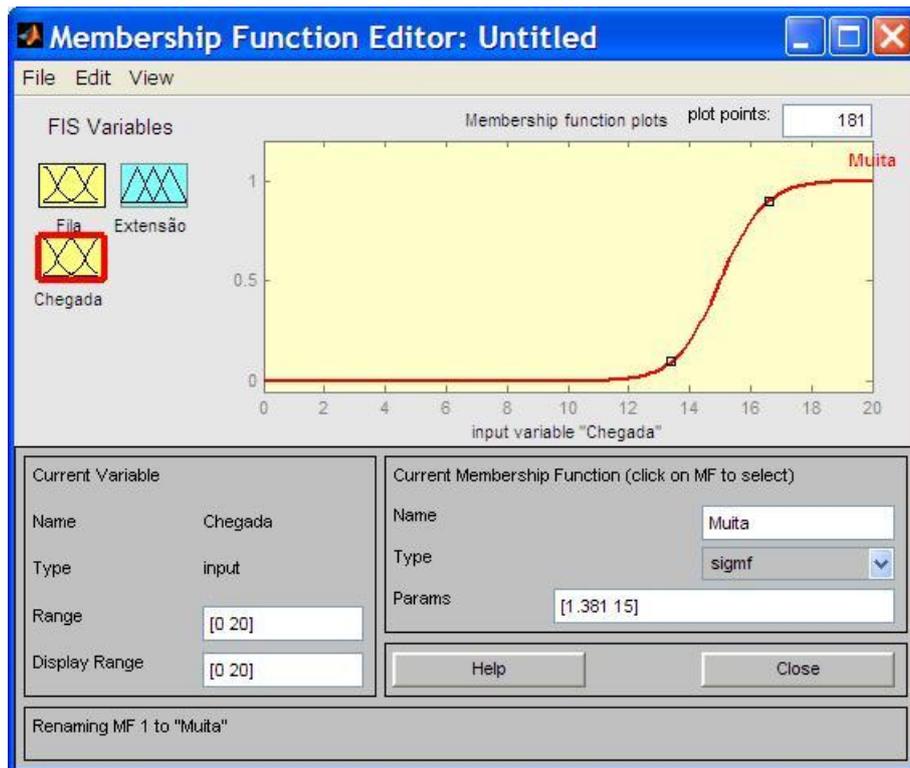


Figura 6.12 – Exemplo de Conjunto *Fuzzy* “Muita”, da Variável Chegada, Definido Por uma Função de Pertinência Sigmoidal

É válido lembrar que neste estudo a base de regras é fixa (Tabela 6.3) e que, por esse motivo, um cromossomo não carrega esta informação consigo. No código do algoritmo esta base de regras já foi definida, e cada vez que o algoritmo acessa a FLT-M, ele insere esses dados na mesma. Caso um usuário desejasse inserir uma base de regras manualmente, a FLT-M apresentaria a seguinte interface a ele (Figura 6.13).

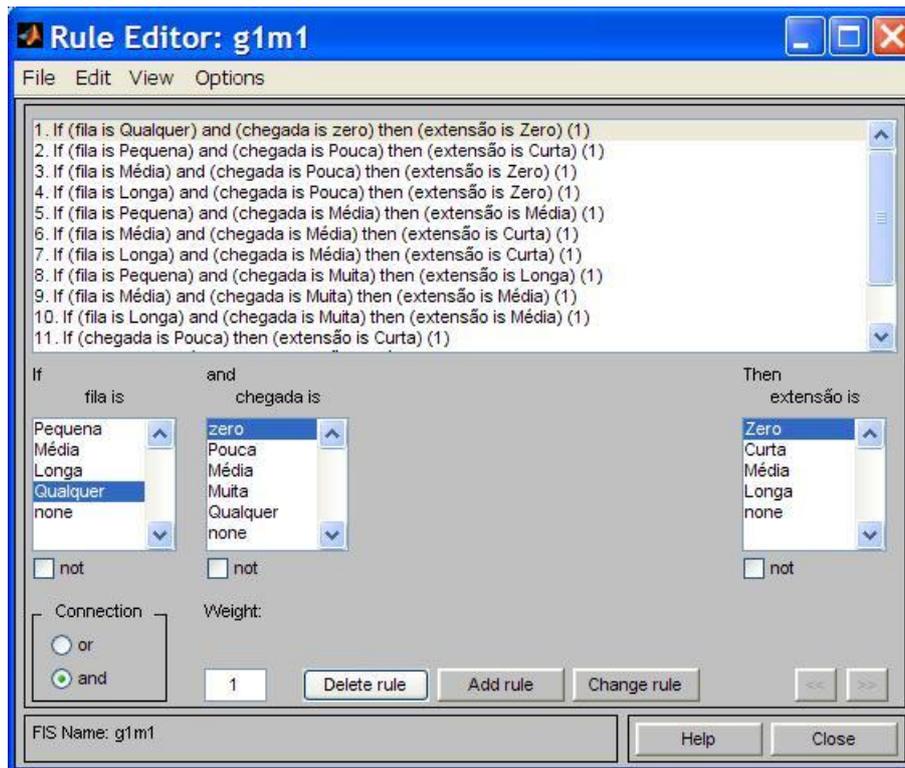


Figura 6.13 – Interface apresentada ao usuário na FLT-M para Inserir a Base de Regras

Após todo esse processo, são fornecidos dois produtos: a tabela de extensões e sua respectiva superfície de controle.

#### 6.4.2. Tabela de Extensões

Para cada cromossomo, a FLT-M fornecerá uma tabela de extensões. Esta nada mais é do que uma matriz 21x21 em que as linhas representam a variável Fila e as colunas representam a variável Chegada. Para cada situação momentânea do tráfego, representada por um par ordenado, a tabela apresenta um valor de extensão a ser fornecido pelo controlador semafórico (expresso em décimos de segundo). O algoritmo salva cada tabela de extensões como um arquivo de texto. A Figura 6.14 ilustra um exemplo de uma tabela de extensões.

19	31	39	46	50	53	57	65	76	87	100	100	102	110	140	166	172	173	174	174
19	31	38	45	50	53	57	65	76	87	100	100	102	110	140	166	172	173	174	174
19	30	38	45	50	53	57	65	76	87	100	100	102	110	140	166	172	173	174	174
19	30	38	45	50	53	57	65	76	87	100	100	102	110	140	166	172	173	174	174
19	30	37	45	49	53	57	65	76	87	100	100	102	110	140	166	172	173	174	174
19	29	37	44	49	53	57	65	76	87	100	100	102	110	140	166	172	173	174	174
19	28	34	40	44	47	51	58	69	79	91	91	93	101	129	153	159	160	161	161
19	27	32	37	40	43	47	53	63	73	84	85	86	93	120	143	149	150	150	150
19	26	30	34	37	39	43	49	58	68	79	79	80	87	113	135	141	142	142	142
19	26	30	33	35	38	41	47	56	66	76	76	78	85	110	132	137	138	139	139
19	25	29	32	35	37	40	46	55	65	75	75	76	84	108	130	135	137	137	137
19	25	29	32	34	37	40	46	55	64	74	74	75	82	106	128	134	135	135	135
19	25	29	32	34	36	40	46	54	64	73	73	73	78	100	123	130	132	133	133
19	25	29	32	35	37	41	46	55	64	73	72	72	74	92	117	127	129	130	130
19	24	28	32	35	37	41	46	55	63	71	70	68	67	82	107	120	123	124	125
19	24	28	32	35	37	40	45	53	61	67	66	63	60	72	96	110	115	116	117
19	23	27	30	32	34	37	42	49	56	62	61	58	55	65	87	102	106	107	108
19	23	26	29	31	33	36	40	47	54	59	58	55	52	61	83	96	101	102	103
19	23	26	28	30	32	35	39	46	52	58	56	53	50	59	80	93	98	99	100
19	23	25	28	30	31	34	39	45	52	57	55	52	49	57	78	92	96	98	98
19	23	25	28	30	31	34	39	45	51	56	55	52	49	57	78	91	96	97	98

Figura 6.14 – Tabela de Extensões (em décimos de segundo)

### 6.4.3. Superfície de Controle

A superfície de controle é a representação gráfica da tabela de extensões. Na Figura 6.1, no Módulo Controlador *Fuzzy*, existem dois produtos: a tabela de extensões e a superfície de controle. Esta divisão, contudo, é apenas de cunho didático, já que o Módulo Superfície de Controle utiliza a própria tabela de extensões para analisar a superfície de controle (ver item 6.6). A FLT-M apresenta a representação gráfica da superfície de controle, no formato mostrado na Figura 4.6.

## 6.5. Módulo Simulador

### 6.5.1. Simulador

Nesta etapa são realizadas as simulações de tráfego. Os dados de entrada para o simulador são a tabela de extensões, gerada no módulo anterior, as características da interseção e do tráfego a ser controlado, definidas no item 6.3.2. O algoritmo insere todos esses dados no simulador, sendo que a tabela de extensões varia de acordo com o cromossomo. As características da interseção são fixas e os volumes do tráfego a ser controlado são de dois tipos: volume alto e volume baixo. O produto desta etapa de simulação é a medida de desempenho escolhida para este trabalho, o atraso médio veicular, para cada cromossomo.

O simulador de tráfego adotado neste trabalho é o UnB-Sitracs (Simulador da operação do tráfego em interseções controladas por semáforos), versão b1.0, desenvolvido pelo Departamento de Ciência da Computação da Universidade de Brasília, como parte do projeto “Controle do tráfego em redes urbanas com o uso de controladores semafóricos *fuzzy*”, conforme já mencionado. O simulador utilizado será apresentado em seguida. Mais informações a respeito deste simulador podem ser encontradas em seu Manual do Usuário, encontrado no sítio <http://sourceforge.net/projects/unb-sitracs/files/>. O arquivo do simulador também é disponibilizado gratuitamente no mesmo sítio.

O UnB-Sitracs é um simulador microscópico que pode ser utilizado para interseções isoladas ou em rede aberta, na qual o tráfego se desloca ao longo de uma via que apresenta duas ou mais interseções controladas por semáforos, operando ou não de forma coordenada. A operação do tráfego pode ser simulada para interseções com controle por tempo fixo e por demanda do tráfego, sendo que no segundo caso as possibilidades são atuado *fuzzy* e semi-atuado.

Como o simulador foi desenvolvido no âmbito do projeto “Controle do tráfego em redes urbanas com o uso de controladores semafóricos fuzzy”, do qual a presente dissertação também faz parte, foi possível uma comunicação efetiva com a sua equipe desenvolvedora, fato este que contribuiu para a evolução do estudo. Os comandos utilizados no algoritmo para inserir os dados de entrada no simulador e obter a saída do mesmo foram cedidos pelos desenvolvedores do simulador. A interatividade entre o simulador e o software Matlab<sup>®</sup> é consequência do fato do simulador ter sido desenvolvido na linguagem de programação Java<sup>®</sup>, e do Matlab<sup>®</sup> ter como uma de suas ferramentas uma máquina virtual capaz de executar programas nessa linguagem.

Para a utilização do simulador é necessário que sejam definidos: geometria da interseção (ou da rede), características do tráfego e dos controladores semafóricos. Como estas definições já foram apresentadas no item 6.3.2, aqui será mostrado somente como elas foram inseridas no simulador. É válido lembrar que o arquivo XML com essas informações é criado uma única vez pelo usuário para que o algoritmo possa então utilizá-lo ao longo do processo.

Em relação à geometria da interseção, o simulador disponibiliza um conjunto de ferramentas para que o usuário possa desenhar a situação desejada. Os elementos que

devem ser definidos para a caracterização da geometria são: largura e extensão das vias, número de faixas de fluxo e posicionamento das faixas de retenção. A Figura 6.15 apresenta a tela principal do UnB-Sitracs com o desenho de uma interseção.

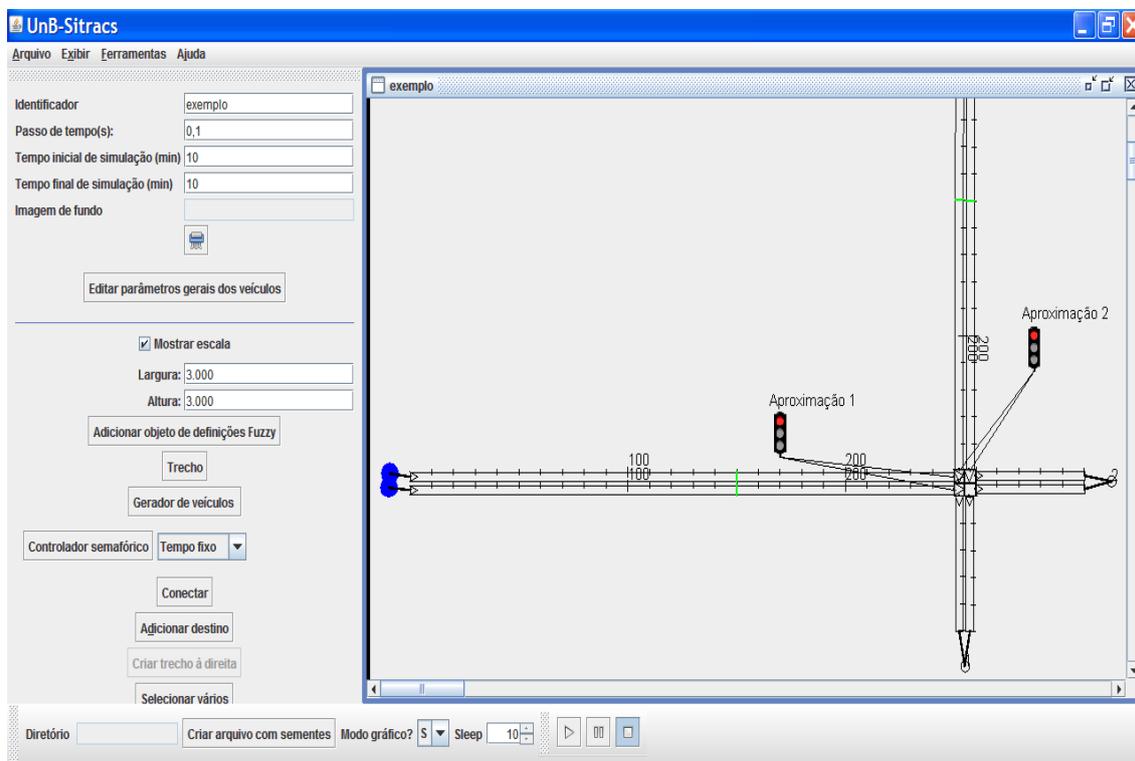


Figura 6.15 – Exemplo de interseção desenhada no UnB-Sitracs

As características do tráfego que devem ser definidas são: origem e destino de tráfego, velocidade, volume de tráfego, composição veicular e parâmetros gerais dos veículos. A origem do tráfego é criada por meio de um elemento denominado gerador. Cada faixa deve ter um gerador e um destino associado a ela. Os dois círculos cheios do lado esquerdo do desenho da Figura 6.15 são geradores, enquanto que os círculos vazios, um do lado direito e o outro na parte inferior da Figura 6.15, são destinos. Várias faixas podem ter um único destino, no entanto, cada faixa deve ter um único gerador.

Quando se clica duas vezes em um gerador, é apresentada uma janela ao usuário para a definição dos elementos citados no parágrafo anterior, com exceção dos parâmetros gerais dos veículos, conforme ilustra a Figura 6.16. Para se realizar a definição dos parâmetros gerais dos veículos, é necessário clicar em “Editar parâmetros gerais dos veículos”, apresentado na tela principal do simulador. Com isso, será apresentada uma janela para a entrada de dados (para maiores informações, ver Manual do Usuário do

UnB-Sitracs). Neste trabalho, adotou-se a mesma caracterização operacional utilizada por Andrade (2009), a qual é apresentada na Figura 6.17.

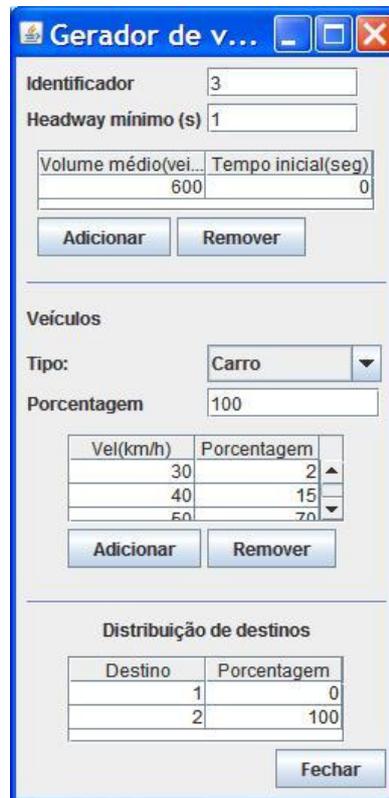


Figura 6.16 – Janela apresentada ao se clicar duas vezes sobre um gerador de veículos

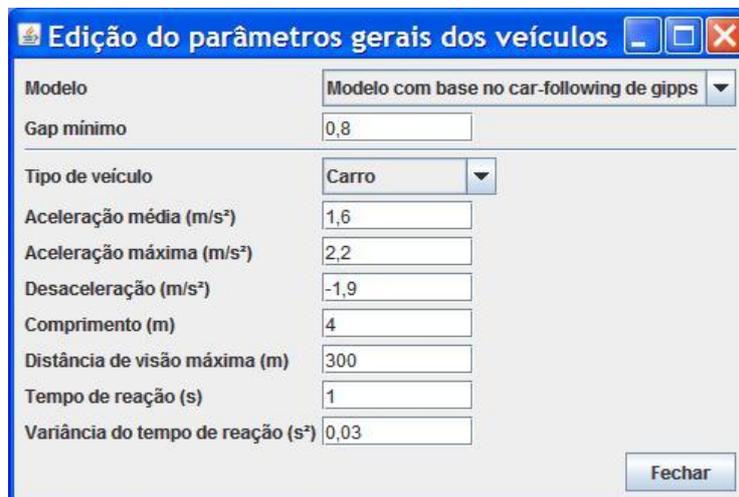


Figura 6.17 – Janela apresentada para definição dos parâmetros gerais dos veículos

Assim como no estudo desenvolvido por Andrade (2009), na definição do controlador semafórico *fuzzy* a ser utilizado no presente trabalho foram adotados os seguintes valores como parâmetros:

- tempo mínimo de verde = 10s;
- tempo de amarelo = 4s;
- tempo de vermelho geral = 1s;
- número máximo de extensões = 5;
- tempo mínimo de extensão = 2s.

Para a definição de um controlador semafórico, clica-se em “Controlador semafórico” (com o tipo desejado selecionado no campo ao lado, isto é, tempo fixo, semi-atuado ou *fuzzy*), e em seguida adiciona-se o controlador ao ponto de conflito em estudo. A Figura 6.18 mostra a janela que é apresentada ao usuário para que sejam inseridas as características de um controlador semafórico *fuzzy*. Destaca-se que o campo “Tabela” é preenchido pelo algoritmo com a tabela de extensões gerada pelo Matlab<sup>®</sup> para cada cromossomo.

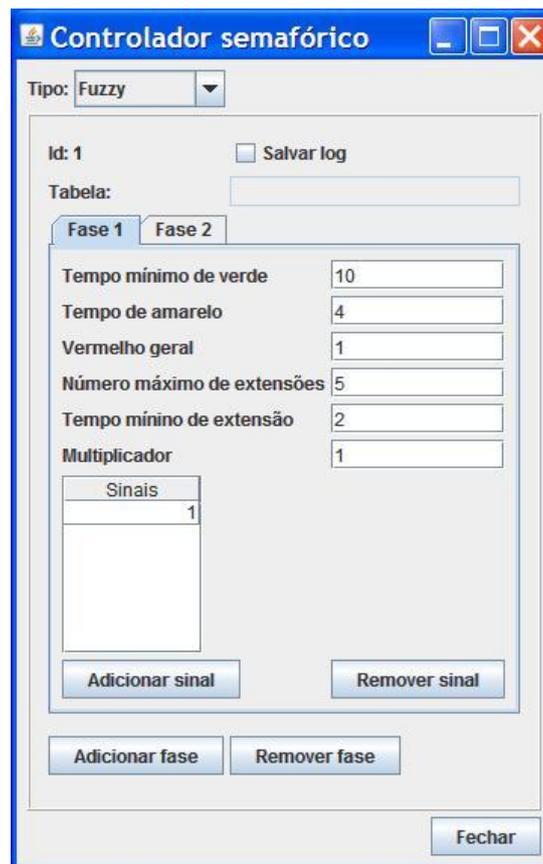


Figura 6.18 – Janela apresentada para definição de um controlador semafórico *fuzzy*

Após o término de cada simulação, o simulador emite um relatório do qual constam as seguintes medidas de desempenho:

- Atraso médio por veículo: média aritmética dos atrasos sofridos pelos veículos, sendo o atraso definido como a diferença entre o tempo que o veículo percorreria o percurso se mantivesse a velocidade desejada e o tempo efetivamente gasto para percorrê-lo.
- Número médio de paradas por veículo: relação entre o total de paradas e o total de veículos, sendo uma parada caracterizada quando o veículo atinge uma velocidade inferior a 3,6 km/h (1 m/s).
- Percentagem de paradas: percentual do total de veículos simulados que sofreu uma ou mais paradas.
- Tempo médio de verde das fases semaforicas: relação entre o total de tempo verde e o número total de vezes em que o verde foi apresentado.
- Velocidade média: relação entre a distância percorrida e o tempo gasto para percorrê-la.

O UnB-Sitracs trabalha com sementes para a simulação. Estas são utilizadas para a geração dos veículos. Com isso, caso duas simulações sejam realizadas com sementes, volumes e geometrias iguais, elas terão os mesmos resultados. Para cada cromossomo, foram realizadas três simulações com três sementes distintas, no intuito de que o resultado obtido não fosse dependente de uma única semente. As três sementes distintas, no entanto, são as mesmas utilizadas para todos os cromossomos.

O atraso atribuído a cada cromossomo é a média dos atrasos de suas simulações, desprezando-se as que estiverem fora do intervalo  $X + 2S$ , em que  $X$  é a média e  $S$  o desvio-padrão. Sabe-se que três simulações para cada cromossomo é um número pequeno, no entanto, devido ao grande período de tempo que cada simulação demanda (para a execução completa do procedimento da Figura 6.1, com três simulações para cada cromossomo, para um volume alto, o algoritmo requer aproximadamente 8 dias), foi necessário a adoção desse valor. Embora não seja ideal realizar poucas simulações para cada cromossomo, os trabalhos de Andrade (2009) e Labanca (2009) mostram que, mesmo para um número maior de simulações, o simulador não gera muitos resultados fora do intervalo acima mencionado.

### 6.5.2. Medida de Desempenho

Pelo fato de ser bastante empregado, o atraso médio veicular foi escolhido como a medida de desempenho para avaliar a fluidez do tráfego na interseção. Conforme exposto, o simulador é o responsável pelo cálculo desse atraso (conforme Equação 6.17). Este é um dos argumentos que são utilizados na função aptidão.

$$AM = \frac{\sum_{i=1}^n Tr_i - Tid_i}{n} \quad (\text{Eq. 6.17})$$

Em que  $AM$  é o atraso médio veicular (em segundos),  $Tr_i$  é o tempo real gasto pelo veículo  $i$  para percorrer uma dada distância (em segundos), e  $Tid_i$  é o tempo ideal para que o veículo  $i$  percorresse essa dada distância (se viajasse na velocidade desejada) (em segundos).

O algoritmo desenvolvido neste trabalho também classifica o atraso de cada simulação de acordo com os parâmetros de classificação no *Highway Capacity Manual – HCM* (TRB, 2000), conforme mostrado na Tabela 6.13. O melhor nível de serviço é aquele que apresenta o menor atraso médio veicular, sendo denominado de A. O nível de serviço F representa a situação mais indesejada, com atraso médio veicular superior a 80 segundos. Os níveis B, C, D e E representam faixas intermediárias.

Tabela 6.13 – Classificação do Atraso Segundo HCM (TRB, 2000)

Atraso Médio Veicular (s)	Nível de Serviço
≤ 10	A
> 10 – 20	B
> 20 – 35	C
> 35 – 55	D
> 55 – 80	E
> 80	F

## 6.6. Módulo Superfície de Controle

### 6.6.1. Avaliação da Superfície de Controle

Para avaliar a suavidade e coerência da superfície de controle, criou-se uma função denominada *penaliza()*. Esta percorre as linhas e colunas da matriz (tabela de extensões)

que gerou a superfície e atribui penalizações a situações indesejadas. A função retorna um valor denominado *penalização*, o qual será somado ao valor do atraso na função *aptidão*.

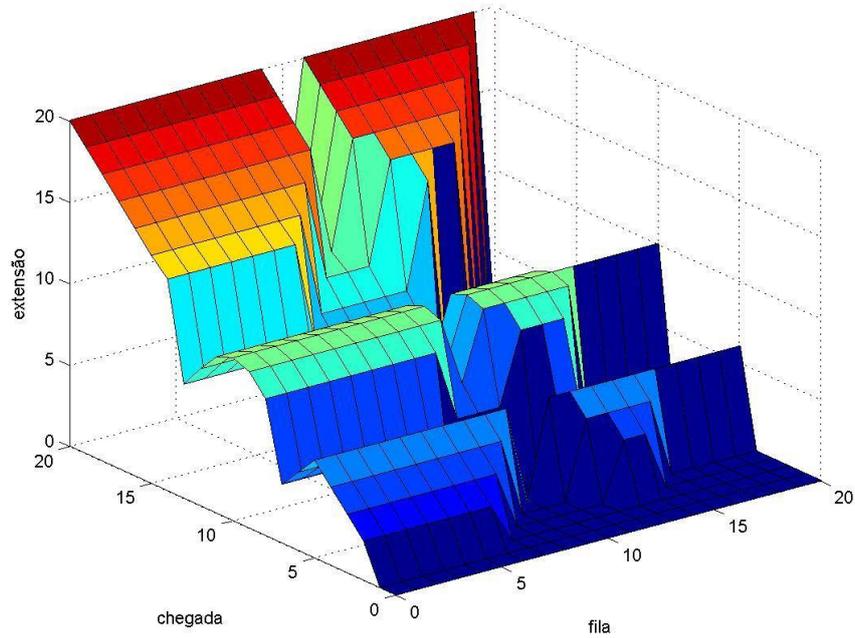
Existem três situações em que a penalização é aplicada: incoerência, não-suavidade e insensibilidade. A incoerência ocorre de duas formas: quando, para uma mesma fila, o valor da extensão diminui com o aumento de chegada, e quando, para uma mesma chegada, o valor da extensão aumenta com o aumento de fila. A penalização por incoerência foi definida como sendo dez vezes o valor da diferença de extensões entre dois pontos sucessivos, quando detectada a incoerência.

A expressão não-suavidade se refere a quando ocorre uma variação coerente, porém muito brusca no valor da extensão devido a uma pequena mudança nas condições do tráfego. Um exemplo, se a extensão dada pelo controlador para fila igual a 5 e chegada igual a 12 é de 10 segundos, e a extensão concedida para fila igual 5 e chegada igual a 11 é de 5 segundos, percebe-se que ocorreu um decréscimo de 5 segundos no valor de extensão devido a uma pequena redução (uma unidade) na chegada. Tal situação é indesejada, pois assim a percepção da dinâmica do controle da interseção por parte dos motoristas pode ser prejudicada.

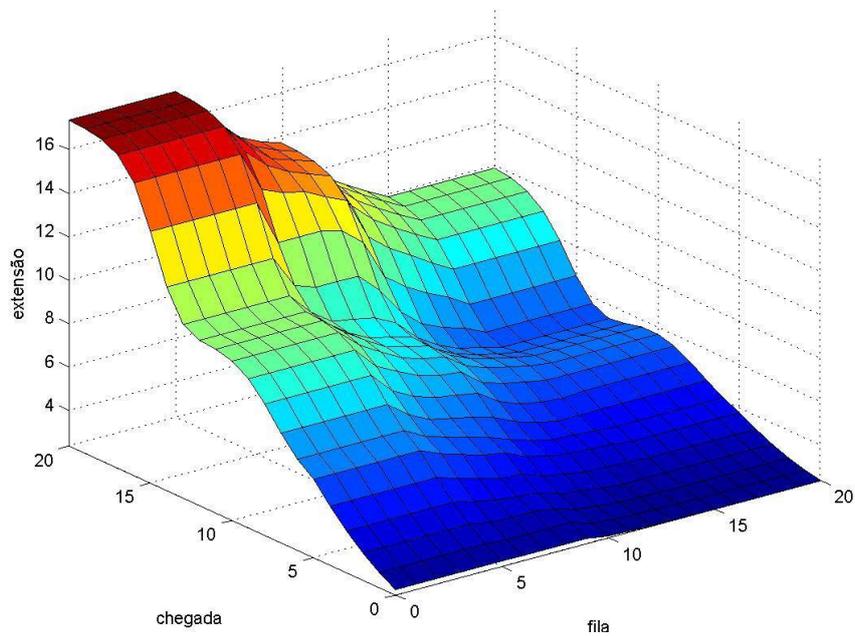
A penalização por não-suavidade depende do quanto a variação entre dois pontos sucessivos é brusca. Se a variação for maior ou igual a 3 segundos, a penalização é de dez vezes o valor da variação. Se for maior que 2 segundos e menor que 3 segundos, a penalização corresponde ao valor da própria variação. Caso a variação seja menor ou igual a 2 segundos, não é aplicada penalização, pois considerou-se este tipo de variação como suave.

A necessidade da penalização por insensibilidade foi constatada durante o processo. Ao longo das gerações, as superfícies tendiam a evoluir para somente dois patamares com um único desnível entre eles. Assim, a penalização era pequena, pois correspondia a um único desnível, já que os patamares não se encaixavam em nenhuma situação de penalização. Assim, para evitar tal fato, penalizou-se também a situação em que extensão tem o mesmo valor para dois pontos sucessivos. A Figura 6.19 ilustra duas superfícies de controle avaliadas pela função *penaliza()*. A primeira (Figura 6.19(a)),

indesejável, recebeu uma penalização de 110.182 e a segunda (Figura 6.19(b)), satisfatória, recebeu uma penalização de 24.910.



(a) Superfície de controle com penalização igual a 110.182



(b) Superfície de controle com penalização igual a 24.910

Figura 6.19 – Superfícies de Controle com suas Respectivas Penalizações

À medida que os primeiros resultados apareceram percebeu-se um problema. A ordem de grandeza dos resultados da penalização era muito maior do que do atraso e, assim, este último era praticamente inexpressivo na função aptidão. A ordem de grandeza das penalizações costumava ser de  $10^4$ , chegando algumas vezes a  $10^5$ . Isto significa que o algoritmo iria evoluir considerando basicamente a qualidade da superfície de controle, sem levar em conta o atraso.

Por isso, decidiu-se dividir o valor das penalizações por 10.000. Logo, uma superfície que tinha uma penalização de 110.000, por exemplo, passaria a ter uma penalização de 11, que acrescida a um suposto atraso de 20 segundos, totalizaria um valor de 31. Esta soma é utilizada na função aptidão, a qual será detalhada no item 6.7.4.

#### *6.6.2. Penalização “P”*

A penalização “P” é o valor descrito no item anterior, sendo o produto final do Módulo Superfície de Controle. Este valor, conforme já exposto é um dos argumentos a serem trabalhados na função aptidão.

### **6.7. Módulo Genético**

#### *6.7.1. Detecção de Indivíduos Problemáticos*

Originalmente, esta etapa foi criada para detectar dois tipos de problemas: os excessos de representação decorrentes da codificação binária (ver item 6.3.5), que não faziam sentido para o algoritmo, e a descontinuidade dos conjuntos. Como os limites dos conjuntos *fuzzy* são variados de acordo com os genes de um cromossomo, é possível que se chegue a uma situação que apresente “vazios” (descontinuidade dos conjuntos) no gráfico. Isto é, casos em que alguns pontos do domínio não tenham grau de pertinência associado a nenhum dos conjuntos da variável considerada.

No entanto, com a redução do espaço de busca descrito no item 6.3.4, e com a adoção da codificação decimal, para solucionar o problema dos excessos, esta etapa tornou-se desnecessária. Isto porque os excessos deixaram de existir e o valor máximo para a variação dos limites dos conjuntos *fuzzy* passou a ser 2, fato que impossibilita uma descontinuidade nos conjuntos das variáveis apresentados na Tabela 6.2.

### 6.7.2. Substituição de Indivíduos Problemáticos

Embora não exista mais a possibilidade de aparecerem indivíduos problemáticos, é válido citar que foi criada a função *eliminaMembroProblematico()*, a qual é responsável por eliminar os indivíduos “problemáticos” que foram previamente detectados (conforme descrito no item 6.7.1), e poderá ser utilizada em outros estudos que trabalhem com maiores variações dos limites dos conjunto *fuzzy*. Esta função tem como argumento de entrada um cromossomo problemático (identificado pelo algoritmo para os casos anteriormente descritos). Este cromossomo é então substituído por outro gerado aleatoriamente. Caso o novo cromossomo também seja problemático, o processo é repetido até que o novo cromossomo não apresente características indesejadas.

### 6.7.3. Geração “i”

A geração “i” corresponde à i-ésima população, depois de substituídos todos os seus cromossomos problemáticos por cromossomos que apresentem características perfeitamente interpretáveis pelo algoritmo.

### 6.7.4. Função Aptidão

Neste trabalho, o cálculo da aptidão é função do atraso médio veicular e da penalização “P” apresentada em 6.6.2. Estes dois valores são somados para cada cromossomo (Equação 6.18), e quanto menor for a soma maior será a aptidão do cromossomo, e vice-versa. Em seguida, para cada geração, estes valores são normalizados no intervalo [0, 10], de forma que o indivíduo mais apto tenha aptidão 10 e o menos apto 0. Assim, a função aptidão  $F(x)$  é definida pela Equação 6.19:

$$S(x) = Atraso(x) + P(x) \quad (\text{Eq. 6.18})$$

Em que  $Atraso(x)$  é o atraso médio veicular do controlador definido pelo cromossomo  $x$  e  $P(x)$  é a penalização da superfície de controle do controlador definido pelo cromossomo  $x$ .

$$F(x) = 10 \cdot \left( \frac{S(x) - S_{maior}}{S_{menor} - S_{maior}} \right) \quad (\text{Eq. 6.19})$$

Em que  $S(x)$  é o mesmo definido na Equação 6.18,  $S_{menor}$  é a menor soma do atraso médio veicular e penalização de um cromossomo na geração e  $S_{maior}$  é a maior soma do atraso médio veicular e penalização de um cromossomo na geração.

Destaca-se que a aptidão de um indivíduo está sempre relacionada ao melhor e ao pior de cada geração (Eq. 6.19). Por exemplo, um indivíduo de aptidão 10 em uma geração pode ter aptidão inferior em outra geração. Assim, o valor de  $F(x)$  só deve ser comparado com outros valores de  $F(x)$  de uma mesma geração. Para se obter o melhor resultado de todas as gerações, deve-se comparar os valores de  $S(x)$  dos indivíduos (Eq. 6.18).

#### 6.7.5. Verificação da Condição de Controle

Nesta etapa, o algoritmo verifica se o número de gerações máximo foi atingido, para as Etapas 1 e 2, apresentadas no item 6.3.4. Quando a última geração da Etapa 1 é simulada e avaliada, esta é encerrada e a Etapa 2 é iniciada. Após as simulações e cálculos de aptidão para a última geração da Etapa 2, a condição de controle é atendida e o processo iterativo chega ao seu fim. Neste trabalho, o número máximo de gerações para cada uma das Etapas é de 100.

#### 6.7.6. Operadores do AG e Elitismo

A seleção foi implementada com o método da roleta, descrito no item 5.4.3. No algoritmo, utilizou-se o recurso da janela, para a qual se adotou o valor 1. Uma vez findada a seleção, a população intermediária (Figura 5.2) está definida. A transformação da população intermediária para a nova população é realizada com o auxílio dos operadores “recombinação” e “mutação”.

Primeiramente, deve ser realizada a escolha de um operador. Cada operador tem uma probabilidade de ser escolhido. Inicialmente, a probabilidade de escolha do operador recombinação é de 0,7 (70%) e a probabilidade de que o operador mutação seja selecionado é de 0,3 (30%) (a soma desses valores deve ser sempre igual a 1). Gera-se um número aleatório entre 0 e 1, caso este seja menor que a probabilidade de escolha do operador recombinação, esta é escolhida, caso contrário, seleciona-se o operador mutação.

Em seguida, verifica-se se o operador escolhido deve ser efetivamente utilizado ou não, de acordo com sua probabilidade de ocorrência. Ressalte-se que **probabilidade de escolha** de um operador e **probabilidade de ocorrência** do mesmo são elementos distintos. O primeiro está relacionado a qual operador será escolhido e o segundo a se o operador é utilizado de fato. Caso um operador seja escolhido, mas sua operação não ocorra no cromossomo(s) selecionado(s), este(s) passará para a nova população inalterado(s).

Se o operador selecionado for a recombinação, são escolhidos aleatoriamente dois cromossomos da população intermediária, os quais serão recombinados entre si e inseridos na nova população. A posição em que a recombinação ocorre também é aleatória (é gerado um número aleatório entre 1 e o tamanho do cromossomo, o qual será o *locus* em que ocorrerá a recombinação). Neste trabalho, uma vez selecionado o operador recombinação, esta sempre ocorrerá, isto ocorre porque a probabilidade de ocorrência de recombinação foi definida como sendo 1.

Caso a mutação é que tenha sido selecionada como operador, escolhe-se aleatoriamente um único cromossomo. O algoritmo percorrerá então cada gene do cromossomo para verificar se realizará a mutação ou não. Para cada gene, é gerado um número aleatório entre 0 e 1; caso este número seja menor que a probabilidade de ocorrência de mutação, aplica-se a mutação. A probabilidade de ocorrência de mutação adotada neste trabalho é de 0,04 (4%). Após esse processo, o cromossomo é inserido na nova população.

Esse processo de escolha de operador é repetido até que a nova população possua o número de indivíduos definidos como o tamanho de uma população, isto é, 100. Chama-se a atenção para uma possível situação em que, depois de inseridos 99 indivíduos na nova população, seja selecionado o operador recombinação. Com isso, seriam inseridos mais dois cromossomos e a nova população passaria a ter 101 indivíduos. Em casos como esse, o algoritmo sempre descarta o último cromossomo que deveria ser inserido na nova população.

No decorrer das gerações, a evolução dos cromossomos tende a manter os genes que apresentam maior aptidão à situação proposta. Com isso, os cromossomos podem ficar muito parecidos ao longo do processo, de maneira que a recombinação não promova novos cromossomos. Por este motivo, a literatura sugere que a probabilidade de escolha

do operador mutação cresça ao longo das gerações, haja vista que a mutação promove a variabilidade genética, e a probabilidade de escolha do operador recombinação diminua.

Sendo assim, definiu-se no algoritmo que a cada 10 gerações, a probabilidade de escolha do operador mutação é acrescida em 0,05 e a probabilidade de escolha do operador recombinação é subtraída em 0,05. Definiu-se, também, valores limites finais para essas probabilidades de escolha, sendo 0,4 o mínimo para a recombinação e 0,6 o máximo para a mutação. A Tabela 6.14 apresenta um resumo de todas as probabilidades adotadas neste trabalho, relacionadas com os operadores.

Tabela 6.14 – Probabilidades Adotadas Relativas aos Operadores

Probabilidade inicial de escolha do operador recombinação	0,7
Probabilidade inicial de escolha do operador mutação	0,3
Probabilidade final de escolha do operador recombinação	0,4
Probabilidade final de escolha do operador mutação	0,6
Probabilidade de ocorrência da recombinação	1
Probabilidade de ocorrência da mutação	0,04

Conforme já mencionado na seção 5.5, não há estudos conclusivos que indiquem quais os melhores parâmetros de um AG. Assim, para a definição dos parâmetros apresentados na Tabela 6.14, bem como do tamanho da população e do número de gerações, foi consultada, além da literatura, a parte da equipe do Departamento de Ciência da Computação da UnB que integra o projeto “Controle do tráfego em redes urbanas com o uso de controladores semafóricos *fuzzy*”, no qual o presente trabalho está incluso.

O algoritmo deste trabalho implementa também o elitismo, garantindo que o melhor resultado de cada geração seja inserido na próxima, caso esta última não apresente um resultado melhor que a anterior. Quando o elitismo ocorre, o pior indivíduo (de menor aptidão) da geração atual é descartado para que o mais apto da geração anterior seja inserido.

### 6.7.7. Nova População

O termo “Nova População” foi criado para denominar a população gerada no item 6.7.6, a qual ainda não teve seus indivíduos problemáticos substituídos. Uma vez realizadas essas substituições, tem-se a nova geração. Como na nova concepção do algoritmo não existe mais a possibilidade de aparecerem indivíduos problemáticos, a nova geração será sempre igual à nova população.

## 6.8. Módulo Final - Resultados

Neste Módulo são apresentados os resultados do processo iterativo realizado pelo algoritmo. O algoritmo gera os seguintes relatórios (do tipo arquivos de texto) para as Etapas 1 e 2:

- cromossomos, agrupados por geração, e seus respectivos atrasos, penalizações, somas do atraso e penalização, aptidões, desvios-padrão dos atrasos das simulações e níveis de serviço;
- atrasos excluídos, por cromossomo (atrasos que ficaram fora do intervalo  $X \pm 2S$ , em que  $X$  é a média e  $S$  o desvio-padrão.);
- cromossomos mais aptos de cada geração;
- cromossomos que tiveram mesmo nível de aptidão do mais apto de cada geração.

## 6.9. Comparação dos melhores resultados

Os melhores resultados serão comparados com resultados obtidos em um controlador semafórico *fuzzy* utilizado no estudo de Andrade (2009). Este é um controlador do tipo Sugeno, cujo modo de implementação do operador “E” foi a interseção, e tem as mesmas variáveis *fuzzy*, conjuntos *fuzzy*, e base de regras utilizadas no presente trabalho. Para tanto, serão realizadas novas simulações no UnB-Sitracs com a tabela de extensões fornecida pelo controlador do estudo de Andrade (2009). Estas novas simulações serão realizadas utilizando-se as mesmas sementes dos geradores de tráfego para os controladores comparados, possibilitando uma comparação imparcial dos controladores.

## 7. APLICAÇÃO DA METODOLOGIA E ANÁLISE DE RESULTADOS

Conforme descrito no Capítulo 6, o algoritmo foi executado 5 vezes para cada nível de volume, alto e baixo. Nas seções a seguir serão apresentados os resultados obtidos em cada execução. Será realizada também a análise destes dados. Na fase de análise, o melhor controlador obtido com o procedimento proposto será comparado ao controlador adotado em estudos anteriores do PPGT (Andrade, 2009). Esta comparação levará em conta o desempenho do tráfego controlado, refletido pela medida de desempenho atraso médio por veículo, calculado pelo UnB-Sitracs para a interseção como um todo. A análise comparativa será feita em separado para os volumes alto e baixo considerados na aplicação do AG proposto.

### 7.1 Resultados Obtidos

#### 7.1.1. Volume Alto

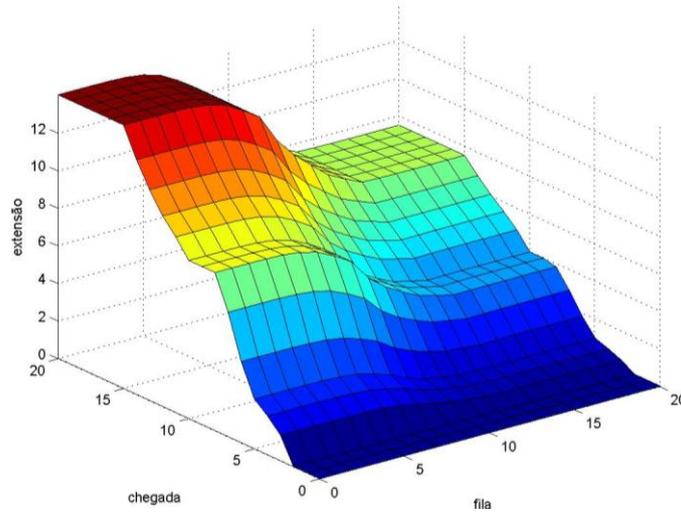
##### a) Cromossomos que Forneceram os Melhores Resultados

A Tabela 7.1 apresenta os cromossomos que apresentaram o melhor resultado em cada uma das 5 execuções do algoritmo para o volume alto, fornecidos ao final da Etapa 2, com seus respectivos atrasos, penalizações e somas (atraso + penalização). A Figura 7.1 ilustra as superfícies de controle de cada um destes cromossomos.

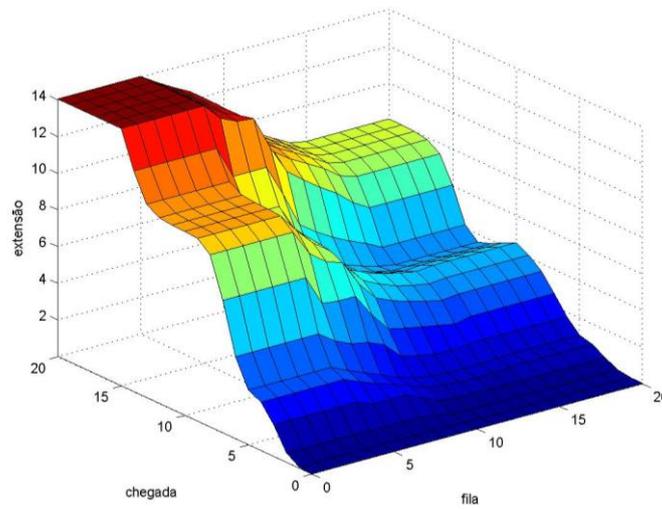
Tabela 7.1 – Cromossomos que Apresentaram os Melhores Resultados para Volume Alto

Semente	Cromossomo com Melhor Resultado (Volume Alto)	Atraso (s)	Penalização	Soma (A+P)
1	1 1 0 2 4 1 2 2 2 0 2 1 2 0 0 1 1 0 1 1 2 1 1 1 1 0 2	20,7582	2,5470	23,3052
2	1 0 1 1 4 1 1 1 2 0 2 0 2 2 0 0 1 1 1 0 0 1 1 2 0 0 2	20,7838	2,6190	23,4028
3	1 0 0 2 4 0 0 2 1 0 2 1 1 2 0 2 1 2 0 0 0 1 1 0 1 0 2	20,8684	2,5804	23,4488
4	1 1 0 2 4 1 1 0 2 0 2 0 0 2 0 1 1 2 0 1 0 1 1 1 0 0 2	20,8175	2,6103	23,4278
5	1 1 0 2 4 0 2 1 2 0 2 1 1 2 0 1 1 0 2 2 2 1 1 1 2 0 2	21,0290	2,4530	23,4820

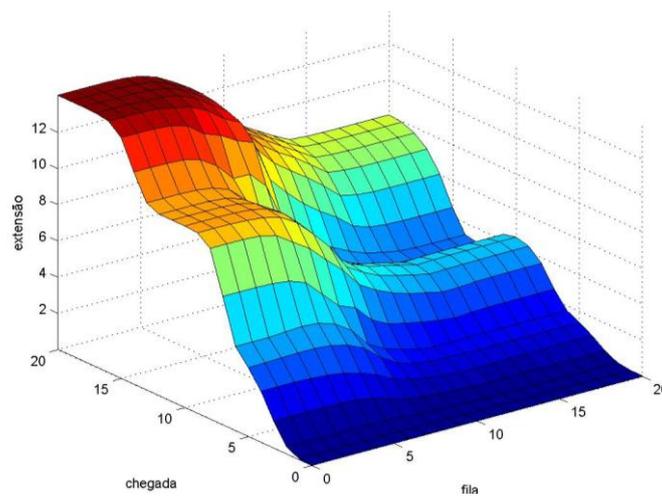
*b) Superfícies de Controle e Penalizações dos Cromossomos de Melhores Resultados*



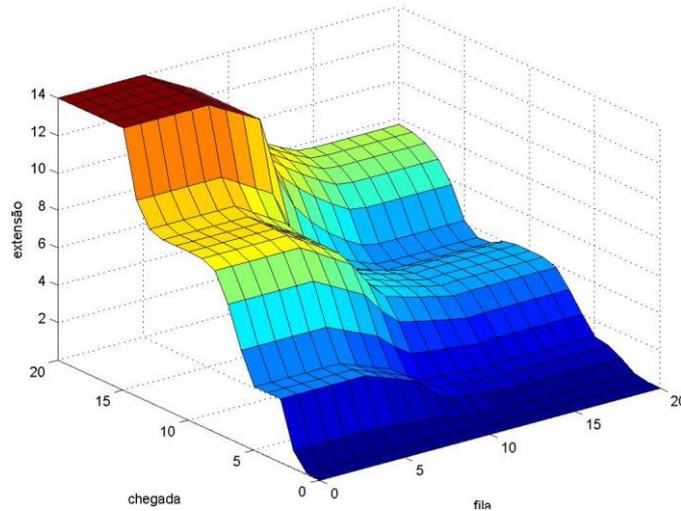
(a) Superfície do melhor cromossomo da semente 1 (Penalização = 2,5470)



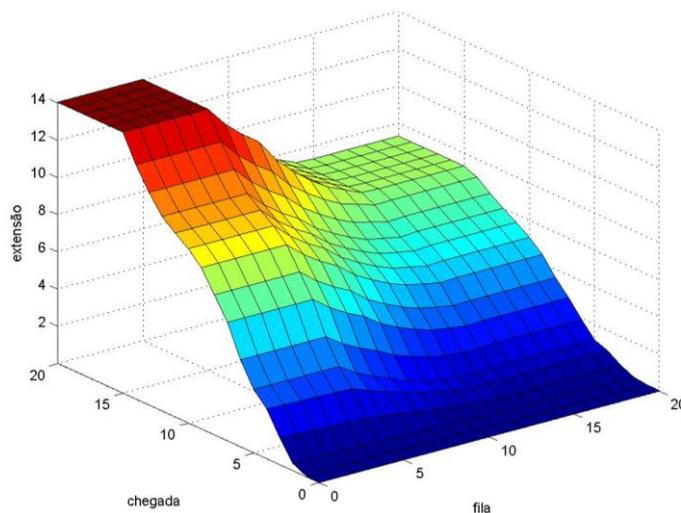
(b) Superfície do melhor cromossomo da semente 2 (Penalização = 2,6190)



(c) Superfície do melhor cromossomo da semente 3 (Penalização = 2,5804)



(d) Superfície do melhor cromossomo da semente 4 (Penalização = 2,6103)



(e) Superfície do melhor cromossomo da semente 5 (Penalização = 2,4530)

Figura 7.1 – Superfícies de Controle dos Cromossomos com Melhores Resultados para Volume Alto com suas Respectivas Penalizações

*c) Características do Melhor Controlador Obtido*

Conforme mostra a Tabela 7.1, o melhor resultado obtido para o volume alto foi o do melhor cromossomo da semente 1, com soma  $S(x) = 23,3052$ . Comparando-se esse controlador com os demais, verifica-se que não só a soma total foi menor do que a dos demais, como também o atraso médio correspondente. O controlador com a melhor superfície foi o produzido pela semente 5 que, no entanto, produziu uma elevação no atraso médio que foi suficiente para tornar o valor da sua função  $S(x)$  superior ao obtido

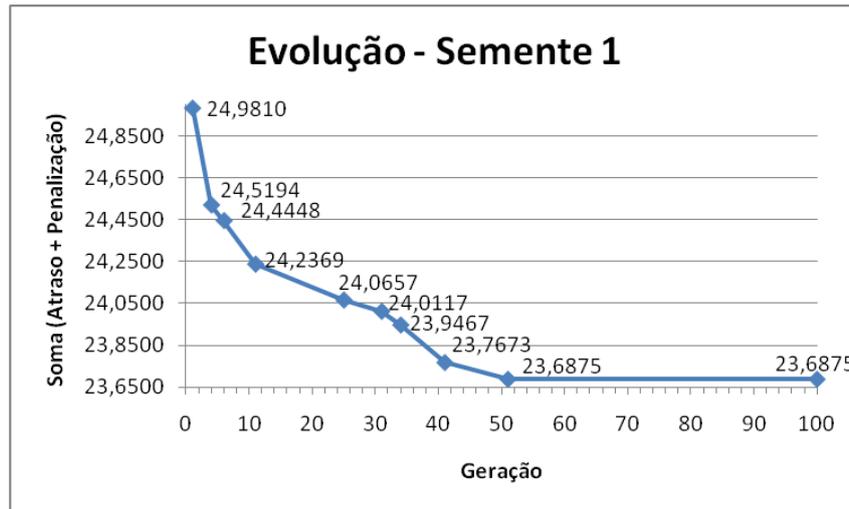
pelo controlador da semente 1. O controlador semafórico *fuzzy* representado pelo cromossomo associado à semente 1 tem as seguintes características:

- tipo do controlador = **Sugeno**;
- modo de implementação do operador “E” = **Prod**;
- método de defuzificação = **Wtaver**;
- tipo de função de pertinência para Fila “Pequena” = **Trapezoidal**;
- variação dos limites do conjunto de Fila “Pequena” = **+2**;
- tipo de função de pertinência para Fila “Média” = **Gaussiana**;
- variação dos limites do conjunto de Fila “Média” = **+2**;
- tipo de função de pertinência para Fila “Longa” = **Trapezoidal**;
- variação dos limites do conjunto Fila “Longa” = **+2**;
- tipo de função de pertinência para Chegada “Zero” = **Trapezoidal**;
- variação dos limites do conjunto Chegada “Zero” = **+2**;
- tipo de função de pertinência para Chegada “Pouca” = **Triangular**;
- variação dos limites do conjunto Chegada “Pouca” = **-2**;
- tipo de função de pertinência para Chegada “Média” = **Trapezoidal**;
- variação dos limites do conjunto Chegada “Média” = **0**;
- tipo de função de pertinência para Chegada “Muita” = **Trapezoidal**;
- variação dos limites do conjunto Chegada “Muita” = **0**;
- tipo de função de pertinência para Extensão “Zero” = **Trapezoidal**;
- variação dos limites do conjunto Extensão “Zero” = **+2**;
- tipo de função de pertinência para Extensão “Curta” = **Trapezoidal**;
- variação dos limites do conjunto Extensão “Curta” = **0**;
- tipo de função de pertinência para Extensão “Média” = **Trapezoidal**;
- variação dos limites do conjunto Extensão “Média” = **0**;
- tipo de função de pertinência para Extensão “Longa” = **Trapezoidal**;
- variação dos limites do conjunto Extensão “Longa” = **+2**.

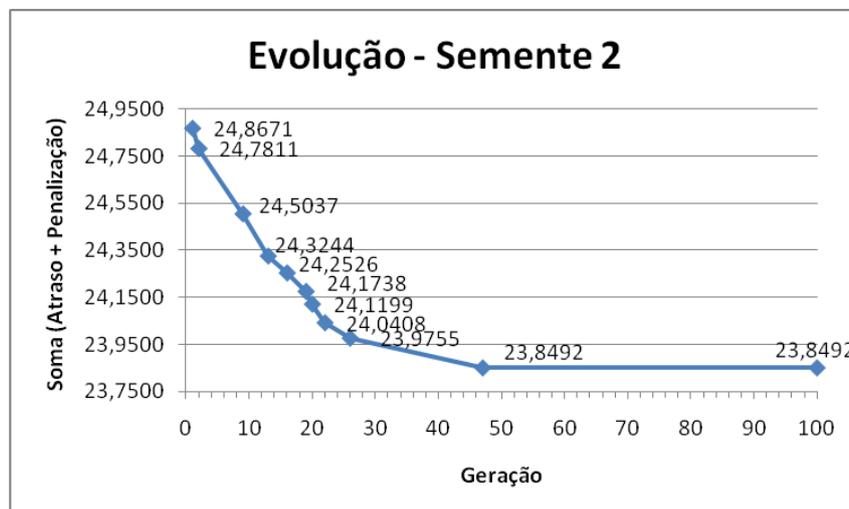
Como o controlador acima é do tipo Sugeno, as variáveis “modo de implementação da função de implicação” e “modo de implementação da agregação” não são consideradas.

d) *Evolução do AG*

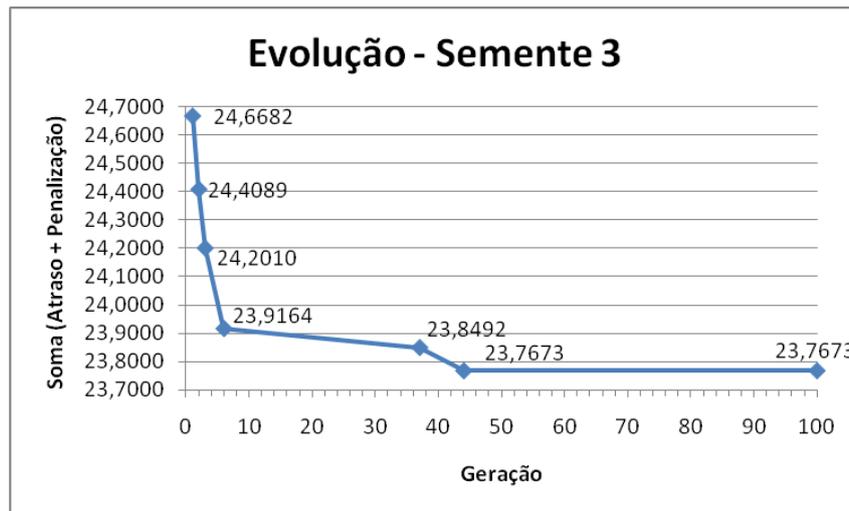
A Figura 7.2 apresenta a evolução do AG, para cada semente, ao longo das 100 gerações da Etapa 1, enquanto que a Figura 7.3 agrega essas evoluções em um único gráfico. Da mesma maneira, a Figura 7.4 ilustra a evolução do AG, para cada semente, na Etapa 2, e a Figura 7.5 agrega essas evoluções em um único gráfico.



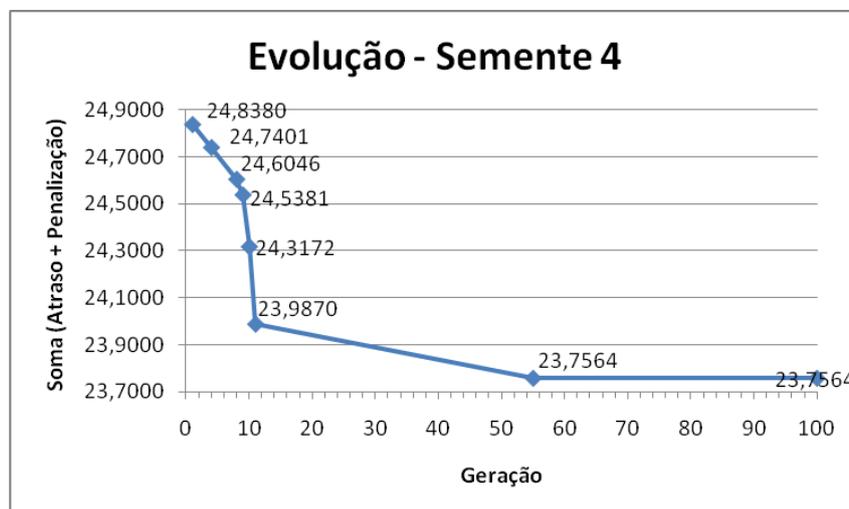
(a) Evolução do AG para semente 1 (Etapa 1)



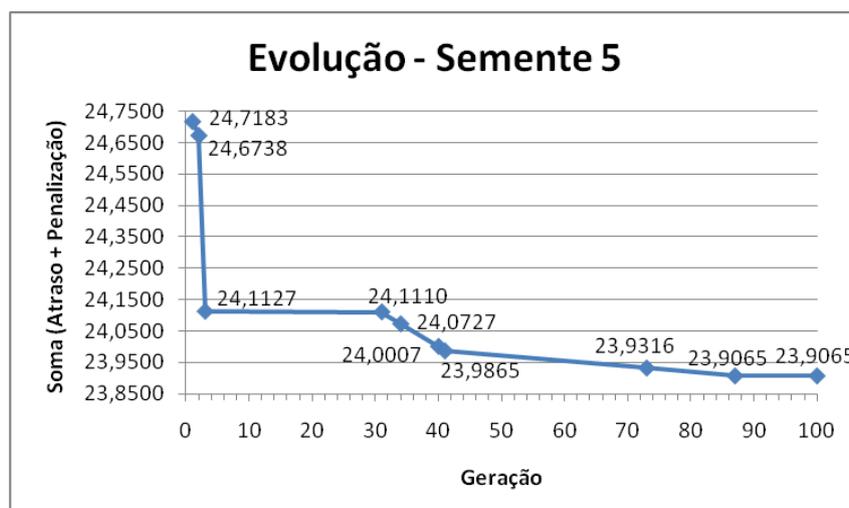
(b) Evolução do AG para semente 2 (Etapa 1)



(c) Evolução do AG para semente 3 (Etapa 1)



(d) Evolução do AG para semente 4 (Etapa 1)



(e) Evolução do AG para semente 5 (Etapa 1)

Figura 7.2 – Evolução do AG para Cada Semente da Etapa 1 (Volume Alto)

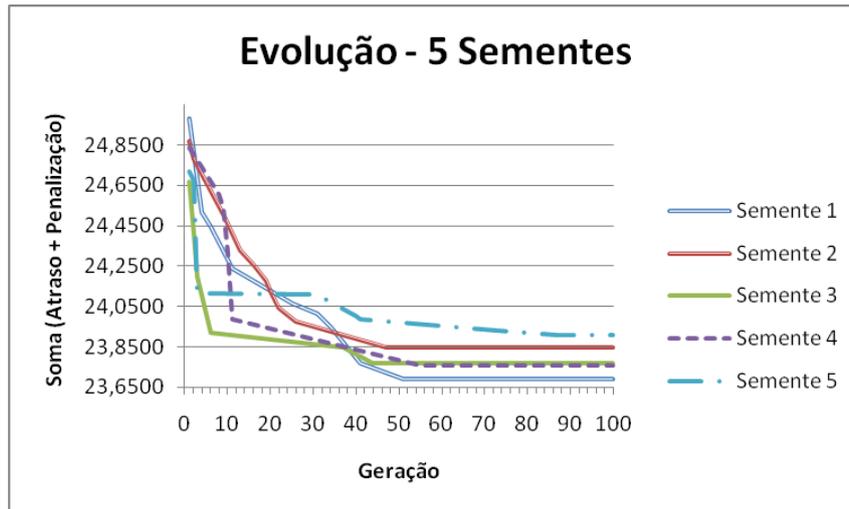
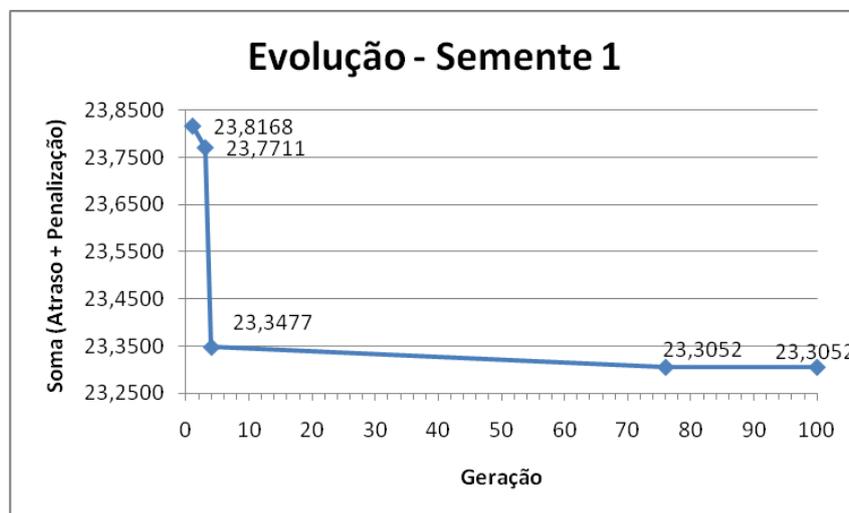
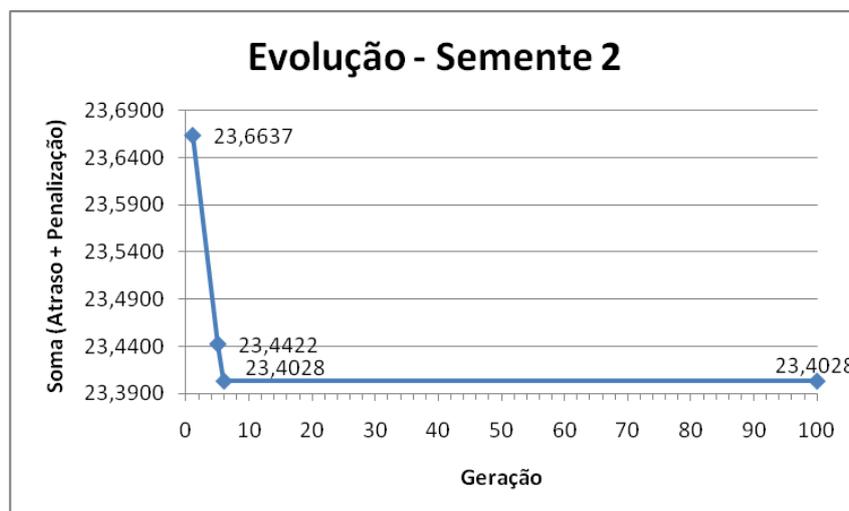


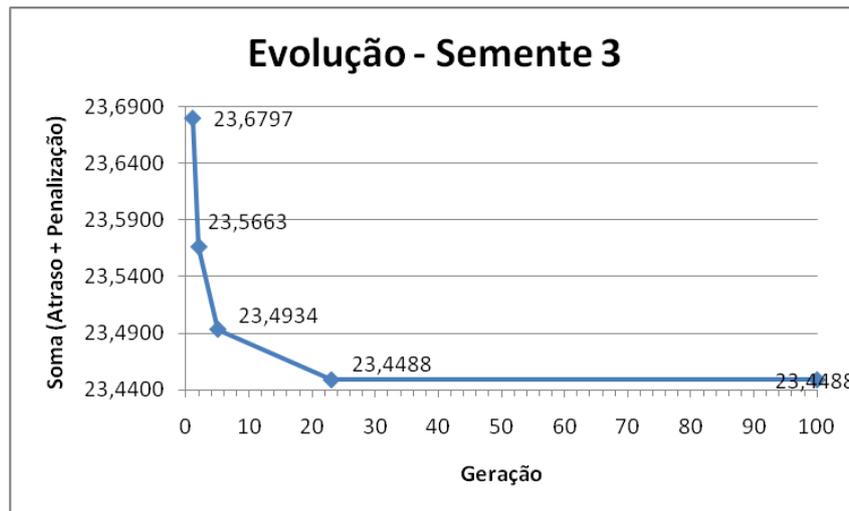
Figura 7.3 – Evolução do AG em Todas as Sementes da Etapa 1 (Volume Alto)



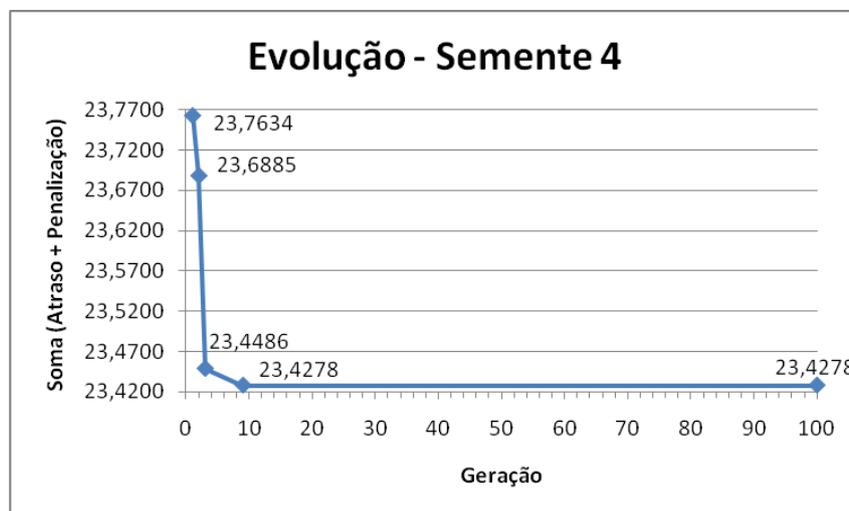
(a) Evolução do AG para semente 1 (Etapa 2)



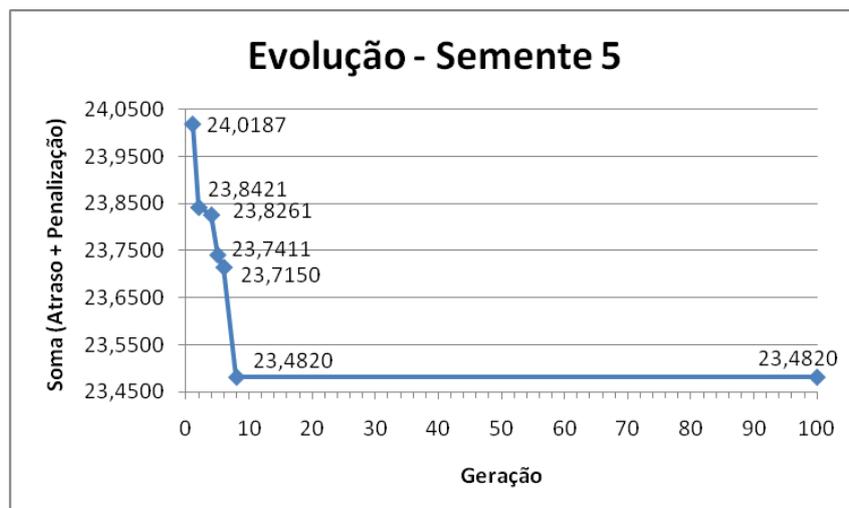
(b) Evolução do AG para semente 2 (Etapa 2)



(c) Evolução do AG para semente 3 (Etapa 2)



(d) Evolução do AG para semente 4 (Etapa 2)



(e) Evolução do AG para semente 5 (Etapa 2)

Figura 7.4 – Evolução do AG para Cada Semente da Etapa 2 (Volume Alto)

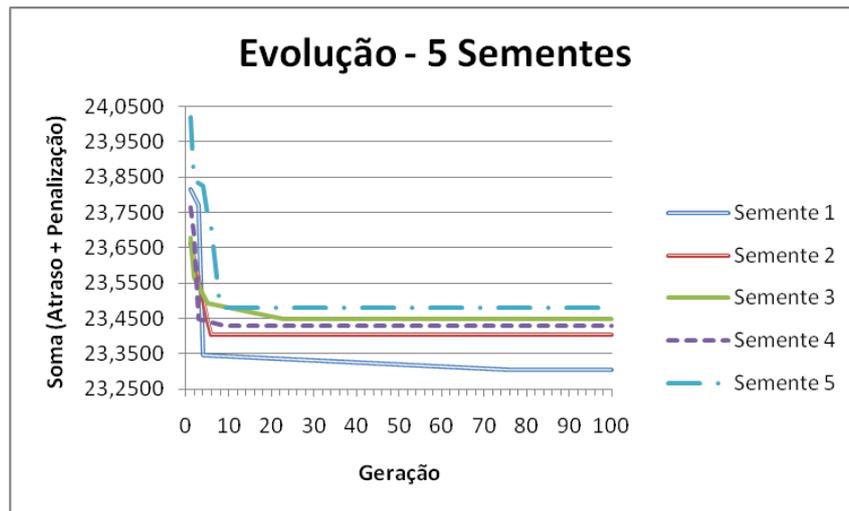


Figura 7.5 – Evolução do AG em Todas as Sementes da Etapa 2 (Volume Alto)

*e) Cromossomos com mesmo desempenho dos melhores*

O algoritmo retorna como melhor controlador o cromossomo que tenha o melhor resultado na última geração. No entanto, ele também informa todos os outros cromossomos, distintos da resposta do algoritmo, que obtiveram o mesmo resultado ao longo do processo. Para o volume alto, foram encontrados outros quatro com o mesmo desempenho da resposta do algoritmo (para a semente 1, a qual forneceu o melhor resultado) (Tabela 7.2). Este resultado revela que ao longo dos patamares de valores idênticos de  $S(x)$ , observados nos gráficos da evolução dos AGs, a solução não se resume obrigatoriamente a um único cromossomo, transferido de uma geração para a outra pelo elitismo.

Tabela 7.2 – Outros Cromossomos que Apresentaram o Mesmo Resultado da Melhor Resposta do Algoritmo para Volume Alto

Cromossomos com Desempenho Igual ao do Melhor																										
1	1	0	2	4	1	2	2	2	0	2	1	2	0	0	1	1	0	1	1	0	1	1	1	1	0	2
1	1	0	2	4	1	2	2	2	0	2	1	2	0	0	1	1	0	1	1	1	1	1	1	1	0	2
1	1	0	2	4	1	2	2	1	0	2	1	2	0	0	1	1	0	1	1	0	1	1	1	1	0	2
1	1	0	2	4	1	2	2	1	0	2	1	2	0	0	1	1	0	1	1	2	1	1	1	1	0	2

*f) Atrasos excluídos*

Em nenhuma das simulações houve atraso excluído, isto é, todos ficaram dentro do intervalo  $X + 2S$ , em que  $X$  é a média e  $S$  o desvio-padrão.

### 7.1.2. Volume Baixo

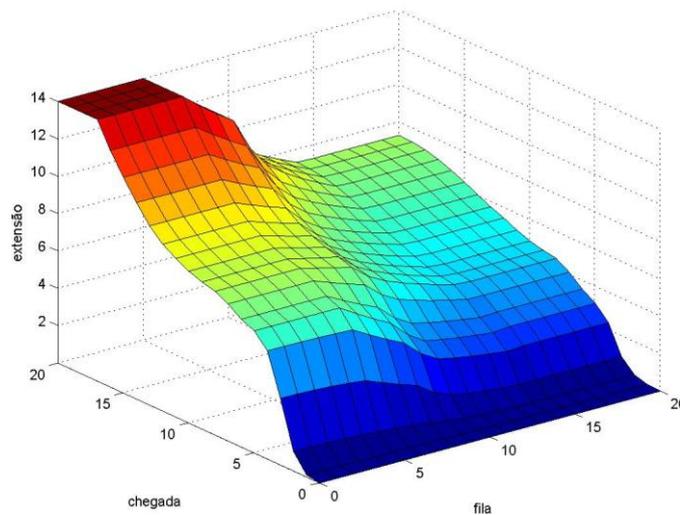
#### a) Cromossomos que Forneceram os Melhores Resultados

A Tabela 7.3 apresenta os cromossomos que apresentaram o melhor resultado em cada uma das 5 execuções do algoritmo para o volume baixo, fornecidos ao final da Etapa 2, com seus respectivos atrasos, penalizações e somas (atraso + penalização). A Figura 7.6 ilustra as superfícies de controle de cada um destes cromossomos.

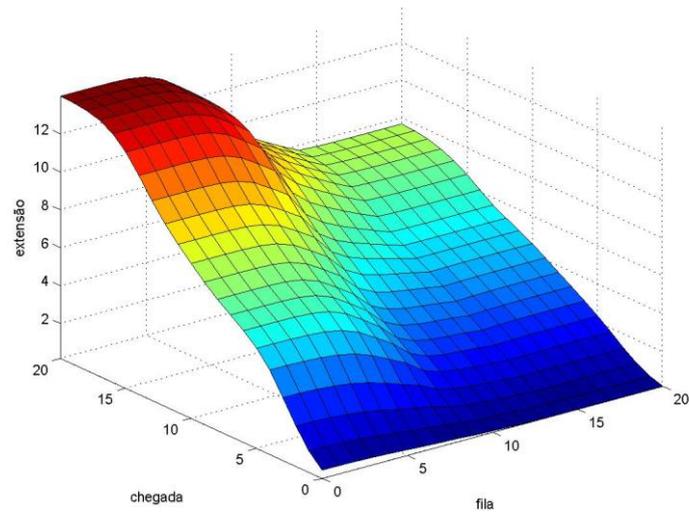
Tabela 7.3 - Cromossomos que Apresentaram os Melhores Resultados para Volume Baixo

Semente	Cromossomo com Melhor Resultado (Volume Baixo)																Atraso (s)	Penalização	Soma (A+P)											
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16														
1	1	1	0	0	4	2	2	0	0	0	2	0	1	2	0	1	2	2	2	2	0	2	2	1	2	0	2	7,5638	2,3680	9,9318
2	1	1	0	1	4	2	1	2	2	0	2	2	2	2	2	2	2	2	2	2	0	2	0	1	0	0	2	7,6311	2,2770	9,9081
3	1	1	0	1	4	2	0	0	2	0	2	2	2	1	2	1	1	2	2	0	2	1	2	1	2	0	2	7,6302	2,3620	9,9922
4	1	1	1	0	4	2	2	0	1	0	2	0	1	2	0	1	2	2	2	0	1	0	2	1	2	0	2	7,5638	2,3680	9,9318
5	1	1	1	0	4	2	2	0	0	0	2	0	1	2	0	1	2	2	2	0	2	2	2	1	2	0	2	7,5638	2,3680	9,9318

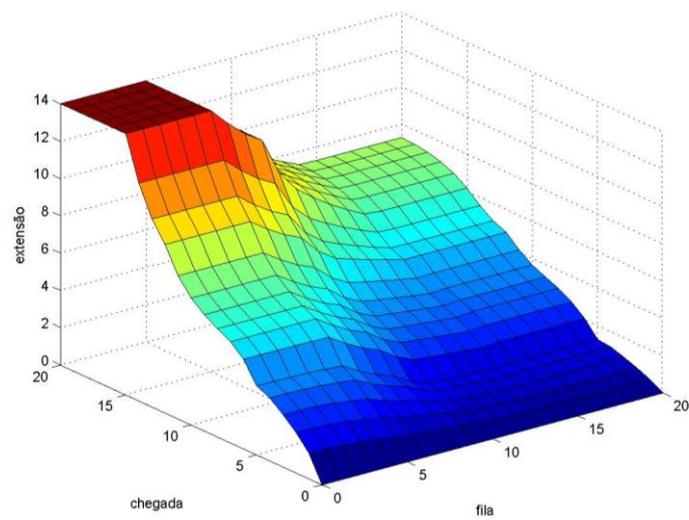
#### b) Superfícies de Controle e Penalizações dos Cromossomos de Melhores Resultados



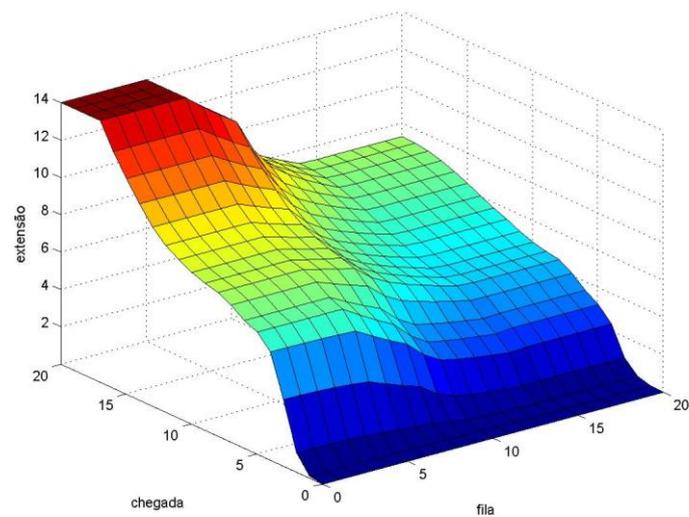
(a) Superfície do melhor cromossomo da semente 1 (Penalização = 2,3680)



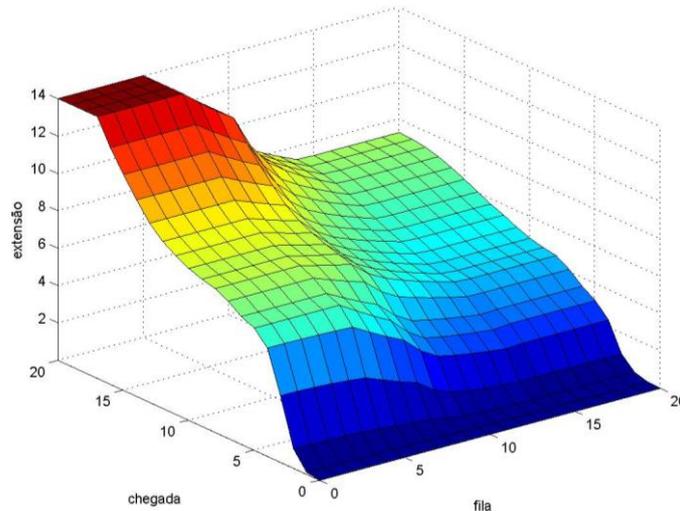
(b) Superfície do melhor cromossomo da semente 2 (Penalização = 2,2770)



(c) Superfície do melhor cromossomo da semente 3 (Penalização = 2,3620)



(d) Superfície do melhor cromossomo da semente 4 (Penalização = 2,3680)



(e) Superfície do melhor cromossomo da semente 5 (Penalização = 2,3680)

Figura 7.6 – Superfícies de Controle dos Cromossomos com Melhores Resultados para Volume Baixo com suas Respectivas Penalizações

### c) Características do Melhor Controlador Obtido

O melhor resultado obtido para o volume baixo foi o do melhor cromossomo da semente 2, com soma  $S(x) = 9,9081$ . Nesse caso, a menor soma não correspondeu ao menor atraso, que foi obtido com os controladores associados às sementes 1, 4 e 5. Esses últimos controladores, embora difiram entre si em um ou mais componentes, produziram superfícies e, conseqüentemente, atrasos idênticos. Mesmo neste caso, a diferença entre o atraso produzido pelo controlador selecionado e o atraso mínimo é bastante reduzida, em torno de 7 centésimos de segundo.

O controlador semafórico *fuzzy* representado pelo cromossomo da semente 2 tem as seguintes características:

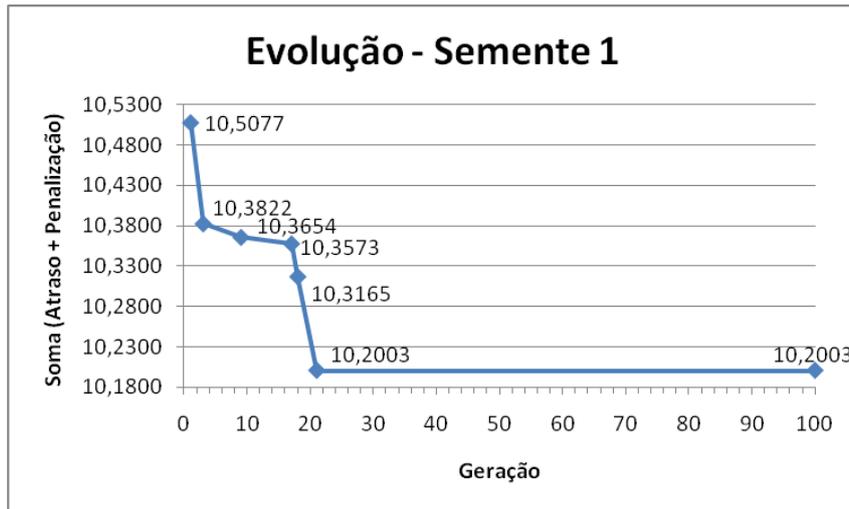
- tipo do controlador = **Sugeno**;
- modo de implementação do operador “E” = **Prod**;
- método de defuzificação = **Wtaver**;
- tipo de função de pertinência para Fila “Pequena” = **Z**;
- variação dos limites do conjunto de Fila “Pequena” = **0**;
- tipo de função de pertinência para Fila “Média” = **Gaussiana**;
- variação dos limites do conjunto de Fila “Média” = **+2**;
- tipo de função de pertinência para Fila “Longa” = **Trapezoidal**;

- variação dos limites do conjunto Fila “Longa” = +2;
- tipo de função de pertinência para Chegada “Zero” = **Z**;
- variação dos limites do conjunto Chegada “Zero” = +2;
- tipo de função de pertinência para Chegada “Pouca” = **Gaussiana**;
- variação dos limites do conjunto Chegada “Pouca” = +2;
- tipo de função de pertinência para Chegada “Média” = **Gaussiana**;
- variação dos limites do conjunto Chegada “Média” = +2;
- tipo de função de pertinência para Chegada “Muita” = **S**;
- variação dos limites do conjunto Chegada “Muita” = +2;
- tipo de função de pertinência para Extensão “Zero” = **Z**;
- variação dos limites do conjunto Extensão “Zero” = -2;
- tipo de função de pertinência para Extensão “Curta” = **Gaussiana**;
- variação dos limites do conjunto Extensão “Curta” = **0**;
- tipo de função de pertinência para Extensão “Média” = **Trapezoidal**;
- variação dos limites do conjunto Extensão “Média” = -2;
- tipo de função de pertinência para Extensão “Longa” = **Trapezoidal**;
- variação dos limites do conjunto Extensão “Longa” = +2.

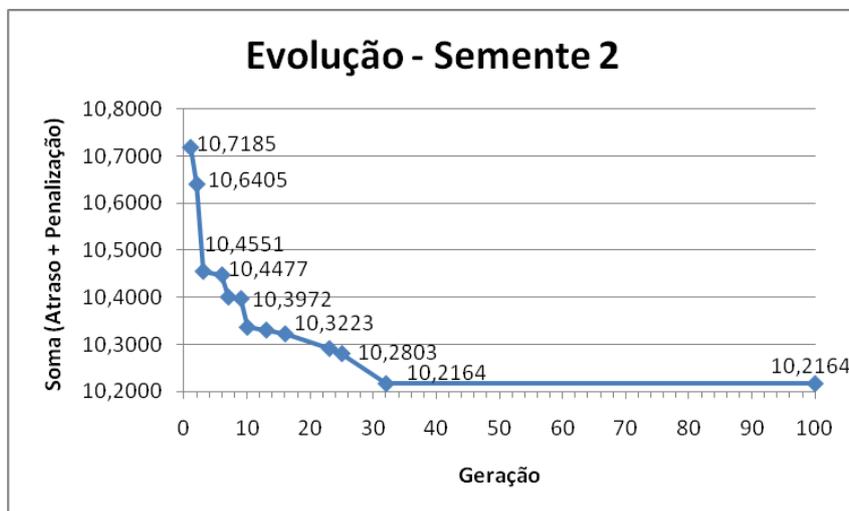
Como o controlador acima é do tipo Sugeno, as variáveis “modo de implementação da função de implicação” e “modo de implementação da agregação” não são consideradas.

#### *d) Evolução do AG*

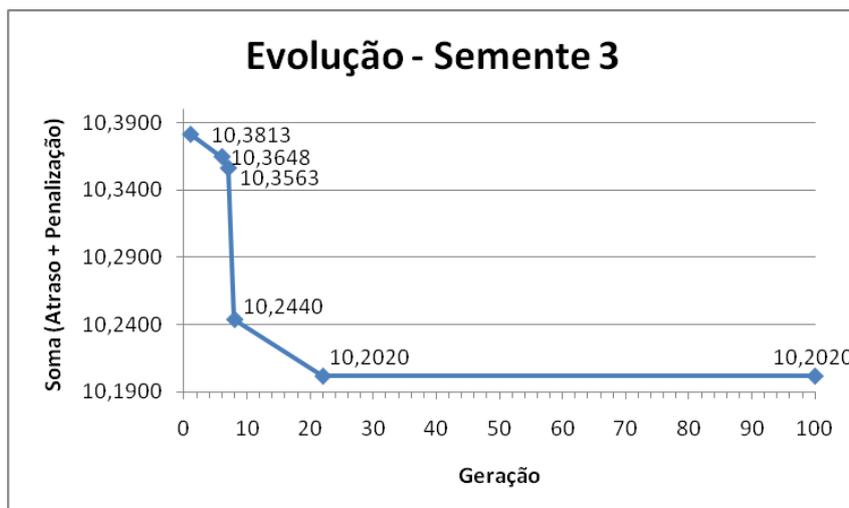
A Figura 7.7 apresenta a evolução do AG, para cada semente, ao longo das cem gerações da Etapa 1, enquanto que a Figura 7.8 agrega essas evoluções em um único gráfico. Da mesma maneira, a Figura 7.19 ilustra a evolução do AG, para cada semente, na Etapa 2, e a Figura 7.10 agrega essas evoluções em um único gráfico.



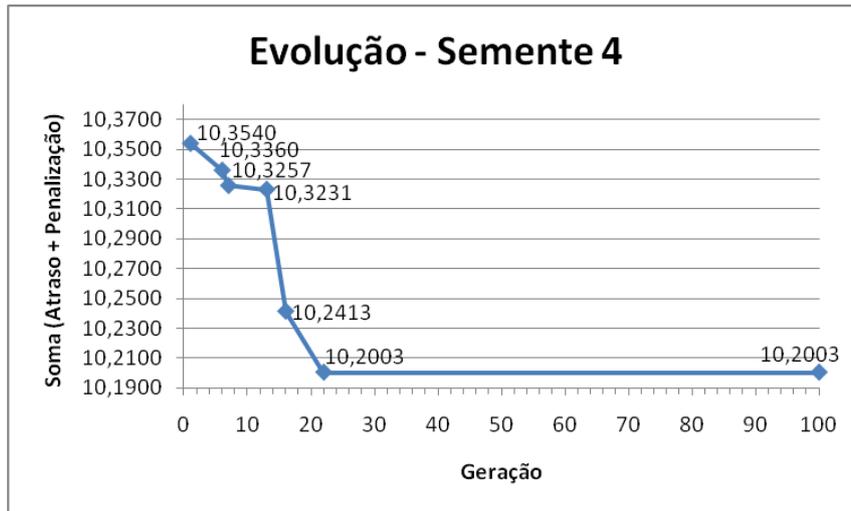
(a) Evolução do AG para semente 1 (Etapa 1)



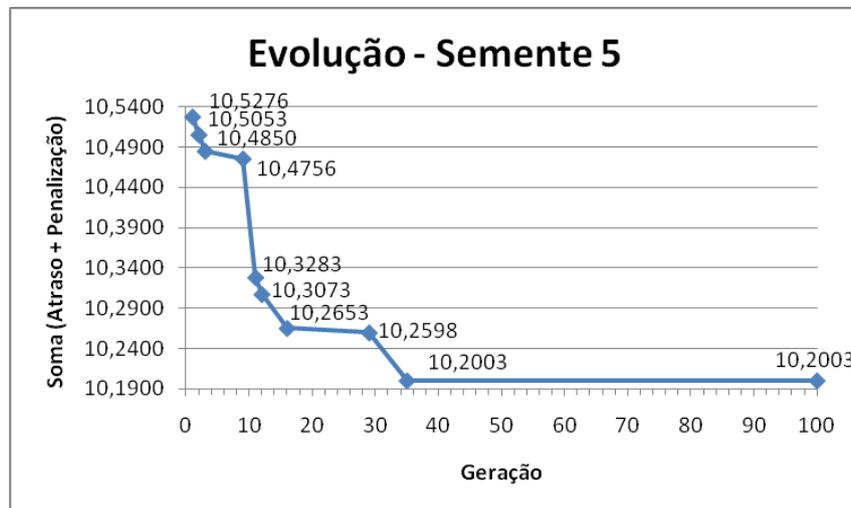
(b) Evolução do AG para semente 2 (Etapa 1)



(c) Evolução do AG para semente 3 (Etapa 1)



(d) Evolução do AG para semente 4 (Etapa 1)



(e) Evolução do AG para semente 5 (Etapa 1)

Figura 7.7 – Evolução do AG para Cada Semente da Etapa 1 (Volume Baixo)

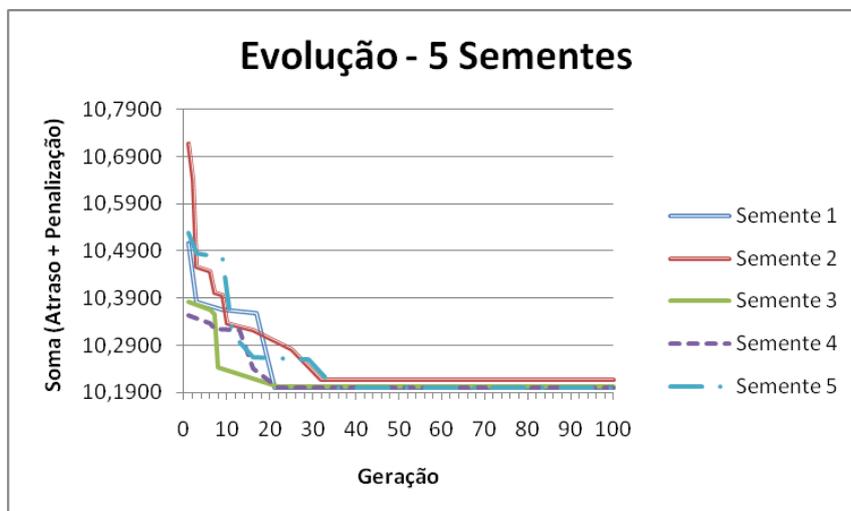
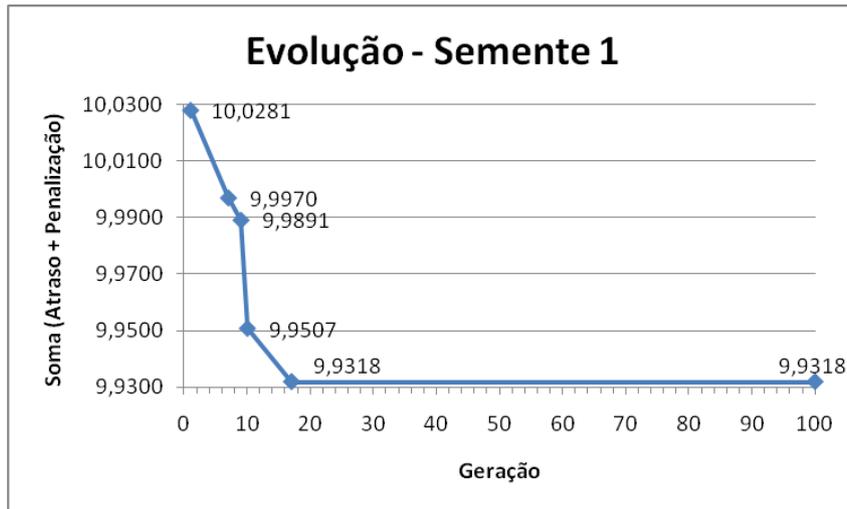
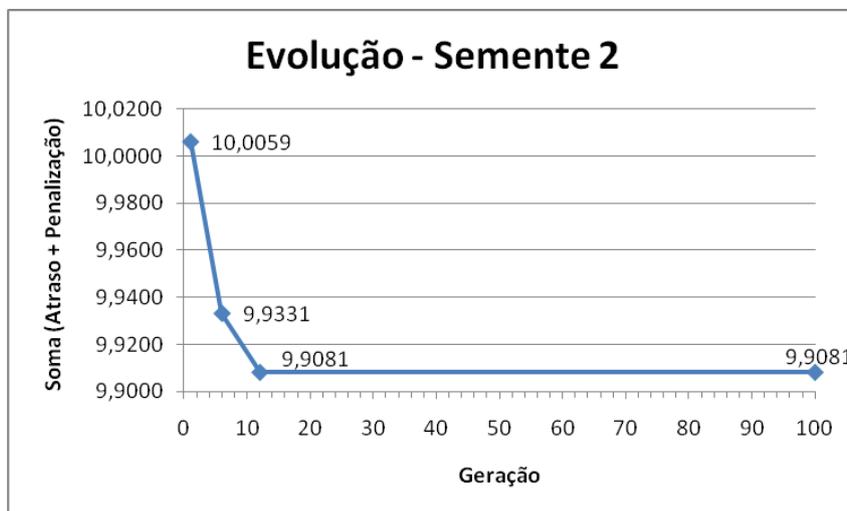


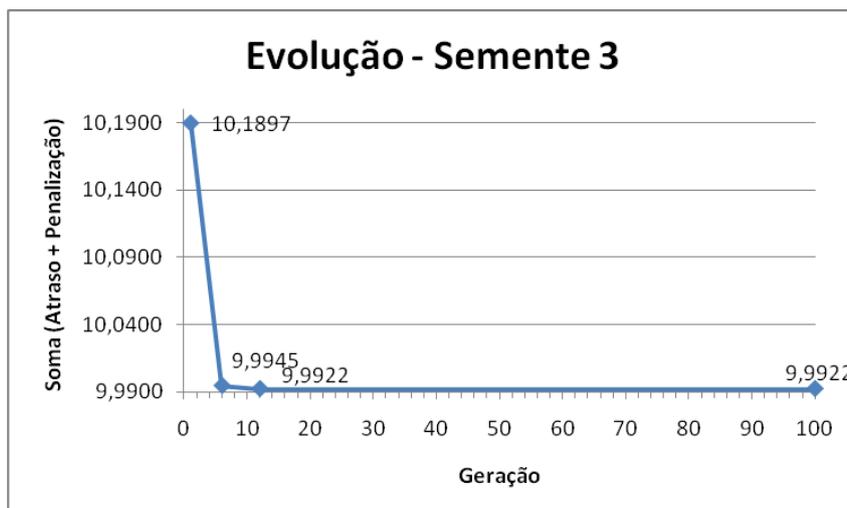
Figura 7.8 – Evolução do AG em Todas as Sementes da Etapa 1 (Volume Baixo)



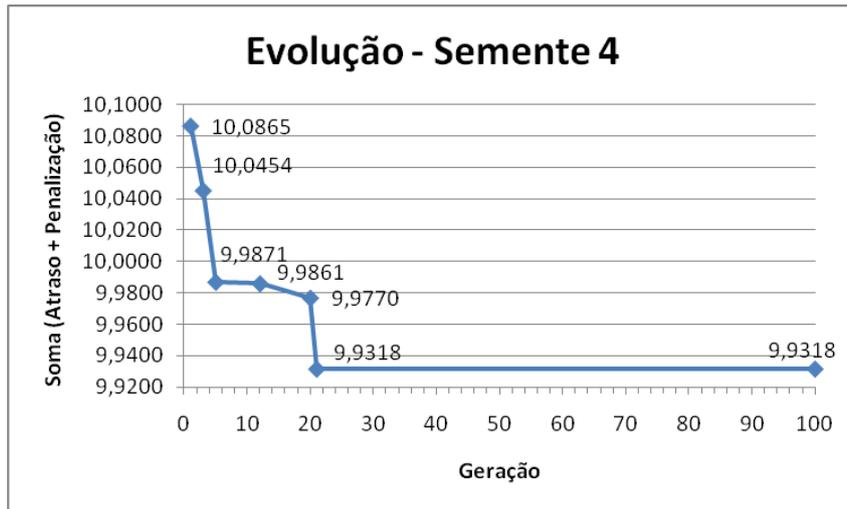
(a) Evolução do AG para semente 1 (Etapa 2)



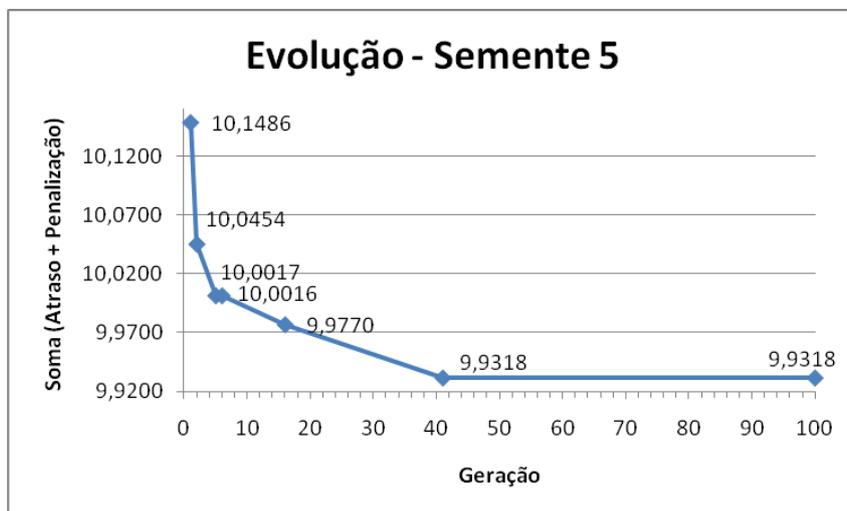
(b) Evolução do AG para semente 2 (Etapa 2)



(c) Evolução do AG para semente 3 (Etapa 2)



(d) Evolução do AG para semente 4 (Etapa 2)



(e) Evolução do AG para semente 5 (Etapa 2)

Figura 7.9 – Evolução do AG para Cada Semente da Etapa 2 (Volume Baixo)

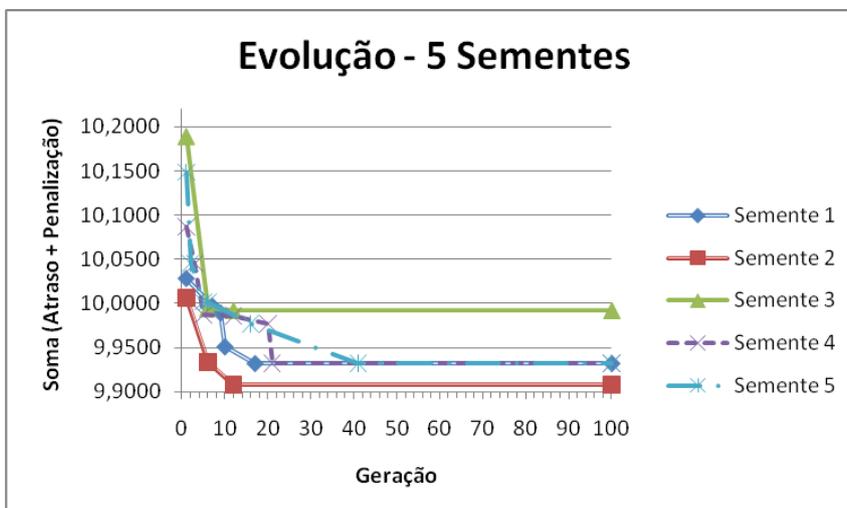


Figura 7.10 – Evolução do AG em Todas as Sementes da Etapa 2 (Volume Baixo)

e) Cromossomos com mesmo desempenho dos melhores

Para o volume baixo, foram encontrados outros dezesseis controladores com o mesmo desempenho da resposta do algoritmo (para a semente 2, a qual forneceu o melhor resultado), os quais são apresentados na Tabela 7.4.

Tabela 7.4 – Outros Cromossomos que Apresentaram o Mesmo Resultado da Melhor Resposta do Algoritmo para Volume Baixo

Cromossomos com Desempenho Igual ao do Melhor																											
1	1	0	1	4	2	1	2	0	0	2	2	2	2	2	2	2	2	2	2	1	2	1	1	0	0	2	
1	1	0	1	4	2	1	2	0	0	2	2	2	2	2	2	2	2	2	2	2	1	1	0	0	2		
1	1	0	1	4	2	1	2	1	0	2	2	2	2	2	2	2	2	2	2	2	1	1	0	0	2		
1	1	0	1	4	2	1	2	0	0	2	2	2	2	2	2	2	2	2	2	0	2	1	1	0	0	2	
1	1	0	1	4	2	1	2	0	0	2	2	2	2	2	2	2	2	2	2	1	2	0	1	0	0	2	
1	1	0	1	4	2	1	2	0	0	2	2	2	2	2	2	2	2	2	2	2	0	1	0	0	2		
1	1	0	1	4	2	1	2	1	0	2	2	2	2	2	2	2	2	2	2	0	2	1	1	0	0	2	
1	1	0	1	4	2	1	2	2	0	2	2	2	2	2	2	2	2	2	2	2	1	1	0	0	2		
1	1	0	1	4	2	1	2	1	0	2	2	2	2	2	2	2	2	2	2	1	2	1	1	0	0	2	
1	1	0	1	4	2	1	2	2	0	2	2	2	2	2	2	2	2	2	2	0	2	1	1	0	0	2	
1	1	0	1	4	2	1	2	2	0	2	2	2	2	2	2	2	2	2	2	1	2	1	1	0	0	2	
1	1	0	1	4	2	1	2	1	0	2	2	2	2	2	2	2	2	2	2	2	0	1	0	0	2		
1	1	0	1	4	2	1	2	0	0	2	2	2	2	2	2	2	2	2	2	0	2	0	1	0	0	2	
1	1	0	1	4	2	1	2	1	0	2	2	2	2	2	2	2	2	2	2	0	2	0	1	0	0	2	
1	1	0	1	4	2	1	2	1	0	2	2	2	2	2	2	2	2	2	2	1	2	0	1	0	0	2	
1	1	0	1	4	2	1	2	2	0	2	2	2	2	2	2	2	2	2	2	1	2	0	1	0	0	2	

f) Atrasos excluídos

Novamente, em nenhuma das simulações houve atraso excluído, isto é, todos ficaram dentro do intervalo  $X + 2S$ , em que  $X$  é a média e  $S$  o desvio-padrão.

**7.2. Análise dos Resultados**

7.2.1. Volume Alto

a) Comparação Estatística entre Controladores

Para que se pudesse realizar a comparação do controlador utilizado por Andrade (2009) com o melhor controlador encontrado para o volume alto, foi feito um teste de hipóteses para verificar se os valores obtidos para o atraso médio por veículo nos dois

controladores eram estatisticamente diferentes. O teste utilizado foi *t de Student*, com nível de significância igual a 0,05, e levou em conta 35 elementos (atrasos obtidos em 35 novas simulações realizadas para cada controlador, com as mesmas sementes, mostrados no Apêndice A).

Nesta aplicação, foram obtidos os seguintes resultados para o atraso médio por veículo:

- melhor controlador selecionado pelo AG proposto: 23,34 segundos/veículo;
- controlador utilizado por Andrade (2009): 39,51 segundos/veículo.

A hipótese de que os dados eram estatisticamente iguais foi rejeitada e, portanto, a comparação pôde ser realizada e mostrou que a redução média de 40,9% do atraso médio na interseção, quando o melhor controlador encontrado para o volume alto foi comparado ao utilizado por Andrade (2009), é estatisticamente significativa para  $\alpha=5\%$ .

Também houve uma pequena melhora na comparação das superfícies de controle. A do controlador de Andrade (2009) (Figura 7.11), quando avaliada pelo método proposto no presente trabalho, obteve penalização de 2,5780, enquanto que a superfície do melhor controlador para volume alto deste estudo obteve um penalização de 2,5470 ( redução de 1,2%).

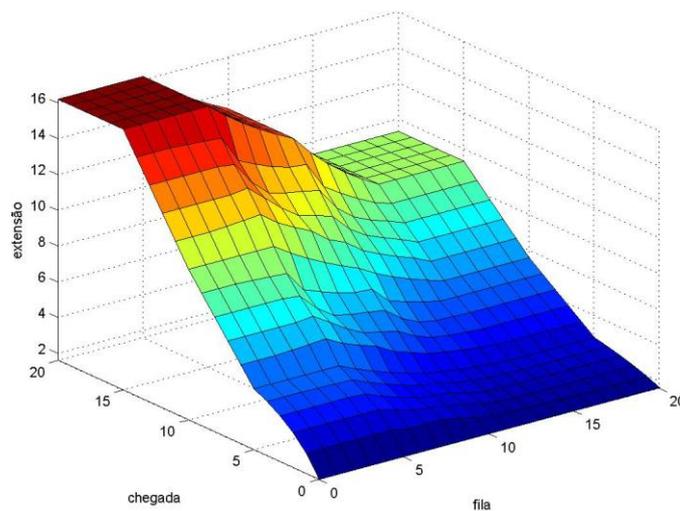


Figura 7.11 – Superfície do Controlador Utilizado por Andrade (2009)

A Tabela 7.5 compara as características do melhor controlador para volume alto desta dissertação com as do controlador utilizado por Andrade (2009).

Tabela 7.5 - Características do melhor controlador encontrado para volume alto e do controlador utilizado por Andrade (2009)

Variável	Melhor Controlador Desta Dissertação (Volume Alto)	Controlador Utilizado por Andrade (2009)	
Tipo do Controlador	Sugeno	Sugeno	
Modo de Implementação do Operador “E”	<i>Prod</i>	<i>Min</i>	
Modo de Implementação da Função de Implicação	Não se aplica	Não se aplica	
Modo de Implementação da Agregação	Não se aplica	Não se aplica	
Método de Defuzificação	<i>Wtaver</i>	<i>Wtaver</i>	
Tipo de Função de Pertinência –Fila	Pequena	Trapezoidal	Triangular
	Média	Gaussiana	Triangular
	Longa	Trapezoidal	Trapezoidal
	Qualquer	Trapezoidal	Trapezoidal
Tipo de Função de Pertinência –Chegada	Zero	Trapezoidal	Triangular
	Pouca	Triangular	Triangular
	Média	Trapezoidal	Triangular
	Muita	Trapezoidal	Trapezoidal
	Qualquer	Trapezoidal	Trapezoidal
Tipo de Função de Pertinência –Extensão	Zero	Trapezoidal	Triangular
	Curta	Trapezoidal	Triangular
	Média	Trapezoidal	Triangular
	Longa	Trapezoidal	Trapezoidal
Limites dos Conjuntos <i>Fuzzy</i> de Fila	Pequena	[0 12]	[0 10]
	Média	[3 17]	[5 15]
	Longa	[8 20]	[10 20]
	Qualquer	[0 20]	[0 20]
Limites dos Conjuntos <i>Fuzzy</i> de Chegada	Zero	[0 7]	[0 5]
	Pouca	[0 8]	[0 10]
	Média	[5 15]	[5 15]
	Muita	[10 20]	[10 20]
	Qualquer	[0 20]	[0 20]
Limites dos Conjuntos <i>Fuzzy</i> de Extensão	Zero	[0 7]	[0 5]
	Curta	[0 10]	[0 10]
	Média	[5 15]	[5 15]
	Longa	[8 20]	[10 20]

### b) Evolução do AG

Nas Figuras 7.2 a 7.5, percebe-se que o algoritmo converge com maior rapidez nas primeiras gerações. A Tabela 7.6 divide, para as Etapas 1 e 2 do controlador selecionado, as 100 gerações em intervalos que possuem mesmas probabilidades de escolha dos operadores de recombinação e mutação, e apresenta qual a porcentagem da redução da soma do atraso com a penalização para cada intervalo.

Tabela 7.6 – Redução Média da Soma, em porcentagem, para Intervalos de Gerações (Volume Alto)

Intervalo de Gerações	Probabilidade de Recombinação	Probabilidade de Mutação	Redução Média	
			Etapa 1	Etapa 2
1 - 10	0,70	0,30	2,24%	1,50%
11 - 20	0,65	0,35	0,48%	0,00%
21 - 30	0,60	0,40	0,26%	0,04%
31 - 40	0,55	0,45	0,25%	0,00%
41 - 50	0,50	0,50	0,32%	0,00%
51 - 60	0,45	0,55	0,53%	0,00%
61 - 100	0,40	0,60	0,08%	0,04%

Tanto na Etapa 1 quanto na Etapa 2, o intervalo em que houve a maior redução foi o das gerações 1 a 10, no qual a probabilidade de escolha do operador recombinação é igual a 0,7 e a do operador mutação é 0,3. Tal informação pode ser útil para futuros trabalhos que desejem utilizar probabilidades de escolha dos operadores recombinação e mutação fixas ao longo das gerações.

Percebe-se ainda, nas Figuras 7.2 a 7.5, que a convergência acontece mais rapidamente nas execuções do AG na Etapa 2. Embora não se possa afirmar conclusivamente, acredita-se que três fatores contribuem para a ocorrência de tal fato. O primeiro é que já houve uma convergência prévia na Etapa 1, e a segunda etapa seria uma espécie de ajuste fino. O segundo é que nesta etapa o número de variáveis trabalhadas é menor do que na Etapa 1. E por último, o fato de que as variáveis trabalhadas na Etapa 2 são todas do mesmo tipo, isto é, variação de limites dos conjuntos *fuzzy*, enquanto que na Etapa 1, são trabalhadas variáveis de diferentes características.

A partir das Figuras 7.3 e 7.5 é possível verificar que o resultado dos AGs não depende do valor do melhor desempenho da população inicial. Isto é, populações iniciais com

menores  $S(x)$  nem sempre resultaram em populações com melhor desempenho ao final das 100 gerações, especialmente no caso da Etapa 1.

*c) Cromossomos com mesmo desempenho dos melhores*

Considerando-se o fato do número de controladores semelhantes aos melhores ser pequeno e sabendo-se que o número de cromossomos distintos avaliados pelo algoritmo pode chegar a 20.000 em cada execução (duas etapas de 100x100 indivíduos cada), percebe-se que a probabilidade de um especialista definir manualmente um controlador com desempenho inferior à resposta do algoritmo genético é muito grande. Reforça-se, portanto, a justificativa do uso do AG neste estudo.

*d) Atrasos excluídos*

O fato de não haver nenhum atraso excluído ao longo das simulações dos cromossomos já era esperado, pois o número de simulações de cada cromossomo para se aferir seu atraso médio era baixo (três). No entanto, quando o melhor controlador obtido foi simulado novamente, 35 vezes, com sementes distintas, pode-se constatar que somente um valor deveria ser desconsiderado, isto é, 2,9% do total. Este baixo percentual vai ao encontro do que foi apontado nos estudos de Andrade (2009) e Labanca (2009), isto é, a confiabilidade dos resultados obtidos no simulador utilizado.

### *7.2.2. Volume Baixo*

*a) Comparação Estatística entre Controladores*

Assim como foi feito para o volume alto, realizou-se um teste de hipóteses para verificar se os valores obtidos no controlador utilizado por Andrade (2009) e no melhor controlador encontrado para o volume baixo eram estatisticamente diferentes. Novamente, o teste utilizado foi *t de Student*, com nível de significância igual a 0,05, para 35 elementos (ver Apêndice A). A hipótese de que os dados eram estatisticamente iguais foi rejeitada e, portanto, a comparação pôde ser realizada.

Os atrasos obtidos para os dois controladores foram:

- melhor controlador selecionado pelo AG proposto: 7,91 segundos/veículo;

- controlador utilizado por Andrade (2009): 9,84 segundos/veículo.

Constata-se, portanto, uma redução média de 19,6% do atraso médio na interseção, quando o melhor controlador encontrado para o volume alto foi comparado ao utilizado por Andrade (2009), e esta diferença é estatisticamente significativa para  $\alpha=5\%$ . Em relação às superfícies de controle, aquela fornecida pelo melhor controlador para volume baixo obteve penalização de 2,2770, enquanto que a do controlador de Andrade (2009) (Figura 7.11), conforme já mencionado, obteve penalização de 2,5780, quando avaliada pelo método proposto no presente trabalho. Percebe-se assim uma redução de 11,7% da penalização e, portanto, uma melhora na superfície de controle.

A Tabela 7.7 compara as características do melhor controlador para volume baixo desta dissertação com as do controlador utilizado por Andrade (2009).

Tabela 7.7 - Características do melhor controlador encontrado para volume baixo e do controlador utilizado por Andrade (2009)

<b>Variável</b>	<b>Melhor Controlador Desta Dissertação (Volume Baixo)</b>	<b>Controlador Utilizado por Andrade (2009)</b>	
Tipo do Controlador	Sugeno	Sugeno	
Modo de Implementação do Operador “E”	<i>Prod</i>	<i>Min</i>	
Modo de Implementação da Função de Implicação	Não se aplica	Não se aplica	
Modo de Implementação da Agregação	Não se aplica	Não se aplica	
Método de Defuzificação	<i>Wtaver</i>	<i>Wtaver</i>	
Tipo de Função de Pertinência -Fila	Pequena	Z	Triangular
	Média	Gaussiana	Triangular
	Longa	Trapezoidal	Trapezoidal
	Qualquer	Trapezoidal	Trapezoidal
Tipo de Função de Pertinência -Chegada	Zero	Z	Triangular
	Pouca	Gaussiana	Triangular
	Média	Gaussiana	Triangular
	Muita	S	Trapezoidal
	Qualquer	Trapezoidal	Trapezoidal
Tipo de Função de Pertinência -Extensão	Zero	Z	Triangular
	Curta	Gaussiana	Triangular
	Média	Trapezoidal	Triangular
	Longa	Trapezoidal	Trapezoidal
Limites dos Conjuntos <i>Fuzzy</i> de <i>Fila</i>	Pequena	[0 10]	[0 10]
	Média	[3 17]	[5 15]
	Longa	[8 20]	[10 20]
	Qualquer	[0 20]	[0 20]
Limites dos Conjuntos <i>Fuzzy</i> de <i>Chegada</i>	Zero	[0 7]	[0 5]
	Pouca	[0 12]	[0 10]
	Média	[3 17]	[5 15]
	Muita	[8 20]	[10 20]
	Qualquer	[0 20]	[0 20]
Limites dos Conjuntos <i>Fuzzy</i> de <i>Extensão</i>	Zero	[0 3]	[0 5]
	Curta	[0 10]	[0 10]
	Média	[7 13]	[5 15]
	Longa	[8 20]	[10 20]

b) *Evolução do AG*

Nas Figuras 7.7 a 7.10, percebe-se mais uma vez que o algoritmo converge com maior rapidez nas primeiras gerações. A Tabela 7.8 divide, para o controlador selecionado, para as Etapas 1 e 2, as 100 gerações em intervalos que possuem mesmas probabilidades de escolha dos operadores de recombinação e mutação, e apresenta qual a porcentagem da redução da soma do atraso com a penalização para cada intervalo.

Tabela 7.8 – Redução Média da Soma, em porcentagem, para Intervalos de Gerações (Volume Baixo)

Intervalo de Gerações	Probabilidade de Recombinação	Probabilidade de Mutação	Redução Média	
			Etapa 1	Etapa 2
1 - 10	0,70	0,30	1,40%	1,17%
11 - 20	0,65	0,35	0,69%	0,16%
21 - 30	0,60	0,40	0,48%	0,09%
31 - 40	0,55	0,45	0,24%	0,00%
41 - 50	0,50	0,50	0,00%	0,09%
51 - 60	0,45	0,55	0,00%	0,00%
61 - 100	0,40	0,60	0,00%	0,00%

Novamente, tanto na Etapa 1 quanto na Etapa 2, o intervalo em que houve a maior redução foi o das gerações 1 a 10, no qual a probabilidade de escolha do operador recombinação é igual a 0,7 e a do operador mutação é 0,3. Informação esta que pode ser útil para futuros trabalhos que desejem utilizar probabilidades de escolha dos operadores recombinação e mutação fixas ao longo das gerações.

Assim como no volume alto, pode-se notar, nas Figuras 7.7 a 7.10, que a convergência acontece mais rapidamente nas execuções do AG na Etapa 2. Conforme já exposto, embora não se possa afirmar conclusivamente, acredita-se que três fatores contribuem para a ocorrência de tal fato: convergência prévia, ocorrida na Etapa 1; menor número de variáveis trabalhadas na Etapa 2; e somente variáveis do mesmo tipo, isto é, variação de limites dos conjuntos *fuzzy*, sendo trabalhadas na Etapa 2.

Do mesmo modo que o observado para o caso do volume alto, as Figuras 7.8 e 7.10 mostram que o resultado do AG ao final das 100 gerações não é diretamente afetado pelo desempenho da população inicial.

### *c) Cromossomos com mesmo desempenho dos melhores*

O fato de que foram encontrados outros dezesseis controladores com o mesmo desempenho da resposta do algoritmo para a semente 2 (a qual forneceu o melhor resultado) confirmam a complexidade de se definir o melhor controlador diretamente, com base em processos de tentativa e erro. Isto é, ao se verificar que diferentes controladores podem apresentar o mesmo resultado (neste caso o melhor resultado), não se pode descartar a possibilidade de existência de outro conjunto de controladores, com resultados inferiores, que também apresentem o mesmo desempenho entre si. Esse tipo de situação poderia induzir o projetista a definir um controlador com desempenho aquém do que seria definido por um AG.

Cita-se novamente o que foi analisado no volume alto. Considerando-se o fato de que o número de controladores semelhantes aos melhores ser pequeno e sabendo-se que o número de cromossomos distintos avaliados pelo algoritmo pode chegar a 20.000 em cada execução (duas etapas de 100x100 indivíduos), percebe-se que a probabilidade de um especialista definir manualmente um controlador com desempenho inferior a um encontrado pelo algoritmo genético é muito grande. Reforça-se, portanto, a justificativa do uso do AG neste estudo.

### *d) Atrasos excluídos*

Nas 35 novas simulações do melhor controlador encontrado para o volume baixo, verificou-se que somente dois valores deveriam ser desconsiderados, isto é, 5,7% do total. Mais uma vez, este valor percentual vai ao encontro do que foi indicado nos estudos de Andrade (2009) e Labanca (2009), no que se refere à confiabilidade dos resultados obtidos no simulador utilizado.

## **7.3. Comparação Entre os Melhores Controladores de Cada Nível de Volume**

A Tabela 7.9 compara as características do melhor controlador para volume alto com as do melhor controlador para volume baixo. Nela, pode-se perceber que os controladores são diferentes, o que sugere que as condições do tráfego realmente influenciaram na resposta do algoritmo, conforme esperado, e que os controladores definidos com a evolução deste refletem as especificidades locais do tráfego.

Tabela 7.9 – Características do melhor controlador encontrado para volume baixo e do melhor controlador encontrado para volume alto.

Variável		Melhor Controlador - Volume Alto	Melhor Controlador - Volume Baixo
Tipo do Controlador		Sugeno	Sugeno
Modo de Implementação do Operador “E”		<i>Prod</i>	<i>Prod</i>
Modo de Implementação da Função de Implicação		Não se aplica	Não se aplica
Modo de Implementação da Agregação		Não se aplica	Não se aplica
Método de Defuzificação		<i>Wtaver</i>	<i>Wtaver</i>
Tipo de Função de Pertinência - Fila	Pequena	Trapezoidal	Z
	Média	Gaussiana	Gaussiana
	Longa	Trapezoidal	Trapezoidal
	Qualquer	Trapezoidal	Trapezoidal
Tipo de Função de Pertinência - Chegada	Zero	Trapezoidal	Z
	Pouca	Triangular	Gaussiana
	Média	Trapezoidal	Gaussiana
	Muita	Trapezoidal	S
	Qualquer	Trapezoidal	Trapezoidal
Tipo de Função de Pertinência - Extensão	Zero	Trapezoidal	Z
	Curta	Trapezoidal	Gaussiana
	Média	Trapezoidal	Trapezoidal
	Longa	Trapezoidal	Trapezoidal
Limites dos Conjuntos <i>Fuzzy</i> de Fila	Pequena	[0 12]	[0 10]
	Média	[3 17]	[3 17]
	Longa	[8 20]	[8 20]
	Qualquer	[0 20]	[0 20]
Limites dos Conjuntos <i>Fuzzy</i> de Chegada	Zero	[0 7]	[0 7]
	Pouca	[0 8]	[0 12]
	Média	[5 15]	[3 17]
	Muita	[10 20]	[8 20]
	Qualquer	[0 20]	[0 20]
Limites dos Conjuntos <i>Fuzzy</i> de Extensão	Zero	[0 7]	[0 3]
	Curta	[0 10]	[0 10]
	Média	[5 15]	[7 13]
	Longa	[8 20]	[8 20]

## 8. CONCLUSÕES E RECOMENDAÇÕES

Diversos trabalhos sobre controle de tráfego em interseções têm mostrado vantagens do uso do controlador semafórico *fuzzy* em relação ao do controlador por tempo fixo, e até mesmo do controlador atuado pelo princípio clássico da extensão. No entanto, a definição do modo de implementação dos componentes de um controlador semafórico *fuzzy* é uma dificuldade encontrada por seu projetista. Isto porque existem inúmeras possibilidades de se implementar esse controlador. Na literatura, percebe-se que grande parte dos controladores desenvolvidos é baseada no de Pappis e Mamdani (1977).

Estudos desenvolvidos no Programa de Pós-Graduação em Transportes da UnB, alguns em parceria com a *Helsinki University of Technology* (Finlândia), têm mostrado que diferentes formas de implementação dos componentes de um controlador semafórico *fuzzy* fornecem resultados distintos, o que justifica a adoção de um procedimento que contribua nessa escolha. Nesse contexto, esta dissertação apresenta um procedimento para a concepção de controladores semafóricos *fuzzy* que use algoritmo genético para definir um modo de implementação dos componentes desses controladores. Este procedimento foi descrito no Capítulo 6 e sua aplicação e avaliação foram apresentadas no Capítulo 7. Nas próximas seções são apresentadas conclusões, limitações e sugestões baseadas no trabalho realizado.

### 8.1. Cumprimento do Objetivo e Verificação da Hipótese

O objetivo do trabalho foi alcançado. O procedimento proposto, com uso de algoritmo genético, permitiu a definição de um modo de implementação dos componentes de um controlador semafórico *fuzzy* cujo desempenho de tráfego na interseção por ele controlada apresentou resultados satisfatórios: reduções no atraso médio da interseção de 40,9%, para volume alto, e 19,6%, para volume baixo, em relação ao controlador adotado em estudos do Programa de Pós-Graduação em Transportes da UnB. Esse controlador foi considerado como referência para esta pesquisa haja vista avaliações positivas da sua atuação apresentadas em diferentes trabalhos.

Além disso, as superfícies de controle fornecidas por estes controladores foram contínuas, suaves e coerentes, possibilitando uma interpretação correta pelo técnico e gerando um controle de tráfego que seja facilmente assimilado pelo motorista. Estas

superfícies, inclusive, foram melhor avaliadas do que a superfície de controle gerada pelo controlador com as quais foram comparadas.

Com isso, aceita-se a hipótese de que um procedimento para a concepção de controladores semafóricos *fuzzy* que use algoritmo genético permite a definição adequada de um modo de implementação dos componentes desses controladores, de modo a permitir que a atuação desse controlador reflita as necessidades específicas do tráfego a ser controlado, em termos de variável de desempenho de tráfego, e forneça uma superfície de controle contínua, suave e coerente.

## 8.2. Principais Conclusões

As principais conclusões obtidas com o desenvolvimento do estudo são:

- o algoritmo genético pode ser aplicado à situação proposta na dissertação. Todas as respostas do procedimento proposto foram satisfatórias, em termos da variável de desempenho de tráfego e da continuidade, suavidade e coerência da superfície de controle, sendo que os resultados obtidos foram melhores do que aqueles fornecidos pelo controlador utilizado como referência para esta pesquisa;
- como todas as respostas do algoritmo são satisfatórias e, inclusive, próximas, o projetista poderá ter a discricionariedade de escolher dentre os melhores controladores aquele que julgar mais apropriado à interseção a ser controlada. Por exemplo, como todos os  $S(x)$ , atrasos e penalizações são bons, em vez de priorizar o controlador de menor  $S(x)$ , poderá priorizar aquele de menor atraso (isto porque tem a garantia de que a superfície de controle também é boa);
- as características de desempenho das populações iniciais não permitem inferir sobre a qualidade dos resultados do AG. Isto é, populações iniciais com cromossomos de bom desempenho não garantem necessariamente os melhores resultados, e populações iniciais com cromossomos de desempenho ruim podem fornecer bons resultados. Exemplo disso é o caso da semente 5 da Etapa 1 para o nível de volume alto, a qual possuía o pior resultado na primeira geração, mas produziu o cromossomo de melhor desempenho na última geração;

- populações iniciais distintas podem fornecer o mesmo resultado, conforme ocorrido com as sementes 1, 4 e 5 da Etapa 1 para o volume baixo;
- a análise dos diagramas de evolução para os diferentes pares de taxas de probabilidade de escolha de recombinação e mutação mostrou que a maior redução média da soma do atraso médio com a penalização aconteceu para os valores de 0,7 e 0,3, respectivamente, tanto para volume alto quanto para volume baixo;
- todos os melhores controladores obtidos nas simulações das sementes são do tipo Sugeno, o que sugere sua vantagem em relação a controladores do tipo Mamdani. Informação esta que é útil, já que grande parte dos controladores encontrados na literatura é baseada no de Mamdani.

### **8.3. Limitações do Estudo**

Ao longo do trabalho, deparou-se com três limitações que merecem destaque. A primeira é que não se pôde realizar o número necessário de simulações por cromossomo para uma análise estatística mais confiável, devido ao longo tempo demandado para as simulações de todos os cromossomos. A análise mais apurada foi realizada somente com os melhores controladores de cada nível de volume.

Outra limitação é que o espaço de busca do estudo, em sua concepção original, era de ordem muito grande. Para que se realizasse uma busca satisfatória no mesmo, seria necessário que se avaliasse um número bem maior de cromossomos. Isto, contudo, demandaria um período de tempo muito maior. Assim, foi necessário reduzir o espaço de busca, com simplificações nos valores que as variáveis poderiam atingir. Acredita-se que estas duas limitações podem ser superadas com o uso de recursos computacionais mais avançados para a execução das simulações, ou uma otimização do tempo de simulação no simulador utilizado.

A terceira limitação do estudo refere-se aos parâmetros do algoritmo genético. A definição do tamanho da população, do número de gerações, das probabilidades de escolha dos operadores e das probabilidades de recombinação e mutação não tem na literatura um respaldo conclusivo que indique quais parâmetros são os melhores para cada situação. Com isso, apesar dos cuidados tomados nas escolhas deste trabalho, não

se garante que os parâmetros utilizados sejam os mais adequados. Esta limitação pode ser mitigada com estudos posteriores que tenham como objetivo calibrar esses parâmetros.

#### **8.4. Sugestões e Recomendações para Futuros Trabalhos**

Tendo em vista o que foi levantado e analisado neste estudo, considera-se oportuno sugerir que futuros trabalhos:

- realizem o mesmo procedimento contemplando alternativas diferentes para as mesmas variáveis de estudo analisadas neste trabalho, e, caso os recursos computacionais permitam, trabalhem com o espaço de busca proposto na concepção original da dissertação;
- analisem qual é a relação existente entre a combinação de parâmetros do AG e os resultados obtidos, de modo a permitir uma melhor calibragem do AG;
- utilizem o algoritmo genético para a definição da base de regras de um controlador semafórico *fuzzy*, buscando-se encontrar resultados melhores do que os já obtidos até então;
- incorporem outras variáveis de desempenho de tráfego na função aptidão, além do atraso médio veicular.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ABREU, V. C. (1999) Desenvolvimento de uma metodologia de distribuição de viagens com aplicação da lógica fuzzy. Dissertação de mestrado. Departamento de Engenharia Civil e Ambiental, Universidade de Brasília, Brasília, 139 p.
- AKIYAMA, T., NAKAMURA, K., SASAKI, T. (1993) Traffic diversion model on urban Express way by fuzzy reasoning. Selected Proceedings of the Sixth World Conference of Transport Research, Lyon 92, pp. 1011-1022.
- AKIYAMA, T., SHAO, C.-F. (1993) Fuzzy mathematical programming for traffic safety planning on an urban express way. *Transportation Planning and Technology* 17, 179-190.
- AKIYAMA, T., TSUBOI, H. (1996) Description of route choice behaviour by multi-stage fuzzy reasoning. Paper presented at the Highways to the Next Century Conference, Hong Kong.
- ANDRADE, M. (2004) Estudo para Aprimoramento da Operação de Controladores Semafóricos Fuzzy. Dissertação de Mestrado, Publicação T.DM-012 A/2004, Departamento de Engenharia Civil, Faculdade de Tecnologia, Universidade de Brasília, DF, 137p.
- ANDRADE, M. (2009) Estudos de elementos de controladores semafóricos *fuzzy* com aplicação para operação em rede aberta. Tese de Doutorado – Universidade de Brasília. Faculdade de Tecnologia. Departamento de Engenharia Civil e Ambiental. Brasília, DF, 238p.
- BIELLI, M., CARAMIA, M., CAROTENUTO, P. (2002) Genetic algorithms in bus network optimization.
- BUSCH, F., CREMER, M., GHIO, A., HENNINGER, T. (1995) A Muti-model approach with fuzzy reasoning for traffic state estimation and incident detection on motorways. In: ERTICO (Ed.), *Proceedings of the First World Congress on Applications of Transport Telematics and Intelligent Vehicle-Highway Systems. Towards an Intelligent Transport Systems*. Artech House, Boston and London, pp. 145-1252.
- CHAKROBORTHY, P., KIKUCHI, S. (1990) Application of fuzzy set theory to the analysis of capacity and level of service of highways. In: Ayyub, B.M. (Ed.), *Proceedings of ISUMA '90, The First International Symposium on Uncertainty Modeling and Analysis*. IEEE Computer Press, College Park, Maryland, pp. 146-150.
- CHAMBERS, L. D. (2001) *The practical handbook of genetic algorithms, applications*, 2ª ed: Chapman & Hall/CRC.
- CHIU, S. (1992) Adaptive Traffic Signal Control Using Fuzzy Logic. *Proceedings of the Intelligent Vehicle Symposium*. Detroit. p. 98-107.

- CZOGALLA, O. (1997) A Fuzzy Logic Application for Traffic Actuated Signal Control. IFAC Algorithms and Architectures in Real Time Control. Vilamoura. Portugal.
- DENATRAN (1984) Manual de Semáforos. Brasília: DENATRAN.
- FOGEL, L. J., OWENS, A. J., e WALSH, M. J. (1966) Artificial Intelligence through Simulated Evolution. Wiley.
- FAVILLA, J., A. MACHION e F. GOMIDE (1993) Fuzzy Traffic Control: Adaptive Strategies. In: Proceedings of the 2nd IEEE International Conference on Fuzzy Systems, Vol 1, San Francisco.
- GADÊLHA, J. C., C. F. G. LOUREIRO e H. C. JÚNIOR (1999) Aplicação de Lógica Fuzzy no Controle Semafórico de Interseções Isoladas. XIII ANPET – Congresso de Pesquisa e Ensino em Transportes. Anais – Artigos Científicos – Vol 1. p. 641-652. São Carlos, SP.
- GOLDBERG, D. E. (1989) Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley. United States of America.
- HAYKIN, S. (1999). Neural Networks: a comprehensive foundation. Practice-Hall, Inc. United States of America.
- HAYKIN, S. (2001) Redes Neurais Princípios e Práticas, 2ª edição, Bookman.
- HOLLAND, J. H. (1975) Adaptation in Natural and Artificial Systems, University of Michigan Press, United States of America.
- JACQUES, M. A. P., J. NITTYMÄKI e M. PURSULA (2002a) Analyzing Different Fuzzy Traffic Signal Controllers for Isolated Intersections. In: TRB 81 ST ANNUAL MEETING. CD Rom. Washington, DC.
- JACQUES, M. A. P.; PURSULA, M.; NITTYMÄKI, J.; e KOSONEN, I. (2002b) The Impact of Different Approximate Reasoning Methods on Fuzzy Signal Controllers. Anais do 13th Mini-Euroconference, 10 a 13 de Junho de 2002, Bari – Itália, p.184 – 192.
- JACQUES, M. A. P.; PURSULA, M.; NITTYMÄKI, J.; e KOSONEN, I. (2002c) Analysis of Different Defuzzyfication Methods Applied to Fuzzy Signal Controller. In: XII CONGRESSO PANAMERICANO DE INGENIERÍA DE TRÁNSITO Y TRANSPORT. Quito – Equador. Avances en Ingeniería de Tránsito e Transporte.
- JANG, J.-S., GULLEY, N. (1995). Fuzzy Logic Toolbox – For use Matlab. Natick, Mauss.: the MathWorks, Inc.
- KALIĆ, M., TEODOROVIĆ, D. (1996) Solving the trip distribution problem by fuzzy rules generated by learning from examples. Proceedings of the XXIII Yugoslav Symposium on Operations Research, Zlatibor, Yugoslavia, pp. 777-780 (in Serbian).

- KALIĆ, M., TEODOROVIĆ, D. (1997a) Trip distribution modeling using soft computing techniques. Paper presented at the EURO XV/INFORMS XXXIV (Book of abstracts, p. 74), Barcelona.
- KALIĆ, M., TEODOROVIĆ, D. (1997b) A soft computing approach to trip generation modeling. Paper presented at the 9th Mini EURO Conference Fuzzy sets in tra • c and transport systems, Budva, Yugoslavia.
- KASABOV, N. K. (1998) Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering. Second printing. Massachusetts Institute of Technology, Cambridge, Massachusetts: The MIT Press, 546p.
- KELL, J. H. e I. J. FULLERTON (1991) Manual of Traffic Signal Design. Second Edition. Institute of Transportation Engineers. Prentice-Hall, Inc. New Jersey, USA.
- KIM, S. (1994) Application of Petri Networks and Fuzzy Logic to Advanced Traffic Management Systems. Ph.D Thesis, Polytechnic University, USA, 139 p.
- KLIR, G. J., YUAN, B. (1995) Fuzzy Sets and Fuzzy Logic: Theory and Applications. Ed. Prentice-Hall. Estados Unidos da América. 574p.
- KÖNÖNEN, V., NIITYMÄKI, J. (2000) New Methods for Traffic Signal Control- Development of Fuzzy Controller. Published in the proceedings of FUZZ-IEEE 2000 conference in San Antonio, Texas, USA, p. 358-362.
- LABANCA, F. G. (2009) Projeto de Controladores Semafóricos Fuzzy com o uso de Algoritmos Genéticos. Monografia de Projeto Final – Universidade de Brasília. Faculdade de Tecnologia. Departamento de Engenharia Civil e Ambiental. Brasília, DF. 72p.
- LARKIN, L. (1985) A fuzzy logic controller for aircraft flight control. In: Sugeno, M. (Ed.), Industrial applications of fuzzy control. Elsevier Science (North-Holland), New York, pp. 87-103.
- LEE, J., ABDULHAI, B., SHALABY, A., Chung, E-H. (2005) Real-time optimization for adaptive traffic signal control using genetic algorithms. Journal of Intelligent Transportation Systems: Technology, Planning, and Operations. Taylor and Francis Inc.
- LEMONGE, A. C. C. (1999) Aplicação de Algoritmos Genéticos em otimização estrutural. XVI, 218p. Tese – Universidade Federal do Rio de Janeiro, COPPE.
- LOTAN, T., KOUTSOPOULOS, H. (1993a) Route choice in the presence of information using concepts from fuzzy control and approximate reasoning. Transportation Planning and Technology 17, 113-126.
- LOTAN, T., KOUTSOPOULOS, H. (1993b) Models for route choice behaviour in the presence of information using concepts from fuzzy set theory and approximate reasoning. Transportation 20, 129-155.

- MAIOLINO, C. E. G., (1992). SEIS - Sistema Especialista para implantação de Semáforos. Dissertação de Mestrado em Transportes. Universidade Federal do Rio de Janeiro. Rio de Janeiro.
- MATHWORKS (2008) Getting Started Guide. The MathWorks, Inc.
- MCSHANE, W. R., ROESS, R. P. (1990) Traffic Engineering. Prentice Hall Polytechnic Series in Traffic Engineering. New Jersey.
- MITCHELL, M. (1998) An introduction to Genetic Algorithms. Eight printing. Massachussetts Institute of Technology, Cambridge, Massachussetts: The MIT Press, 221p.
- NDOH, N.N., ASHFORD, N.J. (1994) Evaluation of transportation level of service using fuzzy sets. Paper presented at the 73rd Annual Meeting, Transportation Research Board, Washington, DC.
- NIITTYMÄKI, J. (1998) Isolated Traffic Signals – Vehicle Dynamics and Fuzzy Control. Thesis. Department of Civil and Environmental Engineering. Helsinki University of Tecnology.
- NIITTYMÄKI, J., PURSULA, M. (2000) Signal Control Using Fuzzy Logic. Fuzzy Sets and Systems, 116, p.11-22.
- NOVAES, A . G. (1978) Métodos de otimização: aplicações aos transportes. Editora Edgard Blucher: transesp. São Paulo.
- NIITTYMÄKI, J., NEVALA, R. (2001). Fuzzy Adaptive traffic Signal Control – Principles and Results. In: Joint 9th IFSA Wolrd Congress and 20th NAFIS International Conference. Vancouver.
- OCHI, L. S., ROCHA M. L. (2000); A new hybrid evolutionary algorithm for the vehicle routing and scheduling problems; Proc. of the Ninth International Conference on Intelligence Systems: Artificial Intelligence Applications for the New Millennium, pp 135-140, Louisville, Kentucky, EUA; Editores: James Graham e David Page; Editora: International Society for Computer & Their Applications – ISCA.
- PACHECO, M. A. (1999) Algoritmos Genéticos: Princípios e Aplicaciones. INTERCON 99: V Congresso Internacional de Ingeniería Electrónica, Elétrica y Sistemas, pp. 11-16, Lima.
- PAPPIS, C. P., MAMDANI, H. (1977) A Fuzzy Logic Controller for Traffic Junctions. IEEE Transactions on Systems, Man and Cybernetics. Vol. SMC-7, N° 10.
- PATTNAIK, S.B., RAMESH KUMAR, K. (1996) Level of service of urban roads based on users' perception. Civil Engineering Systems 14, 87-110.
- PEDROSO, F. F. F. (2006) Modelagem Temporal da Severidade para a Gestão Rodoviária na Prevenção de Acidentes. Dissertação de Mestrado, Publicação T.DM 002-A/2006, Departamento de Engenharia Civil e Ambiental, Universidade de Brasília, Brasília, DF, 128p.

- PIETRANTONIO, H. (1994). Notas de Aula sobre Engenharia de Tráfego. São Paulo/SP: Departamento de Engenharia de Transportes, Escola Politécnica da USP. (Material Didático).
- RECHENBERG, I. (1973) Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution. Frommann-Holzboog, Stuttgart.
- ROSS, T. J. (2008) Fuzzy Logic with engineering applications. Ed. John Wiley & Sons Ltd., 2ª ed., Inglaterra. 628p.
- SAMPAIO, C. C. D. (2004) Determinação de uma Rede Ótima de Transporte Utilizando Algoritmo Genético. Dissertação de Mestrado, Publicação T.DM 001A/2004, Departamento de Engenharia Civil e Ambiental, Universidade de Brasília, Brasília, DF, 108p.
- SANTOS, D. B. L. M. (2003) Procedimento para a Construção dos Conjuntos Fuzzy utilizados em Controladores Semafóricos. Dissertação de Mestrado, Publicação TU.DM 006A/03, Departamento de Engenharia Civil e Ambiental, Universidade de Brasília, Brasília, DF, 109p.
- SAYED, T., ABDELWAHAB, W., NAVIN, F. (1995) Identifying accident-prone locations using fuzzy pattern recognition. Journal of Transportation Engineering 121, 352-358.
- SAYERS, T., ANDERSON, J. e CLEMENT, S. (1999) The Multi-Objective Optimisation of a Traffic Control System. Transportation and Traffic Theory (Proceedings of the Abbreviated Presentations Sessions, 14th ISTTT). Transportations Research Institute, Technion-Israel, Institute of Technology, Haifa, Israel, pp. 153 – 156.
- SCHRETTER, N., HOLLATZ, J. (1996) A fuzzy logic expert system for determining the required waiting period after traffic accidents. Proceedings of the Fourth European Congress on Intelligent Techniques and Soft Computing, Aachen, Germany, pp. 2164-2170.
- SILVA, O. Q. (2003) Modelagem de Algoritmo Genético para Otimização, Previsão e Orientação de Tráfego Viário. Dissertação de Mestrado. Departamento de Ciência da Computação, Universidade de Brasília, Brasília 122p.
- SUGENO, M. (1985). An introductory survey of fuzzy control. Information Sciences 36, p. 59-83.
- TEODOROVIĆ, D., KIKUCHI, S. (1990) Transportation route choice model using fuzzy inference technique. In: Ayyub, B.M. (Ed.), Proceedings of ISUMA '90, The First International Symposium on Uncertainty Modeling and Analysis. IEEE Computer Press, College Park, Maryland, pp. 140-145.
- TEODOROVIĆ, D., BABIĆ, O. (1993) Fuzzy inference approach to the flow management problem in air traffic control. Transportation Planning and Technology 17, 165-178.

- TEODOROVIĆ, D., KALIĆ, M., PAVKOVIĆ, G. (1994) The potential for using fuzzy set theory in airline network design. *Transportation Research* 28B, 103-121.
- TEODOROVIĆ, D., KALIĆ, M. (1995) A fuzzy route choice model for air transportation networks. *Transportation Planning and Technology* 19, 109-119.
- TEODOROVIĆ, D. (1999) Fuzzy logic systems for transportation engineering: the state of the art. *Transportation Research Part A*, vol. 33, p. 337-364.
- TORRES, A. C. S. (2003) Determinação de Rotas Ótimas de Ônibus Urbanos Utilizando Algoritmo Genético. Dissertação de Mestrado, Publicação TU.DM-002-A/2003, Departamento de Engenharia Civil e Ambiental, Universidade de Brasília, Brasília 87p.
- TRABIA, M.B., M. S. KASEKO e M. ANDE (1999) A Two-stage Fuzzy Logic Controller for Traffic Signals. *Transportation Research Part C*, Vol. 7, p.353-367.
- TRB – TRANSPORTATION RESEARCH BOARD (2000). *Highway Capacity Manual*. National Research Council, Washington, DC.
- VAZ, A. M. (2006) Estudos das Funções de Pertinência para Conjuntos Fuzzy Utilizados em Controladores Semafóricos Fuzzy. Dissertação de Mestrado, Publicação T.DM – 005A/2006, Departamento de Engenharia Civil e Ambiental, Universidade de Brasília, Brasília, DF, 158p.
- VUKADINOVIĆ, K., TEODOROVIĆ, D. (1994) A fuzzy approach to the vessel dispatching problem. *European Journal of Operational Research* 76, 155-164.
- VYTHOULKAS, P.C., KOUTSOPOULOS, H.N. (1994) Modeling discrete choice behaviour using concepts from fuzzy set theory, approximate reasoning and neural networks. Research Report TSU 817, ESRC Transport Studies Unit, University of Oxford, Oxford.
- WRIGHT, S. (1931) Evolution in Mendelian populations. *Genetics* 16: p. 97 – 159.
- XU, W., CHAN, Y. (1993a) Estimating an origin±destination matrix with fuzzy weights. Part 1: Methodology. *Transportation Planning and Technology* 17, 127-144.
- XU, W., CHAN, Y. (1993b) Estimating an origin-destination matrix with fuzzy weights Part 2: Case studies. *Transportation Planning and Technology* 17, 145-164.
- YAMAKAWA, T. (1993) A Fuzzy Inference Engine in Nonlinear Analog Mode and its Application to a Fuzzy Logic Control. *IEEE Transactions on Neural Networks* 4:496-520.
- YUANLI, G., SHANMEI, L., LEI, Y., CHUNFU, S. (2006) Study on intelligent control of signalized intersection based on genetic algorithm. *Proceedings of the Conference on Traffic and Transportation Studies, ICTTS*.
- ZADEH, L, A, (1965), *Fuzzy Sets, Information and Control*, Vol, 8, pp.

ZADEH, L, A, (1973), Outline of a New Approach to the Analysis of Complex Systems and Decision Process, IEEE Transactions on Systems, Man and Cybernetics, Vol, SMC-3, N° 1.

# Apêndice A

Resultados das Simulações para  
Comparação entre Controladores

## A.1 – Resultados Obtidos para o Volume Alto

Tabela A-1 – Resultados para Volume Alto (sem desconsiderar nenhum valor)

Semente	Atraso Médio	
	Controlador Andrade (2009)	Melhor Controlador Obtido no AG
1	32,31	21,55
2	36,90	21,30
3	30,44	19,42
4	37,63	24,08
5	43,22	25,64
6	38,26	24,04
7	40,85	22,51
8	38,03	21,43
9	26,70	18,91
10	34,03	19,29
11	37,63	22,16
12	34,83	21,28
13	32,32	22,49
14	33,54	20,94
15	38,61	21,68
16	40,49	22,19
17	37,17	22,55
18	41,61	25,00
19	36,99	24,67
20	40,60	23,82
21	52,83	28,00
22	46,45	27,37
23	51,51	30,12
24	47,45	28,66
25	44,73	26,14
26	50,65	28,93
27	46,02	24,05
28	41,02	23,89
29	34,96	22,39
30	38,24	22,55
31	37,31	22,28
32	43,34	22,94
33	42,35	23,42
34	45,07	23,21
35	40,50	23,91
<b>Soma</b>	1394,59	822,81
<b>Média (X)</b>	39,85	23,51
<b>Desvio Padrão (S)</b>	5,93	2,69

Valores a serem desconsiderados (estão fora do intervalo  $X \pm 2S$ ). Obs: seus correspondentes no outro controlador também foram desprezados.

Tabela A-2 – Resultados para Volume Alto (desconsiderando valores fora do intervalo  $X \pm 2S$ )

Semente	Atraso Médio	
	Controlador Andrade (2009)	Melhor Controlador Obtido no AG
1	32,31	21,55
2	36,90	21,30
3	30,44	19,42
4	37,63	24,08
5	43,22	25,64
6	38,26	24,04
7	40,85	22,51
8	38,03	21,43
9	34,03	19,29
10	37,63	22,16
11	34,83	21,28
12	32,32	22,49
13	33,54	20,94
14	38,61	21,68
15	40,49	22,19
16	37,17	22,55
17	41,61	25,00
18	36,99	24,67
19	40,60	23,82
20	46,45	27,37
21	51,51	30,12
22	47,45	28,66
23	44,73	26,14
24	46,02	24,05
25	41,02	23,89
26	34,96	22,39
27	38,24	22,55
28	37,31	22,28
29	43,34	22,94
30	42,35	23,42
31	45,07	23,21
32	40,50	23,91
<b>Soma</b>	1264,41	746,97
<b>Média</b>	39,51	23,34
<b>Desvio Padrão</b>	4,87	2,37

- Hipótese  $H_0$ : média da diferença entre os atrasos médios = 0 (os resultados são estatisticamente iguais)
- Hipótese  $H_1$ : média da diferença entre os atrasos médios  $\neq$  0 (os resultados são estatisticamente diferentes)
- $t$  crítico =  $t_{31, 0,05} = 2,040$

- $t$  calculado = 28,72
- Rejeita-se  $H_0$ , e aceita-se  $H_1$

## A.2 – Resultados Obtidos para o Volume Baixo

Tabela A-3 – Resultados para Volume Baixo (sem desconsiderar nenhum valor)

Semente	Atraso Médio	
	Controlador Andrade (2009)	Melhor Controlador Obtido no AG
1	9,84	7,49
2	9,32	7,87
3	9,39	7,53
4	9,48	7,60
5	10,48	8,47
6	10,16	8,01
7	9,64	7,77
8	9,78	7,79
9	9,66	7,86
10	9,23	7,68
11	9,71	7,61
12	9,40	7,67
13	9,77	7,64
14	9,59	7,74
15	9,66	8,04
16	9,70	8,10
17	9,60	7,45
18	9,92	7,99
19	10,00	8,22
20	9,40	8,08
21	10,12	8,87
22	9,99	8,08
23	9,79	8,15
24	10,32	8,20
25	10,43	7,88
26	10,09	7,95
27	9,83	7,86
28	10,24	8,62
29	10,29	7,91
30	10,27	7,87
31	9,58	8,12
32	10,06	8,42
33	10,41	8,37
34	9,90	8,02
35	10,16	7,74
<b>Soma</b>	345,21	278,67
<b>Média (X)</b>	9,86	7,96
<b>Desvio Padrão (S)</b>	0,34	0,32

Valores a serem desconsiderados (estão fora do intervalo  $X \pm 2S$ ) Obs: seus correspondentes no outro controlador também foram desprezados.

Tabela A-4 – Resultados para volume baixo (desconsiderando valores fora do intervalo  $X \pm 2S$ )

Semente	Atraso Médio	
	Controlador Andrade (2009)	Melhor Controlador Obtido no AG
1	9,84	7,49
2	9,32	7,87
3	9,39	7,53
4	9,48	7,60
5	10,48	8,47
6	10,16	8,01
7	9,64	7,77
8	9,78	7,79
9	9,66	7,86
10	9,23	7,68
11	9,71	7,61
12	9,40	7,67
13	9,77	7,64
14	9,59	7,74
15	9,66	8,04
16	9,70	8,10
17	9,60	7,45
18	9,92	7,99
19	10,00	8,22
20	9,40	8,08
21	9,99	8,08
22	9,79	8,15
23	10,32	8,20
24	10,43	7,88
25	10,09	7,95
26	9,83	7,86
27	10,29	7,91
28	10,27	7,87
29	9,58	8,12
30	10,06	8,42
31	10,41	8,37
32	9,90	8,02
33	10,16	7,74
<b>Soma</b>	324,85	261,18
<b>Média (X)</b>	9,84	7,91
<b>Desvio Padrão (S)</b>	0,34	0,26

- Hipótese  $H_0$ : média da diferença entre os atrasos médios = 0 (os resultados são estatisticamente iguais)

- Hipótese  $H_1$ : média da diferença entre os atrasos médios  $\neq 0$  (os resultados são estatisticamente diferentes)
- $t$  crítico =  $t_{32, 0,05} = 2,037$
- $t$  calculado = 36,041
- Rejeita-se  $H_0$ , e aceita-se  $H_1$