

**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DO PARANÁ**  
Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial

---

**DISSERTAÇÃO**

apresentada ao CEFET-PR  
para obtenção do título de

**MESTRE EM CIÊNCIAS**

por

**ALEXANDRE FADEL GOBBO**

---

**PROPOSTA DE APLICAÇÃO DE  
SISTEMAS DE INFERÊNCIA NEURO-FUZZY  
PARA OTIMIZAÇÃO DE TRÁFEGO**

---

Banca Examinadora:

Presidente e Orientador:

**PROF. DR. PAULO CÉZAR STADZISZ**

**CEFET-PR**

Examinadores:

**PROF. DR. LUIS ALLAN KÜNZLE**

**UFPR**

**PROF. DR. CESAR AUGUSTO TACLA**

**CEFET-PR**

**PROF. DR. WERNER KRAUS JUNIOR**

**UFSC**

Curitiba, Março de 2005.



**ALEXANDRE FADEL GOBBO**

**PROPOSTA DE APLICAÇÃO DE  
SISTEMAS DE INFERÊNCIA NEURO-FUZZY  
PARA OTIMIZAÇÃO DE TRÁFEGO**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial do Centro Federal de Educação Tecnológica do Paraná, como requisito parcial para a obtenção do título de “Mestre em Ciências” – Área de Concentração: Informática Industrial.

Orientador: Prof. Dr. Paulo César Stadzisz

Curitiba

2005



## SUMÁRIO

<b>LISTA DE FIGURAS .....</b>	<b>V</b>
<b>LISTA DE TABELAS.....</b>	<b>VII</b>
<b>LISTA DE ABREVIATURAS E SIGLAS .....</b>	<b>IX</b>
<b>RESUMO .....</b>	<b>XI</b>
<b>ABSTRACT .....</b>	<b>XIII</b>
<b>1. INTRODUÇÃO .....</b>	<b>1</b>
1.1 CONTEXTO .....	1
1.2 PROBLEMÁTICA.....	2
1.3 MOTIVAÇÃO .....	3
1.4 OBJETIVOS.....	5
1.5 ORGANIZAÇÃO DOS CAPÍTULOS.....	6
<b>2. FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>7</b>
2.1 CONCEITOS EM ENGENHARIA DE TRÁFEGO .....	7
2.1.1 Noções Básicas.....	7
2.1.2 Detectores.....	10
2.1.3 Fluxo de Saturação, Capacidade e Grau de Saturação .....	11
2.1.4 Medidas de Efetividade .....	12
2.1.5 Determinação de Tempo de Ciclo e Tempos de Verde.....	18
2.1.6 Simulação e Otimização .....	21
2.1.7. Estratégias de Controle de Tráfego .....	25
2.1.8. Sistemas de Controle Adaptativo .....	26
2.2 ALGORITMOS GENÉTICOS .....	27
2.2.1 Conceitos Básicos .....	27
2.2.2 Aplicação de GA em Otimização de Tempos Semafóricos .....	29
2.3 ANFIS .....	30
2.3.1. Introdução a Sistemas de Inferência Fuzzy .....	30
2.3.2. A Arquitetura ANFIS .....	31
2.4. CONCLUSÃO .....	33
<b>3. ABORDAGEM PROPOSTA .....</b>	<b>35</b>

3.1 OBJETO DA PROPOSTA.....	35
3.2 ARQUITETURA ATUAL DO SISTEMA DE CURITIBA.....	35
3.3 APRESENTAÇÃO DA ABORDAGEM PROPOSTA .....	43
3.4 JUSTIFICATIVA.....	46
3.5 CONCLUSÃO .....	48
<b>4. IMPLEMENTAÇÃO.....</b>	<b>49</b>
4.1 ETAPAS DE IMPLEMENTAÇÃO.....	49
4.2 ESTIMATIVAS DE EFETIVIDADE DE LINKS COM O ANFIS .....	51
4.3 COMPOSIÇÃO DO ÍNDICE DE DESEMPENHO GLOBAL.....	59
4.4 OTIMIZAÇÃO DE DEFASAGENS COM ALGORITMOS GENÉTICOS.....	60
4.5 CONSTRUÇÃO DO OTIMIZADOR HÍBRIDO .....	62
4.6 CONCLUSÃO .....	64
<b>5. RESULTADOS.....</b>	<b>65</b>
5.1 ANFIS COMO ESTIMADOR DE MEDIDAS DE EFETIVIDADE.....	65
5.2 ALGORITMO GENÉTICO COMO OTIMIZADOR DE DEFASAGENS .....	69
5.3 MECANISMO DE OTIMIZAÇÃO HÍBRIDO COMPLETO.....	70
5.4 ANÁLISE DOS RESULTADOS.....	78
5.5 CONCLUSÃO .....	84
<b>6. DISCUSSÃO E CONCLUSÕES.....</b>	<b>85</b>
6.1 RECAPITULAÇÃO DOS OBJETIVOS DO TRABALHO.....	85
6.2 SUMÁRIO DOS RESULTADOS OBTIDOS .....	86
6.3 CONCLUSÕES.....	88
6.4 TRABALHOS FUTUROS.....	89
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>91</b>

## LISTA DE FIGURAS

<b>Figura 1:</b> Representação de um cruzamento .....	8
<b>Figura 2:</b> Representação de um plano de tráfego .....	9
<b>Figura 3:</b> Modificação de tempos de um cruzamento mantendo-se o tempo de ciclo .....	10
<b>Figura 4:</b> Formação e descarga de filas .....	15
<b>Figura 5:</b> Filas residuais na saturação .....	15
<b>Figura 6:</b> Relação entre tempos de ciclo e atrasos .....	19
<b>Figura 7:</b> Função de chegada de veículos no cruzamento.....	23
<b>Figura 8:</b> Alocação de janela de verde .....	24
<b>Figura 9:</b> Arquitetura ANFIS .....	33
<b>Figura 10:</b> Arquitetura do Sistema Antares.....	36
<b>Figura 11:</b> Diagrama de casos de uso do sistema de controle.....	37
<b>Figura 12:</b> Posicionamento dos detectores .....	38
<b>Figura 13:</b> Gráficos de contagem e ocupação .....	39
<b>Figura 14:</b> Ciclo de aquisição, otimização e comando por sub-área.....	41
<b>Figura 15:</b> Decomposição nos subsistemas otimizador e simulador.....	44
<b>Figura 16:</b> Utilização do simulador Transyt para validação do otimizador .....	45
<b>Figura 17:</b> Diagrama de atividades do mecanismo de otimização .....	46
<b>Figura 18:</b> Planta da subárea Batel.....	50
<b>Figura 19:</b> Representação de um <i>link</i> .....	51
<b>Figura 20:</b> Esquema de generalização dos resultados do FIS .....	55
<b>Figura 21:</b> Esquema de treinamento e estimação do FIS .....	56
<b>Figura 22:</b> Funções de Pertinência .....	57
<b>Figura 23:</b> Superfícies de Resposta do FIS .....	58
<b>Figura 24:</b> Esquema do mecanismo de otimização .....	62
<b>Figura 25:</b> Estimativas de paradas (caso normalizado).....	66
<b>Figura 26:</b> Estimativas de filas (caso normalizado) .....	66
<b>Figura 27:</b> Estimativas de paradas (caso geral).....	68
<b>Figura 28:</b> Estimativas de filas (caso geral) .....	68
<b>Figura 29:</b> Estimativas de atrasos (caso geral).....	68
<b>Figura 30:</b> Evolução do PI para população=20 e gerações=200 .....	70
<b>Figura 31:</b> $PI_{ANFIS}$ e $PI_{TRANSYT}$ em <i>hill-climbing</i> sobre $PI_{ANFIS}$ ( $1^d$ ) .....	72
<b>Figura 32:</b> $PI_{ANFIS}$ e $PI_{TRANSYT}$ em <i>hill-climbing</i> sobre $PI_{ANFIS}$ ( $40^d, 15^d, 1^d$ ).....	72

<b>Figura 33:</b> $PI_{ANFIS}$ e $PI_{TRANSYT}$ em <i>hill-climbing</i> sobre $PI_{ANFIS}$ ( $40^d, 15^d, 1^d, 1^s, 1^d$ ) .....	72
<b>Figura 34:</b> <i>Hill-climbing</i> sobre $PI_{TRANSYT}$ ( $1^d$ ) .....	73
<b>Figura 35:</b> <i>Hill-climbing</i> sobre $PI_{TRANSYT}$ ( $40^d, 15^d, 1^d$ ).....	73
<b>Figura 36:</b> <i>Hill-climbing</i> sobre $PI_{TRANSYT}$ ( $40^d, 15^d, 1^d, 1^s, 1^d$ ) .....	73
<b>Figura 37:</b> Evolução por algoritmo genético do $PI_{ANFIS}$ .....	74
<b>Figura 38:</b> <i>Hill-climbing</i> sobre $PI_{ANFIS}$ ( $40^d, 15^d, 1^d, 1^s, 1^d$ ).....	75
<b>Figura 39:</b> Superfície de resposta para estimativa de atrasos.....	78
<b>Figura 40:</b> Superfície de resposta para estimativa de atrasos ( $sat \leq 1.0$ ) .....	79
<b>Figura 41:</b> Gráfico de contorno para estimativa de atrasos ( $sat \leq 1.0$ ) .....	79
<b>Figura 42:</b> Saturação inicial e após EQUISAT dos 106 Links.....	81
<b>Figura 43:</b> Participação do fluxo primário por link.....	83
<b>Figura 44:</b> Comparação entre estimativas de paradas por link .....	83
<b>Figura 45:</b> Descontinuidade devido à variável cíclica ( $ch$ ).....	84



## LISTA DE TABELAS

<b>Tabela 1:</b> Sumário de benefícios de sistemas ITS .....	3
<b>Tabela 2:</b> Níveis de Serviço.....	14
<b>Tabela 3:</b> Variáveis fuzzy dos FISs e intervalos de validade .....	56
<b>Tabela 4:</b> Valores lingüísticos das variáveis fuzzy .....	57
<b>Tabela 5:</b> Tempos dos cruzamentos após inicialização de splits.....	71
<b>Tabela 6:</b> Tempos dos cruzamentos após otimização .....	75
<b>Tabela 7:</b> Resultados de otimização .....	76
<b>Tabela 8:</b> Tempos de execução .....	77
<b>Tabela 9:</b> Resultados de otimização sobre cenário inicial pré-programado .....	77



## LISTA DE ABREVIATURAS E SIGLAS

ANFIS	- <i>Adaptive Network-based Fuzzy Inference System</i> (Sistema de Inferência <i>Fuzzy</i> baseado em Rede Adaptativa)
EQUISAT	- <i>Equal Saturation</i> (método de determinação de tempos de verde em semáforos estabelecendo-se saturação equivalente nos <i>links</i> )
FHWA	- <i>Federal Highway Administration</i>
FIS	- <i>Fuzzy Inference System</i> (Sistema de Inferência <i>Fuzzy</i> )
GA	- <i>Genetic Algorithm</i> (Algoritmo Genético)
HCM	- <i>Highway Capacity Manual</i>
ITE	- <i>Institute of Transportation Engineers</i>
ITS	- <i>Intelligent Transportation Systems</i>
LOS	- <i>Level of Service</i> (nível de serviço)
MOE	- <i>Measures of Effectiveness</i> (medidas de efetividade para um <i>link</i> )
PI	- <i>Performance Index</i> (índice de desempenho para o sistema de tráfego)
SCATS	- <i>Sydney Coordinated Adaptive Traffic System</i>
SCOOT	- <i>Split Cycle Length Offset Optimization Technique</i>
TRANSYT	- <i>Traffic Network Study Tool</i>
TRB	- <i>Transportation Research Board</i>
TTI	- <i>Texas Transportation Institute</i>
vph	- Veículos por hora, medida de fluxo de veículos



## RESUMO

Neste trabalho propõe-se um método de otimização de tráfego baseado em sistemas de inferência *neuro-fuzzy* (ANFIS). Esse método deve ser aplicável a um contexto de otimização em tempo real e orientado à arquitetura do sistema de controle de tráfego da cidade de Curitiba, considerando suas características e restrições.

O modelo proposto apresenta similaridades com o principal sistema de otimização dinâmica de tráfego existente, denominado SCOOT, empregando o método *hill-climbing* sobre um índice de desempenho calculado pela ponderação das estimativas de eficiência dos *links*. Entretanto, a amostragem de dados de fluxo no sistema de Curitiba, por não fornecer medida das conversões em tempo real, determina que o modelo de simulação seja mais aproximado ao utilizado por ferramentas *offline* de otimização. Empregou-se o Transyt, uma das mais populares ferramentas de simulação e otimização *offline*, na geração de dados de treinamento ao ANFIS e como referência para validação dos resultados.

O alto custo computacional do modelo matemático implementado no Transyt impede sua aplicação na otimização em tempo real de um grande número de cruzamentos sem que se imponha restrições no espaço de busca. Utilizou-se o ANFIS para extrair o conhecimento do simulador, ou seja, aproximar seus resultados. Optou-se pelo ANFIS por apresentar boa precisão e baixo custo computacional e assim sua utilização viabilizou a execução de algoritmos de otimização para um grande número de cruzamentos. Com relação aos processos de otimização, utilizou-se métodos consagrados como o EQUISAT e o *hill-climbing*, mas também foi avaliado o uso de um algoritmo genético para geração de um estado inicial ao *hill-climbing*, em um otimizador híbrido.



## *ABSTRACT*

It is proposed in this dissertation a traffic optimization model based on neuro-fuzzy inference systems (ANFIS) to be applied in real-time optimization, targeting the architecture of the traffic control system deployed in the city of Curitiba, considering its properties and restrictions.

The model presented herein has a similar approach to the best known dynamic traffic optimization system, SCOOT, implementing hill-climbing optimization on a performance index defined as a weighted sum of the links' measures of effectiveness. However, the nature of the traffic flow sampling in Curitiba, by not distinguishing the flow tuning rates, implies the simulation model should be closer to those implemented by offline traffic simulation tools. Transyt, which is one of the most popular offline traffic simulators, was used in order to generate training data and as a reference to validate the results.

Real-time optimization based on the mathematical model of Transyt cannot be applied to systems with a large number of intersections in the current state of technology, due to the high computational costs of the algorithms, unless by imposing restrictions on the search space. ANFIS was used in order to capture the knowledge of the simulator, which means to approximate Transyt's outputs. ANFIS was chosen due to its precision and low execution time. Its estimations made it possible the real-time execution of optimization algorithms for a very large number of intersections. Regarding the optimization processes, well-known methods such as EQUISAT and hill-climbing were used. A hybrid optimization method was also validated using a genetic algorithm to provide an initial state for the hill-climbing method.





## CAPÍTULO 1

### INTRODUÇÃO

#### 1.1 CONTEXTO

O problema do tráfego urbano é uma constante na maioria das cidades do mundo moderno, independentemente de padrões culturais e econômicos. Sua gestão demanda soluções adaptadas à cada realidade, realidades essas particulares e dinâmicas. Em especial nos grandes centros urbanos, as condições do tráfego tendem à rápida saturação das vias e existência de longos períodos de congestionamento. Tal efeito ocorre tanto em países desenvolvidos, com expressiva frota de veículos, como naqueles com menor frota, pois nesses a rede viária também é menor (GARTNER et al., 1999).

O tráfego urbano refere-se aos movimentos que coexistem nas vias públicas de cidades: veículos automotores, de tração animal, bicicletas e pedestres. A problemática do tráfego tem grandes implicações sociais, ecológicas e econômicas, sendo que uma boa administração determina melhorias nos níveis de segurança, redução de custos com combustíveis e desgaste de peças, redução na emissão de gases poluentes, assim como aumento de bem estar e qualidade de vida (WALLACE et al., 1999), (FHWA, 1998).

A engenharia de tráfego é uma disciplina que engloba as várias atividades relacionadas com a administração de tráfego, como projeto estrutural, sinalização, pavimentação e manutenção de vias, levantamentos estatísticos, monitoração e controle de temporização de semáforos. Com o desenho, implantação e administração de sistemas viários eficientes, busca-se atingir os seguintes objetivos:

- Harmonização dos diversos fluxos conflitantes, assegurando a segurança de condutores e pedestres;
- Minimização dos tempos de trajeto, atrasos e paradas, reduzindo assim o desconforto e a frustração dos usuários.
- Minimização de custos operacionais, redução do consumo de combustível e emissões de gases poluentes e, por consequência, do impacto ambiental.

Os grandes centros urbanos do Terceiro Mundo, apresentando rápido aumento da frota de automóveis, falta de recursos para obras de engenharia (como aumento da rede viária,

passagens de nível, viadutos e soluções eficientes de transporte coletivo), e ainda tendência generalizada de congestionamento em vias de quadras curtas, demandam ainda maior atenção no que se refere à correta sincronização dos semáforos.

## 1.2 PROBLEMÁTICA

Grandes melhorias nos padrões de tráfego são obtidas com relativamente pouco investimento inicial, pela correta sincronização dos semáforos através de heurísticas, medições manuais de fluxo e cálculos *offline*. As medições de fluxo usualmente são tomadas em semanas sucessivas sendo considerada, nos cálculos, a média por dia de semana, e por período. Esses valores alimentam um *software* de otimização que gera um estado otimizado de tempos considerando as medidas tomadas. Esse método, denominado “controle a tempos fixos”, é uma solução efetiva somente quando o tráfego segue padrões repetitivos (TTI, 2002). Além disso, sendo estático em sua programação, exige reavaliação da sincronia ao longo dos anos, à medida que o perfil do tráfego muda.

Mecanismos de otimização dinâmica, implementados com diferentes abordagens, buscam tornar o sistema mais eficiente, respondendo automaticamente a variações imprevisíveis nas condições de tráfego, sem demandar um trabalho contínuo de reavaliação de tempos e sem interferência manual dos operadores do sistema.

Sistemas adaptativos de otimização dinâmica são incorporados no conceito de *Intelligent Transportation Systems – ITS* (FHWA, 1998), e buscam reduzir a necessidade de novos investimentos em estrutura viária pelo melhor aproveitamento da estrutura existente (FHWA, 1997). No entanto tais sistemas são de implementação mais cara que o emprego de controle a tempos fixos por demandarem:

- Sistema de controle central;
- Infra-estrutura adequada de comunicação;
- Grande número de detectores de veículos para fornecer os dados necessários aos algoritmos de otimização.

A Tabela 1 apresenta os benefícios obtidos com a utilização de sistemas de ITS, conforme apresentado em (FHWA, 1997). Nela são apresentadas as variações de diferentes indicadores de desempenho, considerando diferentes sistemas ITS instalados nos Estados Unidos.

Este trabalho de dissertação de mestrado foca o problema da otimização dinâmica dos tempos de semáforos. Particularmente, considera-se como objeto de análise (sistema alvo) uma região da cidade de Curitiba denominada “subárea Batel”. Essa região é composta por um conjunto de ruas de grande fluxo em todas as direções, formando uma matriz de quadras curtas com tendência à saturação nos horários de pico. As quadras curtas exigem eficiência na otimização de tempos, pois os pelotões de tráfego não apresentam muita dispersão – que aumenta com a distância entre os semáforos - e dada a facilidade da ocorrência do fenômeno de bloqueio de transversais (a fila de veículos atingindo o cruzamento anterior), o que multiplica os malefícios do congestionamento. A subárea Batel está interligada a um sistema de controle central e possui detectores na maioria de suas aproximações, permitindo a implantação de estratégias de controle adaptativo.

**Tabela 1:** Sumário de benefícios de sistemas ITS

<b>Critério</b>	<b>Impacto</b>
Tempos de Viagem	Diminuição de 8 a 25%
Velocidades Médias	Aumento de 14% a 20%
Número médio de paradas de veículos	Diminuição de até 41%
Atrasos médios por veículos	Diminuição de 17 a 44%
Consumo de combustível	Diminuição de 6 a 13%
Emissão de poluentes (CO)	Diminuição de 5 a 15%

### 1.3 MOTIVAÇÃO

Nos últimos anos, um mecanismo dinâmico de controle de filas vem sendo utilizado na subárea Batel com alguns bons resultados observados. Esse mecanismo é baseado em heurísticas do engenheiro de tráfego e implementado pela configuração de um conjunto de regras baseadas no estado de saturação das aproximações de um cruzamento. Apresenta como vantagem sua escalabilidade e estabilidade. Como pontos fracos tem-se o grande número de parâmetros de configuração e a pouca informação para uma análise objetiva dos benefícios gerados pelo mecanismo.

Este trabalho propõe um modelo de otimização baseado em cálculo de estimativas de desempenho global, aplicável em tempo real a um grande número de cruzamentos, no qual sejam configuradas somente as características físicas das vias e, com um mínimo de

parâmetros iniciais, possa obter uma resposta ótima ou subótima, considerando todas as variáveis de controle.

O modelo de otimização proposto contém conceitos tradicionais de engenharia de tráfego e elementos oriundos de tópicos de inteligência computacional:

- Aplicação de sistemas *neuro-fuzzy* (ANFIS) para viabilizar o cálculo de estimativas de desempenho em tempo real;
- Utilização de algoritmos genéticos para dispensar a necessidade de programação de estados iniciais para o sistema.

O desenvolvimento de tecnologias em controle de tráfego adaptativo é pertinente, dada a importância do tema, a inexistência de tecnologias nacionais completas, o alto custo de sistemas importados e a consideração de características e restrições locais na criação de mecanismos mais eficientes.

A utilização de ferramentas de inteligência computacional mostra-se adequada ao problema da otimização de tráfego:

- Algoritmos genéticos se aplicam à obtenção de soluções em grandes espaços de busca, com incontáveis possibilidades combinatórias de solução, como é a determinação de tempos dos semáforos (FOY, BENEKOHAL e GOLDBERG, 1992);
- Regras *fuzzy* podem ser utilizadas na otimização local do cruzamento (PEDRYCZ e GOMIDE, 1998) - o comportamento do tráfego é "*fuzzy*" por natureza;
- Diversos modelos baseados em redes neurais vêm sendo propostos para a predição de medidas de tráfego (LINT, HOOGENDOORN e ZUYLEN, 2002).

Abordagens utilizando esses paradigmas já existem na literatura, no que se refere à estimação e otimização localizada. Entretanto, a literatura não apresenta um modelo que una esses conceitos em uma otimização global adaptativa.

Alguns aspectos justificam a utilização dessas técnicas. O primeiro é que se trata de um sistema de tempo real que deve ser processado em uma janela fixa de tempo, sincronizada em um ciclo de aquisição de dados, estimação, otimização e comando. Assim, o tempo de processamento deve se restringir a poucos segundos, em configurações que podem envolver um número muito grande de cruzamentos. Métodos de simulação convencionais são muito caros em termos computacionais e permitiriam buscas muito restritas de solução.

O segundo aspecto é que a aplicação desses paradigmas pode reduzir a necessidade de conhecimento especializado do engenheiro de tráfego requerida para a configuração de um sistema. Por exemplo, em um esquema tradicional de otimização *offline* por simulação, melhores resultados são obtidos com a introdução de um estado inicial coerente, o que demanda o trabalho prévio do engenheiro de tráfego (TTI, 2002).

Finalmente, tais técnicas trabalham bem a complexidade de sistemas onde não se tem uma descrição formal precisa e com incerteza associada. Não existe uma solução geral para o tráfego, modelos tornam-se imprecisos se transplantados para outros locais, e, no caso de condições de saturação, complexidade e incerteza são muito aumentadas.

## 1.4 OBJETIVOS

A intenção deste trabalho é estudar e propor contribuições para a otimização de tráfego urbano incluindo os seguintes objetivos gerais:

1. A concepção de um modelo de otimização de tráfego, ou seja, de tempos de semáforos, que viabilize um controle adaptativo e aplicável ao caso da cidade de Curitiba, com as características indicadas na seção 1.3.
2. O desenvolvimento do *software* de um protótipo para o modelo de otimização proposto.
3. A validação dos resultados por ferramentas de simulação.

A validação do modelo proposto se dará por simulações *offline*. Estará fora do escopo desse trabalho a implementação da interface com o sistema de controle de tráfego existente e avaliação do comportamento em situações reais.

O mecanismo de otimização proposto apresentará requisitos de tecnologia dados pelas características do sistema de controle de tráfego de Curitiba, considerando:

- *Hardware* e sistema operacional existente;
- Janela de tempo disponibilizada para cálculos de otimização;
- Formato dos dados de entrada aos algoritmos conforme disponibilizados pelos detectores instalados na região;
- Formato dos dados de saída (tempos de semáforos) compatíveis com os utilizados pelo sistema.

Além disso, busca-se atender os seguintes requisitos de *software*:

- Deverão ser configuradas apenas as características físicas do sistema viário, além de alguns parâmetros mínimos de segurança, minimizando a necessidade de conhecimento especializado;
- O modelo de otimização deverá apresentar alguma métrica de desempenho para sua validação.

## 1.5 ORGANIZAÇÃO DOS CAPÍTULOS

No Capítulo 2 (“Fundamentação Teórica”) define-se os conceitos e a nomenclatura utilizados neste documento, subdivididos em: conceitos em engenharia de tráfego, algoritmos genéticos e sistemas *neuro-fuzzy* (ANFIS). O Capítulo 3 (“Abordagem Proposta”) enumera as propriedades do sistema de controle de tráfego urbano da cidade de Curitiba, delimitando o universo de trabalho pelas suas possibilidades e restrições tecnológicas. A seguir, apresenta e justifica o modelo de otimização proposto. O Capítulo 4 (“Implementação”) detalha os passos de implementação deste modelo. No Capítulo 5 (“Resultados”) apresenta-se e discute-se os resultados obtidos. Por fim, o Capítulo 6 (“Discussão e Conclusões”) contém as conclusões gerais do trabalho e as propostas de trabalhos futuros.

## CAPÍTULO 2

### FUNDAMENTAÇÃO TEÓRICA

Este capítulo traz a fundamentação teórica utilizada na construção deste trabalho de pesquisa. Na seção 2.1 são apresentados conceitos em engenharia de tráfego incluindo definições consagradas e resultados de pesquisas relevantes na área. Busca-se formalizar os aspectos da disciplina necessários ao entendimento dos objetivos enunciados no Capítulo 1. Além disso, é traçado um panorama dos avanços em engenharia de tráfego e são apresentados os principais sistemas de otimização de tráfego existentes. As seções 2.2 e 2.3 apresentam sucintamente duas ferramentas utilizadas na abordagem proposta: algoritmos genéticos e ANFIS, respectivamente. Para melhor compreensão desses temas, bibliografia relevante adicional será sugerida.

#### 2.1 CONCEITOS EM ENGENHARIA DE TRÁFEGO

##### 2.1.1 Noções Básicas

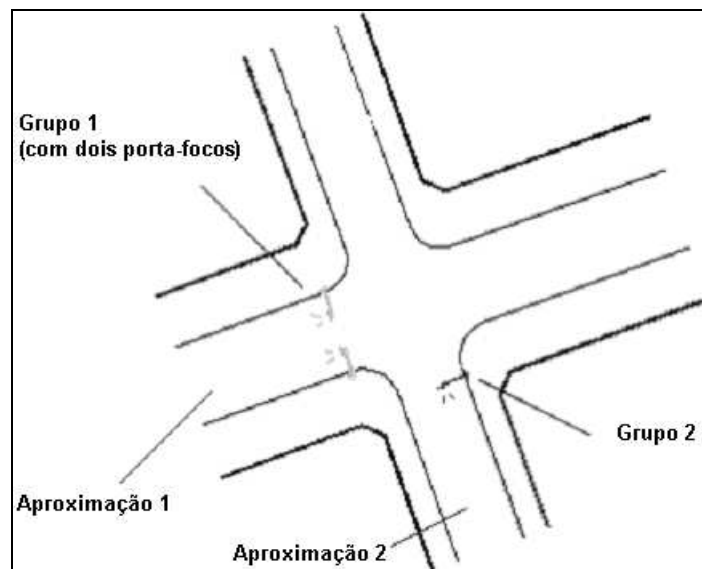
Os conceitos apresentados nesta seção encontram-se na literatura de engenharia de tráfego como em (WEBSTER, 1958), (VALDES, 1978), (WALLACE et al., 1999) e (TRB, 2000) com alguma diferenciação em termos de nomenclatura. Padroniza-se, assim, para este documento, os nomes dos conceitos utilizados, apresentando também o termo equivalente mais usual em inglês.

Um **cruzamento** (*intersection*), representado na Figura 1 em sua configuração mais simples, corresponde a uma intersecção de vias de tráfego com movimentos conflitantes, denominadas **aproximações** (*links*). Tais movimentos são harmonizados através de indicações luminosas providas por grupos focais. A cada movimento é associado um **estágio** de verde (*phase*).

Cruzamentos podem se apresentar em configurações mais complexas, como, entre outros:

- Mais de dois *links* de chegada ou saída;

- *Links* com fluxo nos dois sentidos;
- Sinalização para pedestres;
- Faixa específica para ônibus – em Curitiba é freqüente o uso de uma faixa central para ônibus, com movimento nos dois sentidos, entre faixas para veículos.



**Figura 1:** Representação de um cruzamento

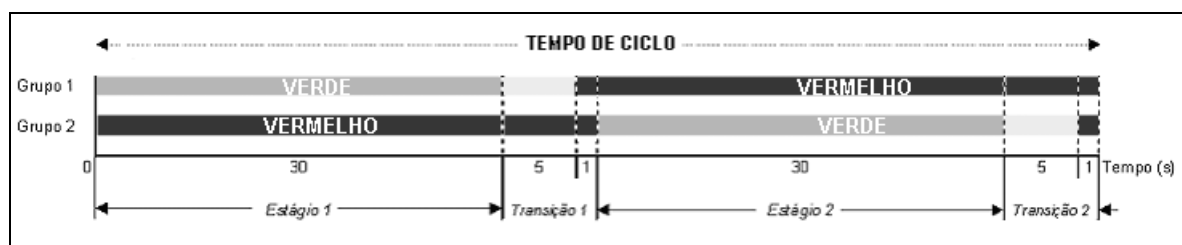
As indicações de um cruzamento são controladas por um equipamento denominado **controlador de tráfego**, normalmente instalado nas proximidades do cruzamento. Esse equipamento comanda as mudanças de cores de acordo com sua programação (ou conforme comandos externos) e impõe uma condição segura no caso de detecção de falhas. Podem ser integrados ao controlador de tráfego diversos sensores (**detectores veiculares**) que realizam aquisição automática de dados do fluxo. Se houver instância superior de controle (**sistema de controle central**), o controlador de tráfego, através de meio de comunicação adequado, informa alarmes e medidas de fluxo, assim como recebe comandos para modificação de seu padrão de funcionamento.

Um conjunto de controladores próximos e interdependentes compõe uma região de controle coerente denominada **subárea**, que pode caracterizar um bairro. Aproximações consecutivas em uma via formam um **corredor** ou **arterial** (arterial).

A Figura 2 representa um possível comportamento de um controlador de tráfego, ou seja, um **plano de tráfego** (*plan*) em seu caso mais simples, comandando dois grupos focais e apresentando as variáveis de um plano de tráfego. Este plano se repetiria ciclicamente até que



outro plano seja selecionado, oferecendo, no exemplo apresentado, tempos de verde e de vermelho idênticos para as duas aproximações (30 segundos). Observa-se um tempo de transição composto de 5 segundos de amarelo e de 1 segundo de segurança em estado vermelho para os dois grupos focais.

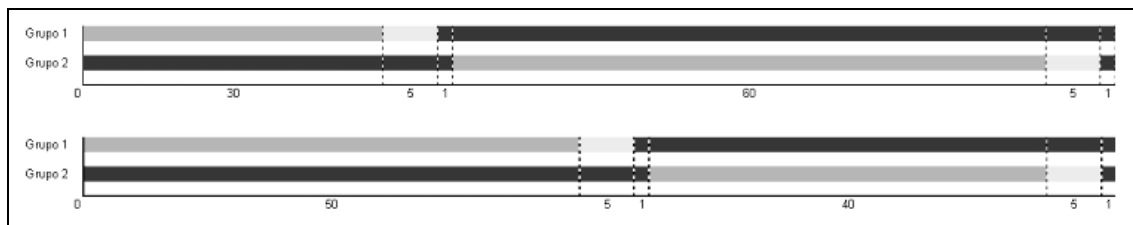


**Figura 2:** Representação de um plano de tráfego

O plano possui variáveis estruturais, como o número de grupos focais e a seqüência de cores. As variáveis de controle, que distinguem diferentes planos de tráfego para um mesmo cruzamento seriam (WALLACE et al., 1999):

- **Tempo de ciclo** (*cycle length*): período de tempo no qual as indicações dos grupos focais retornam à sua condição inicial. Normalmente, esse valor é fixo para um conjunto de cruzamentos adjacentes (em uma subárea) durante um determinado período, de modo que seja possível a sua sincronização.
- **Estágios ou frações de verde** (*splits*): período de tempo no qual se tem uma indicação estável dos grupos focais, com pelo menos um dos grupos em verde. Este tempo pode ser variável, mas dentro de limites de segurança (por exemplo, um **tempo mínimo de verde** de 8 segundos para o estágio).
- **Transições:** conjunto de modificações nos estados dos grupos para a condução de um estágio a outro. Normalmente seu tempo é fixo, sendo um parâmetro de segurança do sistema (por exemplo, 3 segundos de amarelo e 1 segundo de **vermelho de segurança** para todos os grupos).
- **Defasagem** (*offset*): embora não representada no diagrama da Figura 2, esta é uma importante variável de tráfego. Considerando-se duas interseções  $i$  e  $j$ , com os mesmos tempos de ciclo, e a interseção  $i$  como referência, a defasagem da interseção  $j$  é a diferença de tempo entre o início do primeiro estágio da interseção  $j$  e o início do primeiro estágio da interseção  $i$ . Utiliza-se essa variável para se programar a sincronia dos controladores, sem que um estágio precise ser representado decomposto em dois, ao início e ao final do ciclo.

Um **plano de tráfego** refere-se então ao comportamento dinâmico de uma interseção, traduzido nas variáveis: tempo de ciclo, frações de verde e defasagem. Um controlador pode armazenar diversos planos, selecioná-los em função do horário e, também, receber planos voláteis a partir do sistema de controle central. Mantendo-se o tempo de ciclo, para coerência com os demais controladores da região, o plano de tráfego pode ser alterado de duas maneiras: pela modificação dos tempos de estágio (ver exemplo da Figura 3) e pela modificação de sua defasagem. A mudança de tempo de ciclo deve ser simultânea a todos os controladores de tráfego de uma região coordenada (**subárea**). A Figura 3 representa o aumento de tempo do primeiro estágio, e, para manutenção do tempo de ciclo, a redução correspondente do tempo do segundo estágio. Notar que, normalmente, os tempos das transições são mantidos, por serem parâmetros de segurança.



**Figura 3:** Modificação de tempos de um cruzamento mantendo-se o tempo de ciclo

### 2.1.2 Detectores

Detectores veiculares são dispositivos destinados a obter medidas de fluxo. A tecnologia mais usual para obtenção automática de dados de tráfego é a de **laços indutivos** (*inductive loops*), instalados sob o pavimento, por serem relativamente baratos e de construção robusta. Sua geometria e posicionamento diferem de sistema para sistema. Esses detectores realizam medida sobre um ponto da via e, individualmente, disponibilizam as seguintes informações (HALL, 1999):

- **Contagem** (*flow rate*): veículos passando por unidade de tempo, usualmente indicada em veículos por hora (vph)
- **Ocupação** (*occupancy*): percentual de tempo que um ponto na via está ocupado por veículos.

No caso de serem instalados dois sensores sequencialmente (medição de uma secção da via), pode-se ter também a informação de velocidade média.

Além do aprimoramento dos algoritmos de otimização, buscam-se hoje métodos para prover maior qualidade nos dados de detecção. As medidas fornecidas por detectores indutivos são integradas no tempo e não representam uma fotografia instantânea e simultânea do sistema. Idealmente, dever-se-ia saber a cada momento o tamanho exato das filas em cada aproximação do sistema, ou então quanto tempo cada veículo espera na fila (GARBACZ, 2002). Novas tecnologias para levantamento de dados de tráfego baseadas em processamento de imagem têm sido pesquisadas (BOILLOT et al., 1992), mas ainda são experimentais e caras.

### 2.1.3 Fluxo de Saturação, Capacidade e Grau de Saturação

**Fluxo de Saturação** (*Saturation Flow*) é a medida que define a capacidade de uma via sem sinalização semaforica, ou seja, o número máximo de veículos capazes de passar por uma secção da via por unidade de tempo, expressa em veículos de passeio por hora (vph). Este é um dos parâmetros mais importantes na definição das características de uma aproximação, sendo que uma boa estimativa é crítica para qualquer modelo de simulação e/ou otimização.

O fluxo de saturação varia em função de fatores como largura da faixa, presença de veículos pesados, atrito lateral, faixa de estacionamento, paradas de ônibus e conversões (TRB, 2000). Além disso, depende da agressividade dos condutores (WALLACE et al., 1999). Comumente se considera, para vias asfaltadas, um fluxo de saturação de 1500 a 1900 veículos por hora por faixa de rolamento (WALLACE et al., 1999).

Assim, o fluxo de saturação para a via é dado por:

$$fs = fsf \cdot n \quad (1)$$

onde:

- $fs$  é o fluxo de saturação da via (vph);
- $fsf$  é o fluxo de saturação por faixa (vph);
- $n$  é o número de faixas.

**Capacidade** (*Capacity*) representa o fluxo máximo por unidade de tempo de uma via semaforizada, ou seja, é o fluxo de saturação ponderado pelo percentual de tempo de verde para a aproximação  $i$  do cruzamento, sendo dada por:

$$cap_i = fs_i \cdot ( tve_i / tc ) \quad (2)$$

onde:

- $cap_i$  é a capacidade da aproximação  $i$  (vph);
- $fs_i$  é o fluxo de saturação da aproximação  $i$  (vph);
- $tve_i$  é o tempo de verde efetivo da aproximação  $i$  (s);
- $tc$  é o tempo de ciclo (s).

O tempo de verde efetivo é dado por:

$$tve_i = tv_i - tpv_i + tev_i \quad (3)$$

onde:

- $tve_i$  é o tempo de verde efetivo da aproximação  $i$  (s);
- $tv_i$  é o tempo de verde total da aproximação  $i$  (s);
- $tpv_i$  é o tempo perdido para a partida no início do verde na aproximação  $i$  (s);
- $tev_i$  é o tempo de extensão do verde da aproximação  $i$ , na qual o fluxo continua durante a transição (s).

O **Grau de Saturação** (*Saturation Degree*) indica o nível de utilização da capacidade de cada *link*, sendo a relação entre volume e capacidade:

$$sat_i = fm_i / cap_i \quad (4)$$

onde:

- $fm_i$  é o fluxo medido na aproximação  $i$  (vph)
- $cap_i$  é a capacidade da aproximação  $i$  (vph).

#### 2.1.4 Medidas de Efetividade

Existem diversas **Medidas de Efetividade** em tráfego (*Measures of Effectiveness - MOEs*), tanto relativas a um único *link*, como para um sistema. Algumas são definições obtidas por cálculo direto, outras podem ser estimadas por equações (que implementam algum

modelo matemático) ou por um modelo de simulação. São exemplos de medidas de efetividade (WALLACE et al., 1999):

- **Grau de Saturação** (*Degree of Saturation*): descrito na subseção 2.1.3.
- **Número de Paradas** (*Stops*): paradas que o *link* impõe ao fluxo em veículos/hora. Situações de congestionamento podem implicar em um número de paradas maior que o fluxo (mais de uma parada em média por veículo por aproximação, chamado de “falha de ciclo”).
- **Atrasos** (*Delays*): reflete diretamente a eficiência do sistema. É o somatório de tempos de espera de todos os veículos, por *link*, em veículos-hora/hora. O atraso médio é definido em segundos por veículo (s/v), sendo o valor médio de tempo de espera por veículo.
- **Tamanho Máximo de Filas** (*Maximum Back of Queue*): comprimento máximo que as filas atingem durante o ciclo, avaliados em função do tamanho do *link* (sua capacidade de estocagem).
- **Deslocamentos** (*Total Travel*): deslocamentos totais no *link*, sendo o produto do fluxo medido no *link* pelo seu comprimento em quilômetros, conforme Equação 5. O deslocamento total para um subsistema é calculado pelo somatório dos deslocamentos individuais.
- **Tempo de Deslocamento** (*Total Travel Time*): tempos totais de deslocamento no *link*, sendo o produto do fluxo pela soma de atrasos mais tempo de deslocamento considerando a velocidade esperada do fluxo, conforme Equação 6. O tempo de deslocamento total para um subsistema é calculado pelo somatório dos tempos de deslocamento totais individuais.
- **Velocidade Média Efetiva**: é a relação para um conjunto de  $n$  cruzamentos dos deslocamentos totais e dos tempos totais de deslocamento, conforme Equação 7.

$$TT_i = fm_i \cdot cl_i \quad (5)$$

$$TTT_i = fm_i \cdot ((cl_i / v_i) + d_i) \quad (6)$$

$$ve = \Sigma TT_i / \Sigma TTT_i \quad (7)$$

onde:

- $fm_i$  é o fluxo medido na aproximação  $i$  (vph);
- $cl_i$  é o tamanho da aproximação  $i$  (km);
- $d_i$  é o atraso da na aproximação  $i$  (h);

- $TT_i$  é o deslocamento total na aproximação  $i$  (km);
- $TTT_i$  é o tempo de deslocamento total na aproximação  $i$  (h);
- $\Sigma TT_i$  é o deslocamento total em um conjunto de aproximações (km);
- $\Sigma TTT_i$  é o tempo de deslocamento total em um conjunto de aproximações (h);
- $v_i$  é a velocidade média esperada de deslocamento do fluxo na aproximação  $i$  (km/h);
- $ve$  é a velocidade efetiva para um conjunto de aproximações (km/h).

Existe uma medida qualitativa das condições operacionais de um *link*, extensível ao sistema como um todo, definida em função do atraso médio (em segundos por veículo), chamada **Nível de Serviço** (*Level of Service - LOS*), conforme apresentado em (TRB, 1997). Os significados dos valores de LOS são definidos segundo a Tabela 2.

Modelos para medidas de efetividade baseiam-se em premissas quanto ao processo de formação de filas, podendo ser determinísticos ou estocásticos. Sistemas sinalizados usualmente são descritos em modelos determinísticos, que buscam identificar matematicamente os padrões dos fluxos de tráfego. Filas são formadas durante o tempo de vermelho e descarregadas durante o verde.

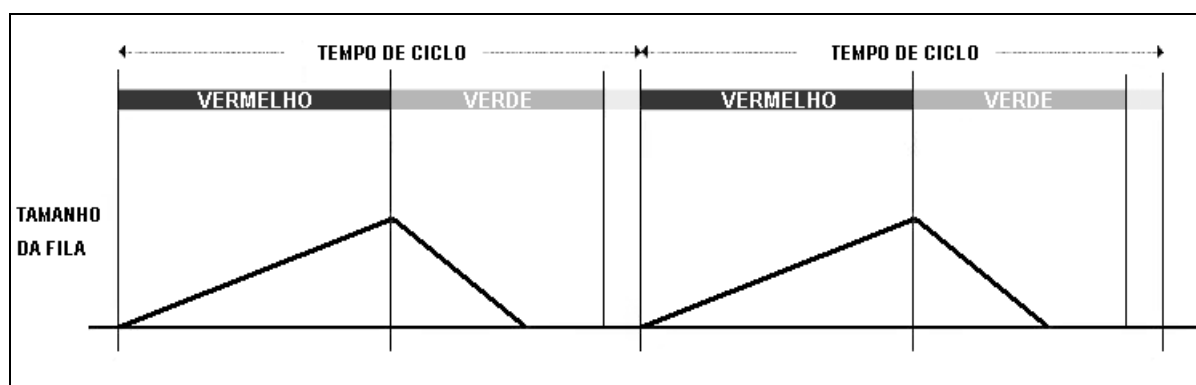
**Tabela 2:** Níveis de Serviço

LOS	Condições Operacionais	Atraso Médio (segundos/veículo)
A	Progressão extremamente favorável, maioria dos veículos chegam com sinal verde e não param.	$\leq 10$
B	Boa progressão.	$>10$ e $\leq 20$
C	Progressão regular, com número significativo de paradas.	$>20$ e $\leq 35$
D	Má progressão com número significativo de paradas e ocorrência de falhas de ciclo ( <i>cycle failure</i> - carros parando mais de uma vez no <i>link</i> ).	$>35$ e $\leq 55$
E	Péssima progressão com freqüentes falhas de ciclo.	$>55$ e $\leq 80$
F	Péssima progressão com muitas falhas de ciclo.	$>80$

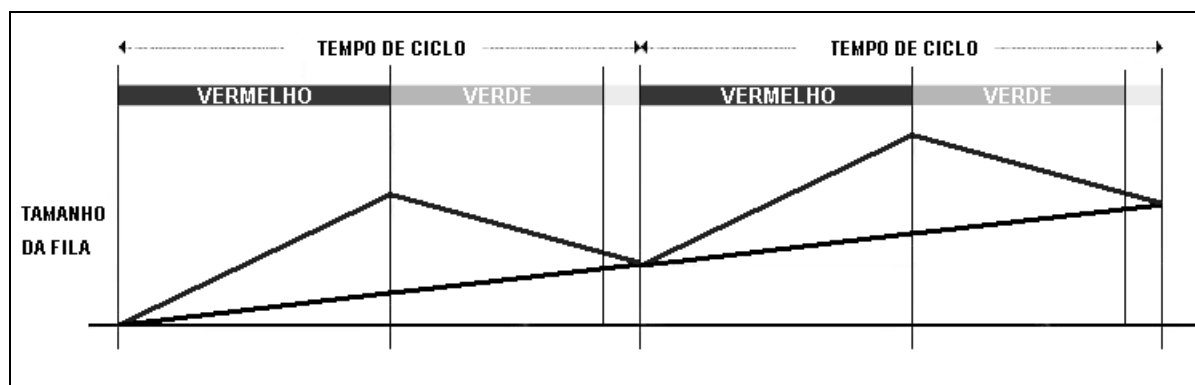
Em condições não saturadas, a fila descarrega totalmente durante o período de verde, como representado na Figura 4. O atraso total equivale à área sob o gráfico do tamanho de fila. Em situações de saturação, o tempo de verde é insuficiente para dissipar a fila formada durante o período de vermelho. Assim, uma fila residual é propagada aos ciclos seguinte.

A Figura 5 indica o processo de propagação de filas residuais. Os triângulos superiores correspondem ao atraso uniforme, enquanto que o atraso devido à saturação é dado pelo

triângulo maior inferior (aumenta a cada ciclo). A soma deles representa o atraso total. Quando as filas estimadas atingem o limite da capacidade de estocagem (o início do *link*) ocorre a situação de congestionamento, que corresponde a uma quebra nos pressupostos de análise. Essa situação é tratada de diferentes formas, sendo que os modelos iniciais consideravam filas como “empilhamento vertical”, não considerando os efeitos do congestionamento. Com ele, a complexidade do sistema é aumentada, e modelos não reproduzem de forma precisa seu comportamento.



**Figura 4:** Formação e descarga de filas



**Figura 5:** Filas residuais na saturação

Webster (1958) propôs o modelo para estimação de atrasos baseados em análise determinística de filas em vias não saturadas que foi a base para todos os modelos subsequentes, dado por.

$$d = \frac{C(1-\lambda)^2}{2(1-\lambda X)} + \frac{X^2}{2v(1-X)} - 0,65 \cdot (c/v^2)^{1/3} \cdot X^{2+5\lambda} \quad (8)$$

onde:

- $d$  é o atraso médio por veículos (s);
- $C$  é o tempo de ciclo (s);

- $\lambda$  é a proporção do ciclo com verde efetivo ( $g/C$ );
- $v$  é a taxa de chegada (vph);
- $c$  é a capacidade por faixa (vph);
- $X$  é a relação volume/capacidade por faixa ( $v/c$ );

Na equação, o primeiro termo representa o atraso médio assumindo chegadas uniformes. O segundo representa atraso adicional devido a um componente randômico de chegadas. O terceiro é um fator de ajuste empírico. Esse modelo apresenta erros crescentes à medida que a saturação aumenta, sendo apropriado somente quando o fluxo ( $v$ ) é significativamente inferior à capacidade ( $c$ ).

Diversos modelos foram apresentados na literatura, sendo que os principais da década de 80 (Transyt 7F, HCM, SOAP) apresentam resultados próximos em condições não saturadas, divergindo na saturação (HAGEN, 1989). Um amplo modelo generalizado foi proposto por Akcelik e Roupail (1993). Em HCM (TRB, 2000) tem-se um modelo que representa as últimas pesquisas na área, dado por:

$$d = d_1 \cdot PF + d_2 + d_3 \quad (9)$$

$$d_1 = \frac{0,5 C (1 - g/C)^2}{1 - \min(1, X) \cdot (g/C)} \quad (10)$$

$$d_2 = 900T [(X-1) + ((X-1)^2 + (8kIX / cT))^{1/2}] \quad (11)$$

$$PF = \frac{(1 - P) f_p}{1 - (g/C)} \quad (12)$$

onde:

- $d$  é o atraso total por veículo (s/v);
- $d_1$  é o atraso uniforme assumindo chegadas uniformes (s/v);
- $PF$  é o fator de ajuste do atraso uniforme para considerar os efeitos da sincronia de sinais sobre o atraso (ou seja, considerando o padrão de chegada);
- $d_2$  é o atraso incremental para considerar efeitos randômicos e de saturação, ajustado para a duração do período de análise e tipo de controle (s/v);
- $d_3$  é o atraso residual para considerar filas de saturação existentes antes do período de análise (s/v);
- $C$  é o tempo de ciclo (s);
- $g$  é o tempo efetivo de verde (s);
- $X$  é o grau de saturação -  $vm/c$ ;



- $c$  é a capacidade por faixa -  $sg/C$  (vph)
- $k$  é um fator de *delay* incremental dependente do tipo de controle - 0,6 para tempos fixos, e variando de 0.04 a 0.5 para controladores atuados com atrasos residuais para considerar filas de saturação existentes antes do período de análise (s/v);
- $I$  é um fator de ajuste para filtragem do fluxo primário da intersecção anterior. Em intersecções isoladas  $I=1,0$ . Se existe sinalização no cruzamento anterior,  $I$  é função do grau de saturação do fluxo primário anterior ( $X_p$ ):
 
$$I=1,0-0,91X_p^{2,68} \text{ se } X_p \leq 1,0 \text{ e } I=0,09 \text{ se } X_p > 1,0;$$
- $\lambda$  é a proporção do ciclo com verde efetivo -  $g/C$ ;
- $T$  é a duração do período de análise (h);
- $P$  é a proporção de veículos chegando no verde;
- $f_p$  é o fator de ajuste de progressão para quando o pelotão chega no verde.

O atraso é o principal MOE para estimação de desempenho do sistema, mas o número de paradas é também útil, principalmente por ser determinante no consumo total de combustível. O número de paradas, embora relacionado com o atraso, não é necessariamente proporcional. Para múltiplos veículos em um corredor, com mesmo atraso, pode-se ter números diferentes de paradas. Fora de saturação é possível estimar as paradas simplesmente pelo número de veículos que sofrem atrasos (ou que chegam no vermelho). Na saturação, a análise é mais complexa, pois veículos podem parar mais de uma vez em uma mesma aproximação e, também, é difícil distinguir “paradas totais” de “paradas parciais” - quando o fluxo se aproxima de uma fila que começa a se deslocar. Em situações como essa, o fluxo pode experimentar apenas redução de velocidade, ou, então, uma ou múltiplas paradas.

Akcelik (1981) apresenta uma estimação desse componente de saturação do número de paradas e do tamanho máximo de filas, dadas por:

$$N_0 = 0.25cTX^2 [(X-1) + ((X-1) + (mX / cT))^{1/2}] \quad (13)$$

$$h = \frac{3240N_0}{C} \quad (14)$$

onde:

- $N_0$  é o componente de saturação de filas;
- $h$  é o componente de saturação de paradas;
- $C$  é o tempo de ciclo (segundos);
- $X$  é o grau de saturação (v/c);

- $c$  é a capacidade por faixa ( $sg/C$ ) (vph);
- $T$  é a duração do período de análise (h).

Os componentes determinísticos de filas e paradas, como definido em (WALLACE et al., 1999), são:

$$N_d = \frac{(X-1) cT}{2} \quad (15)$$

$$h_d = \frac{q \cdot N_d}{c} \quad (16)$$

onde:

- $N_d$  é o componente determinístico de filas;
- $h_d$  é o componente determinístico de paradas;
- $X$  é o grau de saturação ( $v/c$ );
- $c$  é a capacidade por ciclo;
- $T$  é a duração do período de análise (h);
- $q$  é a taxa de chegada de veículos (vph).

### 2.1.5 Determinação de Tempo de Ciclo e Tempos de Verde

A seguinte equação aproxima o tempo de ciclo ideal para um cruzamento (WEBSTER, 1958):

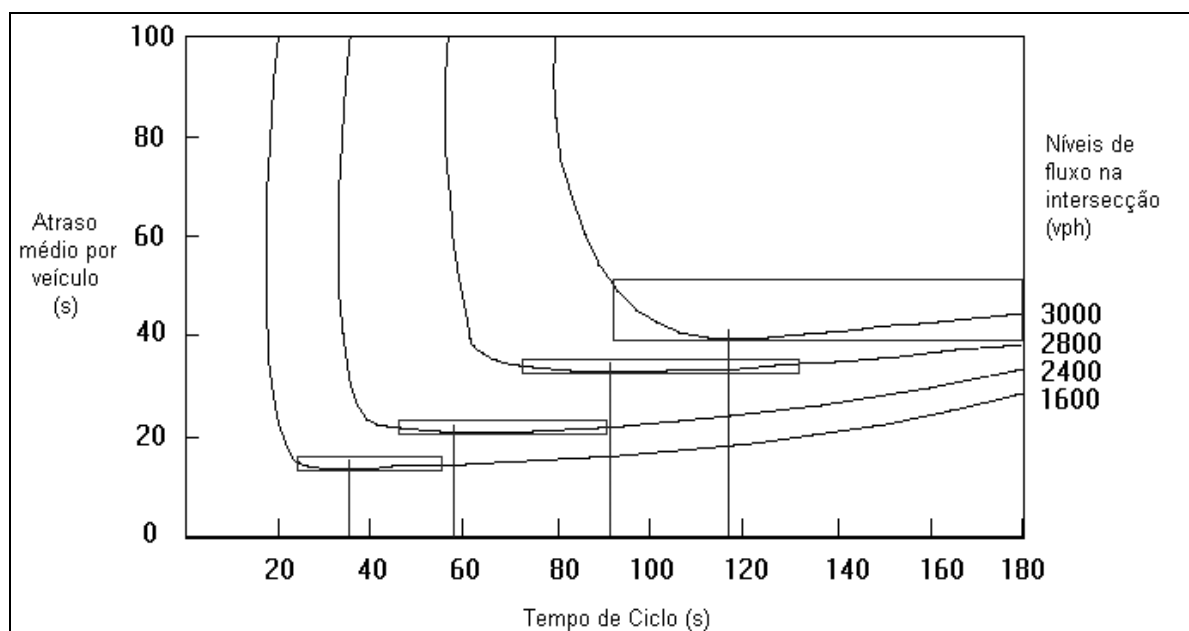
$$TC_0 = \frac{(1.5 * TP) + 5}{1.0 - y_1 - y_2 - \dots - y_n} \quad (17)$$

onde:

- $TC_0$  é o tempo de ciclo ideal;
- $TP$  é o tempo total perdido por ciclo, o somatório de tempos de amarelo e vermelho de segurança;
- $y_n$  é o fluxo dividido pelo fluxo de saturação para a aproximação  $n$  do cruzamento.

O diagrama da Figura 6, tomado de (ITE, 1992), apresenta a relação entre tempos de ciclo e atrasos médios por veículo, para diferentes níveis de fluxo. As linhas verticais apresentam o tempo ideal de ciclo dado pela equação de Webster. Pode-se observar que várias

soluções de ciclo apresentam bons resultados para certo valor de fluxo. Esses ciclos situam-se na faixa representada pelos retângulos e correspondem ao intervalo de  $0,75.TC_0$  a  $1,5.TC_0$ . Assim, não existe grande sensibilidade em desempenho para variações de tempo de ciclo dentro dessa faixa.



**Figura 6:** Relação entre tempos de ciclo e atrasos

Os cruzamentos de uma subárea devem ter tempos de ciclo idênticos (ou múltiplos) para que se possa obter uma sincronia constante em regime, ao longo de diversos ciclos. Este tempo de ciclo único para a subárea pode ser calculado em função do cruzamento mais saturado, buscando-se manter para ele um nível de saturação máximo.

O método mais usual na determinação do tempo de verde de cada grupo é chamado EQUISAT (*Equal Saturation*), proposto por Webster (1958), que estabelece o mesmo nível de saturação para todas as aproximações como forma de maximizar o desempenho do cruzamento. Para implementar a condição de EQUISAT, todas as aproximações de um cruzamento devem apresentar a relação:

$$(f m_i \cdot t c) / (f s_i \cdot t v_i) = k \quad (18)$$

onde:

- $f m_i$  é o fluxo na aproximação  $i$ ;
- $t c$  é o tempo de ciclo;
- $f s_i$  é o fluxo de saturação para a aproximação  $i$ ;

- $tv_i$  é o tempo de verde efetivo na aproximação  $i$ .

O tempo de ciclo é a soma de todos os tempos de verde mais os tempos de transições. Então, segundo Webster, o tempo de verde ideal para a aproximação  $i$  é o dado pela Equação 19.

$$tv_{i0} = (y_i / Y) \cdot (tc - L) + l_i \quad (19)$$

onde:

- $tv_{i0}$  é o tempo de verde dado pela condição de EQUISAT na aproximação  $i$ ;
- $y_i$  é a relação entre fluxo medido e fluxo de saturação, ou seja,  $fm_i / fs_i$ ;
- $Y$  é o somatório de todos os  $y_i$  da intersecção;
- $L$  é o tempo perdido no ciclo (em transições);
- $l_i$  é o tempo perdido na no estágio da na aproximação  $i$  (na transição).

A seguinte equação apresenta a mesma relação considerando tempos de verde efetivos:

$$tv_{i0e} = (y_i / Y) \cdot tc_e \quad (20)$$

onde:

- $tv_{i0e}$  é o tempo de verde efetivo ideal na aproximação  $i$ ;
- $tc_e$  é o tempo de ciclo efetivo (descontado transições).

Nem sempre é possível a implementação de tempos que satisfaçam o princípio de saturações equivalentes, devido a arredondamentos (normalmente a resolução em tempo é de um segundo) e à configuração dos **tempos de verde mínimos**, que são parâmetros de segurança, conforme a Equação 21.

$$tv_i = \max (tv_{i0}, tv_{min}) \quad (21)$$

onde:

- $tv_i$  é o tempo de verde ajustado pelo o tempo de verde mínimo;
- $tv_{min}$  é o tempo de verde mínimo.

Nesse caso ( $tv_i - tv_{i0}$ ) deve ser reduzido dos demais tempos de verde de forma a não alterar os tempos de ciclo.

### 2.1.6 Simulação e Otimização

Simuladores de tráfego estimam as medidas de desempenho ou efetividade para uma rede de cruzamentos (subseção 2.1.4) através da análise de um determinado cenário programado. Um cenário é composto de medidas de fluxo, características físicas das vias (geometria e fluxos de saturação) e dos planos de tráfego de toda a região sob análise - tempos de ciclo, tempos de verde e defasagens.

Simuladores podem ser classificados como microscópicos, mesoscópicos e macroscópicos (ROUPHAIL, TARKO e LI, 1999).

Modelos microscópicos descrevem as entidades do sistema e suas interações em um alto grau de detalhamento, modelando o comportamento individual dos veículos como aceleração, frenagem e mudanças de faixa. São mecanismos de simulação estocástica, com geração randômica dos vários parâmetros. Demandam grande quantidade de dados e muito esforço computacional. Como exemplo de simulador microscópico tem-se o CORSIM (LIEBERMAN, 1997).

Simuladores mesoscópicos implementam modelos determinísticos de simulação que descrevem as entidades do sistema em um alto nível de detalhamento, mas suas interações num nível mais baixo (como, por exemplo, mudanças de faixa). Avaliam o sistema em passos de tempo (usualmente 1 segundo), onde para cada passo é estimado o fluxo entrando no *link*, se deslocando, parando, no caso de sinal vermelho, e avançando no verde. Implementam uma função de dispersão do pelotão para descrever a função de chegada em uma intersecção a partir da função de saída da intersecção anterior. Os primeiros modelos processavam um *link* por vez para todos os passos de tempo do ciclo (*link-based models*). Tais modelos não computavam as relações dos eventos de um ciclo para o ciclo seguinte. Em particular, não computavam o final das filas em relação ao tamanho do *link* (a fila sendo considerada um “empilhamento vertical”) e não consideravam a quebra da capacidade de estocagem de veículos e o fenômeno de bloqueio dos cruzamentos anteriores.

Modelos mais recentes (TRB, 2000) computam todos os *links* em cada passo de tempo (*step-based models*), podendo assim descrever melhor a dinâmica das filas e avaliar o fenômeno de fechamento, apresentando estimativas mais precisas em condições próximas à saturação. A precisão desses modelos depende do número de ciclos simulados, e, por isso, eles demandam maior esforço computacional. A mais importante ferramenta de simulação e otimização desse tipo chama-se Transyt (ROBERTSON, 1969).

Simuladores macroscópicos modelam entidades e interações de forma pouco detalhada. São modelos determinísticos de simulação do comportamento do fluxo de forma agregada, ciclo a ciclo, sendo os menos custosos do ponto de vista de esforço computacional, mas os mais simplistas e imprecisos. Não consideram a dinâmica das filas para situações próximas ao congestionamento. Como exemplo de simulador macroscópico tem-se o CORFLO (LIEBERMAN, 1997).

Embora as equações de Webster (subseção 2.1.5) possam ser utilizadas para determinação de tempos de ciclo e *splits* adequados, não existem equações para estimar diretamente as defasagens em uma malha de cruzamentos. Além disso, as equações de Webster não garantem a otimização dos índices de desempenho e são utilizadas usualmente para gerar a situação inicial de tempos de verde para o método de otimização adotado.

Muitas ferramentas de otimização existentes, como o Transyt, implementam a otimização de tempos baseada em simulação, que se dá pela geração sistemática de cenários, a utilização de simulador ou modelo matemático para cálculo de medidas de efetividade para cada cenário e a seleção do melhor cenário baseado nos critérios do método de otimização utilizado (CHAUDHARY, 2002). Ou seja, é a aplicação de algum algoritmo de busca sobre resultados dados por um simulador. A qualidade da estratégia de otimização adotada depende, então, do algoritmo de busca utilizado e da precisão do modelo de simulação.

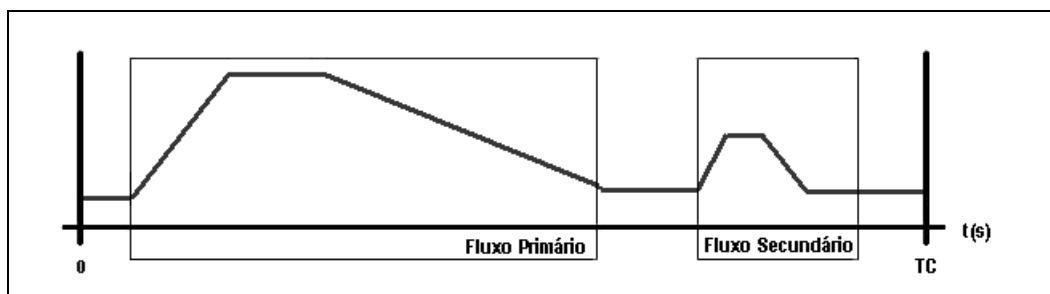
Os mecanismos mais usuais de busca para otimização baseada em simulação são busca exaustiva, *hill-climbing* ou método do gradiente ascendente e algoritmos genéticos (TTI, 2002).

No método de busca exaustiva, computa-se o valor da função objetivo a ser otimizada para todas as possíveis soluções. Dependendo do tamanho do espaço de busca, tempo de cálculo da função objetivo e capacidade de processamento, a busca pode demandar muito tempo para sua conclusão ou ser impraticável. Sem utilizar simplificações para acelerar o tempo de busca, esse método garante a obtenção da solução ótima.

Algoritmos de *hill-climbing* (método do gradiente ascendente) tomam por base um estado inicial e, incrementando e decrementando cada variável de um certo valor (*step size*), determinam, por comparação ao desempenho do estado inicial, a direção da próxima busca. O algoritmo continua até que a situação atual seja melhor que os cenários com incremento e decremento do passo. Usualmente o tamanho do passo nesses algoritmos é variável para acelerar o processo de busca. Algoritmos de *hill-climbing* garantem a obtenção da solução ótima apenas para funções unimodais (que não apresentam máximos ou mínimos locais). Para funções multimodais o algoritmo pode evoluir para um máximo local (subótimo) e a

qualidade da resposta depende do estado inicial apresentado. Algoritmos genéticos, cuja aplicação em tráfego é mais recente, são apresentados na seção 2.2.

Ilustrando um processo de otimização que utilize um simulador mesoscópico, denomina-se **fluxo primário** aquele de maior importância na composição do fluxo chegando ao cruzamento por uma aproximação, normalmente relativo ao movimento que segue adiante no cruzamento anterior. **Fluxos secundários** são então os movimentos de conversão no *link* anterior que também alimentam o *link* em questão. Um modelo de simulação de tráfego implementa uma função que transforma a composição dos fluxos primário e secundários no padrão de chegada do fluxo no cruzamento, através de uma função de dispersão do pelotão (WALLACE et al., 1999), conforme a Figura 7.



**Figura 7:** Função de chegada de veículos no cruzamento

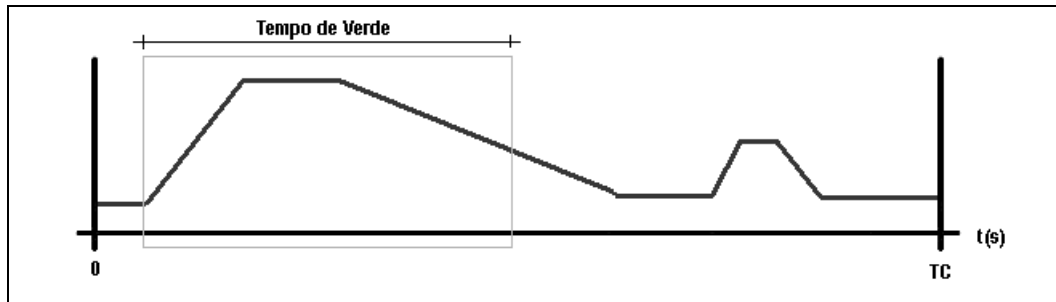
Tipicamente, em um modelo mesoscópico, os fluxos anteriores são discretizados em intervalos de 1 segundo, em sua janela de verde, e aplica-se uma função de estimação para determinar a sua chegada no *link* seguinte.

O padrão de fluxo de chegada, aliado às variáveis de controle no cruzamento (tempos e defasagem), determinará os índices de desempenho para um *link*. Particularmente no que se refere à busca da melhor solução de defasagem para um cruzamento, uma vez fixados os tempos de verde, a questão é como encaixar a janela de verde ao fluxo de chegada de forma a minimizar o número de paradas, os atrasos e o tamanho de filas. Por exemplo, para a Figura 7, um exemplo de solução para a janela de verde seria a ilustrada na Figura 8.

A fixação de um intervalo de verde para um *link* determina a janela de verde para os demais *links* do cruzamento, então todas as aproximações devem ser consideradas na determinação da defasagem do controlador, que deve otimizar o somatório dos índices de desempenho de suas aproximações.

O Transyt é talvez a mais popular ferramenta de simulação e otimização *offline* empregada nas últimas décadas, originalmente desenvolvida por Robertson (ROBERTSON,

1969) na Inglaterra. Nesse período diversas versões aprimoraram tanto seu modelo de simulação como também seus métodos de otimização. O Transyt 7F foi uma versão “americanizada” criada pelo *Transportation Research Center* da Universidade da Flórida em 1981, derivada do Transyt original em sua versão 7.



**Figura 8:** Alocação de janela de verde

O Transyt 7F contém um simulador mesoscópico aliado a um otimizador de tempos. Originalmente seu modelo era do tipo *link-based* e não tratava bem a dinâmica das filas. A partir do Transyt 7F Release 8.2 (1998) um modelo *step-based* foi incluído, aumentando a precisão das estimativas próximas à saturação. Nele, atrasos são estimados de acordo com o modelo de HCM (TRB, 2000) que inclui componentes de atrasos uniformes, randômicos e de saturação (subseção 2.1.4). Assim, as estimativas de atrasos em saturação são mais realistas do que nos modelos anteriores, embora mais imprecisas à medida que a saturação aumenta. Se tamanhos máximos de fila sobrepõem as capacidades de estocagem, penalidades são aplicadas na avaliação de um cenário.

A estratégia de otimização no Transyt consiste em uma busca exaustiva para o tempo de ciclo (se o tempo de ciclo não for configurado como fixo), onde para cada tempo de ciclo os tempos de verde iniciais são estabelecidos por EQUISAT (ou pré-determinados), sendo então realizada otimização por *hill-climbing* dos tempos de verde e defasagens. A função objetivo é uma ponderação parametrizável de estimativas, como atrasos, paradas e filas. Como característica do processo de *hill-climbing*, a qualidade da resposta depende de bons valores de defasagem configurados como condição inicial.

Recentemente, otimização por Algoritmos Genéticos (ver seção 2.2) foi introduzida no TRANSYT 7F, em seu *release* 9 (2003) para otimização de tamanho de ciclo, seqüência de fases, tempos de verde e defasagens, dispensando a entrada de uma condição inicial (conjunto de planos) adequada.



### 2.1.7. Estratégias de Controle de Tráfego

Pode-se classificar os sistemas de controle de tráfego existentes em três tipos: horário, reativo e adaptativo (GARBACZ, 2002).

No controle horário (ou controle a tempos fixos), os planos de tráfego são gerados manualmente ou por otimizador *offline* e pré-programados nos controladores. A geração desses planos é feita levando-se em conta as medidas históricas de fluxo de veículos para diferentes horários, e por dia de semana. A obtenção desses dados pode ser feita por detectores, mas na ausência deles é realizada por observação.

Como vantagem do controle horário tem-se o baixo custo de implantação (não demanda detectores e nem um sistema de controle central) e a simplicidade e segurança pelo seu comportamento determinístico. Evidentemente, o controle horário não permite a adaptação automática a mudanças nas condições de tráfego. Para condições normais e não saturadas, em que o tráfego tem um padrão bastante repetitivo, apresenta bons resultados, mas demanda contínua reavaliação dos planos (a cada 1 a 3 anos) para adaptação às novas condições.

O controle do tipo reativo (ou seleção dinâmica de planos) é similar ao controle horário, mas os planos não são selecionados em função do horário, e sim em função dos fluxos medidos. Este método demanda número moderado de detectores, e permite reação à ocorrência de certas situações anormais, mas está limitada pelo universo de planos gerados *offline*, e, da mesma forma que o controle horário, demanda a reavaliação periódica dos planos; ao longo dos anos os planos se tornam obsoletos.

Por fim, o controle adaptativo (ou geração dinâmica de planos) é mais aprimorado, porém de maior custo, demandando detectores em todas as aproximações. Variáveis dos planos são ajustadas automaticamente pelo sistema de controle em função das condições reais do trânsito.

Um sistema adaptativo busca, pela acomodação automática à variação de demanda, a melhoria dos índices de desempenho do sistema, minimizando as filas, o número de paradas, os tempos de viagem, a emissão de poluentes, entre outros, além de gerenciar os problemas decorrentes de padrões de tráfego não usuais, como em jogos, manifestações e obras na vias públicas.

### 2.1.8. Sistemas de Controle Adaptativo

Os sistemas de controle adaptativo mais populares hoje (GARBACZ, 2002) são o SCOOT (*Split Cycle Length Offset Optimization Technique*) e o SCATS (*Sydney Coordinated Adaptive Traffic System*), que implementam diferentes abordagens.

O SCOOT (HUNT, 1991) é considerado uma versão *online* do Transyt. Sua primeira instalação data de 1979 na Inglaterra, tendo sido muito aprimorado desde então, e sendo hoje aplicado em mais de 150 cidades no mundo (PAPAGEORGIOU, 2003). É um sistema de arquitetura centralizada que, através de modelo de simulação, gera um índice de desempenho por estimativas de atrasos, paradas e filas, a pesos configuráveis. Realiza as alterações nos tempos em uma série de pequenos incrementos, pela comparação das estimativas do cenário atual com cenários alternativos onde os incrementos de tempo são aplicados. Demanda alta disponibilidade da estrutura de comunicação e dos detectores, acomodando até 15% de perda de informação e a partir disso regredindo para modo horário. A cada *link* pode ser atribuído um peso na composição do índice de desempenho.

O modelo de simulação do SCOOT baseia-se em dados de contagem que são enviados ao sistema central a cada quatro segundos com os quais o SCOOT estima a função que descreve o perfil de chegada no cruzamento, o número de veículos chegando no sinal vermelho e o tamanho da fila formada. A velocidade dos veículos e a taxa de descarga das filas são parâmetros de configuração. Os detectores localizam-se no início dos *links* e os dados de ocupação são utilizados para a detecção de congestionamento. Como desvantagem tem-se seu alto custo, pois sua licença é da ordem de centenas de milhares de dólares.

O SCATS (SIMS, 1979) é utilizado em diversas cidades da Austrália, Ásia e Estados Unidos. Ao contrário do SCOOT, o SCATS é executado de forma distribuída, em controladores que comandam subáreas. Não possui um modelo de simulação para otimização de planos, mas é um método de realimentação incremental. Possui biblioteca de *splits* pré-programada sendo escolhido o mais próximo da condição de EQUISAT, pela avaliação do grau de saturação em cada *link*. O tempo de ciclo é determinado para cada região baseado no grau de saturação do cruzamento mais congestionado. Os *splits* representam um percentual do tempo, sendo ponderados para o de ciclo calculado. *Offsets* são interpolados entre valores pré-programados de ciclos, considerando o tempo de ciclo utilizado. A otimização de *offsets* não é tão eficiente quanto à do SCOOT, interrompendo a progressão de pelotões. O SCATS é igualmente sensível à perda de dados de detectores, mas, por ter processamento distribuído,

apresenta menores requisitos de disponibilidade e de poder de processamento do que a arquitetura centralizada baseada em simulação do SCOOT.

Mais recentemente foram desenvolvidas estratégias adaptativas que não consideram explicitamente ciclo, *splits* e defasagem: OPAC, PRODYN, CRONOS e RHODES (PAPAGEORGIU, 2003). Essas estratégias baseiam-se em estágios pré-programados, calculando em tempo real os tempos ideais de mudança de estágio sobre um horizonte de tempo futuro, considerando o instante de tempo atual e o estágio em vigor. Para obter os tempos ótimos para as mudanças de estágio, resolve-se em tempo real um problema de otimização dinâmica, considerando-se restrições como os tempos mínimos e máximos de estágios. O índice de desempenho típico a ser minimizado é o tempo de deslocamento para todos os veículos. O problema de otimização é resolvido para um horizonte de tempo  $H$ , por exemplo, de 60 segundos, utilizando condições iniciais obtidas por medições do fluxo e predição de demanda sobre  $H$ . Entretanto, o resultado obtido é aplicado por um intervalo menor de tempo  $h$ , por exemplo, de 4 segundos, após o qual novos dados são coletados e novo problema é resolvido para um horizonte  $H$ , e assim sucessivamente. Chama-se essa abordagem de horizonte móvel (*rolling horizon*). Devido à complexidade exponencial dos algoritmos, esses métodos, embora conceitualmente aplicáveis ao sistema como um todo, não são aplicáveis em tempo real para mais de uma intersecção. Constituem, portanto, estratégias de otimização descentralizadas, por cruzamento, cujas ações podem ser coordenadas heurísticamente por uma camada de controle superior (PAPAGEORGIU, 2003).

## 2.2 ALGORITMOS GENÉTICOS

### 2.2.1 Conceitos Básicos

Algoritmos genéticos (*genetic algorithms* – GA) são algoritmos de busca baseados no mecanismo de seleção natural, que buscam sistematizar um processo de busca randômica. Originalmente proposto por Holland (1975) hoje é um dos mais explorados métodos de otimização em diversas áreas de pesquisa. Apresenta a vantagem de não tender a máximos locais como ocorre com métodos de gradiente local (*hill-climbing*). Uma apresentação detalhada dos tópicos de GA pode ser encontrada em (GOLDBERG, 1998).

A modelagem de um mecanismo de otimização em GA começa pelo processo de **codificação**: esquema de transformação de variáveis (pontos no espaço de discurso contínuo ou discreto) constituintes de uma solução possível para o problema em cadeias de bits. Cada valor é codificado em um **gene**. Agrupamentos de genes são denominados **cromossomos**. Uma solução possível do problema compõe-se de um ou mais cromossomos e é denominada **indivíduo**. Ou seja, um indivíduo é representado por uma cadeia de bits, divididos em blocos que representam cada valor de uma dada solução do problema. Os métodos de codificação diferem em cada aplicação e compõem a forma de traduzir o conhecimento do problema para o modelo GA (JANG, 1997).

O processo de otimização por GA demanda uma **função de fitness**. Esta é a normalização/translação de alguma função objetivo de forma que seja monotônica, positiva, usualmente em uma escala de 0 a 1. A função de *fitness* é aplicável a cada indivíduo e traduz o grau de eficiência deste indivíduo. Mesmo sem a capacidade de descrever o sistema-alvo de forma precisa, necessita-se uma função que permita avaliar a qualidade de cada resposta (indivíduo), função essa que será otimizada.

No início do processo de otimização é necessário construir uma **população** - conjunto de indivíduos (soluções possíveis do problema) – de forma aleatória. A partir da população inicial, implementa-se a **seleção** dos indivíduos de melhor *fitness*, a recombinação de seus cromossomos (**crossover**), a aplicação de **mutações** aleatórias sobre os cromossomos recombinados e a criação, assim, de outra população (segunda **geração**). O tamanho da população, a forma de seleção, o tipo de *crossover*, a probabilidade de mutação e de *crossover*, o **elitismo** (manutenção dos melhores indivíduos da geração anterior) e número de gerações são parâmetros do mecanismo de GA, determinados por heurísticas e/ou tentativa e erro.

A operação de *crossover* visa recombinar as boas cadeias de codificação de indivíduos de bom *fitness*, para gerar indivíduos ainda melhores. Correspondem a buscas locais. A operação de mutação busca variações aleatórias nas populações, correspondendo a buscas globais. As probabilidades de *crossover* e mutação em cada geração determinam uma busca mais localizada ou não.

Existem diversas variantes, como a evolução de populações em paralelo e recombinação de seus melhores indivíduos, viabilizando otimização em múltiplas máquinas em paralelo nos casos onde grande capacidade de processamento é necessária para cálculo do *fitness*.

Pode-se também combinar GAs com algum método de busca local, e produzir um mecanismo híbrido de otimização. Assim, vale-se da capacidade de busca global do GA aliada à convergência ao máximo local por outro método.

### 2.2.2 Aplicação de GA em Otimização de Tempos Semafóricos

O espaço de busca para otimização de tempos em sistemas de tráfego é muito grande. Ilustrando, considerando uma malha composta por 30 semáforos, supondo um ciclo constante de 90 segundos e os tempos de verde definidos *a priori* em função dos níveis de saturação nos *links* conflitantes, onde somente é necessário determinar a defasagem, tem-se 30<sup>90</sup> possibilidades de resposta que poderiam ser avaliadas em modelo de simulação, o que evidentemente não é computável por método de busca exaustiva.

Otimizadores de planos baseados em simulação como o Transyt otimizam um índice de desempenho através de algoritmo de otimização *hill-climbing*. Esse tipo de busca tem efetividade limitada no espaço de busca total, conduzindo a máximos locais (que podem estar muito longe do máximo global do sistema). Demanda-se, assim, a programação de um conjunto inicial de valores coerente para a obtenção de boa resposta, ou seja, demanda conhecimento especializado.

Algoritmos Genéticos são adequados para otimização nessa dimensão de espaço de busca. Mesmo que não assegurem a obtenção do máximo global, tendem a evitar máximos locais, sem demandar o conhecimento especializado para a determinação do estado inicial a ser otimizado.

A primeira proposta de utilização de GAs na otimização de tempos semafóricos foi apresentada em (FOY, BENEKOHAL e GOLDBERG, 1992), com o objetivo de realizar buscas mais eficientes no enorme espaço de respostas possíveis para sistemas de tráfego. Otimização de tráfego por GA demanda mecanismos de simulação para obtenção de medidas de desempenho e determinação da função de *fitness* para cada solução possível. O artigo citado coloca que o simulador utilizado foi muito simplificado, pois seu foco era apresentar a viabilidade da utilização de GA. A utilização de simuladores mais precisos foi proposta como um avanço. A forma de codificação das variáveis tornava sua execução um tanto complexa para grandes redes (no trabalho era considerada uma rede de apenas 4 cruzamentos). Mais recentemente otimizadores como o Transyt e Passer passaram a incluir GA em suas estratégias de otimização (TTI, 2002).

## 2.3 ANFIS

### 2.3.1. Introdução a Sistemas de Inferência Fuzzy

Sistemas de inferência fuzzy (*fuzzy inference systems* - FIS) são modelos computacionais baseados em conceitos de teoria de conjuntos *fuzzy*, regras *fuzzy* e raciocínio *fuzzy*, compreendendo (JANG, 1997):

- Base de regras: conjunto de regras *fuzzy*;
- Base de dados: definição das funções de pertinência usadas nas regras *fuzzy*;
- Mecanismo de raciocínio: procedimento de inferência sobre as regras e fatos para gerar saídas (ou conclusões) adequadas.

A apresentação de conceitos relativos à teoria de conjuntos *fuzzy* – primeiramente propostos por Zadeh (1965), está além do escopo deste documento e podem ser encontrados em (PEDRYCZ e GOMIDE, 1998).

Regras Fuzzy (ZADEH, 1965) são expressões na forma:

- SE <antecedente> ENTÃO <conseqüente>

Antecedentes e conseqüentes são proposições do tipo “ $x$  é  $X$ ” onde  $x$  é uma variável lingüística e  $X$  um valor *fuzzy* associado à uma função de pertinência. Por exemplo “SE pressão é alta ENTÃO temperatura é baixa”. As regras podem ter múltiplas proposições como antecedente e conseqüente.

Regras *fuzzy* são muito apropriadas para representar o conhecimento lingüístico/empírico, assim como o modo de raciocínio humano que nos fazem capazes de concluir sobre ambientes imprecisos e complexos (JANG, 1993), mesmo sem regras e dados exatos. Outra forma de expressão de regras *fuzzy*, proposta por Takagi e Sugeno (1983) apresentava nos conseqüentes uma expressão não *fuzzy* das variáveis de entrada. Ambos modelos são muito utilizados em aplicações de controle. Além de poderem representar conjuntos de associações imprecisas que constituem as regras que determinam o mecanismo de raciocínio, as regras *fuzzy* (JANG, 1993) podem ser encaradas como uma descrição local de um sistema em consideração, pois representam uma conclusão sobre determinadas regiões do espaço de discurso das variáveis.

Sistemas de inferência *fuzzy* baseiam-se no mecanismo de raciocínio *fuzzy*, ou raciocínio aproximado – raciocínio por regras e fatos imprecisos, ou seja, descritos por funções de pertinência – e, por conseqüência, na regra composicional de inferência (ZADEH, 1973).

Os passos do processo de raciocínio *fuzzy* são (JANG, 1993):

- Fuzzificação: obtenção das funções de pertinência para cada variável de entrada;
- Combinação (por norma-T) das funções de pertinência de entrada nas premissas para obter os valores de ativação de cada regra;
- Geração de um conseqüente para cada regra, de acordo com os valores de ativação;
- Defuzzificação: agregação dos conseqüentes para produção de uma saída não *fuzzy*.

Existem diversos modelos de sistemas *fuzzy* na literatura, dependendo do tipo de regra *fuzzy* e do mecanismo de raciocínio, sendo a principal classificação de FISs:

- Modelo Mamdani: apresenta conjuntos *fuzzy* em antecedentes e conseqüentes das regras. A saída final, representada por conjunto *fuzzy*, resulta da agregação das saídas das diversas regras. Demanda processo de defuzzificação para obtenção de saída não *fuzzy*;
- Modelo Sugeno (TSK): conseqüentes das regras são uma combinação linear das variáveis de entrada mais uma constante, e a saída final é a média ponderada (em função do nível de ativação) da saída de cada regra;
- Modelo Tsukamoto: utiliza funções de pertinência monotônicas nos conseqüentes. Valor não *fuzzy* é induzido pelo nível de ativação de cada regra, e a saída final é obtida pela média ponderada da saída de cada regra.

### 2.3.2. A Arquitetura ANFIS

ANFIS (*Adaptive Network-based Fuzzy Inference System*) é um processo proposto por Jang (1991) de mapeamento de um sistema de inferência *fuzzy* (particularmente no Modelo TSK, embora seja possível a extensão para os demais) em uma rede neural. Ou seja, consiste em construir uma rede neural equivalente a um FIS de forma que seus parâmetros

correspondam às variáveis do FIS (funções de pertinência de entrada e parâmetros das combinações lineares das saídas das regras) e possam ser ajudados por dados de treinamento (dados de entrada e saída, cuja função de transferência busca-se aproximar).

Jang (1991) propõe um mecanismo de treinamento híbrido, baseado no método convencional do gradiente descendente aliado ao método de mínimos quadrados para acelerar o processo de treinamento e evitar a possibilidade do método do gradiente tender a mínimos locais.

Ilustrando, transcreve-se o exemplo dado por Jang (1993) para a tradução de um FIS em uma rede neural. Por simplicidade, são consideradas duas entradas para o sistema ( $x$  e  $y$ ), uma saída ( $z$ ) e uma base de regras composta de duas regras do tipo Sugeno:

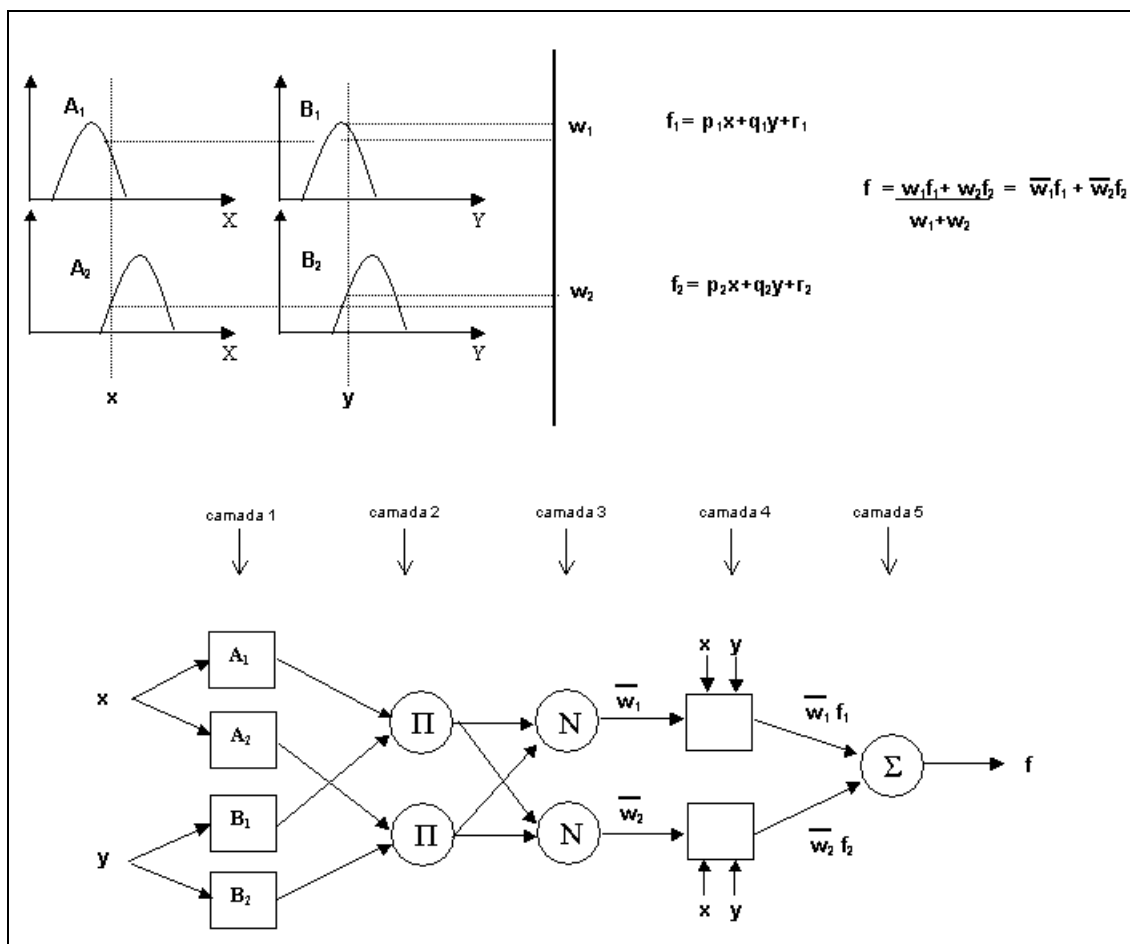
- Regra 1: Se  $x$  é  $A_1$  e  $y$  é  $B_1$  então  $f_1 = p_1x + q_1y + r_1$ .
- Regra 2: Se  $x$  é  $A_2$  e  $y$  é  $B_2$  então  $f_2 = p_2x + q_2y + r_2$ .

A Figura 9 apresenta o FIS proposto, representando o mecanismo de inferência, e a rede neural equivalente, em cinco camadas. Na rede neural equivalente, nodos quadrados são adaptativos, ou seja, possuem parâmetros a serem ajustados no processo de treinamento, enquanto nodos circulares são fixos. As funções dos nodos de cada camada são:

- Camada 1: representa as funções de pertinência das regras;
- Camada 2: multiplicação (ou norma-T para produzir uma operação “E” genérica) entre os valores das funções de pertinência para cada entrada resultando no nível de ativação da regra;
- Camada 3: razão entre nível de ativação da regra e a soma de todos os níveis de ativação;
- Camada 4: produto do conseqüente da regra pelo nível de ativação normalizado;
- Camada 5: somatório dos conseqüentes ponderados pelos níveis de ativação.

O ANFIS propõe aumentar a precisão das respostas de FISs, ao custo de alguma perda de interpretabilidade (as funções de pertinência otimizadas podem carecer de significado). O processo de treinamento da rede neural da Figura 9 equivale à otimização das funções de pertinência de entrada e dos parâmetros de saída para melhor aproximar a resposta do FIS aos dados de treinamento (pares entrada/saída).





**Figura 9:** Arquitetura ANFIS

## 2.4. CONCLUSÃO

Neste capítulo buscou-se detalhar os conhecimentos necessários à consecução dos objetivos deste trabalho de dissertação de mestrado, descritos no Capítulo 1.

Foram apresentados os conceitos de engenharia de tráfego relevantes para a compreensão do problema da otimização de tráfego e identificadas algumas ferramentas de otimização existentes, tanto *online* como *offline*. Essas ferramentas servirão como referência na concepção do modelo proposto e também na sua avaliação. Em particular o SCOOT servirá como paralelo na definição da estratégia de controle adaptativo, enquanto o Transyt será utilizado para validação dos resultados obtidos e na geração de dados de treinamento.

Também foi incluída neste capítulo uma breve apresentação de duas ferramentas utilizadas na abordagem proposta: algoritmos genéticos e ANFIS. Essa abordagem será detalhada no próximo capítulo.



## CAPÍTULO 3

### ABORDAGEM PROPOSTA

Apresenta-se neste capítulo a proposta e o escopo deste trabalho, baseados nos conceitos apresentados no Capítulo 2 e nas características do sistema de controle de tráfego da cidade de Curitiba, descritas na seção 3.2. Busca-se enfatizar os aspectos que tornam a proposta original e as restrições e premissas do sistema que influenciaram sua concepção. A seção 3.3 detalha a abordagem proposta e na seção 3.4 apresenta-se a justificativa das escolhas tomada na sua construção.

#### 3.1 OBJETO DA PROPOSTA

Considerando as restrições do sistema de controle de tráfego de Curitiba apresentadas na seção 3.2, busca-se implementar um modelo de otimização de tempos dos semáforos (planos de tráfego) que viabilize um controle adaptativo. Entende-se por controle adaptativo o ajuste automático dos tempos dos semáforos às variações de demanda, ou seja, às variações dos fluxos de veículos nos diversos *links* do sistema.

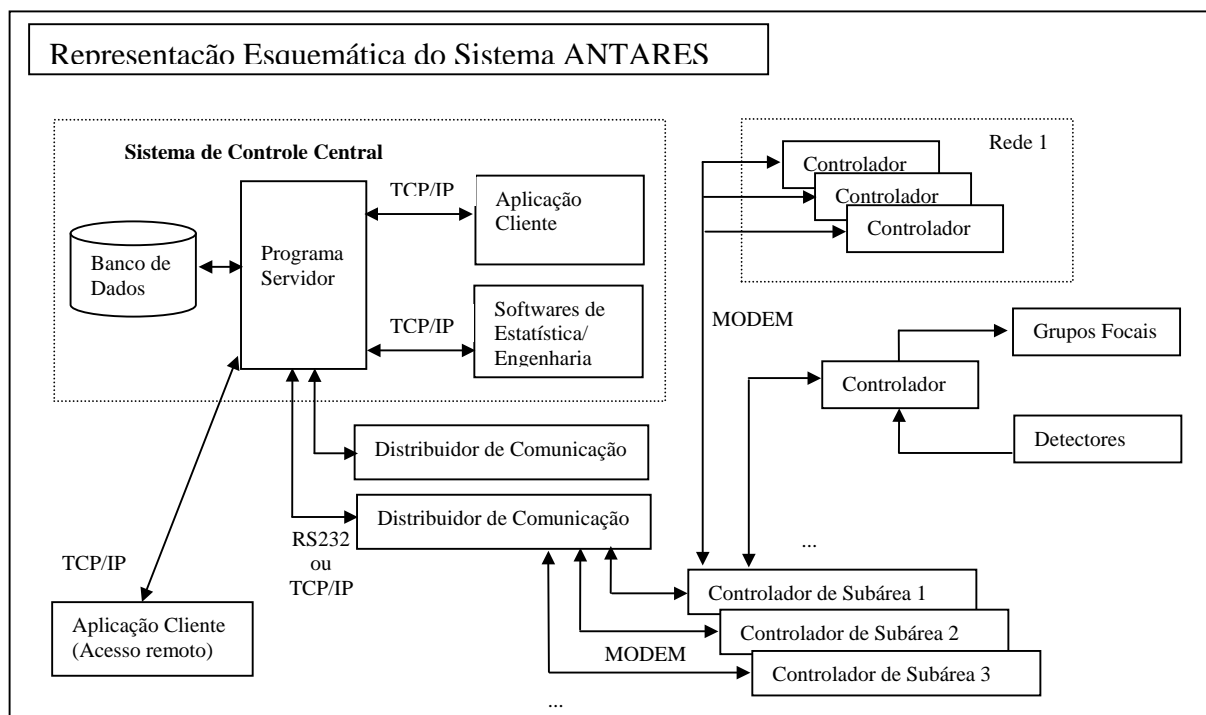
Propõe-se a criação de um modelo que:

- Atue pela otimização de um índice que reflita a eficiência do sistema;
- Demande pouco conhecimento do especialista na sua configuração: idealmente deve-se configurar apenas características físicas das vias e parâmetros de segurança do sistema;
- Atenda restrições de tempo de execução indicadas na seção 3.2 para um grande número de cruzamentos;
- Trabalhe com dados de entrada como fornecidos pelos detectores de Curitiba.

#### 3.2 ARQUITETURA ATUAL DO SISTEMA DE CURITIBA

O sistema de controle de tráfego de Curitiba é composto por equipamentos organizados em três níveis hierárquicos (DATAPROM, 1999): controladores locais,

controladores de subárea e sistema de controle central. Controladores locais são os equipamentos que efetivamente comandam os grupos focais nos cruzamentos e obtêm dados de detectores de veículos. Podem assumir comportamento programado localmente ou comandado por sistema de controle central. Controladores de subárea desempenham a função de roteadores de comunicação e realizam a sincronização dos relógios dos controladores locais de uma subárea. O sistema de controle central comunica-se com os diversos controladores de subárea e, através destes, com os controladores locais, para cumprir os objetivos de monitoração e controle, fornecendo informação e ferramentas às equipes de manutenção e engenharia de tráfego. O sistema de controle central de Curitiba utiliza plataforma PC com Windows 2000 e apresenta arquitetura cliente/servidor. A Figura 10 ilustra a arquitetura geral do sistema.

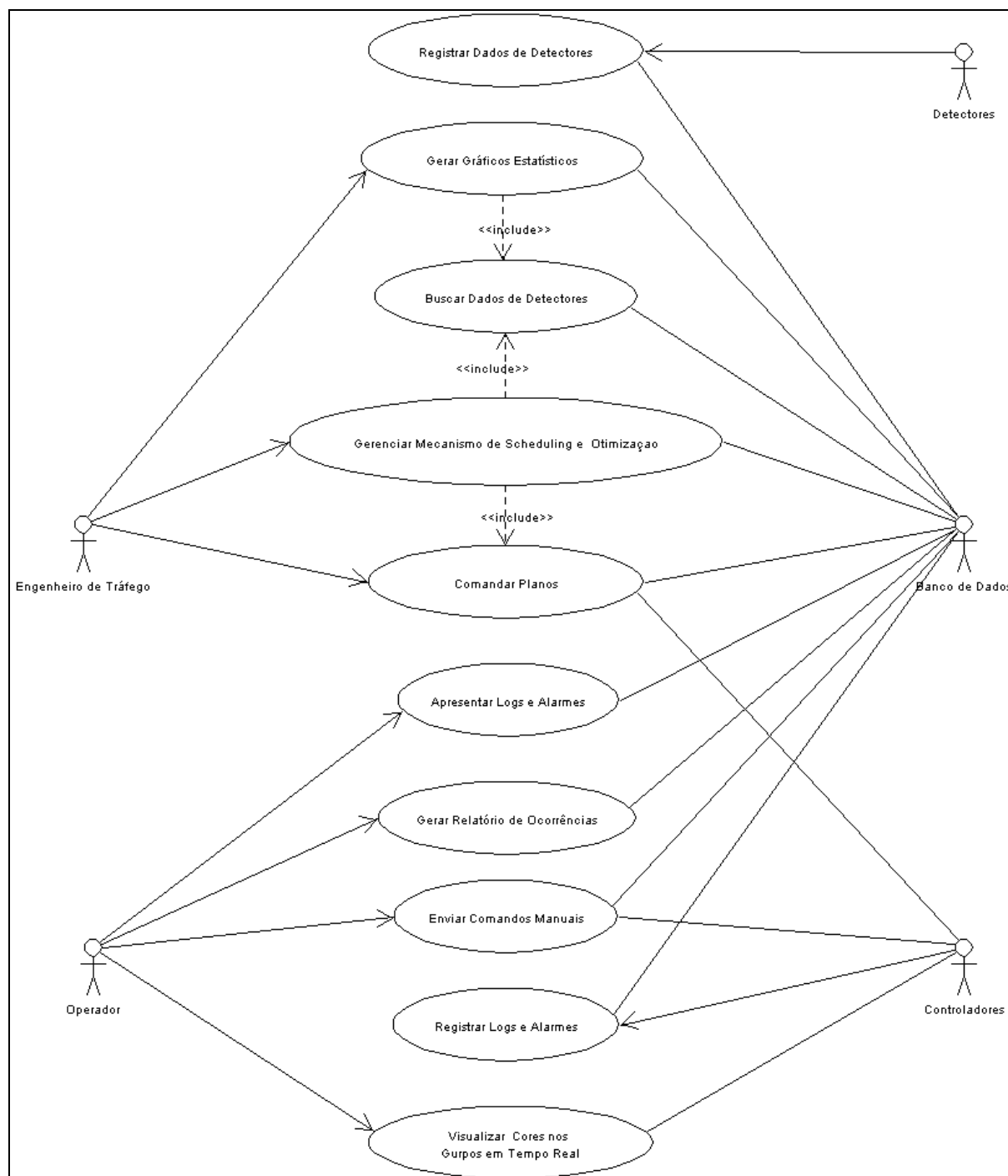


**Figura 10:** Arquitetura do Sistema Antares

A Figura 11 apresenta um diagrama de casos de uso representando os serviços essenciais disponibilizados pelo sistema de controle central.

Diferentes sistemas de controle implementam diferentes formas de instalação de detectores e de tratamento de seus dados. No caso do sistema de Curitiba, a instalação se dá com um laço indutivo por faixa de rolamento, localizado no início do *link*, conforme a Figura 12. Essa escolha foi feita para que a informação de ocupação dos detectores indique se as filas chegaram ao final da aproximação, ou seja, se o *link* está congestionado. Assim, além da

informação de contagem, tem-se a informação de detecção de congestionamento, a partir da qual é computado o somatório de metros congestionados para uma subárea. Essa é a estimativa de desempenho global fornecida automaticamente pelo sistema, utilizada para avaliar o impacto da utilização de diferentes estratégias. A aquisição dos dados de detectores é executada a cada ciclo. Os valores de contagem e ocupação são integrados ao longo do ciclo.

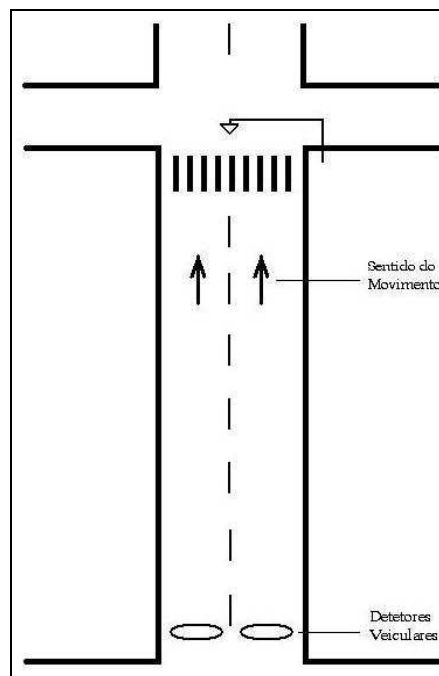


**Figura 11:** Diagrama de casos de uso do sistema de controle

A Figura 13 apresenta as medidas de contagem de veículos por minuto (gráfico superior) e ocupação em percentual de tempo (gráfico inferior) para uma via composta por três faixas de rolamento, ao longo do dia, com escala de tempo horizontal de 0 a 24 horas, e valores registrados a cada 5 minutos. As medidas são dadas pelo sistema de controle de tráfego de Curitiba – Sistema Antares – para medições efetuadas em 15 de setembro de 1999.

Considerando a forma de instalação de detectores em Curitiba (no final dos *links*), a Figura 13 nos permite observar que:

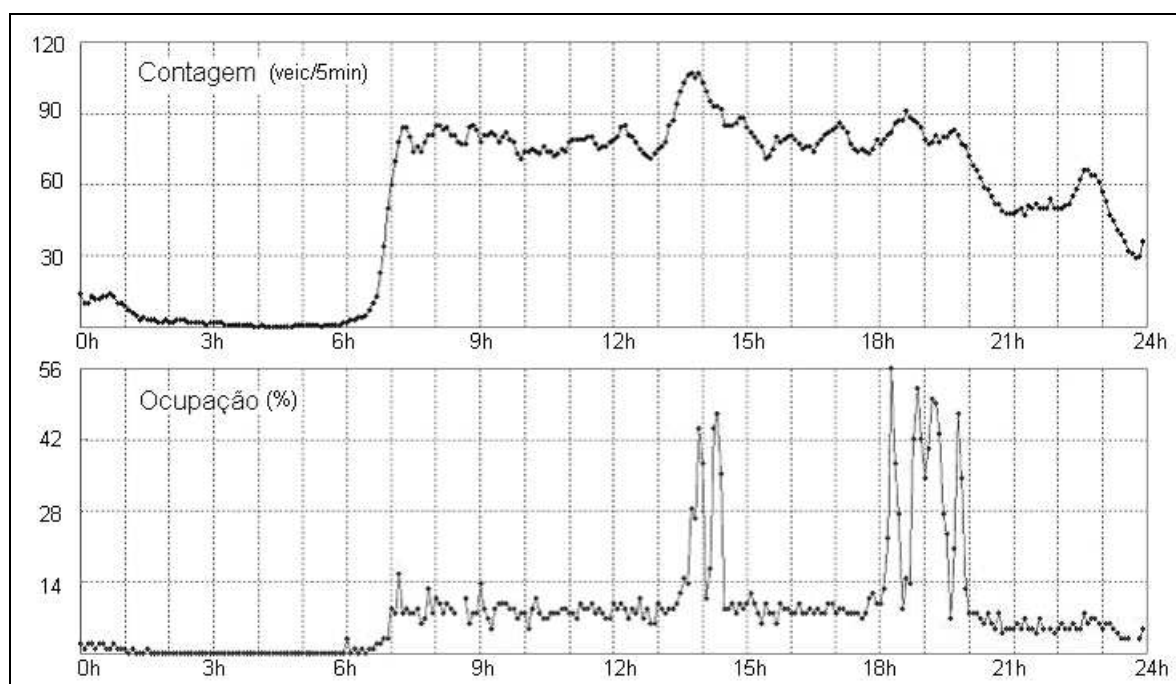
- O valor de contagem de veículos nunca supera certo patamar equivalente à capacidade do *link*;
- Valores de ocupação apresentam picos correspondentes às situações de congestionamento (momentos em que a fila atinge o início do *link* e pára sobre o detector);
- Valores de ocupação e contagem apresentam relação linear enquanto fora da condição de congestionamento. No congestionamento a relação pode se inverter, com picos locais de ocupação correspondendo a vales de contagem;
- Pode-se determinar um limiar para o valor de ocupação, que sinalize a situação de congestionamento (por exemplo, 20%).



**Figura 12:** Posicionamento dos detectores

O esquema de otimização dinâmica hoje existente é baseado em controle de filas, onde regras são configuradas para cada cruzamento, considerando os estados dos detectores circundantes (se estão congestionados ou não) para a alteração gradativa e sistemática dos tempos de verde, beneficiando os tempos dos estágios relativos às aproximações congestionadas em detrimento das não congestionadas.

As características dos dados de detectores impõem algumas restrições. Primeiramente, a informação, tanto de contagem como de ocupação, é integrada no ciclo: não se tem a curva do perfil do fluxo ao longo do tempo. No sistema SCOOT (HUNT, 1991) a aquisição é feita a cada quatro segundos, o que permite ao modelo ser preditivo com relação à formação de filas, estimando o número de veículos que chegam durante o tempo de vermelho (e que sofrem paradas) e possibilitando ao sistema atuar sobre as variáveis dos controladores dentro do mesmo ciclo. No caso de Curitiba, medições somente poderão ser consideradas para otimizações no ciclo seguinte à aquisição.



**Figura 13:** Gráficos de contagem e ocupação

Outra limitação é que não se dispõe de informação em tempo real do percentual do volume que é originário de conversões. Como não se pode decompor o fluxo global nos movimentos parciais – fluxo primário e fluxos secundários – tem-se menor precisão nas estimativas de desempenho, pois se deve assumir esses valores em termos de médias históricas. Isso não seria necessário se a aquisição fosse feita em pequenos intervalos de

tempo, como faz o SCOOT, pois assim seria possível estimar as filas sem a informação de conversões, simplesmente utilizando o perfil do fluxo medido.

O sistema de controle de tráfego de Curitiba implementa atualmente dois mecanismos de otimização:

1. Totalização dos valores de contagem nos diversos sentidos de uma região. Esses valores são verificados em uma tabela que relaciona faixas de valores de fluxo e planos comandados. Essa abordagem implementa uma estratégia de seleção dinâmica de planos. Como apresentado na subseção 2.1.7, esse tipo de estratégia apresenta vantagens em relação à seleção horária de planos, mas demanda trabalho constante de reavaliação da base de dados de planos e, também, a correta configuração de inúmeros parâmetros por especialista, baseado em heurísticas.
2. Controle de filas, sendo implementado em regras baseadas nos dados de ocupação. Esses dados provêm uma indicação “binária” de congestionamento de cada aproximação (se a fila atingiu o detector) obtida comparando os dados de ocupação com limiares pré-estabelecidos. A partir da comparação do estado das diferentes aproximações, são estabelecidas regras para incrementos/decrementos de tempo nos estágios, de forma a beneficiar as aproximações congestionadas. O método oferece resultados, mas com a limitação de responder somente quando a situação de congestionamento já é uma realidade.

Dadas as características dos processos de aquisição de dados e comando de planos, a reavaliação de planos deve estar contida em uma janela de tempo de poucos segundos. O diagrama de estados representado na Figura 14 descreve os processos executados no Sistema de Controle Central pela *thread* de gerenciamento dos mecanismos de *scheduling* (seleção em função do horário) e otimização de planos. Esse mecanismo sincroniza as atividades de:

- Varredura de dados de detectores;
- Determinação de planos;
- Comando de planos.

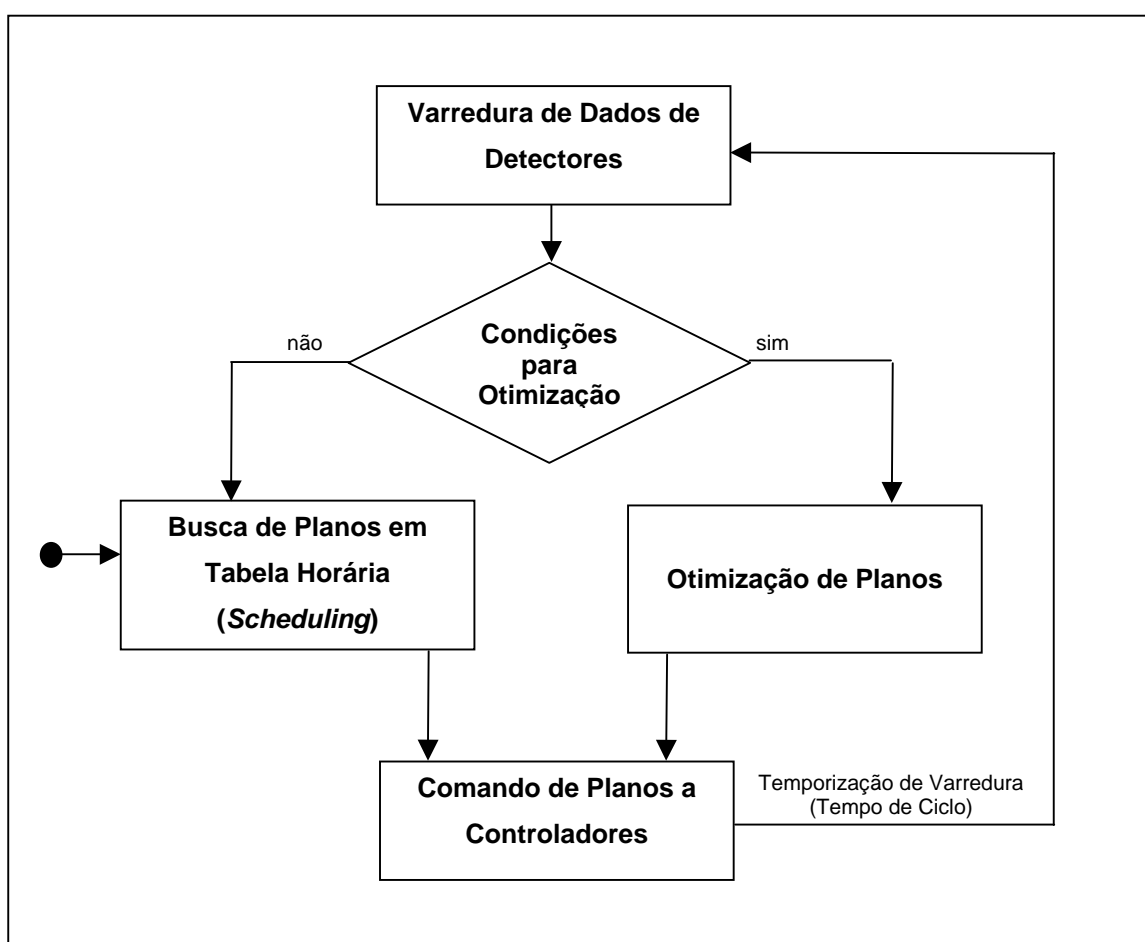
Para que o sistema possa responder a cada ciclo, essas atividades são executadas seqüencialmente a cada período de varredura. A amostragem dos dados de detectores deve ser executada de forma a totalizar dados em um período igual ao tamanho do ciclo, pois de outra



forma os valores sofreriam variações em função dos intervalos de tempo em que fossem colhidas.

As condições para que seja executada a otimização dinâmica são:

- Mecanismo de otimização dinâmica deve estar habilitado na configuração do sistema para o período em questão;
- Inexistência de comando manual do operador inibindo o mecanismo;
- Inexistência de controladores em falha de comunicação;
- Inexistência de detectores em falha;
- Inexistência de temporizações de proteção em curso.



**Figura 14:** Ciclo de aquisição, otimização e comando por sub-área

Pode existir tolerância a certo número de falhas em controladores ou detectores, absorvidas pelo sistema que passa a se valer de dados históricos. Nesse caso, podem ser configurados cruzamentos críticos, para os quais qualquer falha conduza imediatamente ao modo de seleção horária.

Quanto aos requisitos temporais de execução, cada execução do algoritmo de otimização representado na Figura 14 deve ter duração máxima igual ao tempo de ciclo da subárea e assim deve-se observar a relação dada pela Equação 22.

$$T_o < TC - (T_a + T_c) \quad (22)$$

onde:

- $T_o$  é o tempo de execução dos algoritmos de otimização;
- $TC$  é o tempo de ciclo;
- $T_a$  é o tempo de aquisição de dados;
- $T_c$  é o tempo de comando dos planos gerados.

Considerando-se o menor tempo de ciclo utilizado em Curitiba ( $TC = 60s$ ), e que são alocados 20 segundos aos processos de varredura de detectores e de comando de planos ( $T_a = 20s$  e  $T_c = 20s$ ), tem-se um tempo de execução máximo de 20 segundos para os algoritmos de otimização ciclo a ciclo. Como estes processos são executados em *thread* própria, não interferem na execução das demais tarefas do sistema de controle, como gerenciamento da comunicação TCP/IP aos clientes conectados, acesso à base de dados, controle de comunicação com equipamentos, entre outros. Essas tarefas não são caras em termos computacionais – por estarem usualmente aguardando I/O - pouco interferindo no desempenho da *thread* de otimização.

As janelas de tempo alocadas aos processos de varredura de detectores e comando de planos são relativamente grandes porque:

- Existe o gargalo da comunicação por modem 2400bps, a partir do equipamento distribuidor de comunicação, para cada subárea.
- As mensagens concorrem com as geradas por outras *threads* do sistema.
- Reserva-se tempo para retransmissões aos equipamentos que não responderem.

Assim, dispõe-se de apenas 20 segundos para executar os algoritmos de otimização de todas as subáreas do sistema. Curitiba apresenta hoje 20 subáreas, o que implica que para que se possa ter o controle adaptativo em toda a cidade, o tempo de processamento do algoritmo de otimização deve ser da ordem de um segundo por subárea.

Isso vale para a execução ciclo a ciclo, onde são executadas buscas em um espaço menor de soluções, pois não se pode variar os tempos de forma abrupta em pequenos

intervalos. Reavaliações globais são executadas em períodos mais longos, por exemplo a cada hora, considerando um tempo maior de aquisição de dados, e, nesse caso, algoritmos mais pesados (por exemplo algoritmos genéticos) poderiam ser executados, processados em *thread* independente da aquisição.

### 3.3 APRESENTAÇÃO DA ABORDAGEM PROPOSTA

Uma proposta de otimização de tráfego para o sistema de Curitiba deve possuir técnicas de otimização das variáveis utilizadas no sistema: tempo de ciclo, frações de verde e defasagens. Para atender aos requisitos temporais e tecnológicos apresentados na seção 3.2, e a premissa de conceber um modelo que demande apenas a configuração das características físicas das vias e de parâmetros de segurança, propõe-se um modelo baseado em:

- Estimativas de medidas de efetividade do *link* (número de paradas, atrasos médios e filas) dadas por sistema de inferência *fuzzy* mapeado em rede neural (ANFIS);
- Cálculo de índice de desempenho global da subárea (PI) sobre as estimativas do ANFIS, parametrizável em função da importância relativa das diferentes medidas de efetividade e de pesos configuráveis para cada *link*;
- Tempos de ciclo determinados em função da equação de Webster (subseção 2.1.5) para o cruzamento mais saturado, ou em função do horário;
- Frações de verde inicializadas pela condição EQUISAT (subseção 2.1.5) e otimizadas por *hill-climbing* sobre a estimativa do índice de desempenho global;
- Determinação de defasagens por otimizador híbrido (algoritmo genéticos e *hill-climbing*) sobre o índice de desempenho global.

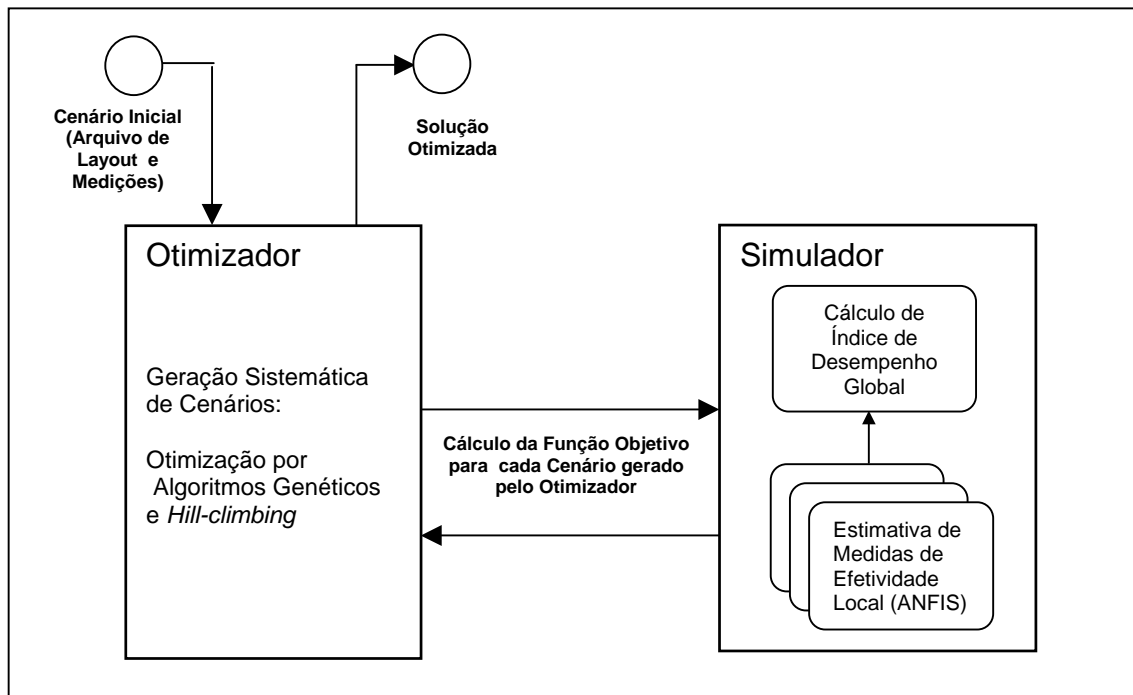
O treinamento do ANFIS é realizado em função de pares de dados de entrada e saída. Idealmente, esse treinamento seria feito com dados reais, oriundos de medições sistemáticas. Entretanto, o escopo deste trabalho não comporta os custos materiais e temporais de tal pesquisa. Serão usados, para o processo de treinamento, dados gerados pelo simulador Transyt (subseção 2.1.6), cujas estimativas podem ser consideradas satisfatórias.

Considera-se que os tempos de ciclo devem ser os mesmos para todos os cruzamentos de uma subárea e que somente pode-se valer de otimização dinâmica do tempo de ciclo se a subárea for isolada. A subárea na qual será testada o modelo de otimização apresenta subáreas

adjacentes operando a tempos fixos e, então, o ciclo será definido em função do horário para que seja possível a manutenção da sincronia nas regiões de fronteira. As defasagens dos controladores de fronteiras também devem ser constantes nesse caso.

O mecanismo de otimização proposto pode ser decomposto em dois subsistemas, conforme representado no diagrama da Figura 15:

- Otimizador: realiza os procedimentos de modificação sistemática de cenários (conjuntos de variáveis de controle), de forma a maximizar o valor de uma função objetivo dada pelas variáveis de controle;
- Simulador: realiza a avaliação de determinado cenário, computando o valor da função objetivo, em função de suas variáveis de controle. Esse valor é denominado índice de desempenho (*Performance Index - PI*).

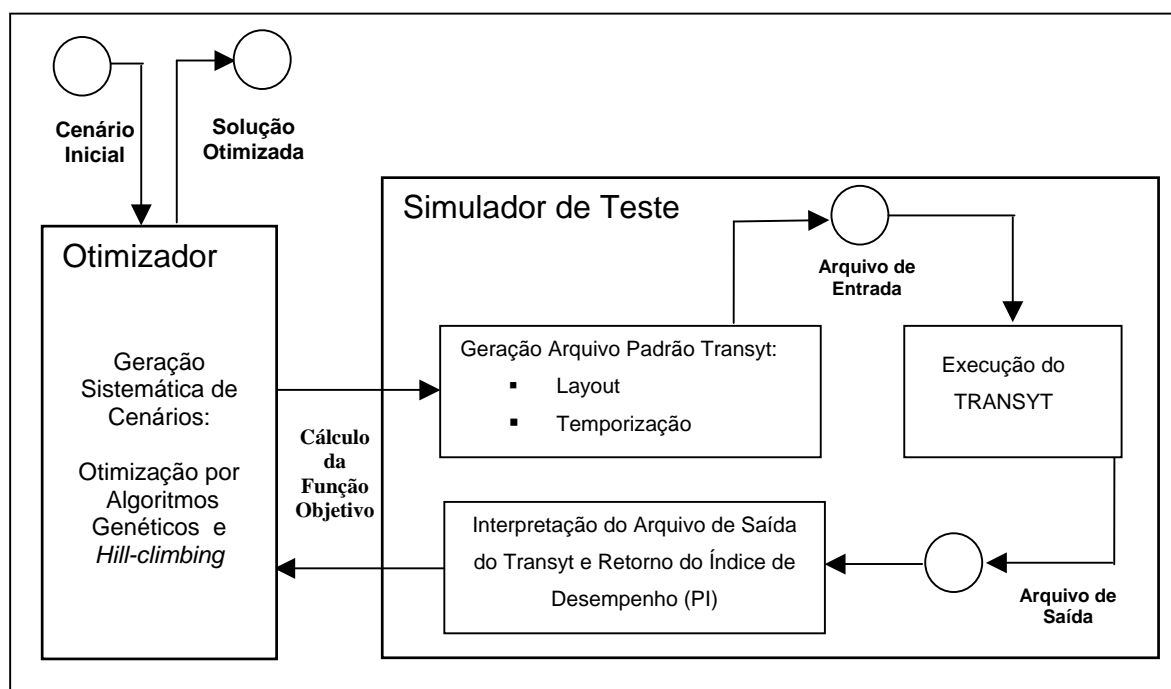


**Figura 15:** Decomposição nos subsistemas otimizador e simulador

Para validação do subsistema de otimização, nas etapas iniciais do desenvolvimento do protótipo, será utilizado um simulador de teste que obterá o índice de desempenho a partir do Transyt (subseção 2.1.6). A comunicação com o Transyt se dá por arquivos texto (arquivos de entrada e saída). Para cada cenário, é gerado um arquivo de entrada. A seguir, a execução do Transyt é comandada em modo simulação. Por fim, é interpretado o arquivo de saída gerado, obtendo-se assim o índice de desempenho para o dado cenário. A Figura 16 ilustra esse mecanismo.

A abrangência de cada ciclo de reavaliação é variável, conforme ilustrado no diagrama de atividades da Figura 17 que sumariza o processo de otimização executado a cada período de reavaliação. Não são detalhados processos correlacionados fora do escopo deste trabalho como:

- Aquisição de dados de detectores;
- Gerenciamento de falhas de detectores e falha de comunicação: os controladores passam a operar em modo local, seguindo programação pré-definida;
- Envio de planos gerados;
- Atuações locais: definidas ao nível de controlador, não são consideradas no processo de otimização aqui apresentado. Como exemplo têm-se botoeiras para pedestres, demandas de passagem de nível e priorização de transporte coletivo.



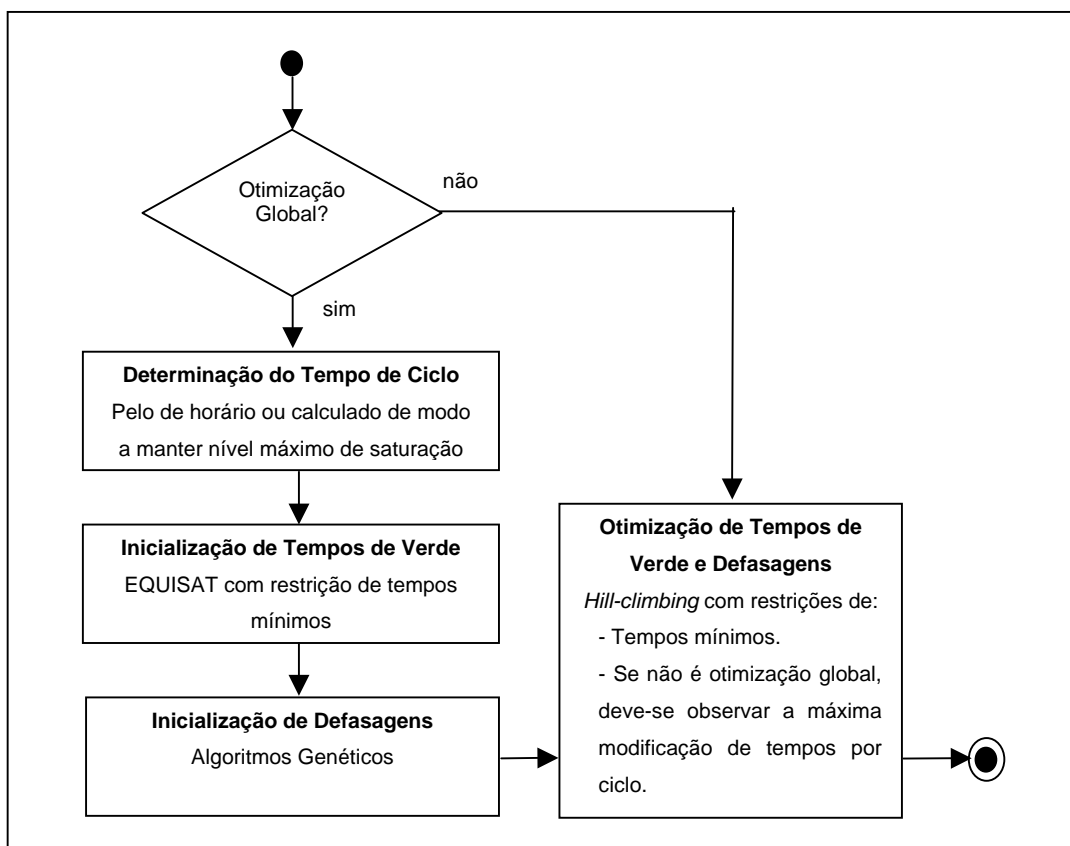
**Figura 16:** Utilização do simulador Transyt para validação do otimizador

Na Figura 17 pode-se observar duas possibilidades de otimização:

- Otimização Global: realiza-se uma busca sem restrições incluindo algoritmos genéticos e inicialização de frações de verde. Não deve ser executada em pequenos intervalos de tempo, pois em cada mudança abrupta de condições de sincronia observa-se efeitos transitórios - necessita-se de alguns ciclos para que o fluxo se

acomode ao novo padrão. Realiza-se em intervalos maiores, por exemplo, a cada 60 minutos, sendo esse um parâmetro de configuração do sistema.

- **Otimização Local:** como é realizada a cada ciclo, é limitada a um parâmetro de configuração que define a variação máxima de defasagens e *splits* por ciclo. Sugere-se que o valor desse parâmetro seja de 2 a 6 segundos por ciclo, para prover o sistema de maior estabilidade. Realiza-se apenas otimização por *hill-climbing* com um passo de tamanho 1 e número de passos limitado à máxima variação permitida.



**Figura 17:** Diagrama de atividades do mecanismo de otimização

### 3.4 JUSTIFICATIVA

Propõe-se a utilização do ANFIS para a estimativa de medidas de efetividade, pois modelos matemáticos de simulação precisos apresentam alto custo computacional (subseção 2.1.6) inviabilizando sua aplicação em tempo real. O treinamento de um sistema de inferência *fuzzy* no modelo ANFIS busca extrair o conhecimento dos modelos de simulação,

reproduzindo suas saídas com baixo custo computacional, embora com algum erro associado. Escolheu-se o ANFIS como aproximador ao invés de, por exemplo, um modelo baseado puramente em rede neural, para se ter um bom compromisso entre precisão das estimativas, baixo tempo de execução e interpretabilidade dos resultados. Além disso, tem-se a simplicidade do processo de treinamento, que pode ser realizado com o uso de ferramenta amplamente utilizada, o Matlab.

As estimativas de medidas de efetividade serão úteis também para a atividade de monitoramento, fornecendo aos engenheiros de tráfego índices de desempenho que permitam comparar os resultados de diferentes técnicas de otimização e de diferentes parametrizações. Atualmente, apenas uma métrica de desempenho existe, a totalização de metros congestionados (seção 3.2). A abordagem por utilização de um sistema de inferência é de fácil inclusão no sistema de controle existente, evitando os processos de comunicação entre diferentes aplicações (sistema de controle e simulador externo).

Tem-se ainda com o ANFIS a flexibilidade para estabelecer a precisão pretendida. O compromisso entre precisão e tempo de processamento é definido pelo número de regras. Pode-se treinar sistemas de inferência com número crescente de regras, e, assim, buscar a maior precisão possível para que os processos de otimização possam ser executados na janela de tempo disponível, considerando as características do *hardware* utilizado.

No caso de realização de extensivo trabalho de pesquisa em campo, sistemas de inferência poderiam inclusive vir a ser treinados com dados reais (e não obtidos por simulação). Da mesma forma, outros simuladores que venham a apresentar melhores resultados podem ser utilizados, como, por exemplo, simuladores microscópicos.

Na otimização de defasagens, propõe-se o emprego de uma etapa prévia de otimização por algoritmos genéticos para evitar a necessidade de uma inicialização coerente dos tempos para o *hill-climbing*, dispensando o conhecimento especializado para entrada de estados iniciais e melhorando os resultados calculados dinamicamente.

Basear a codificação dos algoritmos genéticos somente na defasagem torna sua execução mais eficiente. Considera-se aqui não ser necessário incluir os tempos de verde na codificação, pois o método EQUISAT (subseção 2.1.7) conduz a bons resultados, alcançando um subótimo em poucos passos por algoritmo *hill-climbing*, o que apresenta um custo computacional muito menor.

O modelo apresentará poucos parâmetros de configuração além das características físicas:

- Intervalo de tempo mínimo entre otimizações globais (tempo de proteção) onde se executa a inicialização de tempos de verde, otimização com algoritmos genéticos e *hill-climbing* sem restrição de variação máxima por ciclo;
- Máximas variações de tempo permitidas ciclo a ciclo;
- Tempos mínimos de verde;
- Pesos relativos das medidas de efetividade e de cada cruzamento na composição do índice de desempenho global.

### 3.5 CONCLUSÃO

Buscou-se nesse capítulo propor uma de estratégia de controle adaptativo de tráfego aplicável ao caso de Curitiba, considerando as características do sistema existente. A proposta é baseada no cálculo de um índice de desempenho para o sistema baseado em estimativas de medidas de efetividade tradicionais, apresentadas na subseção 2.1.4, mas obtidas pelo uso do ANFIS (seção 2.3) e não por modelos matemáticos existentes na literatura.

Considerou-se na elaboração do otimizador a estratégia adotada pelo SCOOT (HUNT, 1991), o principal sistema de otimização dinâmica existente. Esse otimizador não poderia ser diretamente aplicado ao caso de Curitiba sem uma reformulação completa da estrutura instalada, devido às características dos detectores veiculares e do *hardware* utilizado no sistema de controle central, sendo muito proibitivo em termos de custos.

Entretanto, algumas das características do SCOOT foram incluídas no modelo proposto, como o controle centralizado, a otimização por *hill-climbing* de um índice de desempenho global baseado em medidas de efetividade dos *links*, a parametrização da importância relativa de *links* e de medidas de efetividade e, finalmente, a limitação das variações de tempo por ciclo. A abordagem proposta apresenta como principais diferenças a utilização de medidas de contagem totalizadas por ciclo (assim como os simuladores *offline*), a utilização do ANFIS (para ter-se maior escalabilidade devido a seus baixos tempos de processamento) e a inclusão da etapa de otimização por algoritmos genéticos.

No próximo capítulo será detalhada a metodologia para implementação da proposta.



## CAPÍTULO 4

### IMPLEMENTAÇÃO

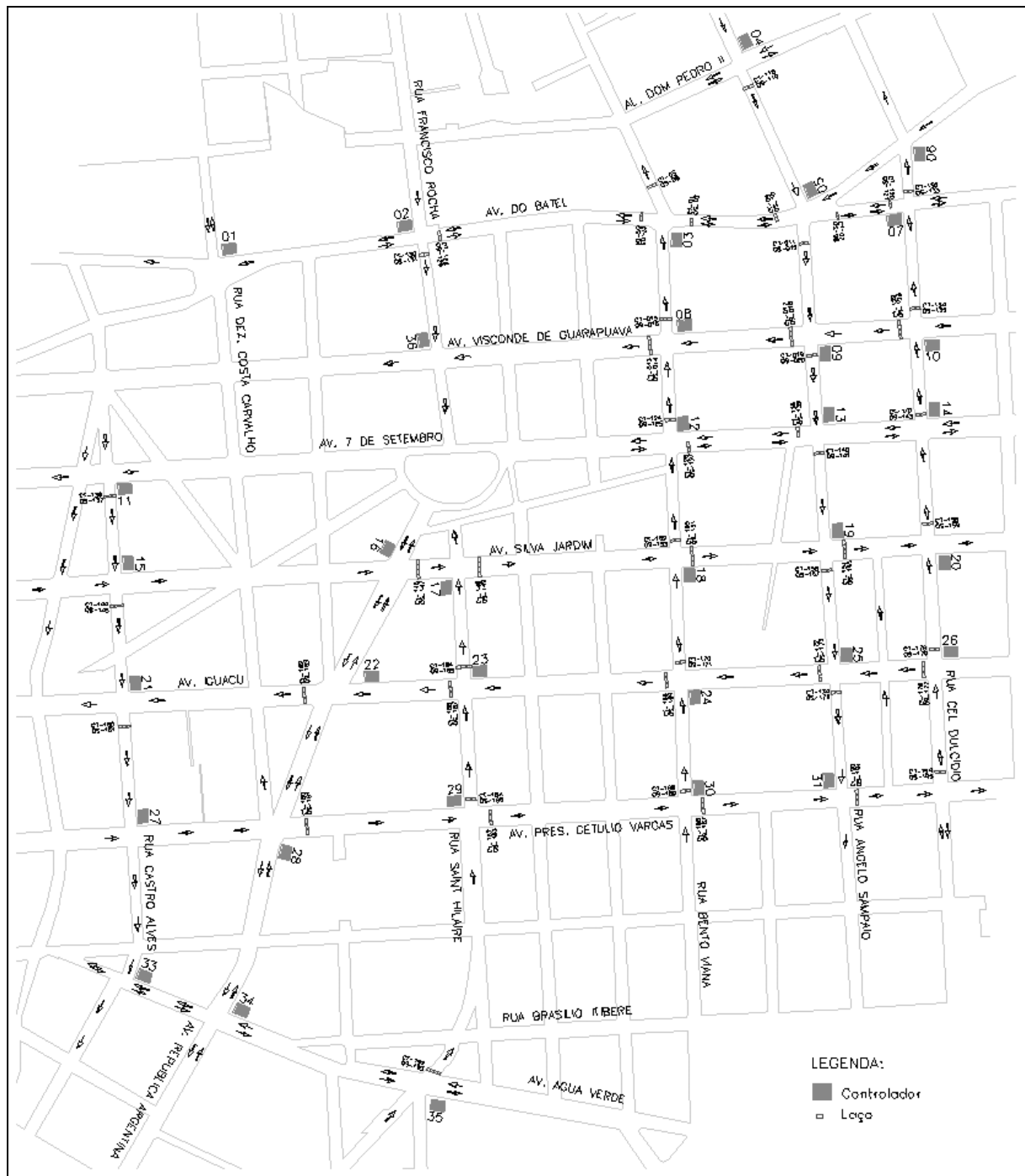
Descreve-se neste capítulo a metodologia para implementação do modelo de otimização proposto no Capítulo 3. A seção 4.1 apresenta a decomposição do modelo em subsistemas e nos passos necessários à implementação, definindo a estrutura do plano de projeto. Nessa seção também é apresentado o desenho físico da região utilizada no processo validação da proposta. As seções seguintes apresentam o detalhamento da implementação de cada subsistema. Por fim, a seção 4.6 comenta a integração dos subsistemas no modelo de otimização completo.

#### 4.1 ETAPAS DE IMPLEMENTAÇÃO

O trabalho de implementação foi dividido nas seguintes etapas de desenvolvimento:

1. Construção do estimador do índice de desempenho global:
  - Implementação de sistemas de inferência *fuzzy* (ANFIS) para estimativa de medidas de efetividade de *links*;
  - Composição do índice de desempenho global a partir das estimativas de medidas de efetividade dos *links*;
  - Validação das medidas obtidas com os sistemas de inferência.
2. Construção do otimizador:
  - Implementação de otimizador de defasagens de planos de tráfego com algoritmos genéticos (GA);
  - Desenvolvimento do otimizador híbrido com a inicialização de tempos de verde por EQUISAT, inicialização de defasagens pelo otimizador GA e etapa posterior de *hill-climbing* sobre defasagens e tempos de verde;
  - Validação do otimizador.
3. Integração do simulador e estimador:
  - Execução para diferentes cenários;
  - Análise dos resultados.

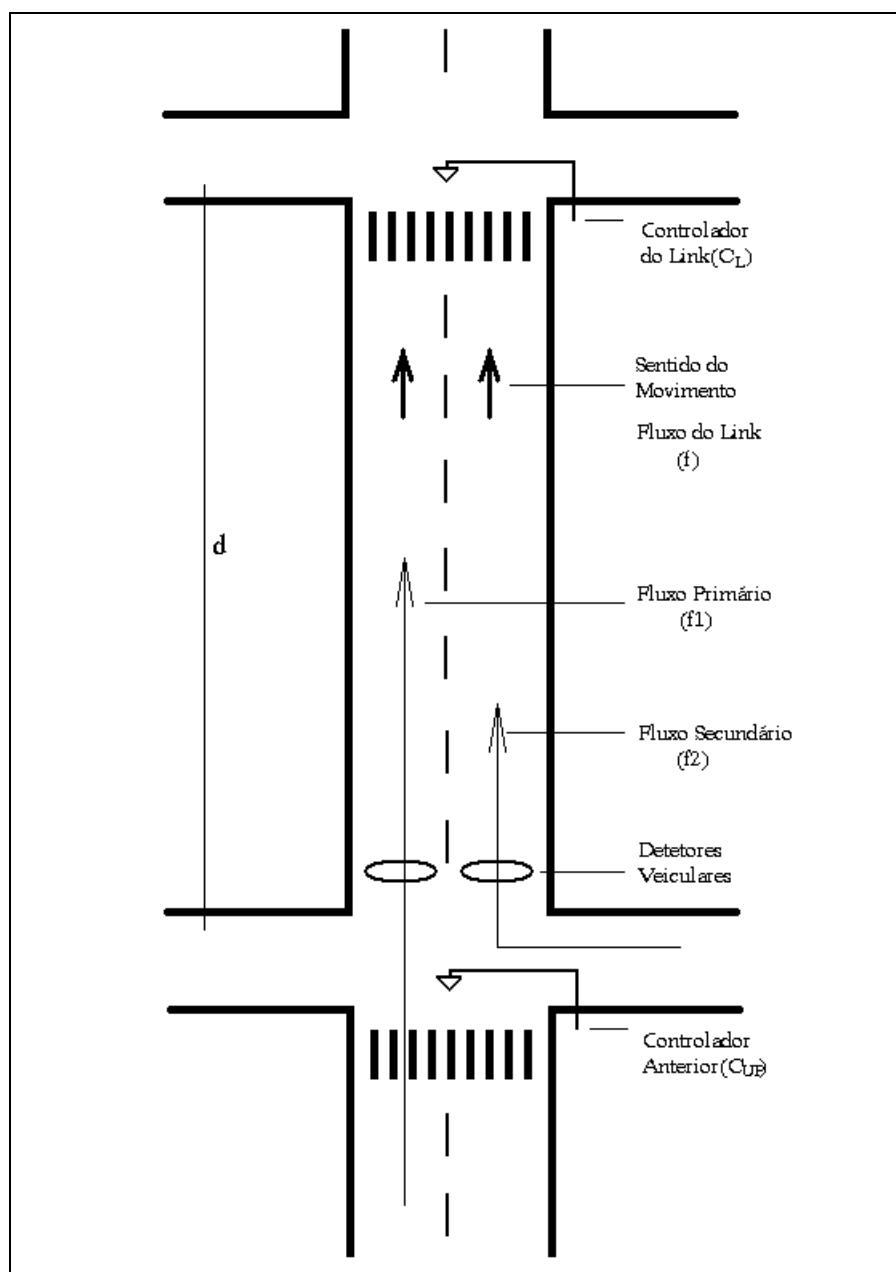
Foi considerado neste trabalho o sistema de controle de tráfego da cidade de Curitiba, em particular a subárea Batel, composta de 36 cruzamentos e apresentando detectores veiculares suficientes para viabilizar um controle adaptativo. A Figura 18 apresenta a subárea Batel, incluindo o sentidos das vias, os detectores (pequenos retângulos nas vias) e os controladores semafóricos (retângulos grandes nas esquinas).



**Figura 18:** Planta da subárea Batel

#### 4.2 ESTIMATIVAS DE EFETIVIDADE DE *LINKS* COM O ANFIS

Nessa etapa busca-se a estimação rápida das medidas de efetividade (subseção 2.1.4) de um *link* do sistema, representado pelo diagrama da Figura 19. Essas medidas serão utilizadas posteriormente na construção de um índice de desempenho global a ser otimizado.



**Figura 19:** Representação de um *link*

Para o *link* genérico apresentado na Figura 19, pode-se considerar uma medida de efetividade  $p$  como função das variáveis:

$$p = f(tc, d, vm, fm, fs, tv, d0, tv1, d1, tvi, di), i=2, \dots, n \quad (23)$$

onde:

- $tc$ : tempo de ciclo;
- $d$ : tamanho do *link*;
- $vm$ : velocidade média no *link*;
- $fm$ : fluxo medido no *link*
- $fs$ : fluxo de saturação do *link*;
- $tv$ : tempo de verde no *link*;
- $d0$ : defasagem do início do verde no *link*;
- $tv1$ : tempo de verde do fluxo primário no *link* anterior;
- $d1$ : defasagem do início do verde do fluxo primário no *link* anterior;
- $tvi$ : tempo de verde dos fluxos secundários nos *links* anteriores;
- $di$ : defasagem do início do verde dos fluxos secundários nos *links* anteriores.
- $n$ : número de movimentos que alimentam o link.

Propõe-se a implementação de sistemas de inferência *fuzzy* para estimação de medidas de efetividade necessárias aos algoritmos de otimização, baseados em variáveis lingüísticas representando as variáveis da equação acima, ou associações entre elas. Para facilitar a representação, considera-se aqui a defasagem para cada início de verde ao invés de defasagem da intersecção acrescida do instante de início de verde no ciclo.

Em um primeiro momento é necessário gerar uma massa de dados de treinamento. Para tanto, será utilizado o Transyt (subseção 2.1.6), cujos resultados se adaptam bem aos dados reais. Esta massa de dados treinará então um FIS do tipo TSK (Takagi-Sugeno-Kang) mapeado em rede neural (ANFIS), cujos parâmetros serão ajustados no Matlab utilizando algoritmo *backpropagation* aliado a um método de mínimos quadrados.

O problema inicial é o número de variáveis. Nota-se na Equação 23 que para um *link* alimentado por  $n$  movimentos tem-se  $7+2n$  variáveis.

Um número elevado de regras tornaria o treinamento do FIS muito lento, assim como sua execução, considerando que desejamos aplicá-lo ao problema de otimização em tempo real. Busca-se a princípio diminuir o número de variáveis, associando-as, para tornar o FIS

mais eficiente, mesmo a custo de alguma perda de precisão. Para isso algumas simplificações foram adotadas, considerando as características do sistema alvo:

1. A solução será treinada para tempos de ciclo de 90 segundos. Este é o ciclo em Curitiba para os horários de pico. Curitiba utiliza outros dois tempos de ciclo, em horários de menor fluxo: 60 e 70 segundos. Para esses casos outros FISs devem ser gerados. A Equação 23 é assim reduzida para:

$$pc = f(d, vm, fm, fs, tv, d0, tv1, d1, tvi, di) , i=2, \dots, n \quad (24)$$

onde:

$pc$ : medida de efetividade para tempo de ciclo igual a 90s.

2. Para cada *link*, considera-se o início do seu verde como referência de tempo ( $d0=0$ ), eliminando assim uma variável e obtendo-se a equação:

$$pc = f(d, vm, fm, fs, tv, tv1, d1, tvi, di) , i=2, \dots, n \quad (25)$$

3. Será desconsiderada a decomposição dos fluxos alimentadores do *link*. Assim, somente será considerado o fluxo primário, sendo que os demais serão incluídos neste. Isso será feito por não se possuir informação dos fluxos de conversões, apenas o valor integralizado dados pelos detectores veiculares (seção 3.2). Para considerar a decomposição, seria necessário valer-se de médias históricas do percentual de conversões, o que enriqueceria o modelo, mas aumentaria o número de variáveis. Nas situações próximas à saturação, os fluxos secundários são marginais e sua inclusão dentro do fluxo principal não tende a alterar muito o resultado. Para situações longe de saturação, assumir que todo o fluxo origina-se do mesmo pelotão principal é impreciso, mas supõe-se aqui um erro tolerável dado que, na grande maioria dos casos, o volume oriundo de conversões corresponde a menos de 20% do fluxo. Para o sistema alvo de quadras curtas, com fluxos relevantes na maioria das vias, a maior eficiência do sistema se dá para a otimização dos corredores, ou seja, no sentido de minimizar as paradas dos fluxos que seguem adiante. Considera-se que esta simplificação não invalida a estratégia de controle já que essa característica tende a favorecer os tempos dos fluxos que seguem adiante em detrimento das conversões, o que é o comportamento esperado. A Equação 25 reduz-se para:

$$pc1 = f(d, vm, fm, fs, tv, tv1, d1) \quad (26)$$

4. Não será considerado o tempo de verde do *link* anterior ( $tv1$ ) como uma variável, mas sim o tempo mínimo necessário para alimentar o fluxo medido, obtendo-se:

$$pc2 = f(d, vm, fm, fs, tv, dl) \quad (27)$$

5. A seguir, associa-se as variáveis  $d$ ,  $vm$  e  $dl$  em uma variável indicadora da sincronia, ou seja, do instante de chegada do pelotão principal em relação à abertura de verde. Essa variável será denominada  $ch$  dada por:

$$ch = dl - (d/vm) \quad (28)$$

O modelo continua genérico quanto à velocidade média e ao tamanho das quadras. Não se perde precisão com esta suposição, pois para simulação usualmente é considerado que a velocidade no *link* é constante, assumindo-se que o tempo de reação à abertura de sinal somada ao tempo de aceleração aproximadamente equivale ao fluxo que extrapola o tempo de verde (durante a sinalização do amarelo). Deve-se considerar, para o caso da medida de filas, que a fila não deve ser superior ao tamanho do *link* (eliminado da Equação 27) e assim uma função será acrescida à saída para limitar este resultado nesse caso:

$$Pc3 = f(fm, fs, tv, ch). g(d) \quad (29)$$

onde:

- $g(d) = \min(pc2, d)$  para estimação de filas e  $g(d) = 1$  nos outros casos.

6. Finalmente, unificando  $fm$  e  $fs$  trabalha-se não em termos de fluxo total, mas sim com a relação entre fluxo medido e fluxo de saturação ( $fm/fs$ ). Optou-se por normalizar o fluxo por faixa de rolamento. Para uma dada situação, primeiramente é calculado o fluxo por faixa, a seguir é executado o FIS e, por fim, retorna-se o resultado normalizado ao  $fs$  real, corrigido por uma função  $h(fs, fm)$ :

$$pc4 = f(fmn, tv, ch). g(d). h(fs, fm) \quad (30)$$

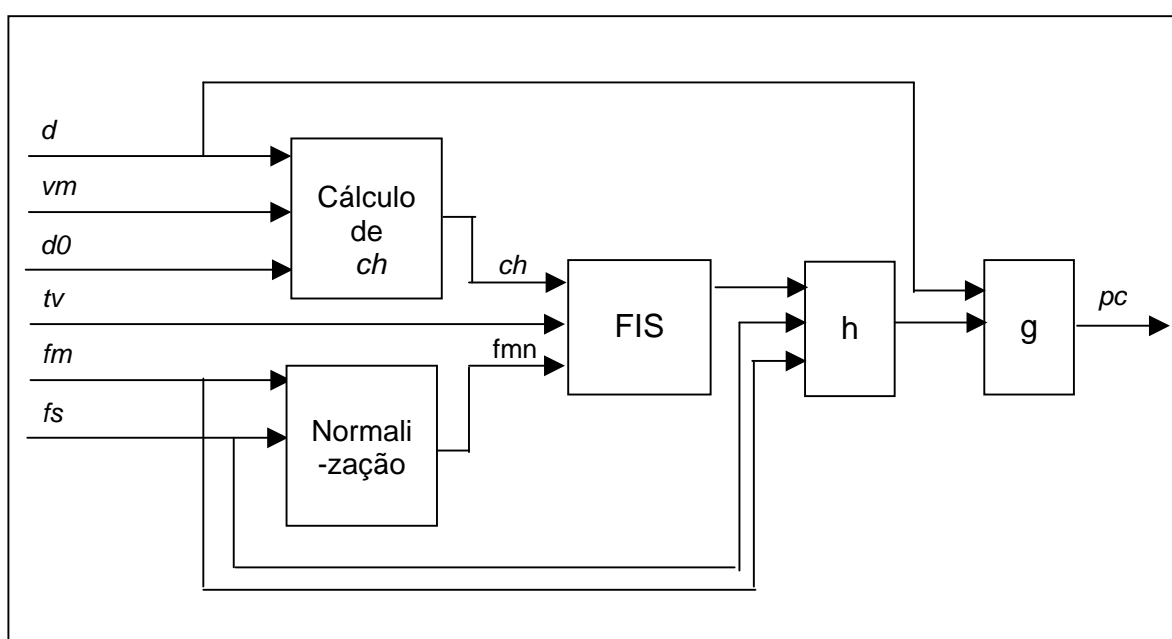
onde:

- $fmn =$  fluxo normalizado:  $fm / (fs / 1800)$ ;
- $h(fs, fm) = 1$  para atrasos médios e  $h(fs, fm) = fs/1800$  nos outros casos.

Em termos práticos, a normalização implica em um pequeno erro devido a se desprezar as diferenças de atritos laterais (menores nas faixas centrais). Considera-se aqui um fluxo de saturação de 1800 veículos por hora por faixa, como ilustrado na subseção 2.1.3,

sendo esse o valor *default* utilizado nos processos de otimização *offline* na cidade de Curitiba. Não haverá desvios relativos ao serem utilizados *links* com diferentes fluxos de saturação, pois se estará sempre normalizando (e retornando) em termos de 1800vph. Poderia-se trabalhar simplesmente em termos da relação  $fmn=fm/fs$ , mas decidiu-se manter a constante 1800 para se ter, posteriormente, maior clareza na visualização das variáveis *fuzzy*.

O esquema geral do modelo utilizado para obtenção de cada medida de efetividade é indicado na Figura 20 (assumindo cálculo prévio para se determinar  $d_0$  em função da defasagem dos cruzamentos, e do início dos estágios de verde).



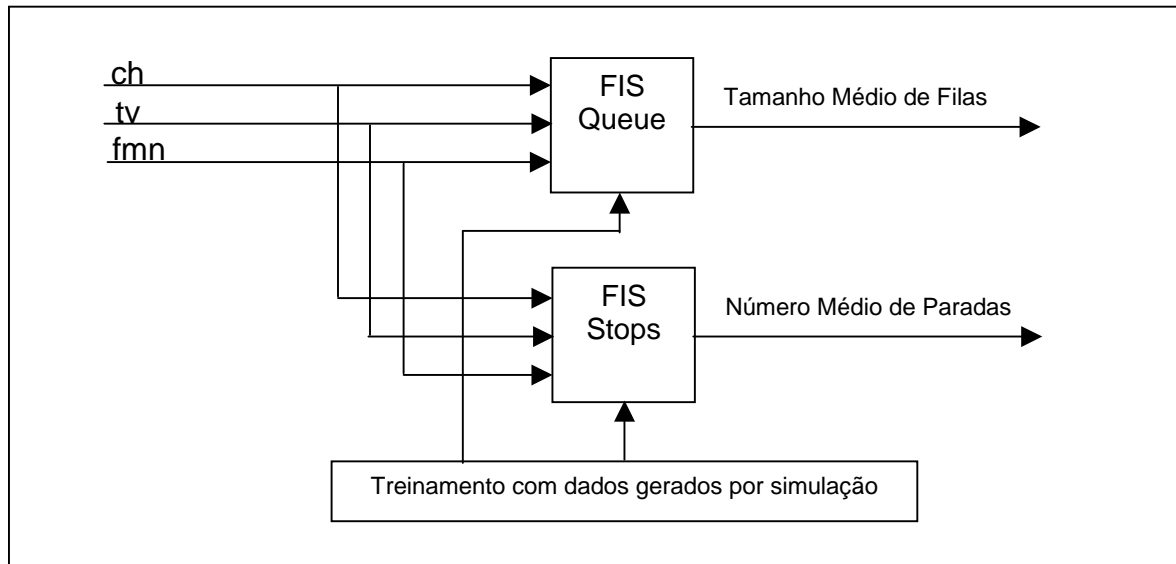
**Figura 20:** Esquema de generalização dos resultados do FIS

O objetivo específico desta etapa é, conforme indicado na Figura 21, a utilização do ANFIS para estimar-se, para um *link* genérico, o tamanho médio de filas, o número de paradas e o atraso médio considerando um tempo de ciclo de 90 segundos e com a restrição de considerar o *link* alimentado por um único movimento principal (fluxo primário).

Foram gerados diversos arquivos com dados de treinamento de diferentes tamanhos nos seguintes métodos:

- Aleatório (valores de entrada aleatoriamente tomados no intervalo de validade);
- Divisão regular do espaço (valores de entrada tomados em intervalos regulares cobrindo todo o espaço de discurso das variáveis);

- Divisão ortogonal do espaço, mas com maior concentração nas faixas de valores mais usuais.



**Figura 21:** Esquema de treinamento e estimação do FIS

As variáveis do sistema e seus intervalos de validade são indicados na Tabela 3. O método de geração de dados teve impacto nos resultados obtidos. Os melhores resultados (menores erros nos dados de verificação) foram alcançados com a utilização de dados de treinamento gerados por divisão ortogonal do espaço com maior concentração nas faixas de valores mais usuais, mas com dados de verificação e *check* gerados aleatoriamente. Um bom compromisso entre precisão do resultado e tempo de treinamento foi obtido para massa de dados constituída de 1000 a 1500 pares de valores de entrada/saída.

**Tabela 3:** Variáveis *fuzzy* dos FISs e intervalos de validade

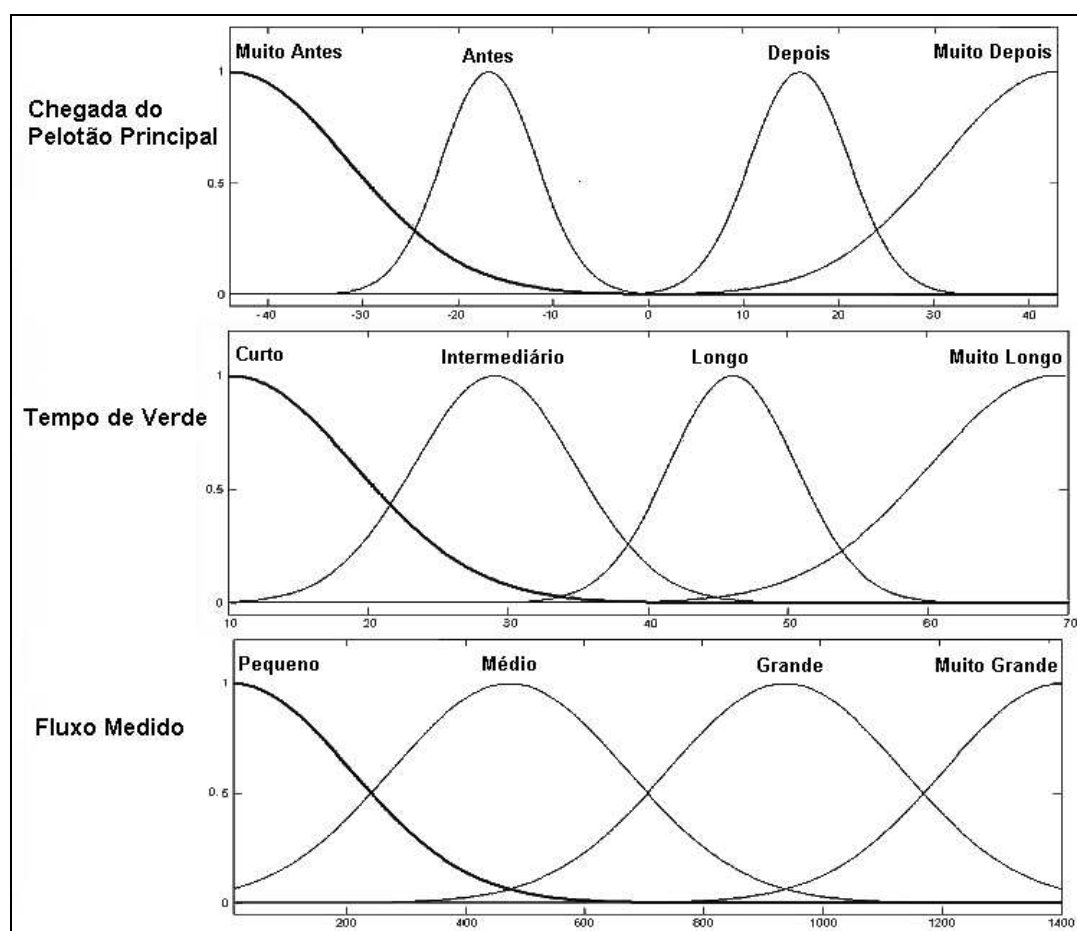
Variável	Símbolo	Intervalo	Observações
Chegada do pelotão principal em relação ao início do verde (s)	<i>ch</i>	-45 a 45 ( $-tc/2$ a $tc/2$ )	$ch=0$ significa que o primeiro veículo chega no semáforo no exato momento em que ele abre (caso não haja fila residual). Como essa variável é periódica, seu valor para $ch=-45$ é igual ao de $ch=45$ .
Tempo de verde (s)	<i>tv</i>	10 a 70	Considera-se um tempo mínimo de segurança por estágio de 10s. Para ciclo de 90 segundos e considerando cada transição durando 5 segundos, o maior tempo individual de estágio será de 70s.
Fluxo Medido (vph)	<i>fm</i>	0 a 1800	Fluxo máximo (1800) indica que valor medido é igual ao fluxo de saturação da via.



Outro aspecto importante é a determinação das funções de pertinência e da partição do espaço em variáveis fuzzy. Bons compromissos entre precisão, tempo de treinamento e tempo de execução foram obtidos para uma divisão por *grid* com quatro funções de pertinência por variável, totalizando 64 regras. A Tabela 4 apresenta os valores das variáveis lingüísticas. Suas funções de pertinência, otimizadas pelo ANFIS, são indicadas na Figura 22.

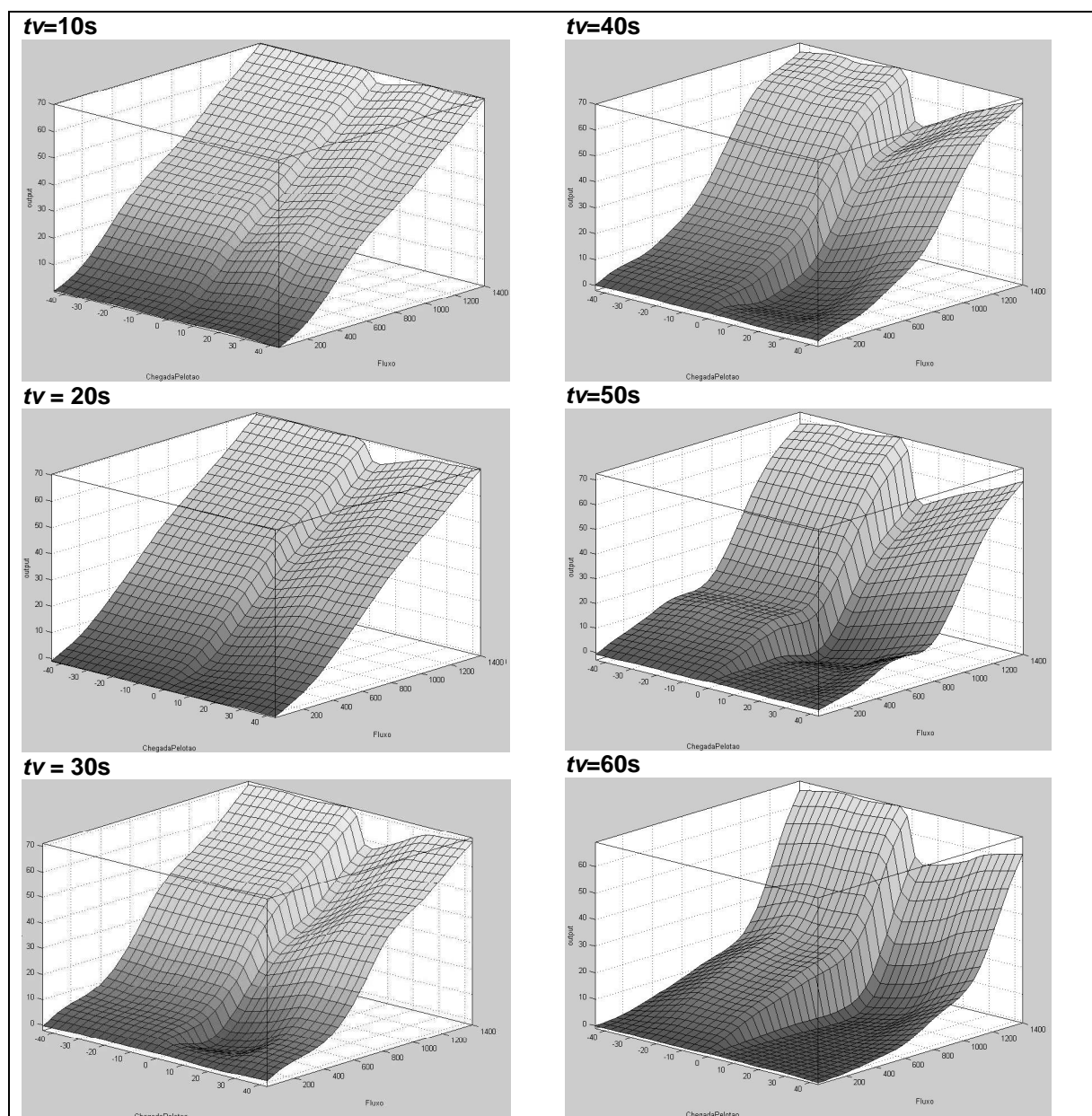
**Tabela 4:** Valores lingüísticos das variáveis fuzzy

Variável	Valores
Chegada do Pelotão Principal (em relação ao início do verde)	Muito Antes Antes Depois Muito Depois
Tempo de verde	Curto Intermediário Longo Muito Longo
Fluxo Medido	Pequeno Médio Grande Muito Grande



**Figura 22:** Funções de Pertinência

As funções de pertinência das variáveis foram estabelecidas inicialmente como gaussianas em divisão regular do espaço, e modificadas pelo ANFIS durante o processo de treinamento. Treinou-se o FIS até que os erros dos dados de *check* parassem de decrescer, evitando assim o *overfitting* (MATHWORKS, 2002). Para tamanhos de fila, ao final do treinamento no Matlab, o erro médio do ANFIS foi de 2,52 para os dados de treinamento e de 2,71 para os dados de verificação. As superfícies de resposta do FIS para esse caso (estimação de filas) são dadas na Figura 23, em função do fluxo medido e do instante de chegada do pelotão, considerando, em cada gráfico, o tempo de verde constante. Assim pode-se representar graficamente uma função de três variáveis.



**Figura 23:** Superfícies de Resposta do FIS

Os gráficos da Figura 23 não representam resultados de estimação ( $pc$ ), mas a saída do FIS como ilustrado na Figura 20, e por isso não estão no Capítulo 5, onde são apresentados os resultados do trabalho, tendo sido gerados pelo próprio Matlab após o treinamento. Com a utilização do módulo “fis.c” do Matlab (Fuzzy Logic Toolbox) pode-se embutir facilmente os FISs gerados no Matlab em aplicações e comparar resultados gerados por simulação com os calculados pelo uso do FIS.

#### 4.3 COMPOSIÇÃO DO ÍNDICE DE DESEMPENHO GLOBAL

A função objetivo para o processo de otimização foi determinada pela geração de um índice de desempenho global para a subárea ( $PI$ ). Como tem-se um problema de otimização multi-objetivo, esse índice é usualmente composto por uma ponderação de diferentes medidas de efetividade por *link*. A mesma abordagem é adotada por otimizadores *offline* como o Transyt, e por sistemas de otimização dinâmica como o SCOOT (HUNT, 1991).

Diferentes otimizadores podem implementar diferentes índices de desempenho. Em geral permitem que o usuário escolha a composição desse índice e a parametrização dos pesos relativos das medidas que o constituem. As estimativas mais empregadas são as medidas de atrasos e paradas. Considera-se como exemplo a formação do PI tomada de (WALLACE et al., 1999):

$$PI = \sum_{i=1}^n w_i (D_i + w_s.S_i + w_q.bq_i \cdot (q_i - qc_i)^2) \quad (31)$$

onde:

- $w_i$  é o peso do *link*  $i$ ;
- $D_i$  é o atraso total no *link*  $i$  (v-hr);
- $S_i$  é o número de paradas no *link*  $i$  (veh/hr);
- $w_s$  é a importância relativa das paradas;
- $w_q$  é a importância relativa do tamanho das filas;
- $bq_i$  é igual a 1 se  $q_i > qc_i$  ou 0, caso contrário.
- $q_i$  é o tamanho máximo de filas no *link*  $i$ ;
- $qc_i$  é a capacidade no *link*  $i$ ;

Mais usualmente, as filas são desconsideradas na estimação do índice ( $wq=0$ ), conforme:

$$PI = \sum_{i=1}^n w_i (D_i + ws.S_i) \quad (32)$$

De modo geral, para avaliação do mecanismo de otimização, será utilizada a Equação 32, não considerando o componente relativo a tamanho de filas, para se ter uma execução mais rápida do algoritmo (assim são executados apenas dois FIS por *link*). Na avaliação não serão utilizadas importâncias relativas de links ( $w_i=1$ ), e será fixado o valor de  $ws$  em 0.00194. Dessa forma temos uma importância relativa maior do componente de atrasos, em geral na ordem de 70% do valor do  $PI$ .

#### 4.4 OTIMIZAÇÃO DE DEFASAGENS COM ALGORITMOS GENÉTICOS

Propõe-se a utilização de algoritmo genético para otimização unicamente de defasagens, conforme justificado na seção 3.4, considerando os demais parâmetros (tempo de ciclo e tempos de verde) determinados *a priori*. Essa otimização constituirá uma inicialização dos valores de defasagens para uma etapa posterior de otimização por *hill-climbing*. O problema será expresso em termos de otimização de um índice de desempenho global, conforme apresentado na seção 4.3.

O conjunto de variáveis compreende as defasagens das  $n$  interseções do sistema,  $d_1$  a  $d_n$ . A defasagem pode variar de zero até o tempo de ciclo, com precisão de um segundo. Será utilizado um tempo de ciclo de 90 segundos. Cada valor de defasagem é codificado em 7 bits. Assim, a codificação do gene é dada por:

$$d_x = b_{6x} b_{5x} b_{4x} b_{3x} b_{2x} b_{1x} b_{0x} \quad (33)$$

$$def_x = dx * TC / 128 \quad (34)$$

$$C = d_1 d_2 \dots d_n \quad (35)$$

onde:

- $d_x$  é o gene relativo ao cruzamento  $x$ , composto de 7 bits;
- $def_x$  é a defasagem do cruzamento  $x$ ;

- $TC$  é o tempo de ciclo;
- $C$  é o cromossomo, justaposição dos genes relativos a cada cruzamento.

Tem-se um espaço de busca de  $36^{90}$ , visto que trabalha-se com um conjunto de  $n=36$  interseções, para tempo de ciclo de 90 segundos.

Deve-se atentar aos blocos construtivos do cromossomo (“*building blocks*”). Cada valor de defasagem é correlacionado com as defasagens das interseções adjacentes e, assim, seus genes deveriam estar consecutivos no cromossomo. Como os cruzamentos estão dispostos em uma matriz e cada interseção está vinculada com as adjacentes nos dois sentidos, o *crossover* poderia prever a possibilidade de manutenção de blocos verticais e horizontais (*crossover* matricial), o que, no entanto, não foi aqui considerado por simplicidade, visto que uma matriz representando as defasagens dos cruzamentos seria esparsa. Busca-se, entretanto, manter os cruzamentos adjacentes das principais vias em genes sucessivos.

Não existem restrições para o problema. Podem ser fixadas as defasagens na fronteira da região a ser otimizada, para garantir sincronia com as regiões adjacentes, mas isso não seria uma restrição, e sim uma imposição *a priori*, que simplesmente diminuiria o espaço de busca.

A função objetivo será dada pelo índice de desempenho global (PI) da subárea. Em um primeiro momento, para validar o algoritmo genético, esse índice será dado por simulador externo, o Transyt. Na integração do otimizador proposto, o índice será gerado utilizando-se o ANFIS conforme apresentado na seção 4.3. A troca de dados com o Transyt se dá por arquivos texto. Como o Transyt também realiza a otimização por *hill-climbing*, ele será utilizado para se avaliar os resultados de otimização obtidos com algoritmos genéticos.

A Função de *Fitness* é dada por:

$$F(x) = PI(x) / PIMax \quad (36)$$

onde:

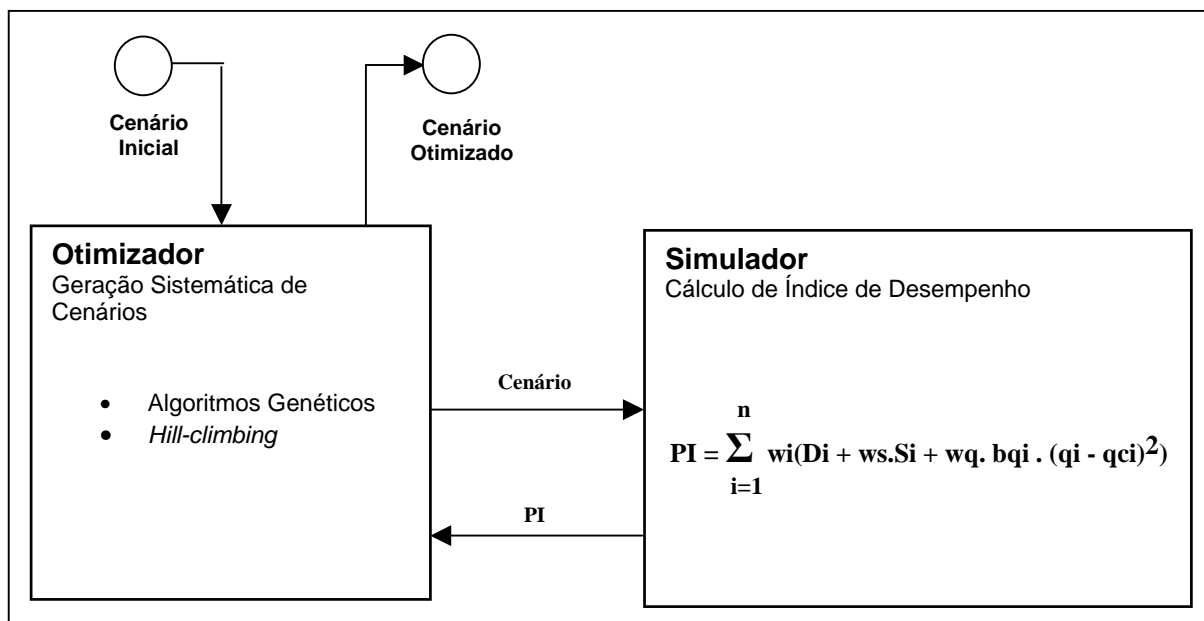
- $F(x)$  é a Função de *Fitness*;
- $PI(x)$  é o índice de desempenho global calculado para o cenário;
- $PIMax$  é o maior índice de desempenho estimado.

Desenvolveu-se um conjunto de classes genéricas para a implementação do GA, que apresenta gráficos da evolução a cada geração, estatísticas e permite a configuração dos diversos parâmetros de GA como:

- Número de gerações;
- Tamanho da população;
- *Generation Gap* (inverso do grau de elitismo);
- Tipo de seleção (roleta ou linear);
- Faixa de seleção (percentual de indivíduos selecionáveis);
- Tipo de *crossover* (simples, duplo ou uniforme);
- Probabilidade de *crossover*;
- Tipo de mutação (uniforme ou variável em função da posição do bit);
- Probabilidade de mutação.

#### 4.5 CONSTRUÇÃO DO OTIMIZADOR HÍBRIDO

O mecanismo de otimização, apresentado na Figura 24, é uma variação sistemática de cenários, com o cálculo do PI para cada um deles, e a escolha do melhor resultado. No caso, o melhor resultado é a solução de menor PI, por ele ser uma ponderação de medidas de ineficiência (atrasos, paradas, filas), a qual busca-se minimizar.



**Figura 24:** Esquema do mecanismo de otimização

Utilizando um modelo híbrido de otimização, por algoritmos genéticos aliado ao *hill-climbing*, aliou-se o poder de convergência desse último à robustez do primeiro. Assim, o algoritmo genético, apresentado na seção 4.4, executa um papel de inicializador das defasagens do sistema. Os processos do otimizador híbrido são representados na Figura 17.

A inicialização de *splits* é feita pelo algoritmo EQUISAT, descrito na subseção 2.1.5, por ser de fácil e rápida execução e por conduzir a um ponto próximo do ótimo no que se refere à alocação de tempos de verde. O tempo de ciclo é determinado *a priori*, por seleção horária ou de forma a manter um grau máximo de saturação no cruzamento mais saturado. Todos os tempos apresentam uma resolução de um segundo.

Após a determinação dos valores iniciais, defasagens e tempos de verde são otimizados por *hill-climbing* cuja implementação aqui foi baseada naquela apresentada em (WALLACE et al., 1999).

Ao contrário do algoritmo genético, o método *hill-climbing* é determinístico. Deve-se configurar um certo número de passos de otimização, de diferentes tamanhos. Utiliza-se passos de tamanho variável para acelerar os processos de busca e procurar evitar a convergência a máximos locais.

Os valores de passos aqui empregados serão:

- Otimização de defasagens:            40<sup>d</sup>    15<sup>d</sup>    1<sup>d</sup>
- Otimização de splits:                1<sup>s</sup>
- Otimização de splits e defasagens:    40<sup>d</sup>    15<sup>d</sup>    1<sup>d</sup>    1<sup>s</sup>    1<sup>d</sup>
- Busca local:                            1<sup>dl</sup>    1<sup>sl</sup>    1<sup>dl</sup>

onde:

- Valores seguidos de <sup>d</sup> significam passos em defasagem;
- Valores seguidos de <sup>s</sup> significam passos em *split*;
- Valores são dados em percentual do ciclo, se passo é diferente de 1;
- Para passo igual a 1 toma-se um segundo;
- Valores seguidos de <sup>l</sup> são limitados pelo parâmetro de máxima variação ciclo a ciclo: o algoritmo de *hill-climbing* é idêntico, mas interrompe o incremento de passos quando a variável apresentar variação maior que a dada pelo parâmetro.

O algoritmo utilizado, no caso das defasagens é:

```

PARA TODOS OS PASSOS
  PARA TODOS OS CRUZAMENTOS
    CALCULA-SE O PI (ÍNDICE DE DESEMPENHO) DO SISTEMA
    ADICIONA-SE O VALOR DO PASSO À DEFASAGEM E CALCULA-SE O NOVO PI
    SE PI MELHOROU
      ADICIONA-SE VALOR DO PASSO ENQUANTO PI MELHORAR
    SE PI PIOROU
      SUBTRAI-SE VALOR DO PASSO ENQUANTO PI MELHORAR

```

O valor da defasagem deve ser limitado à faixa de zero ao tempo de ciclo. Assim, em cada passo, se a defasagem for calculada como negativa ou maior que o tempo de ciclo, deve ser somado ou subtraído dela o tempo de ciclo, respectivamente.

O algoritmo para otimização de *splits* é:

```

PARA TODOS OS PASSOS
  PARA TODOS OS CRUZAMENTOS
    PARA TODOS OS SPLITS
      CALCULA-SE O PI (ÍNDICE DE DESEMPENHO) DO SISTEMA
      ADICIONA-SE O VALOR DO PASSO AO SPLIT E CALCULA-SE O PI
      SE PI MELHOROU
        ADICIONA-SE VALOR DO PASSO ENQUANTO PI MELHORAR
      SE PI PIOROU
        SUBTRAI-SE VALOR DO PASSO ENQUANTO PI MELHORAR

```

Cada alteração no tempo do *split* deve obedecer às seguintes restrições:

- Tempos mínimos devem ser respeitados;
- O tempo de ciclo deve ser mantido. Assim, para cada incremento em tempo em um *split*, um decremento correspondente deve ser feito nos demais *splits*.

## 4.6 CONCLUSÃO

Buscou-se validar cada subsistema isoladamente antes de proceder-se à sua integração para minimizar a complexidade e conseqüentemente reduzir o risco do projeto. Após a validação em separado do algoritmo genético, da otimização por *hill-climbing* e do algoritmo EQUISAT, eles foram unificados no otimizador híbrido, ainda utilizando as estimativas do Transyt para cálculo do índice de desempenho de cada cenário.

Validaram-se as estimativas do ANFIS em comparação aos resultados de simulação, registrando-se os erros obtidos. Em seguida, verificou-se o desvio do índice de desempenho global gerado pelo ANFIS em relação aos calculados pelo Transyt.

Por fim, o sistema foi unificado, sendo o índice de desempenho do ANFIS empregado no otimizador final. Os resultados obtidos nas validações parciais, assim como no sistema integrado, são apresentados no Capítulo 5.



## CAPÍTULO 5

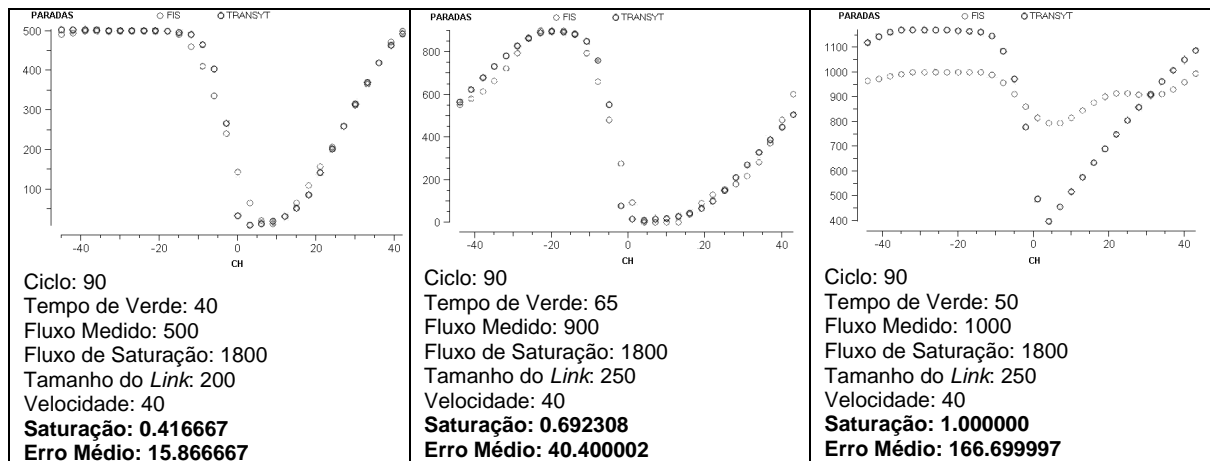
### RESULTADOS

São apresentados neste capítulo os resultados obtidos, primeiramente nas validações parciais de subsistemas e a seguir na avaliação do modelo proposto integrado. A seção 5.1 apresenta a validação das estimativas de medidas de efetividade de *links* obtidas com o ANFIS. A seção 5.2 avalia a utilização isolada do algoritmo genético na otimização de defasagens, comparando-o ao método *hill-climbing*. A seção 5.3 apresenta os resultados obtidos com o sistema integrado: a otimização híbrida (algoritmo genético e *hill-climbing*) sobre o índice de desempenho global calculado com o uso das estimativas do ANFIS. Como existe uma infinidade de cenários de avaliação possíveis, foram aqui reproduzidos alguns casos particulares que traduzem um comportamento usual. No caso da avaliação do sistema integrado, sempre foram consideradas no cenário as características físicas da região de avaliação, a subárea Batel, representada na Figura 18. Concluindo, a seção 5.4 apresenta a análise dos resultados obtidos.

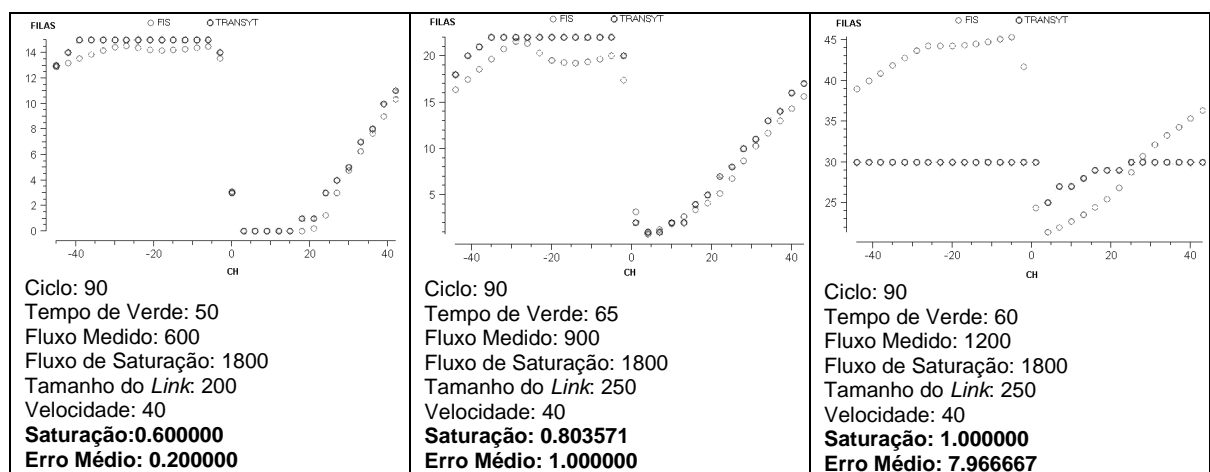
#### 5.1 ANFIS COMO ESTIMADOR DE MEDIDAS DE EFETIVIDADE

Primeiramente avaliou-se as saídas dos FISs (medidas normalizadas conforme indicado na Figura 20). As Figuras 25 e 26 apresentam gráficos de resposta do FIS (em cinza) e do Transyt (em preto) em função da variável de sincronia *ch* (Equação 28) para estimativas de número de paradas e tamanhos de filas, com níveis de saturação crescente. Foram escolhidos cenários aleatórios, com o fluxo de saturação igual a 1800 veículos. Os parâmetros utilizados na execução de cada gráfico estão nele indicados, bem como nível de saturação e o erro médio. Os FISs gerados possuem 64 regras e o tempo de execução para cada um deles foi de 50 microssegundos em microcomputador PC Pentium II 450MHz.

Pode-se observar que os valores estão bem ajustados nos dois primeiros gráficos, com a qualidade da estimativa decrescendo à medida que o *link* tende à saturação. Este comportamento não é inesperado. Quando o *link* se aproxima da saturação (*sat* tende a 1, conforme subseção 2.1.3), o modelo de simulação do Transyt apresenta variações abruptas mais difíceis de ajustar.



**Figura 25:** Estimativas de paradas (caso normalizado)



**Figura 26:** Estimativas de filas (caso normalizado)

O FIS relativo às filas não considera o tamanho do *link* sendo que, na realidade, os resultados da estimação do tamanho máximo de filas deveriam estar limitados à capacidade de estocagem. Se assim fosse, os erros seriam menores. Nota-se nos gráficos da Figura 26 que maiores erros na estimativa de filas se referem aos casos próximos à saturação, exatamente quando a estimativa de tamanho de filas é maior do que o tamanho do *link*.

A estimativa de filas aqui apresentada é dada em número de veículos, então para se ter o valor em metros utiliza-se a Equação 37.

$$fm = (fv / nf) \cdot dv \quad (37)$$

onde:

- $fm$  é a fila em metros;
- $fv$  é a fila em veículos;
- $nf$  é o número de faixas ( $fs / 1800$ );
- $dv$  é o distanciamento médio entre veículos, sugere-se considerar 7,5m.

Para se obter os resultados generalizados das estimativas de efetividade deve-se multiplicar a saída do FIS pela função  $h(\mathbf{fs}, \mathbf{fm})$  – que extrapola os resultados para um fluxo de saturação qualquer, já que a estimativa do FIS é normalizada - e aplicar a função  $g(\mathbf{d})$  – limitando a saída ao comprimento do *link* - conforme indicado na Equação 30 e representado na Figura 20.

As Figuras 27, 28 e 29 apresentam resultados de estimação do ANFIS (em cinza) e do Transyt (em preto) utilizando parâmetros aleatórios nas situações de fluxo médio, alto e próximo da saturação. Esses gráficos não foram traçados em termos da chegada do pelotão principal em relação ao início do verde (variável *ch*), mas sim em termos de defasagem (*offset*) dos inícios de verde. Retorna-se, assim, aos valores de tempo do simulador original (e do respectivo modelo matemático), sendo que a chegada do pelotão (*ch*) foi gerada para o FIS em termos de velocidade média, defasagem e tamanho do *link*, conforme a Equação 28. Assim como para o caso normalizado, os resultados obtidos estão bem ajustados quando fora de saturação, confirmando a validade do processo de eliminação das variáveis *fs* e *fm* pela normalização. Erros mais significativos somente ocorreram em situações próximas à saturação.

Pode-se notar uma descontinuidade nos gráficos de estimativas do ANFIS (como na Figura 29, gráfico da direita) que se refere ao problema de representação de uma variável cíclica: a defasagem incrementa de apenas 1 segundo, mas a variável fuzzy de chegada (representando *ch*) passa de “muito antes” para “muito depois”. Essa questão será tratada posteriormente na análise dos resultados obtidos (seção 5.4).

Fora de saturação, os erros médios das medidas de paradas são menores de 4% do fluxo total de veículos. O erro médio das filas em metros, calculado conforme a equação 37, é menor de 3% do tamanho do *link*. Por fim, os erros médios das medidas de atraso correspondem a menos de 5% do tempo de ciclo. A avaliação dos erros é apresentada na seção 5.4.

As medidas de efetividade dos *links* estimadas pelo ANFIS, aqui apresentadas, são utilizadas na composição do índice de desempenho global (PI) do sistema conforme a Equação 32. O índice de desempenho global obtido será posteriormente utilizado nos processos de otimização, sendo sua aplicação descrita na seção 5.3.

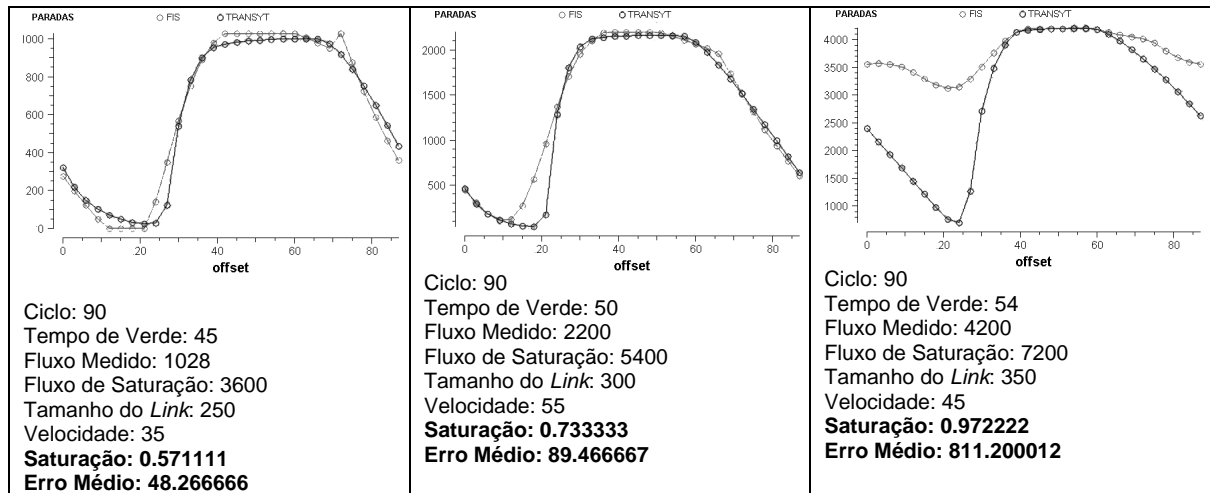


Figura 27: Estimativas de paradas (caso geral)

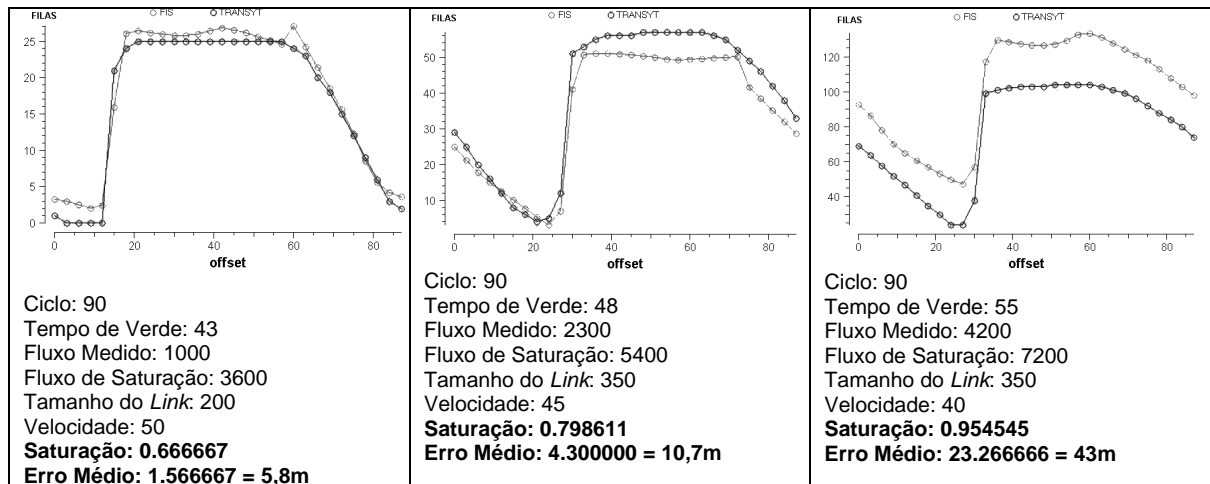


Figura 28: Estimativas de filas (caso geral)

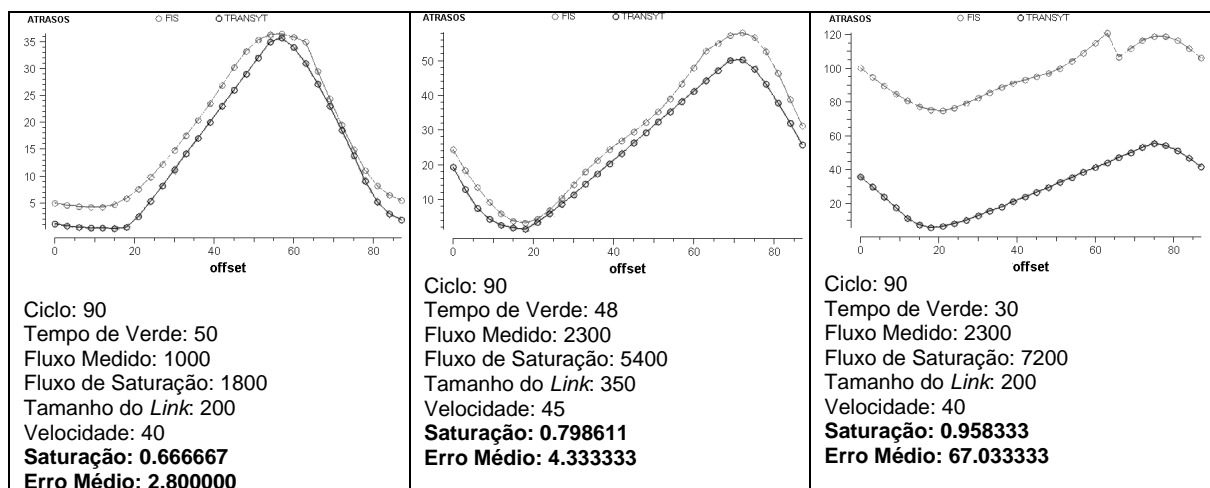


Figura 29: Estimativas de atrasos (caso geral)

## 5.2 ALGORITMO GENÉTICO COMO OTIMIZADOR DE DEFASAGENS

Validou-se o algoritmo genético de otimização de defasagens em separado, valendo-se do índice de desempenho calculado pelo Transyt na composição da função de *fitness*. A melhor configuração de parâmetros para o algoritmo genético, considerando tempo de processamento e resultados obtidos, foi a seguinte:

- Número de populações: 1
- Tamanho da população: 20
- Número de gerações: 100
- “*Generation gap*”: 95%
- Tipo de Seleção: Linear
- Tipo de *Crossover*: Simples
- Probabilidade de *Crossover*: 80%
- Probabilidade de Mutação: entre 0,5%

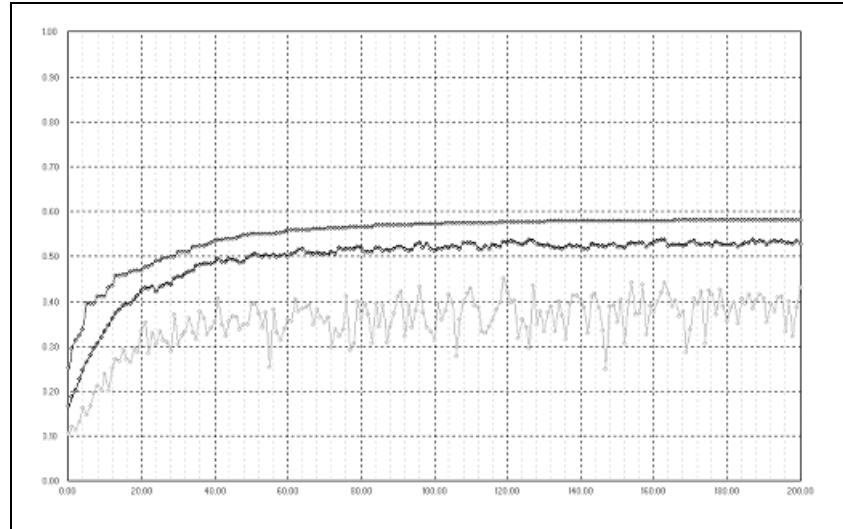
Observa-se na Figura 30, plotados para cada geração, os valores máximo, médio e mínimo do índice de desempenho (PI), em uma execução típica do algoritmo genético, utilizando a configuração de variáveis acima, mas em 200 gerações.

A variação dos parâmetros do algoritmo não apresentou melhoria significativa nos resultados. Percebe-se que para um número de gerações acima de cem, o algoritmo evolui pouco. Conseguiu-se uma maior garantia no resultado utilizando um conjunto de populações evoluindo independentemente e combinadas ao final (mas com maior custo computacional).

A utilização de algoritmos genéticos (GA) em geral não conduz a resultados melhores que o método *hill-climbing*, apesar do esforço computacional ser muito superior, demandando número elevado de simulações. Resultados comparativos são apresentados na seção 5.3, em termos de tempo de processamento e poder de otimização. Entretanto, o resultado é positivo nos seguintes aspectos:

- O GA mostrou-se capaz de otimizar sistemas de tráfego a níveis satisfatórios.
- Os resultados obtidos com GA foram bastante repetitivos, indicando a robustez do método.
- Como o GA não demanda a introdução de um estado inicial, pode ser utilizado previamente ao *hill-climbing* para inicializar as defasagens e assim conduzir a melhores resultados sem intervenção humana.

- Quanto ao tempo de processamento, a aplicação de GA pode ser viável no contexto de tempo real com a utilização de métodos mais rápidos de simulação e cálculo da função de *fitness* – o objetivo deste trabalho.



**Figura 30:** Evolução do PI para população=20 e gerações=200

### 5.3 MECANISMO DE OTIMIZAÇÃO HÍBRIDO COMPLETO

Tendo sido implementados o mecanismo de otimização e o cálculo de índice de desempenho global (PI) baseado no ANFIS, conforme a Equação 32, procedeu-se à execução do algoritmo em diferentes configurações, e em diferentes cenários, para avaliação da eficiência do método. Como neste trabalho não é implementada a interface para o sistema de controle real, a avaliação se dará em *offline*, tomando os resultados do Transyt como referência para inferir a qualidade de estimativas e dos algoritmos de otimização. Assim, cada cenário otimizado será avaliado tanto em termos da estimativa do ANFIS como do Transyt. Considera-se a eficiência real da otimização sobre as estimativas do ANFIS pela melhora do PI estimado pelo Transyt. Será utilizada aqui a seguinte notação para indicar o PI de um cenário:

- $PI_{ANFIS}$ : índice de desempenho de determinado cenário calculado pelas estimativas de medidas de efetividade de *links* dadas pelo ANFIS.
- $PI_{TRANSYT}$ : índice de desempenho para determinado cenário calculado pelo Transyt.

Inicialmente, parte-se da entrada de um cenário inicial composto de:

- Características físicas das vias representadas na Figura 18;
- Medições de detectores (medidas históricas registradas pelo sistema de controle em determinado dia, no intervalo de tempo da análise – horário de pico);
- Variáveis de tráfego (*splits* e defasagens) não inicializadas.

Adotando tempo de ciclo de 90s, com todas as defasagens iguais a zero e dividindo o tempo uniformemente entre os *splits*, tem-se  $PI_{ANFIS}=1951$  ( $PI_{TRANSYT}=1289$ ). Aplicando-se o EQUISAT para a inicialização dos *splits*, obtém-se o estado inicial para os processos de otimização, sendo obtido  $PI_{ANFIS}=786$  ( $PI_{TRANSYT}=754$ ), e os tempos para os 36 cruzamentos da subárea Batel apresentados na Tabela 5.

**Tabela 5:** Tempos dos cruzamentos após inicialização de *splits*

Cruzamento	Offset	Splits	Cruzamento	Offset	Splits	Cruzamento	Offset	Splits
1	0	41 25 09	13	0	30 50	25	0	43 37
2	0	38 42	14	0	41 39	26	0	50 30
3	0	27 46 06	15	0	39 41	27	0	30 50
4	0	24 56	16	0	46 34	28	0	37 43
5	0	29 20 26	17	0	45 35	29	0	45 35
6	0	27 53	18	0	32 48	30	0	31 49
7	0	31 41 03	19	0	37 43	31	0	35 45
8	0	39 41	20	0	49 31	32	0	38 42
9	0	38 42	21	0	32 48	33	0	42 38
10	0	41 39	22	0	56 24	34	0	51 29
11	0	17 63	23	0	48 32	35	0	41 39
12	0	34 46	24	0	58 22	36	0	42 38

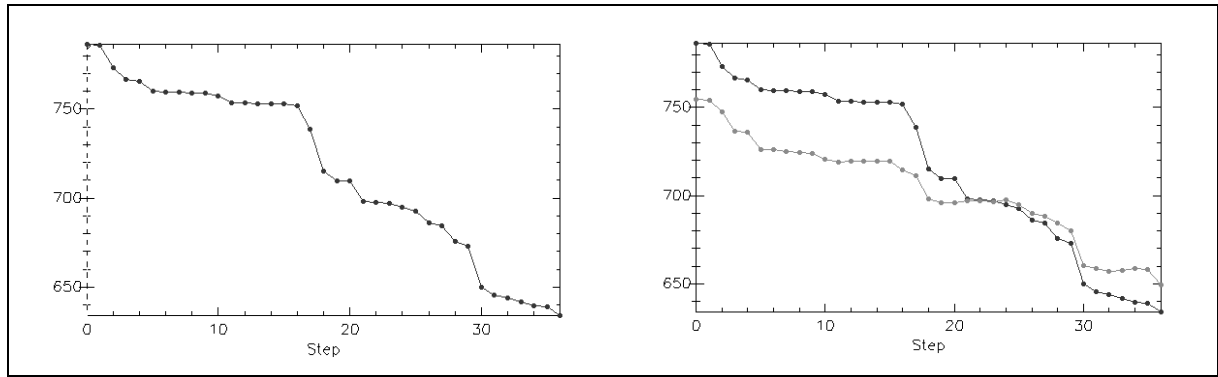
Aplicando-se o algoritmo de *hill-climbing* sobre as defasagens com um único tamanho de passo ( $1^d$ ) obtém-se o gráfico à esquerda da Figura 31. Nele foi plotado o PI após a execução da sucessão de passos de tamanho 1 em cada um dos 36 cruzamentos. O resultado final é  $PI_{ANFIS}=634.2$  ( $PI_{TRANSYT}=649.2$ ). O gráfico à direita apresenta a mesma função (em preto) e também a medida calculada pelo Transyt (em cinza) ao final da sucessão de passos em cada cruzamento.

A Figura 32 é análoga para a otimização de defasagens com os tamanhos de passo:

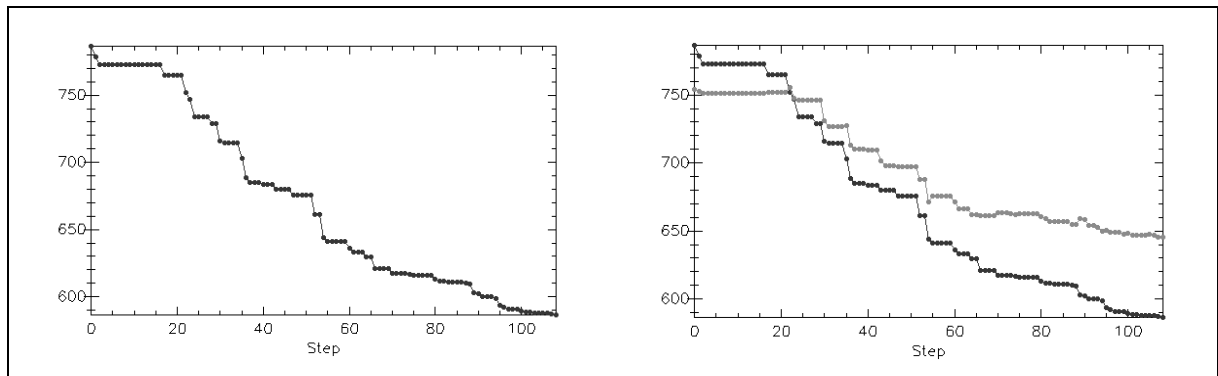
- $40^d$   $15^d$   $1^d$

A Figura 33 é análoga para a otimização de defasagens com os tamanhos de passo:

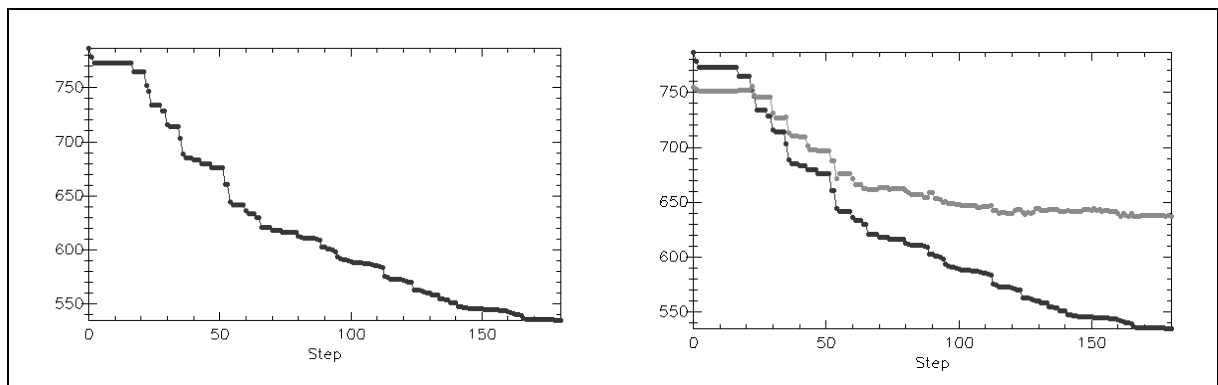
- $40^d$   $15^d$   $1^d$   $1^s$   $1^d$



**Figura 31:**  $PI_{ANFIS}$  e  $PI_{TRANSYT}$  em *hill-climbing* sobre  $PI_{ANFIS}$  ( $1^d$ )



**Figura 32:**  $PI_{ANFIS}$  e  $PI_{TRANSYT}$  em *hill-climbing* sobre  $PI_{ANFIS}$  ( $40^d, 15^d, 1^d$ )

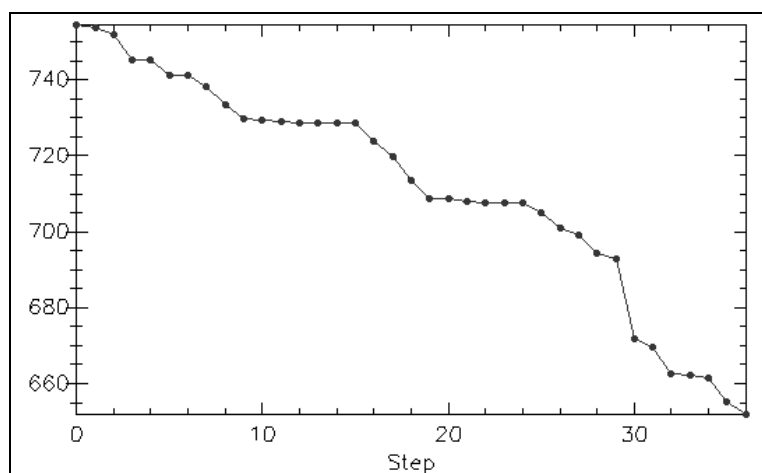


**Figura 33:**  $PI_{ANFIS}$  e  $PI_{TRANSYT}$  em *hill-climbing* sobre  $PI_{ANFIS}$  ( $40^d, 15^d, 1^d, 1^s, 1^d$ )

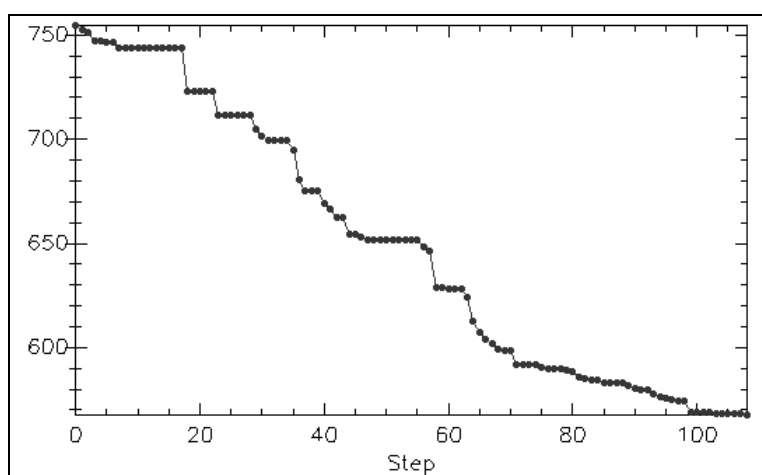
Para que se possa comparar o benefício da utilização do ANFIS, é representado nas figuras 34, 35 e 36 a execução do mesmo algoritmo *hill-climbing* sobre o cálculo do PI original do Transyt ( $PI_{TRANSYT}$ ), para os mesmos casos, respectivamente:

- $1^d$
- $40^d \quad 15^d \quad 1^d$
- $40^d \quad 15^d \quad 1^d \quad 1^s \quad 1^d$

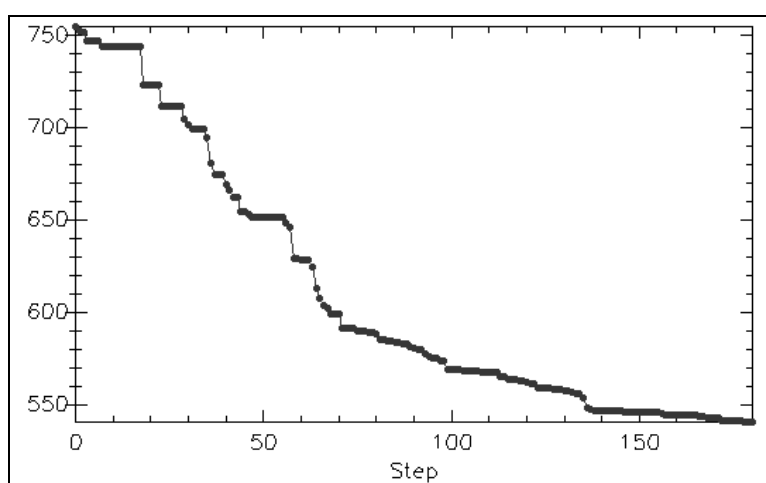




**Figura 34:** Hill-climbing sobre  $PI_{TRANSYT}$  ( $1^d$ )



**Figura 35:** Hill-climbing sobre  $PI_{TRANSYT}$  ( $40^d, 15^d, 1^d$ )



**Figura 36:** Hill-climbing sobre  $PI_{TRANSYT}$  ( $40^d, 15^d, 1^d, 1^s, 1^d$ )

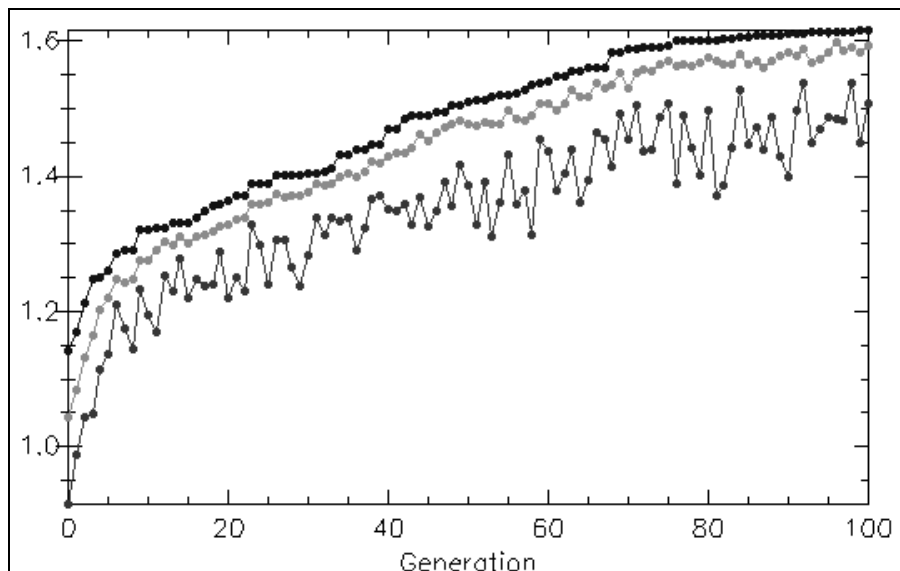
A figura 37 apresenta uma curva típica da evolução do algoritmo genético sobre o cenário inicial, plotando os valores máximo, médio e mínimo por geração. A curva não indica o PI original porque é necessária uma função de *fitness* crescente ao algoritmo, dada por:

$$F(x) = F(x) = 1000 / PI(x) \quad (38)$$

onde:

- $F(x)$  é a função de *fitness*;
- $PI(x)$  é o índice de desempenho calculado para o cenário.

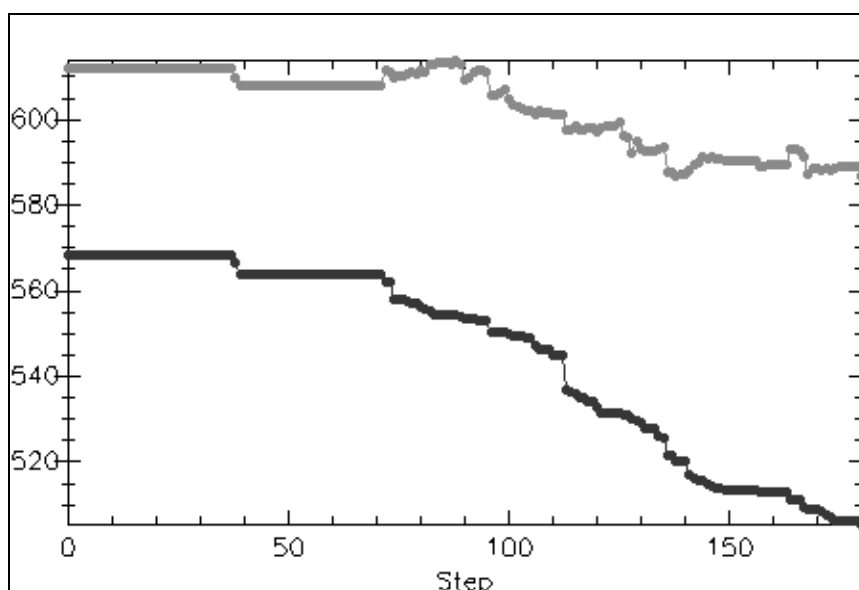
O algoritmo genético não é determinístico, mas conduziu a resultados em geral da ordem de  $PI_{ANFIS}=610$  e  $PI_{TRANSYT}=620$ .



**Figura 37:** Evolução por algoritmo genético do  $PI_{ANFIS}$

A Figura 38 apresenta a melhora do PI com o *hill-climbing* após a execução do algoritmo genético, utilizando como tamanhos de passo:  $40^d$ ,  $15^d$ ,  $1^d$ ,  $1^s$  e  $1^d$  e estimação com  $PI_{ANFIS}$ . Assim, essa figura representa o resultado obtido pela aplicação completa da estratégia de otimização, executada com o PI gerado pelas estimativas do ANFIS:

- Determinação do tempo de ciclo;
- Inicialização de *splits* com EQUISAT;
- Inicialização de defasagens com o algoritmo genético;
- Aplicação do *hill-climbing* com os passos:  $40^d$ ,  $15^d$ ,  $1^d$ ,  $1^s$ ,  $1^d$



**Figura 38:** *Hill-climbing* sobre  $PI_{ANFIS}$  ( $40^d, 15^d, 1^d, 1^s, 1^d$ )

Na Figura 38, a curva em preto representa o  $PI_{ANFIS}$ , enquanto a curva em cinza representa o  $PI_{TRANSYT}$ . Os valores são plotados após a execução de cada seqüência de passos, em cada cruzamento. Ilustrando, após a otimização os tempos para os 36 semáforos são apresentados na Tabela 6.

**Tabela 6:** Tempos dos cruzamentos após otimização

Cruzamento	Offset	Splits	Cruzamento	Offset	Splits	Cruzamento	Offset	Splits
1	43	42 22 11	13	83	34 46	25	73	41 39
2	21	35 45	14	84	38 42	26	66	54 26
3	80	28 45 06	15	82	38 42	27	1	30 50
4	80	24 56	16	82	41 39	28	54	41 39
5	69	39 10 26	17	83	40 40	29	64	45 35
6	4	24 56	18	22	30 50	30	18	28 52
7	85	36 36 03	19	70	37 43	31	68	36 44
8	83	39 41	20	78	42 38	32	22	43 37
9	80	39 41	21	73	34 46	33	7	41 39
10	78	42 38	22	38	55 25	34	18	45 35
11	0	19 61	23	24	46 34	35	75	40 40
12	84	36 44	24	89	57 23	36	19	43 37

A Tabela 7 resume os resultados obtidos, para os diversos tipos de otimização testados (*hill-climbing*, algoritmo genético e híbrido), sempre partindo de um cenário inicial com todas as defasagens iguais a zero e com os *splits* inicializados por EQUISAT. No cenário inicial,  $PI_{ANFIS} = 786.1$  e  $PI_{TRANSYT} = 754.40$ . A coluna “ANFIS” apresenta os PIs finais obtidos com a otimização sobre a estimativa do ANFIS ( $PI_{ANFIS}$ ). Ela é subdividida apresentando o valor do  $PI_{ANFIS}$  otimizado, bem como do  $PI_{TRANSYT}$  calculado para o cenário final otimizado, para que

se possa avaliar o benefício real alcançado, medido pelo sistema de referência. A coluna “TRANSYT” apresenta os resultados alcançados com os mesmos algoritmos de otimização utilizando a função objetivo dada pela estimativa do Transyt. A diferença entre o valor dessa coluna e o valor dado em “ANFIS /  $PI_{TRANSYT}$ ” indica a perda no poder de otimização com o emprego das estimativas do ANFIS, que evidentemente apresentam desvios em relação às estimativas originais do Transyt.

**Tabela 7:** Resultados de otimização

MÉTODOS Cenário Inicial: Defasagens=0 Splits=EQUISAT $PI_{ANFIS} = 786.1$ $PI_{TRANSYT} = 754.40$	ANFIS		TRANSYT
	Resultado Final $PI_{ANFIS}$	Resultado Final $PI_{TRANSYT}$	Resultado Final $PI_{TRANSYT}$
HC: 1 <sup>d</sup>	634.22	649.20	652.00
HC: 40 <sup>d</sup> 15 <sup>d</sup> 1 <sup>d</sup>	586.68	645.70	567.80
HC: 40 <sup>d</sup> 15 <sup>d</sup> 1 <sup>d</sup> 1 <sup>s</sup> 1 <sup>d</sup>	535.10	637.10	540.90
AG: população=20 gerações=100 (pior resultado em 10 execuções)	637.07	638.20	629.30
AG: população=20 gerações=100 (melhor resultado em 10 execuções)	608.49	610.80	573.40
AG + HC: 1 <sup>d</sup>	585.93	609.70	578.20
AG + HC: 40 <sup>d</sup> 15 <sup>d</sup> 1 <sup>d</sup>	578.66	617.50	569.40
AG + HC: 40 <sup>d</sup> 15 <sup>d</sup> 1 <sup>d</sup> 1 <sup>s</sup> 1 <sup>d</sup>	524.37	586.90	542.90

A Tabela 8 apresenta a comparação dos tempos de processamento do Transyt em relação ao ANFIS, considerando a execução de:

- Cálculo de PI;
- Otimização por GA com população de tamanho 20 e 100 gerações;
- Otimização por HC com os passos: 40<sup>d</sup>, 15<sup>d</sup>, 1<sup>d</sup>.

Os tempos foram medidos em microcomputadores Intel Celeron 400MHz e AMD Athlon 2.4GHz. São comparados os seguintes casos:

- Transyt no modo “*Single Cicle Link-wise*”;
- Transyt no modo “*Single Cicle Step-wise*”;
- Transyt no modo “*Multi Cicle Step-wise*” com tempo de simulação de 60 minutos;
- ANFIS utilizando estimativas de atrasos, paradas e filas no cálculo do PI;
- ANFIS utilizando apenas estimativas de atrasos e paradas no cálculo do PI.

No caso de otimização por algoritmos genéticos, o tempo de processamento é proporcional ao número de cenários avaliados (tamanho da população x número de gerações).

No caso do *hill-climbing* não, pois em cada incremento de passo somente o PI do cruzamento modificado é reavaliado, não sendo necessário recalcular todos os demais. Enquanto o algoritmo genético apresenta tempo de execução fixo, dado pelos parâmetros, o tempo para execução do *hill-climbing* é variável, pois não se sabe de antemão quantos passos serão dados.

**Tabela 8:** Tempos de execução

MÉTODO	Cálculo de PI 36 controladores		Otimização Alg. Gen população=20 gerações=100		Otimização <i>Hill-Climbing</i> passos = 40, 15, 1	
	Tempo (ms) Intel Celeron 400MHz	Tempo (ms) AMD Athlon 2.4GHz	Tempo (ms) Intel Celeron 400MHz	Tempo (ms) AMD Athlon 2.4GHz	Tempo (ms) Intel Celeron 400MHz	Tempo (ms) AMD Athlon 2.4GHz
Transyt: Single Cycle Link-wise	220 (*)	50 (*)	940.000	-	19.000	-
Transyt: Single Cycle Step-wise	1200 (*)	150 (*)	2.900.000	-	156.000	-
Transyt: Multi Cycle Step-wise: 60min	5300 (*)	900 (*)	11.100.000	-	530.000	-
ANFIS: Paradas+Atrasos+Filas	16	1.5	11.000	1.100	460	25
ANFIS: Paradas+Atrasos	11	1.0	8.000	900	350	20

(\*) Quando o Transyt é utilizado como um processo externo à aplicação para a obtenção do índice de desempenho deve-se adicionar 250ms (Celeron 400MHz) ou 200ms (Athlon 2.4GHz) no tempo de cada cálculo do PI, que corresponde à criação de arquivo de entrada, lançamento do processo do Transyt e leitura do arquivo de saída.

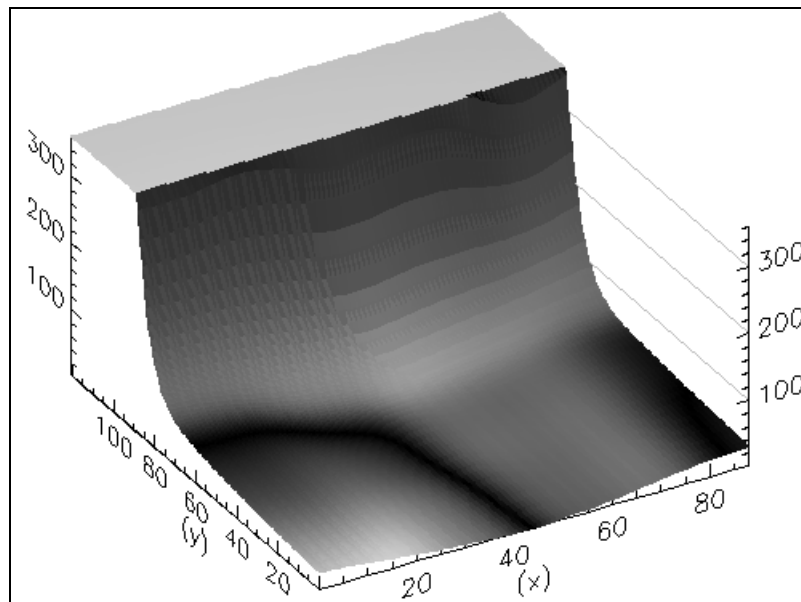
A Tabela 9 fornece uma referência para análise da viabilidade da utilização de algoritmos genéticos para substituir o conhecimento do especialista na geração de cenários iniciais favoráveis em otimização dinâmica. Nela são apresentados os resultados de otimização sobre um cenário inicial otimizado pelo Transyt após inicialização das variáveis por engenheiro de tráfego, com medidas históricas diferentes das utilizadas no processo de otimização agora executado. Pode-se comparar os dados com os da Tabela 7, onde o mecanismo de otimização foi executado sem nenhum conhecimento prévio do sistema.

**Tabela 9:** Resultados de otimização sobre cenário inicial pré-programado

MÉTODO Cenário Inicial estabelecido por conhecimento especialista (previamente otimizado considerando outra condição de tráfego) $PI_{ANFIS} = 693.16$ $PI_{TRANSYT} = 603.9$	ANFIS		TRANSYT
	Resultado Final $PI_{ANFIS}$	Resultado Final $PI_{TRANSYT}$	Resultado Final $PI_{TRANSYT}$
HC: 1 <sup>d</sup>	606.25	577.50	558.00
HC: 40 <sup>d</sup> 15 <sup>d</sup> 1 <sup>d</sup>	574.52	583.90	555.79
HC: 40 <sup>d</sup> 15 <sup>d</sup> 1 <sup>d</sup> 1 <sup>s</sup> 1 <sup>d</sup>	547.07	582.40	533.20

#### 5.4 ANÁLISE DOS RESULTADOS

Inicia-se a análise pelas estimativas de medidas de efetividade de *links*. A Figura 39 apresenta a superfície de resposta do mecanismo de estimativa de atrasos baseado no ANFIS. Nela foi considerado um tempo de verde de 40s para o *link*. No eixo  $x$  tem-se a defasagem, e no eixo  $y$  o fluxo dividido por 10. Considerando o fluxo de saturação de 1800vph, e o ciclo de 90s, o *link* apresenta grau de saturação maior que 1.0 para fluxos maiores de 800vph (corresponde a atrasos maiores que 200s e à parte seccionada do gráfico). Pode-se observar que a variação da estimativa nesse caso é exponencial, e assim as distorções aumentam à medida que se aproxima da saturação.

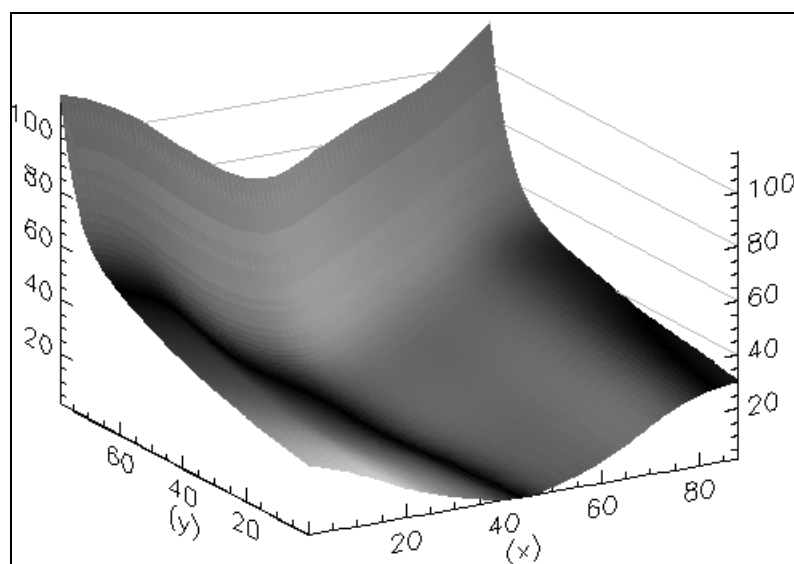


**Figura 39:** Superfície de resposta para estimativa de atrasos

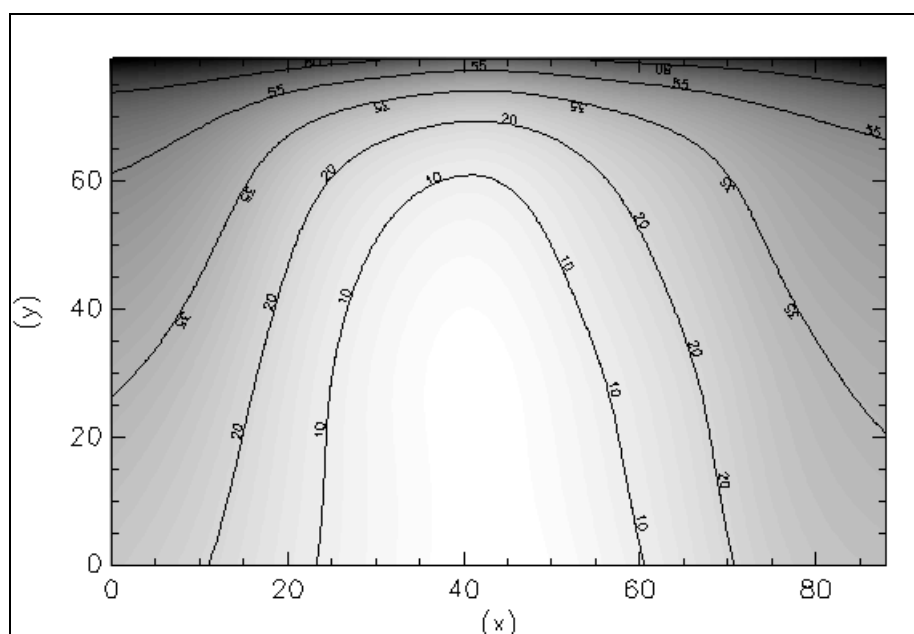
A mesma superfície é reproduzida na Figura 40, mas limitada ao intervalo de validade dos dados, ou seja, à região com o grau de saturação menor ou igual a 1.0. No treinamento os valores foram concentrados nessa faixa, pois não haveria sentido ter-se precisão nas estimativas em casos a princípio não reproduzíveis na prática.

A Figura 41 reproduz os mesmos dados da Figura 40 sob forma de gráfico de contorno. O eixo  $x$  representa a defasagem, e o eixo  $y$  o fluxo dividido por 10. Com o fluxo variando de 0 a 800, o grau de saturação varia de 0 a 1.0. Utilizando a definição de LOS - *Level of Service* (subseção 2.1.4) sobre a Figura 41, pode-se ver o grande impacto da correta

sincronia dos sinais. Pode-se assegurar uma boa progressão, com LOS A (atrasos  $\leq 10$ s) e B (atrasos  $\leq 20$ s) para um grau de saturação de até 0,87 (fluxo igual a 700). E pode-se ter uma progressão regular, com LOS C (atrasos  $\leq 35$ s) para um grau de saturação de até 0,93 (fluxo igual a 750).



**Figura 40:** Superfície de resposta para estimativa de atrasos ( $\text{sat} \leq 1.0$ )



**Figura 41:** Gráfico de contorno para estimativa de atrasos ( $\text{sat} \leq 1.0$ )

Conclui-se que para grau de saturação de até 0,93 seria possível ao sistema adaptativo manter o *link* operando em condições boas ou razoáveis. Acima desse valor é impossível,

independentemente da sincronia aplicada, tendendo o *link* rapidamente ao congestionamento. É evidente que normalmente não se pode aplicar o ponto ótimo ao *link*, pois se deve otimizar o somatório dos atrasos em todos os *links* do cruzamento. Essa análise permite concluir que as ações de otimização não apresentariam efetividade para graus de saturação maiores que 0,93. Além disso, a partir desse valor também se perde precisão nas estimativas, conforme apresentado na seção 5.1, na mesma faixa de valores onde o simulador de referência utilizado não apresenta qualidade na simulação (WALLACE et al., 1999).

Conforme apresentado nas Figuras 27, 28 e 29 e comentado na seção 5.1, os erros médios das estimativas do ANFIS foram pequenos em relação ao Transyt quando fora de saturação (grau de saturação menor que 0,95). Em saturação observou-se desvios por vezes grandes em relação aos dados originais, mas deve-se salientar que:

- Na saturação, os simuladores não garantem precisão nas suas estimativas.
- Maior precisão pode ser obtida pelo aumento do número de regras, se sua execução não violar as restrições de tempo.
- A simplificação que causou maior imprecisão foi desconsiderar-se os fluxos originados de conversões, incluindo-os no fluxo primário.
- O processo de otimização consiste em busca nas variáveis de defasagem e *splits*. Como se pode observar nos gráficos, picos e vales são bem definidos em relação às medidas de referência. Assim pode-se esperar que seja viável a execução de métodos de otimização valendo-se dessas estimativas.

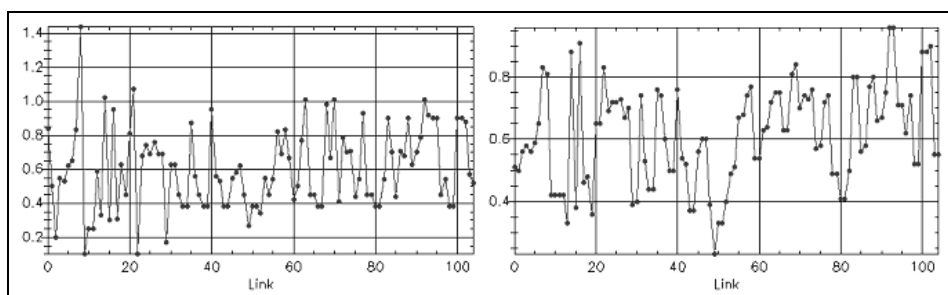
Considerando os processos apresentados na seção 5.3, observa-se primeiramente que, no caso de defasagens e *splits* não inicializados, a estimativa do índice de desempenho pelo ANFIS apresenta grande variação em relação ao Transyt ( $PI_{ANFIS}=1951$  e  $PI_{TRANSYT}=1289$ ). Isso não é uma surpresa, pois como os *splits* são aí definidos com uma divisão uniforme do tempo, muitos *links* apresentam grau de saturação maior que 1.0, e, como visto, nesse caso as estimativas são imprecisas.

Com a simples aplicação da condição de EQUISAT aos *splits*, as medidas do índice melhoraram sensivelmente ( $PI_{ANFIS}=786$  e  $PI_{TRANSYT}=754$ ), e a estimativa do ANFIS torna-se mais realista. Vale recordar que o índice é uma medida de ineficiência, e o sistema é tão mais eficiente quanto menor for o índice de desempenho (seção 4.3).

Alguns métodos de otimização *offline* ou *online* implementam unicamente o EQUISAT para a determinação de *splits*, pois apresenta resultados próximos aos obtidos com



etapas posteriores de otimização. A Figura 42 apresenta, para cada um dos 106 *links* do sistema, o grau de saturação calculado antes e depois da aplicação do EQUISAT. Vê-se uma melhoria sensível nos níveis de saturação, sendo que apenas dois *links* permanecem acima de 0,93. O grau de saturação não é alterado nas posteriores modificações de *offsets*, mas apenas nas alterações de tempos de verde.



**Figura 42:** Saturação inicial e após EQUISAT dos 106 Links

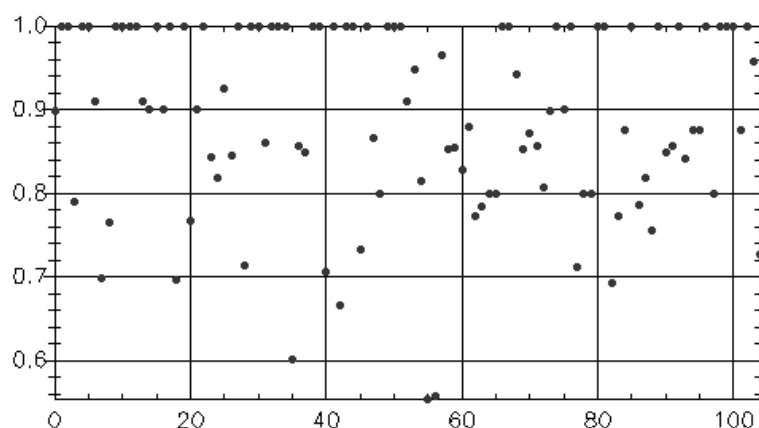
Como as estimativas dadas pelo ANFIS possuem erro associado em relação às medidas dadas pelo Transyt, não se pode esperar que a otimização baseada no PI gerado pelo ANFIS conduza a resultados iguais ou melhores em termos de  $PI_{TRANSYT}$ , em comparação com a aplicação do mesmo método diretamente sobre  $PI_{TRANSYT}$ . Embora a evolução do PI seja similar, com os algoritmos conduzindo a PI finais na mesma ordem de grandeza (como se pode observar nos gráficos das Figuras 31, 32 e 33, em comparação com os das Figuras 34, 35 e 36) os ganhos em desempenho são menores se avaliado o estado final calculado pelo Transyt ( $PI_{TRANSYT}$ ), conforme as Figuras 31, 32, 33 e 38 e a Tabela 1. Ou seja, uma solução ótima calculada em termos de  $PI_{ANFIS}$  não é um ótimo em termos de  $PI_{TRANSYT}$ . Como a referência de efetividade no processo de otimização é  $PI_{TRANSYT}$  então pela sua variação avalia-se a qualidade da otimização com a aplicação do método de estimativas pelo ANFIS.

A média da melhoria do índice de desempenho (ponderação de atrasos e paradas) nas diferentes modalidades testadas de *hill-climbing* sobre  $PI_{TRANSYT}$  é de 16%. O benefício médio nas mesmas situações otimizando  $PI_{ANFIS}$ , com resultado avaliado em termos de  $PI_{TRANSYT}$ , é de 10%. Esses valores fornecem uma dimensão da perda de poder de otimização. Entretanto, o ganho em tempo de processamento é significativo, sendo de 54 vezes no caso do *hill-climbing* e 117 vezes para o algoritmo genético (essa diferença sugere que a implementação do algoritmo *hill-climbing* neste projeto foi menos eficiente do que a implementação do Transyt). O ganho em tempo é muito maior ainda se comparado com o Transyt operando em modo “*step-wise*” ou “*step-based*” (subseção 2.1.6).

Verifica-se na Tabela 7 que, de modo geral, se alcançou melhores resultados com a inicialização de defasagens por algoritmo genético. Pela comparação das Tabelas 7 e 9, o emprego de algoritmos genéticos mostrou-se menos eficiente do que a utilização de pré-programação gerada por um especialista para definir o estado inicial dos tempos para o *hill-climbing*. Entretanto, apresenta a vantagem de poder ser executado sem intervenção humana, e pode possibilitar uma melhor acomodação a situações não usuais quando executado dinamicamente. Embora o algoritmo genético demande um esforço computacional muito grande, se utilizar-se as estimativas do ANFIS para cálculo do índice de desempenho e processadores acima de 2,8GHz, consegue-se execução em menos de um segundo para uma subárea de 36 cruzamentos, com os parâmetros do algoritmo utilizados neste trabalho.

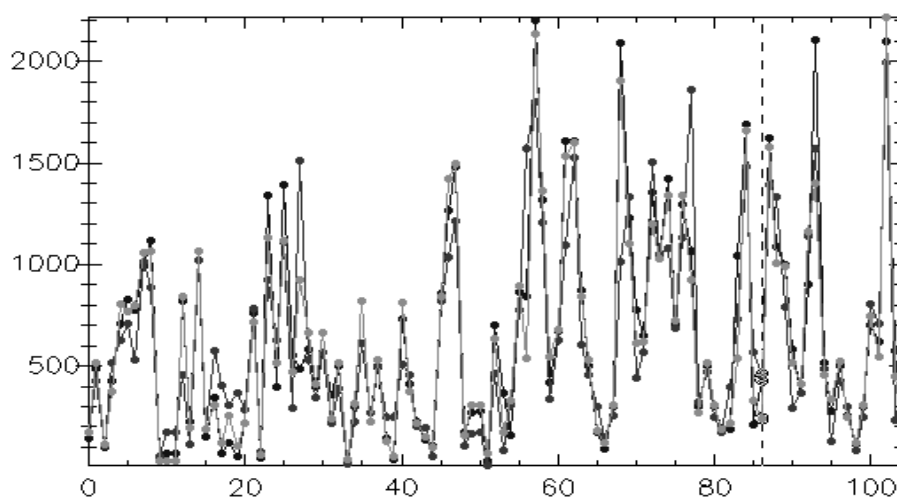
Apesar da capacidade de otimização reduzida, a utilização do ANFIS possibilita a otimização em contexto de tempo real de um grande número de cruzamentos. Mesmo com a etapa de aplicação de algoritmos genéticos, estima-se, pelos dados apresentados na Tabela 8, que até 20 subáreas com 36 cruzamentos cada podem ser processadas na janela alocada de 20s com a tecnologia atual. Isso seria impraticável pela utilização de um simulador como o Transyt ou o cálculo pelo modelo matemático utilizado por ele, definido em (TRB, 2000), principalmente se em modo “*step-wise*” (mais preciso e custoso).

O fator limitante na capacidade de otimização é o erro das estimativas. Observa-se que o maior erro não é dado pela utilização do ANFIS em si, mas pelas simplificações das variáveis envolvidas nos processos de determinação de variáveis *fuzzy* e geração dos dados de treinamento. A normalização realizada em relação ao fluxo (trabalha-se com valores de fluxo por faixa) acarreta pequeno desvio. A supressão das variáveis de distância, velocidade média e defasagens, substituída por uma única variável que indica a chegada do pelotão principal (*ch*), praticamente não introduz erro. Dependendo da parametrização do Transyt, o resultado é idêntico ou muito próximo. Por outro lado, desconsiderar a decomposição dos fluxos introduz erro significativo, que tende a ser maior à medida que diminui a participação do fluxo primário na composição do fluxo total. A Figura 43 plota a participação do fluxo primário sobre o fluxo total para todos os 106 *links* do sistema.



**Figura 43:** Participação do fluxo primário por link

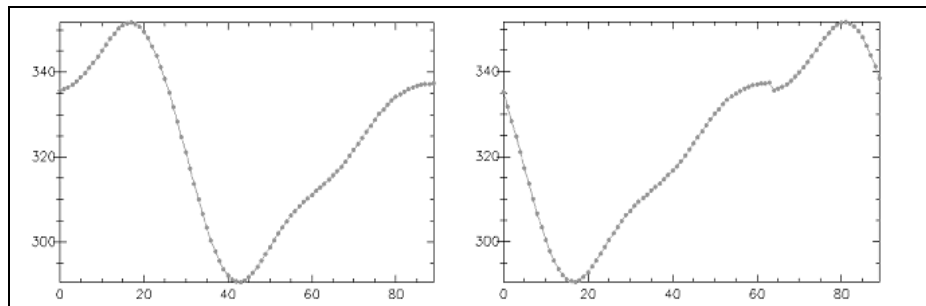
Os erros do ANFIS em relação aos dados de treinamento são pequenos. A Figura 44 plota as estimativas de paradas para os 106 *links* analisados, dadas pelo Transyt, pela função de geração de dados de treinamento, e pelo ANFIS. O erro médio entre valores do Transyt e os dados pela função de geração de dados de treinamento são da ordem de 17%, enquanto que destes para as estimativas do ANFIS é de 2,3%. Assim, o ANFIS mostrou-se um excelente aproximador e o método de otimização poderia ser mais eficiente se fosse concebida uma tradução em variáveis *fuzzy* que não levasse à perda da informação de conversões. O erro do ANFIS poderia ainda ser menor pelo aumento do número de regras, com o respectivo aumento do custo computacional.



**Figura 44:** Comparação entre estimativas de paradas por link

Outra característica do sistema proposto que limita a otimização é a natureza cíclica da variável de sincronia, com a variável *fuzzy* adotada assumindo valores “muito antes”, “antes”,

“depois” e “muito depois”. Para o ciclo de 90s, os instantes de chegada -45s e 45s são equivalentes e tem-se que o mesmo instante de chegada pode corresponder a “muito depois” ou “muito antes”. A Figura 45 apresenta à esquerda o gráfico de resposta de atraso para um único *link* com tempo de verde de 30s e fluxo de 680, em função da defasagem. Essa função, para um único link, deveria ser monotônica, e assim parece ser quando a chegada do pelotão ocorre no início do verde para uma defasagem de 45s (o centro do gráfico). O gráfico da direita reproduz uma situação equivalente, mas para a chegada do pelotão coincidindo com o início do verde em uma defasagem de 20s. Pode-se notar a descontinuidade no gráfico. Essa característica pode interferir na busca por *hill-climbing*, fazendo-a estagnar no ponto errado (no ponto de defasagem igual a 65s, nesse caso).



**Figura 45:** Descontinuidade devido à variável cíclica (*ch*)

## 5.5 CONCLUSÃO

Neste capítulo foram apresentados os resultados obtidos com a implementação do protótipo da proposta e, a seguir, a análise detalhada desses resultados. A análise deu-se em relação a estimação de desempenho assim como otimização de cenários. Buscou-se levantar dados para confirmar a aplicabilidade do modelo proposto, tanto em termos de requisitos temporais como em qualidade dos resultados de estimação e otimização. No próximo capítulo a avaliação do modelo será resumida e serão apresentadas as conclusões gerais do trabalho.

## CAPÍTULO 6

### DISCUSSÃO E CONCLUSÕES

Este capítulo é o fechamento deste trabalho de dissertação de mestrado. Na seção 6.1 são recapitulados os objetivos gerais do trabalho e os principais aspectos da proposta. A seguir, a seção 6.2 sumariza, qualitativamente, os resultados apresentados no Capítulo 5, enumerando elementos que possibilitem determinar a validade da abordagem. As seções seguintes trazem as conclusões gerais do trabalho e sugestões para continuidade desta pesquisa.

#### 6.1 RECAPITULAÇÃO DOS OBJETIVOS DO TRABALHO

Principiou-se este trabalho com o objetivo de criar um método de otimização dinâmica de tráfego aplicável ao sistema de controle da cidade de Curitiba, baseado na minimização de um índice de desempenho definido por uma ponderação de medidas de ineficiência dos *links*. Apresentou-se um sistema de otimização dinâmica de referência (SCOOT) que também opera segundo a mesma filosofia de otimização de um índice de desempenho, calculado por modelos matemáticos, com a utilização de algoritmo tipo *hill-climbing*.

Entretanto, o sistema de controle de Curitiba não apresenta a informação em tempo real do fluxo originário de conversões, dispondo apenas das contagens de veículos totalizada a cada ciclo. Já o SCOOT registra as medidas de contagem discretizadas em pequenos intervalos de tempo, obtendo assim o perfil do fluxo que alimenta o *link*. A partir desse perfil, aplicando-se uma função de dispersão de pelotão, o SCOOT obtém o perfil do fluxo de chegada no cruzamento e, considerando os tempos assumidos no cenário dado para o semáforo em questão, determina o percentual de veículos que chega no vermelho, o número de paradas e os atrasos esperados.

No caso da utilização de valores de contagem totalizados no ciclo, é necessário inferir o perfil do fluxo de chegada no mesmo espírito dos otimizadores *offline* como o Transyt. Otimizadores *offline* também operam com a otimização de um índice de desempenho, mas são alimentados, além das características físicas, apenas com a contagem de veículos por hora, pois os dados são usualmente obtidos a partir de medições manuais. Em simuladores *offline* o

perfil do fluxo de chegada é estimado pelas condições de defasagem apresentadas pelo cenário, pelo percentual do fluxo total gerado por conversões em cada *link* e por premissas quanto ao comportamento do fluxo, valendo-se de um modelo matemático. O percentual do fluxo originado de cada conversão é medido manualmente e configurado no simulador como parâmetro fixo, para cada conversão que alimenta cada *link*.

Dessa forma, o objetivo básico da proposta seria a aplicação de um algoritmo similar ao Transyt em tempo real. O principal problema dessa abordagem seria o tempo de execução de tais algoritmos, que inviabilizaria o controle dinâmico em sistemas com grande número de cruzamentos. Na proposta deste trabalho manteve-se os mesmos métodos de otimização empregados em sistemas de otimização existentes (*hill-climbing* e algoritmos genéticos), mas buscou-se uma alternativa para um cálculo mais rápido do índice de desempenho, de forma que uma solução otimizada pudesse ser alcançada na janela de tempo disponível para o comando de planos ciclo a ciclo a partir do sistema de controle central.

A abordagem proposta foi a utilização do ANFIS para a estimativa de medidas de efetividades de *links*, cuja ponderação determina o índice de desempenho do sistema. Apresentou-se um método de tradução das variáveis do problema em um modelo ANFIS de pouca complexidade e baixo custo computacional, possibilitando-se executar os algoritmos de otimização em tempos compatíveis com os requisitos do sistema.

O ANFIS pode ser encarado como um aproximador genérico, cujo erro é tanto menor quanto maior for seu número de regras. Seu treinamento foi realizado com dados de efetividade de *links* gerados pelo próprio Transyt, sistema cujo comportamento busca-se reproduzir, embora a mesma abordagem poderia ser aplicada valendo-se de dados obtidos por medições sistemáticas de sistemas reais. Optou-se por reduzir-se o número de variáveis e regras para se ter melhores tempos de resposta, embora se prevendo alguma perda na qualidade das estimativas, e, por consequência, na efetividade da otimização.

## 6.2 SUMÁRIO DOS RESULTADOS OBTIDOS

Tendo em vista os objetivos deste trabalho e considerando a análise detalhada da seção 5.4, pode-se enumerar os seguintes resultados principais da pesquisa:

- Demonstrou-se que o mecanismo de controle adaptativo pode ser efetivo para *links* que apresentem grau de saturação de até 93%.

- A abordagem proposta com geração de estimativas por sistemas *neuro-fuzzy* (ANFIS) viabilizou a execução *on-line* de algoritmos de otimização para um grande número de cruzamentos.
- O ANFIS mostrou-se um excelente aproximador, com erro médio de 2,3% em relação a valores obtidos com a função de geração de dados de treinamento.
- A tradução do modelo do *link* para variáveis *fuzzy* buscou gerar um FIS de baixa complexidade. Várias variáveis que definem o desempenho de um *link* foram simplificadas ou agregadas em uma mesma variável *fuzzy*. Algumas dessas simplificações mostraram-se adequadas, não comprometendo os resultados. A simplificação que implicou em maior erro foi a desconsideração dos fluxos secundários, incluídos no fluxo principal. Por isso, embora o ANFIS apresente baixo erro em relação aos dados de treinamento, os dados de treinamento apresentaram erro significativo em relação ao simulador de referência (Transyt), sendo o erro médio das estimativas da ordem de 17%.
- Apesar da variação nas estimativas em relação ao Transyt, ao serem executados algoritmos de otimização sobre o índice de desempenho calculado com o ANFIS, observou-se a melhora do índice de desempenho dado pelo Transyt no cenário otimizado. Isso confirma que as estimativas do ANFIS podem ser empregadas em um processo de otimização com baixo esforço computacional, embora com alguma perda associada na capacidade de otimização.
- Observou-se que, para cenários extremamente desfavoráveis (não inicializados), os resultados de otimização são próximos aos obtidos com o Transyt, conduzindo a uma grande redução nas estimativas de atrasos, paradas ou combinação desses. Para cenários mais ajustados, obteve-se uma ordem de grandeza menor na redução dessas medidas (em torno de 16%). Nesse caso observou-se que a avaliação pelo Transyt dos cenários inicial e otimizado aponta um benefício da ordem de 10%, e assim estimou-se uma perda da ordem de 37% em capacidade de otimização com o emprego das estimativas do ANFIS.
- Por outro lado, o tempo de execução dos algoritmos de otimização com o uso do ANFIS correspondeu a 0,6% do tempo obtido com o Transyt em modo *Single Cycle Step-wise*.

### 6.3 CONCLUSÕES

Como conclusão geral, os resultados obtidos indicam a viabilidade da utilização de sistemas *neuro-fuzzy* para a otimização dinâmica de tráfego em situações não saturadas. Embora o modelo de estimação proposto implique erros nas estimativas e menor efetividade no processo de otimização em relação à utilização do modelo do Transyt, permite sua aplicação a um grande número de cruzamentos pelo seu baixo tempo de processamento. Em sistemas de controle de tráfego compostos por um grande número de cruzamentos, otimizadores baseados no modelo do Transyt não poderiam ser empregados em tempo real por não atenderem às restrições de tempo dadas pelo sistema.

Com relação à utilização de algoritmos genéticos, observou-se que, apesar do seu custo computacional muito mais elevado, via de regra não obtém por si só resultados melhores do que algoritmos tipo *hill-climbing* na otimização de defasagens. Entretanto, a otimização por *hill-climbing* é mais efetiva se precedida pela otimização com algoritmos genéticos, pois assim se evita que o algoritmo *hill-climbing* permaneça estagnado em máximos locais. A viabilidade da aplicação de algoritmos genéticos em tempo real depende, então, do tempo de cálculo do índice de desempenho (função de *fitness*). Mostrou-se que é possível sua aplicação, sobre o índice calculado com o ANFIS, em momentos de reavaliação global de tempos, que ocorre tipicamente em intervalos maiores que uma hora. Isso não é restritivo, visto que otimizações por algoritmo genético não podem ser executadas ciclo a ciclo por gerar mudanças bruscas na defasagem, impedindo a estabilização do sistema.

Quanto à aplicabilidade no sistema de controle de tráfego da cidade de Curitiba, mostrou-se que as 20 subáreas podem ser atendidas por algoritmos executados em tempo real com a capacidade de processamento atual. O mecanismo provê saídas (*splits*, defasagens) compatíveis com as comandadas pelo sistema de controle e demanda entradas conforme disponibilizadas por ele – contagens de veículos totalizadas no ciclo. Por fim, o sistema apresenta pouca parametrização, sendo basicamente alimentado por arquivos de *layout* físico já utilizados no sistema de otimização *offline* existente, além de poucos parâmetros de segurança. Assim, supõe-se que o método seja de fácil inserção no sistema de controle da cidade de Curitiba.

Quanto à aplicabilidade em outros sistemas, a abordagem somente é justificada em sistemas de controle que possuam apenas os dados integrados por ciclo (ou período maior de



tempo). Em sistemas que possuem dados de detectores discretizados em pequenos intervalos de tempo, outras abordagens são mais adequadas, como a adotada pelo SCOOT.

#### 6.4 TRABALHOS FUTUROS

Como primeira sugestão de continuidade desta pesquisa propõe-se o enriquecimento das estimativas do ANFIS através do desenvolvimento de um modelo de representação *fuzzy* que considere a decomposição de fluxos, mas ainda atendendo aos requisitos temporais para sua execução. Desta forma, supõe-se que os erros sejam menores e que os resultados de otimização sejam mais efetivos. Além disso, para ser genérico, o sistema ANFIS deveria incluir o tempo de ciclo como variável.

Em segundo lugar, sugere-se a realização de uma análise do comportamento dinâmico do sistema de otimização, considerando condições de fluxo variáveis no tempo, para se estimar o benefício obtido com a aplicação em sistemas reais. Assim seria possível avaliar se os resultados esperados seriam compatíveis aos apresentados na Tabela 1, que apresenta uma média de resultados obtidos em diferentes sistemas de otimização dinâmica nos Estados Unidos, quanto à melhoria de índices de desempenho.

A seguir, propõe-se a inclusão do mecanismo de otimização proposto no sistema de controle de tráfego operante e a verificação do seu comportamento dinâmico em situações reais. Essa avaliação poderia ser feita também por estimativas de medidas de desempenho, mas principalmente através de medições em campo como, por exemplo, comparativos de tempos de trajetos e medições reais de filas, para casos com e sem controle dinâmico em condições de fluxo similares.

Por fim, sugere-se a formação de um índice de desempenho que reflita o custo operacional do sistema, como uma função de consumo de combustível e desgaste de peças. Existem modelos que estimam esses valores para cada *link* em função de distância percorrida, atrasos e paradas. Assim, a função a ser otimizada (custo) representaria um valor com significado prático, mais compreensível pela administração do sistema de transporte, fornecendo uma percepção mais clara dos benefícios obtidos com a otimização dinâmica.



## REFERÊNCIAS BIBLIOGRÁFICAS

- AKCELIK, R. **Traffic Signals: Capacity and Timing Analysis**. ARRB Research Report 123, Australia, 1981.
- AKCELIK, R., ROUPHAIL, N. **Estimation of Delays at Traffic Signals for Variable Demand Conditions**. Transportation Research Board, 1993.
- BOILLOT, F. et al. **Optimal signal control of urban traffic networks**. Sixth International Conference on Road Traffic Monitoring and Control, IEE London, 1992.
- CHAUDHARY, N., KOVVALI, V., ALAM, S. **Guidelines for Selecting Signal Timing Software**. Texas Transportation Institute, 2002.
- DATAPROM. **Sistema ANTARES - Manual de Operação**. Dataprom – Equipamentos e Serviços de Informática Industrial Ltda, 1999.
- FHWA. **Developing Traffic Signal Control Systems Using the National ITS Architecture**. US Department of Transportation, Federal Highway Administration, 1998.
- FHWA. **ITS Benefits: Continuing Successes and Operational Test Results**. US Department of Transportation, Federal Highway Administration, 1997.
- FOY, M., BEBEKOHAL, R. and GOLDBERG, D. **Signal Timing Determination Using Genetic Algorithms**. Transportation Research Record 1365, National Academy Press, 1992.
- GARBACZ, M. **Adaptive Signal Control: What To Expect**. 2002.
- GARTNER, N, MESSER, C., RATHI, A. **Traffic Flow Theory**. Capítulo 1, Oak Ridge National Laboratory, 1997.
- GOLDBERG, D. **Genetic Algorithms in Search, Optimization and Machine Learning**. Addison-Wesley, 1998.
- HAGEN, L., COURAGE, K. **Comparison of Macroscopic Models for Signalized Intersections**. Transportation Research Board, 1989.

- HALL, F. **Traffic Stream Characteristics**. Capítulo 2 em Gartner et al. Oak Ridge National Laboratory, 1997.
- HOLLAND, J. **Adaptation in natural and artificial systems**. University of Michigan Press, 1975.
- HUNT, P.B. **SCOOT - A Traffic Responsive Method of Coordinating Signals**. Department of Transport, Crowthorne, 1981.
- ITE. **Traffic Engineer Handbook**. Institute of Transportation Engineers. Prentice-Hall, 1992.
- JANG, J.S.R. **ANFIS: Adaptive-Network-Based Fuzzy Inference System**. IEEE Transactions on Systems, Men and Cybernetics, 1993.
- JANG, J.S.R. **Fuzzy modeling using generalized neural networks and Kalman filter algorithm**. Ninth Nat.Conf. Artificial Intelligence, 1991.
- JANG, J.S.R., SUN, C.T., MIZUTANI, E. **Neuro-Fuzzy and Soft Computing**. Prentice Hall, 1997.
- LIEBERMAN, E., RATHI, A.K. **Traffic simulation**. Capítulo 10 em Gartner et al. Oak Ridge National Laboratory, 1997.
- LINT, J., HOOGENDOORN S., ZUYLEN, H. **State Space Neural Networks for Freeway Travel Time Prediction**. 2002.
- MATHWORKS. **Fuzzy Logic Toolbox Help**. Matlab 6.5, The MathWorks Inc, 2002.
- McSHANE, W., PRASSAS, E., ROESS, R. **Traffic Engineering**. Prentice Hall, 1998.
- PAPAGEORGIOU, M. et al. **Review of road traffic control strategies**. Proceedings of the IEEE 91, 2003.
- PEDRYCZ, W., GOMIDE, F. **An Introduction to Fuzzy Sets**. MIT Press, 1998.
- ROBERTSON, D. **TRANSYT-A Traffic Network Study Tool**. Road Research Laboratory, Crowthorne, Berkshire,U.K., 1969.
- ROUPHAIL, N., TARKO, A., LI, J. **Traffic flow at signalized intersection**. Capítulo 9 em Gartner et al. Oak Ridge National Laboratory, 1997.

- SIMS, A.G. **SCATS: the Sydney Coordinated Adaptive Traffic System**. Engineering Foundation Conference on Research Priorities in Computer Control of Urban Traffic Systems, 1979.
- TAKAGI, T., SUGENO, M. **Derivation of Fuzzy control rules from human operator's control actions**. 1983.
- TRB. **Highway Capacity Manual 2000**. Transportation Research Board, 2000.
- TRB. **Highway Capacity Manual Special Report 209**. Transportation Research Board, 1997.
- TTI. **Passer V User's Documentation** . Texas Transportation Institute, 2002.
- VALDES, A. **Ingenieria de Trafico**. Ed. Dossat, Madrid, 1978.
- WALLACE, C.E. et al. **TRANSYT-7F release 8.2 Users Guide**. University of Florida, Transportation Research Center, 1999.
- WEBSTER, F.V. **Traffic Signals Settings**. Road Research Technical Paper 39, London, 1958.
- ZADEH, L.A. **Fuzzy sets**. Information and Control, 1965.
- ZADEH, L.A. **Outline of a new approach to the analysis of complex systems and decision processes**. 1973.

