



**HENRIQUE DEZANI**

**OTIMIZAÇÃO DO FLUXO DO TRÁFEGO URBANO MODELADO EM REDES DE  
PETRI UTILIZANDO ALGORITMO GENÉTICO**

**CAMPINAS  
2012**



**UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO**

**HENRIQUE DEZANI**

**OTIMIZAÇÃO DO FLUXO DO TRÁFEGO URBANO MODELADO EM REDES DE  
PETRI UTILIZANDO ALGORITMO GENÉTICO**

**Orientador: Prof. Dr. Furio Damiani**

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas para obtenção do título de Doutor em Engenharia Elétrica. Área de concentração: Eletrônica, Microeletrônica e Optoeletrônica.

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA TESE  
DEFENDIDA PELO ALUNO HENRIQUE DEZANI  
E ORIENTADA PELO PROF. DR. FURIO DAMIANI

Assinatura do Orientador

---

**CAMPINAS  
2012**



FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

D533o	<p>Dezani, Henrique Otimização do fluxo do tráfego urbano modelado em redes de Petri utilizando algoritmo genético / Henrique Dezani. --Campinas, SP: [s.n.], 2012.</p> <p>Orientador: Furio Damiani. Tese de Doutorado - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.</p> <p>1. Engenharia de tráfego. 2. Sistemas Inteligente de controle. 3. Redes de Petri. 4. Algoritmo genético. 5. Otimização. I. Damiani, Furio, 1943-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.</p>
-------	---

Título em Inglês: Optimization of urban traffic stream modeled in Petri net using genetic algorithm

Palavras-chave em Inglês: Traffic Engineering, Intelligent control systems, Petri nets, Genetic algorithm, Optimization

Área de concentração: Eletrônica, Microeletrônica e Optoeletrônica

Titulação: Doutor em Engenharia Elétrica

Banca examinadora: Norian Marranghello, Carlos Magnus Carlson Filho, Leandro Tiago Manera, Romis Ribeiro de Faissol Attux

Data da defesa: 05-12-2012

Programa de Pós Graduação: Engenharia Elétrica

## COMISSÃO JULGADORA - TESE DE DOUTORADO

**Candidato:** Henrique Dezani

**Data da Defesa:** 5 de dezembro de 2012

**Título da Tese:** "Otimização do Fluxo do Tráfego Urbano Modelado em Redes de Petri Utilizando Algoritmo Genético"

Prof. Dr. Furio Damiani (Presidente): \_\_\_\_\_  
Prof. Dr. Norian Marranghello: \_\_\_\_\_  
Prof. Dr. Carlos Magnus Carlson Filho: \_\_\_\_\_  
Prof. Dr. Leandro Tiago Manera: \_\_\_\_\_  
Prof. Dr. Romis Ribeiro de Faissol Attux: \_\_\_\_\_

*Para minha família.*

## AGRADECIMENTOS

Ao meu orientador, professor Furio Damiani, pela inspiração concebida e compartilhada e pela confiança depositada no desenvolvimento deste trabalho.

Aos meus pais, Rui Dezani e Lucianete Mendes Dezani, ao meu irmão e família, pelo amor e carinho que sempre priorizaram minha educação e acreditaram em mim, apoiando-me e auxiliando-me, para que eu pudesse realizar meus trabalhos e sonhos.

Ao professor Norian Marranghello, da Universidade Estadual de São Paulo, pelo constante incentivo, sempre indicando, com extrema sabedoria, a direção a ser tomada nos meus estudos.

Ao professor Luis Gomes, da Universidade Nova de Lisboa, pelo auxílio no desenvolvimento da minha tese de doutorado e por proporcionar-me uma grande experiência acadêmica no período em que estive em Portugal.

Aos meus professores do doutorado, especialmente os professores Romis Ribeiro de Faissol Attux, Fernando José Von Zuben e Alice Maria B. H. Tokarnia, que sempre lecionaram aulas excepcionais e inspiraram a busca pelo conhecimento.

Ao professor Carlos Magnus Carlson Filho, que sempre me inspirou nas áreas acadêmica, profissional e pessoal, sendo um ótimo professor e amigo.

À Jaqueline Bisson, pela grande ajuda prestada durante todos os anos em que estive realizando o doutorado.

Aos meus amigos de trabalho da Faculdade de Tecnologia de São José do Rio Preto, em especial aos professores Eliana Magrini Fochi, Adriano Luís Simonato, Evanivaldo Castro Silva Jr., Mariângela Cazetta, Djalma Domingos da Silva, Luciene Cavalvanti Rodrigues, José Aparecido de Aguiar Viana e Waldir Barros Fernandes Jr.

Ao Centro Estadual de Educação Tecnológica Paula Souza, pelo incentivo ao desenvolvimento do meu doutorado.

À CAPES, pelo suporte financeiro para a realização do meu doutorado “sanduíche” e participação dos congressos internacionais.

Aos meus amigos, pelo apoio e paciência, durante todos esses anos de pesquisa.

A Deus, por estar sempre comigo e iluminar o meu caminho.

*“Computer science is no more about computers than astronomy is about telescopes”*  
(Edsger W. Dijkstra)

## RESUMO

O controle do tráfego urbano constitui um importante tópico de pesquisa nos dias atuais. De fato, a cada dia aumenta a quantidade de veículos trafegando nas vias urbanas, as quais nem sempre são modificadas para a melhoria do tráfego. A existência de transportes públicos também não resolve o problema, visto que estes disputam o espaço nas vias com os demais veículos. Na literatura, foram encontrados diversos trabalhos que tentam resolver este problema reajustando o tempo semafórico das interseções ou, de maneira isolada, definindo rotas para um veículo. Neste projeto é apresentada uma nova maneira de abordar o problema, por meio do desenvolvimento de um sistema que otimiza o fluxo do tráfego urbano, em tempo real, ao explorar as possíveis rotas para cada veículo trafegando em uma determinada área urbana, utilizando análises em rede de Petri, e, então, a partir de um Algoritmo Genético, definir as rotas que diminuam o tempo e/ou tamanho das filas formadas nestas vias. Ainda neste trabalho, diversos testes são efetuados sobre o sistema desenvolvido, comprovando sua eficiência em relação às técnicas existentes.

**Palavras-chave:** Controle de Tráfego, Sistema de Transporte Inteligente, Redes de Petri, Algoritmo Genético.

## **ABSTRACT**

*Urban traffic control is an important research topic in the present days. In fact, everyday, the amount of vehicles traveling on urban roads increases. Moreover, such roads are not always modified for traffic improvement. Also, the availability of public transportation does not solve the problem, since they compete for space on the roads with other vehicles. Several papers have been found in the literature that attempt to solve this problem by either readjusting the traffic light time in intersections or isolately defining routes for a vehicle . This thesis presents a new way to approach the problem through the development of a system that optimizes the urban traffic stream, in real time, exploring possible routes for each vehicle traveling on a given urban area using Petri net analysis, and then, with a Genetic Algorithm, defines routes which reduce the time and/or size of the queues formed in such lanes. Also in this work, several tests are performed on the developed system, proving its efficiency with respect to existing techniques.*

**Keywords:** *Traffic Control, Intelligent Transportation Systems, Petri nets, Genetic Algorithm.*

## LISTA DE FIGURAS

Figura 1. Frota de automóveis entre janeiro de 2009 e julho de 2012. Fonte: DETRAN .....	1
Figura 2. Tempo de arranque dos veículos parados .....	7
Figura 3. Aplicação do Google Traffic.....	9
Figura 4. Exemplo de uma rede de Petri (a) transição habilitada (b) após disparo da transição...	11
Figura 5. Uma rede de Petri representando atividades sequenciais e paralelas.....	12
Figura 6. Processo de recombinação .....	16
Figura 7. Processo de mutação .....	16
Figura 8. Estrutura do tráfego urbano.....	18
Figura 9. Grafo com os vértices (vias) e arestas (caminhos permitidos) controlados pelo sistema .....	19
Figura 10. Modelo em rede de Petri do grafo apresentado na Figura 9 .....	20
Figura 11. Modelo em rede de Petri T-Temporizada dos semáforos de uma interseção .....	21
Figura 12. Árvore de alcançabilidade gerada com base na rede de Petri apresentada na Figura 10. ....	23
Figura 13. Matriz de incidência reversa .....	24
Figura 14. Matriz de incidência direta.....	25
Figura 15. Matriz característica.....	26
Figura 16. População do Algoritmo Genético.....	27
Figura 17. Processo de recombinação .....	28
Figura 18. Lugares percorridos pela marca no decorrer do tempo usando $R_D$ .....	30
Figura 19. Lugares percorridos pela marca no decorrer do tempo usando $R_I$ .....	31
Figura 20. Quantidade de marcas nos lugares no decorrer do tempo usando a rota $R_D$ .....	32
Figura 21. Quantidade de marcas nos lugares no decorrer do tempo usando as combinações de rotas. ....	33
Figura 22. Mapa das vias urbanas controladas pelo sistema .....	34
Figura 23. Modelo da rede de Petri utilizada como modelo das vias urbanas controladas .....	35
Figura 24. Quantidade de veículos nos lugares no decorrer do tempo usando a rota fornecida pelo algoritmo de menor caminho de Dijkstra .....	36
Figura 25. Quantidade de veículos (4 rotas) no decorrer do tempo usando Algoritmo Genético .	37



Figura 26. Quantidade de veículos (2 rotas) no decorrer do tempo usando Algoritmo Genético .	38
Figura 27. Quantidade de veículos no decorrer do tempo utilizando algoritmo de menor caminho de Dijkstra.....	39
Figura 28. Quantidade de veículos no decorrer do tempo utilizando algoritmo de menor caminho de Algoritmo Genético .....	40

**LISTA DE ABREVIATURAS E SIGLAS**

AG	Algoritmo Genético
ATCS	Adaptive Traffic Control System
CPN	Coloured Petri Net
CPN ML	Coloured Petri Net ML
DETRAN	Departamento Estadual de Trânsito
SCOOT	Split Cycle and Offset Optimization Technique
SML	Standard ML (Meta Language)
TRL	Transport Research Laboratory
UTCS	Urban Traffic Control System
V2V	Vehicle to Vehicle
V2I	Vehicle to Infrastructure
ITS	Intelligent Transportation System
ATIS	Advanced Traveler Information System

## SUMÁRIO

<b>Lista de figuras .....</b>	<b>ix</b>
<b>Lista de abreviaturas e siglas .....</b>	<b>xi</b>
<b>1 Introdução.....</b>	<b>1</b>
1.1 Objetivos .....	3
1.2 Justificativa .....	3
<b>2 Fundamentação Teórica .....</b>	<b>4</b>
2.1 Engenharia de Tráfego Urbano .....	4
2.2 Redes de Petri.....	10
2.3 Algoritmos Genéticos .....	15
<b>3 Desenvolvimento .....</b>	<b>18</b>
3.1 Modelo em rede de Petri do Tráfego Urbano .....	19
3.2 Análise das redes de Petri .....	22
3.2.1 Árvore de Alcançabilidade.....	22
3.2.2 Matrizes de Incidência e Equações de Estados.....	24
3.3 Algoritmo Genético .....	26
<b>4 Resultados e Discussões .....</b>	<b>29</b>
4.1 Rede de Petri com 19 lugares.....	29
4.1.1 Entrada de apenas de um veículo .....	30
4.1.2 Entrada uniforme de veículos.....	31
4.2 Rede de Petri com 50 lugares.....	33
4.2.1 Entrada Uniforme de Veículos .....	35
4.2.2 Entrada Uniforme de Veículos em Vias diferentes .....	39
<b>5 Conclusões.....</b>	<b>42</b>
<b>Referências.....</b>	<b>44</b>

# 1 INTRODUÇÃO

Todos os dias, as pessoas deparam-se com os problemas causados pelo aumento de veículos trafegando nas vias urbanas. Esses problemas estão diretamente ligados à qualidade de vida das pessoas e à preservação do meio ambiente, já que o aumento de veículos gera congestionamentos, e, quanto mais veículos existirem e por quanto mais tempo estiverem ligados, maior será a emissão de poluentes na atmosfera e mais tempo as pessoas aguardarão pelo destino dentro dos veículos.

Na Figura 1, é mostrado um gráfico gerado com as informações disponibilizadas pelo Departamento Estadual de Trânsito de São Paulo (DETRAN-SP) sobre a frota de automóveis na cidade de São Paulo no período de janeiro de 2009 a julho de 2012 [DETRAN, 2012]. A partir deste gráfico verifica-se que, a cada mês, cresce a quantidade de automóveis nessa cidade.

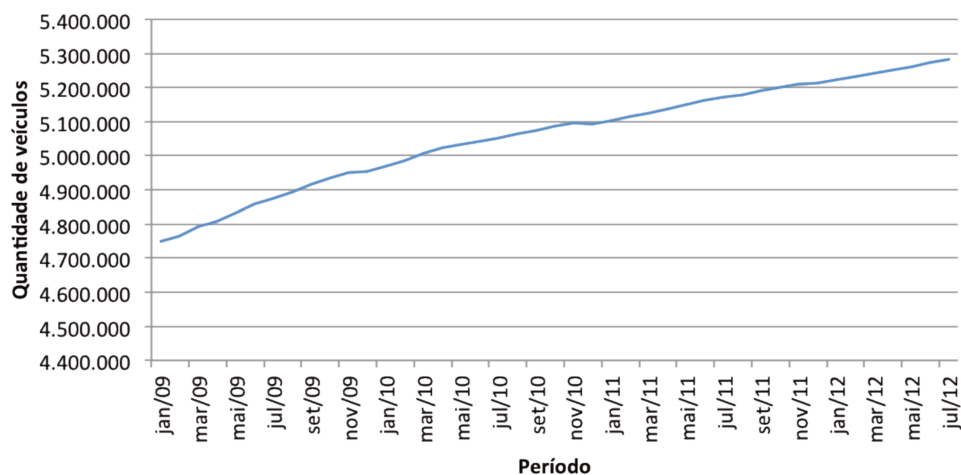


Figura 1. Frota de automóveis entre janeiro de 2009 e julho de 2012. Fonte: DETRAN

Mesmo com o aumento do número de automóveis trafegando nas vias urbanas, estas vias não são modificadas de maneira a suportar a crescente quantidade de veículos, seja por questões de falta de estrutura para modificá-las, como espaço físico, seja por ausência de iniciativas dos órgãos competentes. Independentemente das razões, o crescente aumento na quantidade de automóveis induz a criação de controles de tráfego urbano que aproveitem a estrutura existente, para agilizar o tráfego dos veículos nas vias urbanas.

Existem transportes coletivos que colaboram para a diminuição da quantidade de automóveis nessas vias, porém, em alguns casos, esses serviços são precários ou insuficientes

para atender a demanda de pessoas que precisam trafegar nas cidades. Uma possível solução para esse problema seria um maior investimento nos transportes públicos, garantindo confiabilidade do serviço. Ainda assim, não há como controlar os carros trafegando nas vias, o que pode ocasionar para os transportes públicos urbanos, tais como ônibus, um atraso ao perfazer seu trajeto.

Dado esse problema, é observada a necessidade da criação de sistemas de controle de tráfego urbano que visem auxiliar os motoristas, a fim de amenizar esta situação. A indicação da melhor rota para um motorista pode ser facilmente executada a partir de sistemas de navegação embutidos nos veículos, os quais informam sua posição atual e o trajeto de menor distância, ou menor tempo, até o seu destino. Alguns desses sistemas utilizam informações fornecidas por controladores de tráfego ou sistemas na Internet para evitar algumas vias que possam estar congestionadas. Entretanto, tais sistemas definem as rotas baseando-se apenas na decisão de um motorista e, mesmo evitando locais já congestionados, podem, ao fornecer rotas alternativas sem considerar o caminho dos demais motoristas, congestionar outras vias.

Para que o sistema possa ser colaborativo, de maneira que os veículos possam saber a rota dos demais veículos trafegando nas vias para, então, definir rotas ótimas para todos, é necessário que as informações de localização atual e destino pretendido desses veículos sejam fornecidas para um sistema central. A troca de informações entre veículos e infraestrutura (*Vehicle-To-Infrastructure – V2I*) tem sido muito utilizada pelos Sistemas Avançados de Informação aos Motoristas (*Advanced Traveler Information Systems – ATIS*), os quais se utilizam de tecnologias emergentes, tais como Internet, dispositivos móveis e comunicação sem fio, para definir tais trajetos com base em informações obtidas do ambiente. Os ATIS fazem parte dos Sistemas de Transporte Inteligente (*Intelligent Transportation Systems – ITS*), cujo foco é a sustentabilidade e a integração de sistemas já implantados e novos sistemas para coordenação dos sistemas de transporte de maneira segura e eficiente, realizando monitoramento, controle e provisionamento de informações relevantes ao tráfego.

Em relação à estrutura do tráfego urbano, considera-se que ela é dirigida a eventos e assíncrona e sua evolução no tempo depende da interação de diversos eventos discretos, tais como a entrada e a saída de veículos nas interseções e a completa execução do ciclo dos semáforos. Considerando então o tráfego urbano como um sistema de eventos discretos, o uso

das redes de Petri torna-se conveniente por suas características assíncronas e paralelas e seu conjunto de técnicas que permite analisar o comportamento da rede.

Uma vez que a entrada e a saída de veículos nas áreas de tráfego urbano são dinâmicas em relação ao tempo e que os motoristas precisam obter informações rápidas para tomarem suas decisões sobre a trajetória, justifica-se a utilização dos Algoritmos Genéticos, os quais se utilizam do conceito de esquemas e de suas interações probabilísticas, baseadas nas terminologias da biologia genética, para encontrar soluções satisfatórias dentro de um espaço de soluções definido *a priori* e em um tempo hábil.

## 1.1 OBJETIVOS

Tem-se por objetivo neste projeto a otimização do fluxo do tráfego urbano com base nas informações obtidas de recurso já existente, tal como o tempo de verde dos semáforos, e nas informações obtidas dos veículos, por meio da comunicação entre eles e a infraestrutura, tais como suas localizações atuais e seus destinos pretendidos.

No âmbito específico, tem-se por objetivo desenvolver um modelo de representação do tráfego urbano, utilizando as redes de Petri de Alto Nível, e um sistema de tempo real, utilizando o Algoritmo Genético que, por sua vez, utilizará as análises da rede de Petri de Alto Nível para indicar quais as melhores rotas para todos os veículos em percurso dentro de uma área de tráfego urbana.

## 1.2 JUSTIFICATIVA

Diante dos problemas atuais, como o tempo excessivo de espera em filas de veículos, os gastos consideráveis com combustíveis, os riscos durante a espera em algumas vias inseguras e outros fatores que prejudicam a circulação de pessoas dentro das cidades, surge a necessidade de criar sistemas que diminuam o tempo global que os veículos gastam para percorrer seus caminhos, não somente com base no histórico do fluxo, mas também em informações obtidas em tempo real.

## 2 FUNDAMENTAÇÃO TEÓRICA

Para a elaboração do trabalho proposto, fez-se necessário o estudo de três tópicos fundamentais. O primeiro tópico teve foco na engenharia do tráfego urbano, para que fosse possível o entendimento e a análise dos problemas atuais, permitindo assim sua solução. Para isso, pesquisaram-se os conceitos fundamentais e os projetos relacionados à área. Obtido o conhecimento sobre o tráfego urbano, fez-se necessária a definição de um modelo para descrição e análise do tráfego. Nesse ponto, utilizaram-se as redes de Petri de alto nível, as quais são muito empregadas na modelagem de sistemas paralelos. Por fim, o estudo dos Algoritmos Genéticos fez-se necessário para que uma proposta de melhoria do fluxo do tráfego urbano em tempo real fosse elaborada. Esses assuntos são abordados nas três seções subsequentes.

### 2.1 ENGENHARIA DE TRÁFEGO URBANO

O fluxo de tráfego é composto por motoristas e veículos que interagem uns com os outros, com os elementos físicos das vias urbanas e com o ambiente em geral. Visto que o comportamento dos motoristas e as características dos veículos variam, os veículos nem sempre seguem o fluxo exatamente da mesma maneira. Além disso, dois fluxos de tráfego nas mesmas circunstâncias não se comportarão da mesma maneira, já que o comportamento dos motoristas varia de acordo com o local e com o hábito.

As instalações do tráfego são separadas em duas categorias: fluxo contínuo e fluxo interrompido [Roess, 2011]. As instalações de fluxo contínuo como, por exemplo, as rodovias, não possuem interrupções no fluxo do tráfego, e, portanto, o comportamento do fluxo do tráfego é baseado apenas na interação entre veículos, estrada e ambiente em geral.

Já nas instalações de fluxo interrompido, são incorporadas interrupções externas ao seu projeto e operação. A mais significativa e operacional interrupção externa é o semáforo, que, de maneira alternada, inicia e para um determinado fluxo do tráfego, criando assim uma fila de veículos progredindo por uma via.

A principal diferença entre fluxo contínuo e interrompido é o tempo. Enquanto nas instalações de fluxo contínuo as vias estão sempre disponíveis para os veículos, nas instalações

de fluxo interrompido, o movimento dos veículos é periodicamente interrompido pela indicação vermelha do semáforo, tornando o controle desta última instalação mais complexo que o da primeira.

De acordo com Roess (2011), a unidade fundamental no projeto e na temporização do semáforo é o ciclo, o qual consiste em uma rotação completa por todas as indicações providas. O tempo, em segundos, para efetuar essa rotação completa é chamado de duração do ciclo. Em geral, os veículos recebem a indicação da cor verde para se movimentarem uma vez a cada ciclo.

O intervalo é um período de tempo no qual não há alterações na indicação do semáforo. O intervalo representa a menor unidade de tempo descrita no ciclo do semáforo e é composto pelos tipos:

- Intervalo de mudança: É representado pela indicação amarela do semáforo e corresponde à transição da indicação verde para a indicação vermelha. Este intervalo permite a um veículo, que não pode parar no final do intervalo verde, atravessar com segurança a interseção.
- Intervalo de folga: Este intervalo também faz parte da transição do intervalo verde para o intervalo vermelho, entretanto, durante este intervalo todos os semáforos participantes da interseção possuem a indicação vermelha (“*all red*”).
- Intervalo verde: Cada movimento recebe um intervalo verde durante o ciclo do semáforo. Durante este intervalo, os veículos possuem permissão para se movimentarem.
- Intervalo vermelho: Cada movimento também tem um intervalo com a indicação vermelha do semáforo durante o ciclo. Neste intervalo não é permitida a movimentação dos veículos na interseção.

Existem três tipos de operação do semáforo numa determinada interseção: operação com tempo predeterminado, operação semiatuada e operação totalmente atuada. Nas operações com tempo predeterminado, a duração do ciclo, a sequência da fase e o tempo de cada intervalo são constantes, isto é, cada ciclo do semáforo segue o mesmo plano predeterminado. Já, nas operações semiatuadas são utilizados detectores nas interseções para contar a quantidade de veículos nas vias menos movimentadas e se, e somente se, houver veículos nessas vias ocorre a mudança para o intervalo verde; caso contrário, sempre a via mais movimentada terá o intervalo verde. Por fim, o último tipo de operação é o totalmente atuado, no qual cada via é monitorada a



partir de detectores, e os intervalos verdes são alocados de acordo com as regras definidas utilizando-se as informações capturadas [Roess, 2011].

Um dos programas de temporização de semáforos muito utilizado é o TRANSYT, desenvolvido originalmente pelo Transport Research Laboratory (TRL) da Inglaterra, em 1968. O TRANSYT, atualmente na versão 14.0.5, é um programa de computador *off-line* para projeto, modelagem e estudo de interseções isoladas ou conectadas na rede do tráfego urbano. Esse programa gera o tempo dos intervalos dos semáforos que operam com tempo predeterminado usando técnicas de otimização tais como *Hill Climbing* e *Simulated Annealing* e tendo como entrada o histórico do fluxo das vias [Wong, 2002], [Transport Research Laboratory, 2011].

Entre as tecnologias utilizadas no tipo de operação de semáforos totalmente atuados há os sistemas de controle de tráfego adaptativos (*Adaptive Traffic Control Systems – ATCS*), nos quais o tempo dos intervalos dos semáforos em interseções individuais é modificado em tempo real, com base na detecção de dados sobre o fluxo [Roess, 2011].

O SCOOT (*Split Cycle and Offset Optimization Technique*), também desenvolvido pelo Transport Research Laboratory da Inglaterra, consiste em um ATCS. Este programa monitora, em tempo real, o fluxo do tráfego e, utilizando sistema similar ao TRANSYT para otimizar o fluxo do tráfego, realiza pequenos ajustes no tempo do semáforo, de maneira a diminuir o tempo de espera dos veículos dentro da rede [Robertson, 1991]. Atualmente o SCOOT está operando em 200 cidades ao redor do mundo, inclusive em São Paulo, na interseção da Avenida Rio Branco com a Avenida Norma Gianotti [SCOOT, 2011].

Na literatura, existem trabalhos sobre o controle dos semáforos com base no tempo gasto pelos veículos durante seu percurso. No trabalho de Balan e Luke (2006) é descrito um sistema no qual cada veículo possui créditos durante o percurso. À medida que o veículo permanece parado no intervalo vermelho, ele acumula créditos e, ao longo do caminho, quando passa pelo intervalo verde, seus créditos são retirados. Dessa maneira os semáforos verificam quais vias possuem mais créditos e definem o intervalo verde para estas. Com isso, adquire-se uma igualdade no tempo percorrido durante os percursos semelhantes.

Em outros trabalhos da área, foram estudadas formas de controlar o tráfego urbano tendo veículos autônomos nas vias. Esses trabalhos apontam ideias de como organizar o tráfego, porém ainda não são aplicáveis hoje, visto que é necessária a criação de veículos capazes de se locomoverem de forma autônoma. Senda et al. (2008) descrevem a criação do controle do tráfego

sem a utilização de semáforos, no qual cada veículo autônomo possuiria uma forma de detectar os demais veículos no cruzamento e, então, calcular o tempo correto para atravessar tal interseção sem que haja a colisão com outros veículos. Quando houvesse um impasse, o veículo com maior prioridade teria a preferência.

Ghali e Smith (1995) entendem que o controle simples do tráfego urbano em algumas vias não é suficiente para evitar os congestionamentos, visto que a quantidade de veículos colocados nas ruas cresce a cada dia. Uma das sugestões dos autores seria trabalhar também com os hábitos dos motoristas, ou seja, conscientizá-los a percorrerem outros caminhos sempre que possível para evitar os congestionamentos. Se várias pessoas mudarem seus hábitos durante o trajeto, pode ocorrer de não haver congestionamento em algumas vias onde antes havia.

O controle das interseções sinalizadas é complexo e não há modelos únicos para descrever tais interseções. Entretanto, na literatura, existem alguns modelos para representar o tráfego de maneira simples, para que seja possível a aplicação de técnicas de otimização no fluxo do tráfego. Um dos modelos que melhor se aplicam a este problema é a rede de Petri, descrita na seção 2.2.

Em fluxo de tráfego interrompido, tal como o das vias urbanas, onde há semáforos que colaboram para a parada de veículos, o tempo de arranque desses veículos é um fator a ser considerado na otimização, visto que, quanto mais vezes eles pararem, mais vezes terão o tempo de aceleração inicial gasto. O tempo médio de arranque tende a se tornar constante, conforme é apresentado na Figura 2, visto que o primeiro motorista consome um tempo maior para verificar a abertura do semáforo e atuar na aceleração do que o segundo motorista.

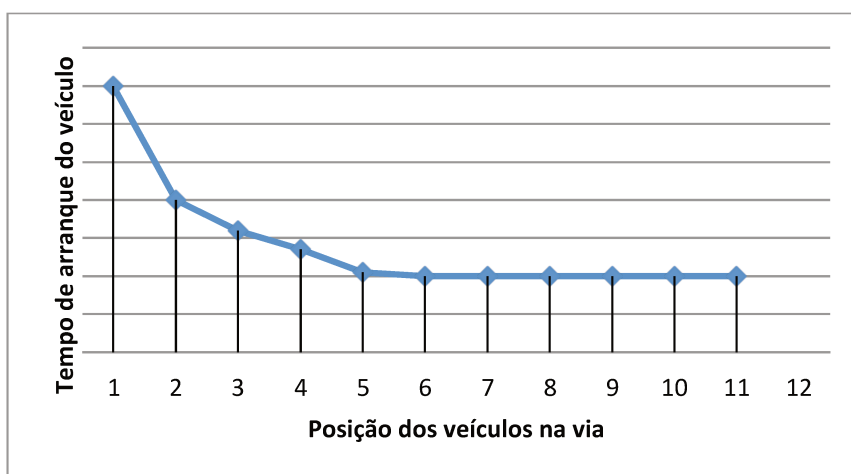


Figura 2. Tempo de arranque dos veículos parados

A queda no tempo de arranque tende a ocorrer até o quarto ou quinto veículo e, em condições normais, se torna constante. Esse valor constante é chamado de saturação de arranque, representado pela letra  $h$ , e possui um valor conhecido na literatura de 2,4 segundos por veículo [Roess, 2011].

O tempo inicial perdido é o tempo adicional gasto no arranque dos veículos e consiste no tempo utilizado pelos motoristas para reagir ao intervalo verde do semáforo e acelerar seus veículos. Esse tempo adicional é calculado por meio da Equação 1, onde o  $\Delta_i$  representa o tempo adicional gasto para cada arranque  $i$ .

Equação 1. Cálculo do tempo inicial perdido

$$l_1 = \sum_i \Delta_i$$

Uma vez definidos os valores do tempo de arranque e do tempo inicial perdido, é possível calcular o tempo de intervalo verde necessário para despachar uma fila com  $n$  veículos utilizando a Equação 2,

Equação 2. Cálculo do tempo de verde necessário para despachar uma fila de veículos

$$T_N = l_1 + n \cdot h$$

onde  $T_n$  é o tempo requerido para mover uma fila com  $n$  veículos através da interseção,  $l_1$  é o tempo inicial perdido,  $n$  é a quantidade de veículos na fila e  $h$  é o tempo de saturação de arranque.

Com base na Equação 1 e na Equação 2, é possível verificar que, quanto maior for a fila de veículos, ou quanto mais vezes um veículo parar, maior será o tempo gasto para despachá-los e, portanto, a otimização do fluxo do tráfego urbano pode ser feita evitando, ou diminuindo, essas situações.

De maneira a melhorar o fluxo dos veículos e, conseqüentemente, a vida dos motoristas e passageiros, e com o advento de novas tecnologias, surgiram os Sistemas de Transporte Inteligente, cujo principal objetivo é, segundo Florea (2007) e Ni (2007), a integração dos motoristas, veículos e sistemas de transporte, com o propósito de contribuir para a segurança no tráfego, criando um ambiente com menos poluições e um sistema de transporte mais eficiente, explorando a infraestrutura existente. Tal proposta se torna viável pela telemática aplicada ao transporte, que integra e implementa tecnologias de telecomunicação e informática no âmbito da

indústria de transporte, tais como comunicação entre veículos e infraestrutura por meio de dispositivos móveis, amplamente utilizados pela sociedade em geral. Um exemplo dessa integração é dada pela aplicação Google Traffic [Google, 2012], a qual se utiliza das informações de localização enviadas pelos dispositivos móveis dos motoristas que utilizam o Google Maps, para mostrar em seus mapas a situação do trânsito, em tempo real, conforme pode ser visto na Figura 3. A comunicação entre os veículos e uma infraestrutura (V2I) permite que os veículos transmitam suas informações, tais como posição, velocidade e trajetória para um servidor central que realize as manipulação e análises em tais dados [Miller, 2008].



Figura 3. Aplicação do Google Traffic

A aplicação do Google Traffic pode ser considerada como um Sistema Avançado de Informação ao Motorista (*Advanced Traveler Information System – ATIS*), o qual tem por objetivo a integração de várias tecnologias, tais como Internet, telefones celulares e comunicação via rádio, para auxiliar os motoristas na tomada de decisões sobre suas trajetórias. A informação

provida por estes sistemas consiste, geralmente, de rotas ou notificações de incidente ou situação do fluxo do tráfego, e pode ser acessada antes ou durante as viagens [Kumar, 2005].

## 2.2 REDES DE PETRI

As redes de Petri foram criadas por Carl Adam Petri em 1962, em sua tese de doutorado, intitulada “Comunicação com autômatos” (em alemão, *Kommunikation mit Automaten*), a qual foi defendida na Universidade de Darmstadt, na Alemanha. Essas redes foram desenvolvidas com o intuito de descrever e estudar, gráfica ou matematicamente, diversos sistemas de processamento de informações que possuam características concorrentes, assíncronas, distribuídas e paralelas [Murata, 1989].

As redes de Petri têm sido empregadas em uma grande variedade de aplicações devido à sua generalidade. Elas podem ser aplicadas em qualquer área que possa ser descrita com grafos de fluxo e que precise de representações de atividades paralelas ou concorrentes. Algumas áreas que utilizam as redes de Petri são: análise de protocolos de comunicação, sistemas distribuídos, programas concorrentes e paralelos, sistemas de controle industrial e de fabricação, sistemas de evento discreto, sistemas operacionais e compiladores, sistemas de informação empresariais, linguagens formais, lógicas de programação e modelos de decisão [Murata, 1989]. Com foco em nosso projeto, há trabalhos relevantes que utilizam redes de Petri para análise e controle do tráfego urbano, tais como [Di Febbraro et al, 2004], [Tzes et al, 1996] e [Dotoli e Fanti, 2006], os quais são descritos com mais detalhes no final desta seção.

As redes de Petri podem ser definidas tanto com formalismo matemático quanto graficamente. Matematicamente, a rede de Petri pode ser descrita como uma quintupla de elementos  $PN = (P, T, F, W, M_0)$ , onde:

$P$  são os lugares da rede;

$T$  são as transições da rede;

$F \subseteq (P \times T) \cup (T \times P)$  são os arcos da rede;

$W: F \rightarrow \mathbb{N}^+$  são os pesos dos arcos e

$M_0$  é a marcação inicial.

Graficamente, conforme mostrada na Figura 4, a rede de Petri contém os elementos lugar, transição, arcos e marcas. Os lugares podem ser interpretados como estados da rede, informando alguma condição, e são representados por um círculo. As transições representam as ações da rede e são representadas por um retângulo preenchido. Os arcos são utilizados para conectar os lugares às transições. As marcas adicionadas nos lugares são utilizadas para indicar se uma condição associada ao lugar é verdadeira.

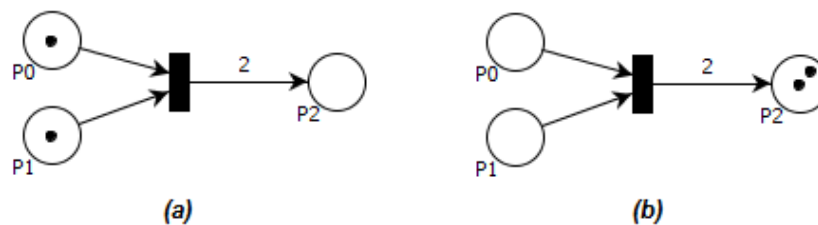


Figura 4. Exemplo de uma rede de Petri (a) transição habilitada (b) após disparo da transição

Os arcos não podem ter nas extremidades dois elementos básicos do mesmo tipo, como, por exemplo, um lugar ligado a outro. Além disso, os arcos podem conter pesos, os quais informam a quantidade de marcas que devem ser retiradas dos lugares de entrada das transições ou a quantidade de marcas que devem ser criadas nos lugares de saída das transições.

Para que uma transição esteja habilitada a disparar, é necessário que todos os lugares de entrada (também chamados de pré-conjunto) possuam a quantidade de marcas necessárias, a qual é definida por um inteiro positivo e informada nos arcos que vão dos lugares para a transição. Quando a transição está habilitada e é disparada, as marcas dos lugares de entrada são removidas, de acordo com o peso dos arcos que ligam os lugares nas transições, e são criadas marcas nos lugares de saída (também chamados de pós-conjunto), de acordo com o peso dos arcos que ligam a transição aos lugares. Uma interpretação pode ser dada pela indicação de recursos ou dados disponíveis para processamento nos lugares de entrada que, quando processados, ou seja, quando ocorre o disparo da transição, fornece recursos ou dados para os lugares de saída. Na Figura 4 (a), a transição está habilitada a ser disparada, pois P0 e P1 possuem pelo menos uma marca, e o peso dos arcos dos lugares de entrada (P0 e P1) à transição, quando ocultos, equivale a um. Quando habilitada, ao disparar a transição, conforme mostrado na Figura 4 (b), uma marca do lugar P0 e uma marca do lugar P1 são removidas e duas marcas no lugar P2 são criadas, de acordo com o peso definido no arco da transição ao lugar P2.

A característica importante da rede de Petri é a modelagem de atividades sequenciais e paralelas. Na rede de Petri mostrada na Figura 5 pode-se definir uma sequência de disparos de transições, a qual é dita como sequência, pois T3 só estará habilitada após o disparo da transição T2. Já as atividades paralelas podem ocorrer, por exemplo, nas transições T2 e T4, as quais estão habilitadas e podem ser disparadas ao mesmo tempo. Essas características são importantes na modelagem de sistemas de controle de tráfego, visto que há atividades sequenciais, por exemplo, a ordem em que as luzes dos semáforos são acessas, e atividades paralelas, por exemplo, dois semáforos funcionando em paralelo em cruzamentos próximos.

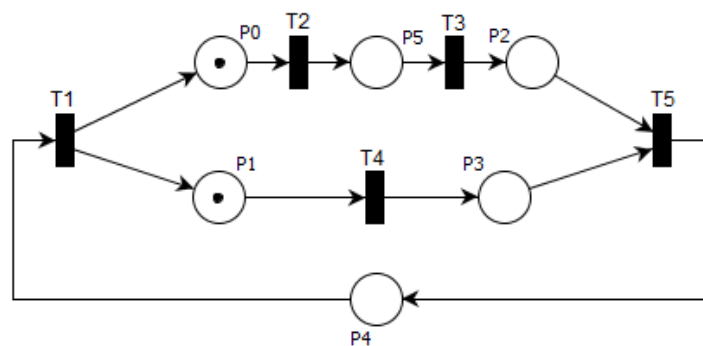


Figura 5. Uma rede de Petri representando atividades sequenciais e paralelas

O conceito de tempo, essencial para o desenvolvimento deste projeto, não foi explicitamente dado na definição original das redes de Petri. Entretanto, para os problemas de avaliação de desempenho e programação de sistemas dinâmicos, é necessário e útil introduzir o tempo de espera associado às transições e/ou marcas no modelo [Murata, 1989].

Para as redes de Petri com tempo discreto, o tempo é associado ou aos lugares ou às transições. É comum associar o tempo aos lugares quando se deseja manter um estado durante um período. Por outro lado, é comum associar o tempo às transições quando se deseja definir um tempo de espera para a mudança de estado da rede [David, 2001].

Neste projeto, são associados tempos discretos às transições que realizam a mudança dos intervalos do semáforo. Por exemplo, ao ser inserida uma marca no lugar que representa o intervalo verde do semáforo, esta marca só será consumida pela transição que realiza a mudança do intervalo verde para o intervalo amarelo após o tempo definido nesta transição, ou seja, caso o tempo seja igual a 18 segundos, a marca permanecerá 18 segundos no lugar que representa o intervalo verde do semáforo. Neste projeto, são associados ainda tempos discretos às marcas, representando o tempo gasto pelos veículos para percorrer uma via.

A rede de Petri Colorida (*Coloured Petri Net* – CPN), considerada uma rede de Petri de Alto Nível, é uma linguagem gráfica para a construção de modelos de sistemas concorrentes e de evento discreto que combina as características da rede de Petri original, a qual fornece os fundamentos da notação gráfica e matemática para modelar concorrência, sincronismo e comunicação, com a capacidade da linguagem de programação de alto nível, chamada CPN ML (*CPN Meta Language*), a qual é baseada na linguagem de programação funcional Standard ML [Jensen, 2009].

A utilização das redes de Petri Coloridas permite a criação de modelos complexos, mas menores, visto que é possível associar tipos de dados às marcas e funções às transições e arcos. A utilização da rede de Petri Colorida é pertinente neste projeto pois, com ela, podem-se diferenciar os veículos, representados por marcas, adicionando estruturas que definem as rotas e o tempo de percurso.

Uma vez criado um modelo em rede de Petri, é possível aplicar diversas análises no modelo a fim de identificar, por exemplo, alcançabilidade das marcações, conflitos entre transições e *deadlocks*.

A análise por enumeração dos estados é talvez a técnica mais comum para análise das redes de Petri. Ela é baseada na construção do espaço de estados resultante da exaustiva execução da rede, ou seja, da sua árvore de alcançabilidade.

A árvore de alcançabilidade de uma rede de Petri consiste de um grafo direcionado e etiquetado. Cada um dos seus vértices é uma marcação alcançável da rede e cada um dos seus arcos representa a execução da transição para atingir a marcação pretendida. A árvore de alcançabilidade é dada então por um par  $RT = (V, E)$ , no qual  $V$  é um conjunto de todos os vértices distintos, e etiquetados com uma indicação da marcação correspondente, e  $E$  é um conjunto de todos os arcos, etiquetados com o nome da correspondente transição, e que identificam a mudança entre duas marcações alcançáveis da rede.

Uma vez que a árvore esteja construída, é possível determinar todas as propriedades da rede de Petri considerada, levando em conta a marcação inicial usada para gerar o espaço de estados. A marcação é chamada alcançável numa rede de Petri se existir uma sequência finita de disparo de transições que conduzem a ela a partir da marcação inicial.

A geração do espaço de estados é o primeiro passo da arquitetura de verificação do modelo. Como no mundo real as aplicações lidam usualmente com um grafo de espaço de estado



muito completo, tendo milhões de estados, este processo requer extensivos recursos computacionais, com longo tempo de processamento e alto consumo de memória [Moutinho, 2011].

Existem alguns trabalhos na área de modelagem e otimização do tráfego urbano utilizando as redes de Petri. Em Tzes et al. (1996) é apresentado um modelo em rede de Petri da rede de transporte urbana, no qual esta é modelada como um sistema a eventos discretos, utilizando as propriedades de paralelismo e sincronização presentes nas redes de Petri.

No trabalho de Di Febbraro (2004), uma rede urbana de intersecções sinalizadas é modelada com uma rede de Petri Híbrida, na qual o comportamento dos veículos é descrito por meio de um modelo orientado por tempo e os semáforos são representados dinamicamente por um modelo a eventos discretos. A partir desse modelo, o problema de coordenar vários semáforos de maneira a otimizar o tempo percorrido por alguns tipos especiais de veículos, tais como veículos públicos e de emergência, é resolvido a partir da alteração dinâmica do tempo de disparo das transições a partir do percurso dado.

No trabalho de Vásquez et al. (2010), é apresentado um modelo de uma única interseção utilizando redes de Petri Híbridas para que possa ser analisado o comportamento dinâmico do tráfego a partir da obtenção do tempo de espera dos veículos nas filas, que são representadas por lugares com valores contínuos. Nesse modelo, os intervalos do semáforo na interseção são representados por lugares com valores discretos.

Os trabalhos relatados anteriormente utilizam redes de Petri de baixo nível. Em Dotoli e Fanti (2004), a rede de tráfego urbano é modelada utilizando-se uma rede de Petri Colorida em que os lugares representam as ruas e interseções, e as marcas representam os veículos.

Nos trabalhos de Qu (2010) e Liu (2009), são utilizadas redes de Petri T-Temporizada e P-Temporizada, respectivamente, para definir a menor rota em relação ao tempo. No primeiro trabalho, as transições da rede têm um custo associado, o qual é calculado pela análise das imagens tiradas das vias urbanas, e o tempo total de viagem é considerado como a soma desses custos, obtidos por meio de simulação. No segundo trabalho, os lugares da rede de Petri correspondem aos arcos de um grafo e contêm tempos associados, correspondente ao tempo utilizado por um veículo para percorrer uma via, e as transições correspondem aos vértices do grafo. Dessa maneira, a simulação é feita sobre o modelo em rede de Petri, e o tempo que a marca consome para alcançar seu destino final é somado.

Os trabalhos relatados anteriormente utilizam-se apenas da simulação da rede de Petri para atingir seus objetivos. Entretanto, o trabalho proposto neste documento diferencia-se e inova, por utilizar as análises da rede de Petri de alto nível na definição, em um curto espaço de tempo, das possíveis rotas de todos os veículos trafegando na área urbana controlada. Uma vez definidas as rotas, realiza-se a otimização do tempo gasto nos percursos dos veículos de maneira global, definindo o melhor conjunto de rotas. A otimização utiliza-se do Algoritmo Genético, o qual é apresentado na seção 2.3.

## 2.3 ALGORITMOS GENÉTICOS

Os Algoritmos Genéticos (AG) são um ramo dos algoritmos evolucionários, ou seja, são algoritmos baseados numa metáfora do processo biológico de seleção natural, apresentada na teoria de Charles Darwin e também na genética moderna [Goldberg, 1989]. Nesses algoritmos, populações de indivíduos são criadas e submetidas aos operadores genéticos de seleção, recombinação e mutação, a fim de combinar as características dos indivíduos sobreviventes de maneira a retornar uma solução para o problema [Linden, 2008], mantendo um balanço entre eficiência e a eficácia necessárias para solucionar problemas em diversas áreas.

A primeira etapa do Algoritmo Genético tem a função de definir uma população inicial, a qual deve contemplar diversos pontos no espaço de soluções, diferentemente de outros métodos de busca e otimização, que trabalham sobre um único ponto. Cada indivíduo da população é estruturado como um vetor, chamado de cromossomo, no qual cada posição representa um locus contendo um gene, cujo valor pode ser predeterminado (alelos).

No Algoritmo Genético clássico, trabalha-se com a geração da população (mecanismo de seleção) usando-se uma forma probabilística, na qual, a partir do seu *fitness*, um indivíduo tem maior probabilidade de ser selecionado. O *fitness* representa a função objetivo que se deseja otimizar. Um dos métodos utilizados para a seleção de um indivíduo na população é o método de seleção da Roleta. Neste método, a probabilidade de selecionar um cromossomo (indivíduo da população), é diretamente proporcional ao seu valor de *fitness*. Cada cromossomo  $x_i$ ,  $i = 1, \dots, N$ , da população é alocado para uma parte da roleta, cujo tamanho ocupado é proporcional ao seu

valor de *fitness*. Assim, seleciona-se um elemento deste conjunto, de maneira aleatória, até que todos os indivíduos de uma população sejam definidos [Castro, 2006].

Após definida uma população inicial, são realizadas várias iterações até que se atinja um valor ótimo com base na função de *fitness*. A cada iteração, os indivíduos da população sofrem recombinação para que possam gerar descendentes. A recombinação em Algoritmo Genético representa a reprodução de novos indivíduos a partir da troca do material genético, conforme mostrado na Figura 6. Desta maneira, uma das soluções é determinar um pedaço do primeiro cromossomo para ser passado ao filho, e preencher os demais genes com parte do segundo cromossomo, sem que os elementos sejam repetidos.

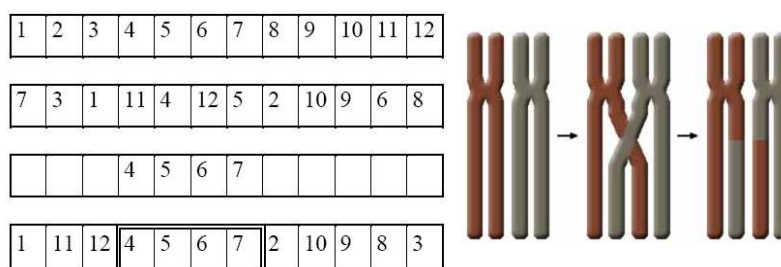


Figura 6. Processo de recombinação

Para que haja uma variação nas soluções, pode-se aplicar o conceito de mutação, segundo o qual um ou mais genes têm seus valores alterados, conforme mostrado na Figura 7 [Castro, 2006].

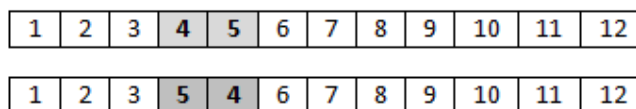


Figura 7. Processo de mutação

Gerada a próxima população, os indivíduos sofrem novamente o processo de seleção natural com base na função de *fitness*. Este algoritmo pode ser executado até que se encontre uma solução ótima, porém tal solução pode ser global ou simplesmente local.

A quantidade de iterações não é conhecida, podendo-se trabalhar com um número predeterminado (10, 100, 1000) ou criar funções que determinam uma quantidade a partir da convergência da população para um único indivíduo, ou seja, quando a população possui em sua maioria todos os indivíduos iguais. A aplicação do algoritmo genético não determina que a

solução será sempre ótima e nem mesmo que, uma vez encontrada a solução ótima, esta será encontrada novamente se o algoritmo for executado outra vez. Este conceito se dá pela estrutura probabilística do algoritmo genético, que, diferentemente de algoritmos determinísticos, como o método de *Hill Climbing*, sua convergência para um resultado ótimo nem sempre pode ser determinado após certo número de iterações.

Portanto, com a utilização do Algoritmo Genético, tendo os parâmetros bem definidos, de maneira a permitir que um conjunto de rotas eficientes fosse incluído na população e se propagasse durante as gerações, obteve-se a melhoria no fluxo de todos os veículos na rede de tráfego urbano a partir da procura no espaço de soluções dado. O tamanho da população e a quantidade de rotas para cada um dos veículos, assim como a quantidade de gerações e o critério de parada são explicados no Capítulo 3, e os resultados obtidos a partir da execução do Algoritmo Genético, comprovando sua eficiência na resolução deste problema, são apresentados no Capítulo 4.

### 3 DESENVOLVIMENTO

De acordo com o texto apresentado na introdução deste trabalho, a atuação dos veículos sobre o fluxo do tráfego urbano deve ser feita em tempo real e, para isso, deve ser elaborado um sistema para definir as possíveis rotas de cada veículo a partir de suas posições atuais, e então selecionar o conjunto de rotas que agilize o fluxo global do sistema de trânsito. Este sistema faz uso das técnicas de análise das redes de Petri, com a obtenção, em tempo real, das informações obtidas do fluxo do tráfego urbano, e do algoritmo genético, como estrutura para definir a melhor solução a partir das análises da rede de Petri obtidas.

Conforme pode ser visto na Figura 8, para cada veículo que começa a trafegar numa área urbana monitorada, o sistema desenvolvido opera na definição de sua rota considerando que esta parta da posição atual até o destino informado pelo motorista.

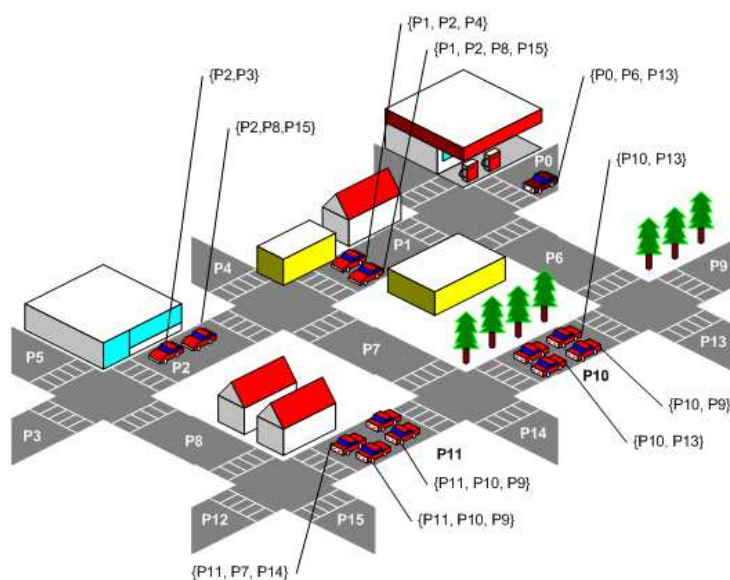


Figura 8. Estrutura do tráfego urbano

Após definir as possíveis rotas para cada veículo, o sistema incorpora as informações reais do fluxo do tráfego, tais como as rotas já definidas pelos demais veículos e os tempos dos semáforos, e define a melhor rota com base no tempo total de viagem.

Para o desenvolvimento deste sistema utilizou-se a linguagem de programação Java [Sierra, 2005], a qual é amplamente usada e difundida na maioria dos dispositivos móveis,

facilitando sua futura implantação. Todas as simulações do sistema foram feitas utilizando os sistemas operacionais Windows, da empresa Microsoft, e Mac OS, da empresa Apple, em um computador com processador Core i5 de 1.8 GHz e 4GB de memória RAM.

### 3.1 MODELO EM REDE DE PETRI DO TRÁFEGO URBANO

A primeira etapa para alcançar o objetivo proposto consiste na modelagem das instalações do tráfego urbano usando uma rede de Petri de alto nível, a qual foi escolhida por permitir modelar o sistema com marcações individuais, as quais contêm informações que são utilizadas para o disparo das transições. Neste trabalho, estas informações consistem no tempo permitido para que a marca possa ser consumida por uma transição e uma lista com os caminhos, lugares, que deverá percorrer na rede. Analogamente, o tempo representa o intervalo utilizado por um veículo para percorrer uma determinada via e o caminho representa sua rota na cidade.

Na Figura 10, é apresentado o modelo em rede de Petri de Alto Nível equivalente ao grafo apresentado na Figura 9. Neste modelo, cada lugar corresponde a uma via urbana e pode conter zero ou um número máximo, limitado à sua capacidade, de marcas, no caso, veículos, e cada arco possui peso unitário.



Figura 9. Grafo com os vértices (vias) e arestas (caminhos permitidos) controlados pelo sistema

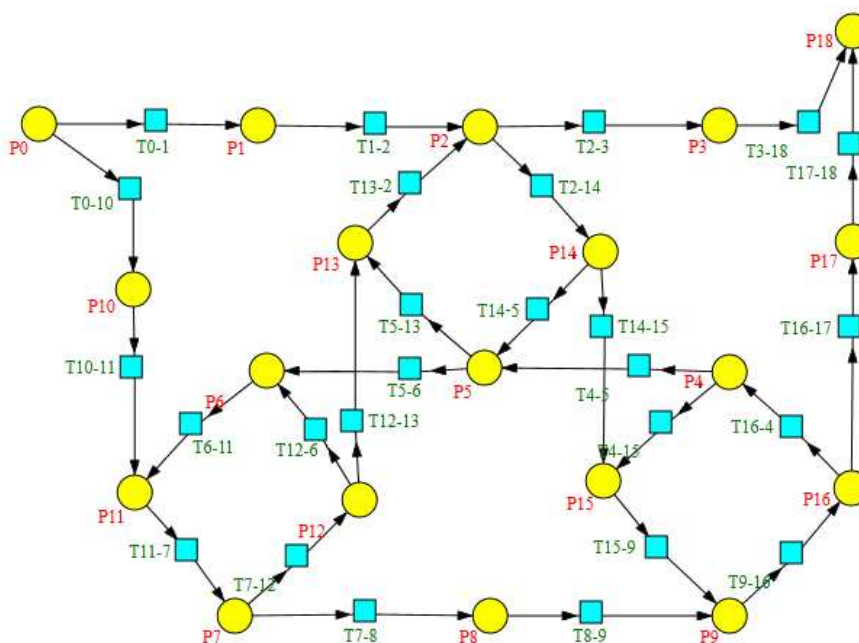


Figura 10. Modelo em rede de Petri do grafo apresentado na Figura 9

Os conflitos gerados com o uso de marcas booleanas são resolvidos quando se incorporam tipos de dados a elas. Por exemplo, tendo uma marca no lugar P0 da rede dada na Figura 10, ambas as transições T0-1 e T0-10 podem ser disparadas; com a inclusão de um vetor indicando quais os lugares que a marca deverá percorrer e uma função de guarda nas transições, a qual é responsável por habilitar a transição de acordo com as informações contidas na marca; quando uma marca tem sua rota para P1, apenas a transição T0-1 será disparada e se a rota é para P10, apenas a transição T0-10 será disparada.

Levando em conta a semântica da execução considerada pelo modelo da rede de Petri de Alto Nível, em que o disparo de uma transição é determinístico dependendo da condição de habilitação associada a ela, há a possibilidade de disparar em paralelo as transições T0-1 e T0-10, mas apenas no caso do lugar P0 conter duas marcas com o mesmo tempo de disparo associado, mas com diferentes rotas. Isso corresponde às vias urbanas, nas quais há duas faixas de veículos. Na Figura 11 é apresentado o modelo da rede de Petri T-Temporizada referente aos semáforos da interseção dada pela entrada em P2, em que ambas as transições T1-2 e T13-2 podem ser disparadas.

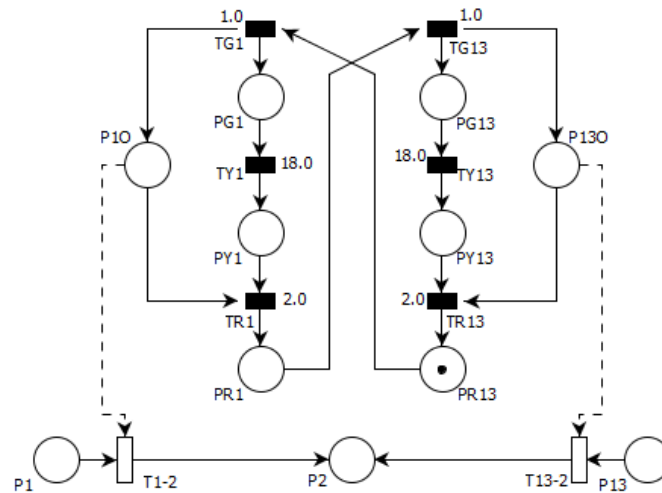


Figura 11. Modelo em rede de Petri T-Temporizada dos semáforos de uma interseção

Em cada interseção na rede apresentada na Figura 10 é adicionado um modelo equivalente ao apresentado na Figura 11 e não são mostrados apenas para efeito de simplificação do modelo. Os lugares PG1, PY1 e PR1 correspondem, respectivamente, aos intervalos verde, amarelo e vermelho do semáforo S1. Já os lugares PG13, PY13 e PR13 correspondem, respectivamente, aos intervalos de verde, amarelo e vermelho do semáforo S13. Quando o semáforo S1 contiver uma marca no lugar PG1 ou PY1, também haverá uma marca no lugar P10, o qual será utilizado para habilitar a transição T1-2. O mesmo ocorre para o semáforo S13 e a transição T13-2. É importante ressaltar que o arco que conecta P10 à transição T1-2 e o arco que conecta o lugar P130 à transição T13-2 são arcos de teste e, portanto, o disparo das transições T1-2 ou T13-2 não remove a marca dos lugares P10 e P130.

As transições TG1 e TG13 são responsáveis pela mudança do intervalo vermelho para o intervalo verde dos semáforos S1 e S13, respectivamente, e têm um tempo de 1 segundo associado para que a transição, quando habilitada, possa disparar. Este intervalo também é parte da transição do intervalo verde para o intervalo vermelho do semáforo oposto que será aberto, mantendo assim, durante este período, ambos os semáforos no intervalo vermelho.

Já as transições TY1 e TY13 são responsáveis pela mudança do intervalo amarelo e somente serão disparadas 18 segundos após estarem habilitadas, mantendo assim as marcas nos lugares de entrada (PG1 ou PG13) disponíveis durante este período. Após serem disparadas, as transições TY1 e TY13 geram uma marca nos lugares PY1 e PY13, respectivamente, as quais indicam que os semáforos estão no intervalo amarelo.



Por fim, as transições TR1 e TR13 são disparadas 2 segundos após estarem habilitadas, consumindo as marcas PY1 e PY13, respectivamente, e criando uma marca nos lugares PR1 e PR13, as quais indicam que os semáforos estão no intervalo vermelho. As transições TR1 e TR13 estarão habilitadas para disparar apenas se houver uma marca no lugar P1O ou P13O e, quando disparadas, removerão essas marcas, desabilitando o disparo das transições T1-2 e T13-2, respectivamente. Este intervalo permite que os veículos que não puderam parar no fim do intervalo verde possam atravessar a interseção em segurança.

## 3.2 ANÁLISE DAS REDES DE PETRI

A definição das possíveis rotas de um veículo, dada sua posição atual, é feita a partir da análise do espaço de estados, conforme exposta na seção 3.2.1. Uma vez definidas as rotas de todos os veículos, o cálculo do tempo utilizado no percurso dos veículos até seus destinos é feito a partir da simulação da rede de Petri, a qual consiste no cálculo das equações de estado da matriz de incidência, conforme será explicado na seção 3.2.2, e que será utilizada como a função de *fitness* do Algoritmo Genético, explicado na seção 3.3.

### 3.2.1 ÁRVORE DE ALCANÇABILIDADE

No primeiro estágio do sistema desenvolvido, todas as possíveis rotas da origem até o destino são definidas a partir da busca pelo espaço de estados gerado com base no modelo da rede de Petri IOPT utilizando a ferramenta IOPT Tool [Pereira, 2011a]. A escolha desta ferramenta se faz por seu desempenho, o qual permite a geração de um grande espaço de estados em pouco tempo, por exemplo, 50 milhões de estados podem ser gerados, dependendo do modelo, em apenas 15 segundos [Moutinho, 2011], [Pereira, 2011b]. Além disso, um modelo de rede de Petri IOPT permite associar condições externas para o disparo das transições, podendo, por exemplo, desabilitar uma interseção em particular a partir de um evento externo se um incidente for notificado no seu lugar de saída. A notificação pode ser feita por um motorista, ou pedestre, para um serviço central, ou de maneira automatizada, conforme descrito no trabalho de Kong (2006),

no qual um algoritmo utilizando lógica nebulosa é aplicado na identificação de incidentes em algumas vias urbanas baseado nas informações do tráfego.

A marcação inicial para a geração do espaço de estados consiste em uma marca no lugar atual onde o veículo se encontra e zero marca nos demais lugares da rede. Sempre que existir um conflito, a análise da rede define duas possíveis marcações, isto é, considerando P0 como o lugar inicial, as possíveis transições habilitadas são T0-1 e T0-10. Desta maneira é possível verificar todas as possíveis rotas utilizando a árvore de alcançabilidade, conforme mostrada na Figura 12.

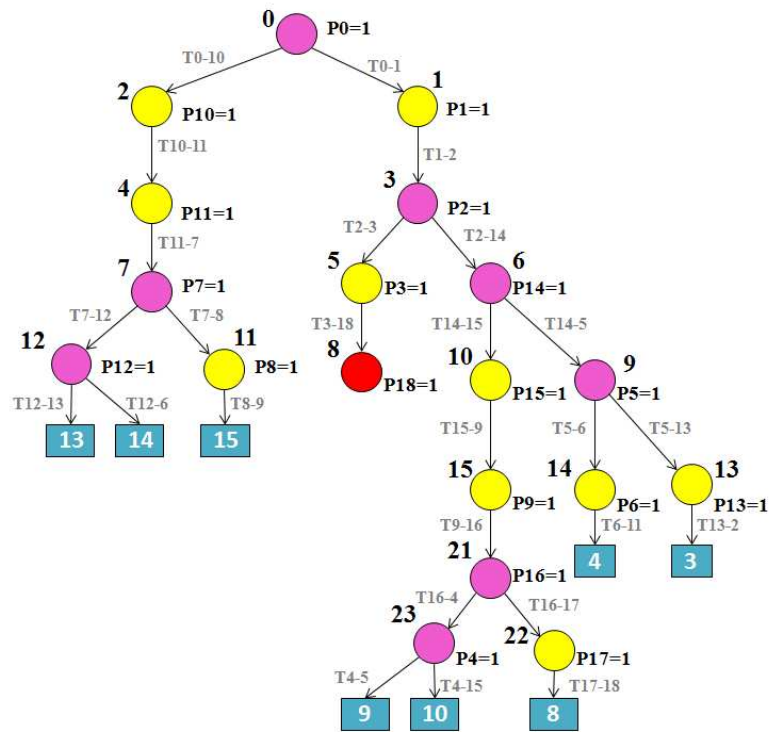


Figura 12. Árvore de alcançabilidade gerada com base na rede de Petri apresentada na Figura 10.

A árvore de alcançabilidade da Figura 12 é montada a partir da execução exaustiva da rede. Um conflito indica que duas ou mais transições estão habilitadas, mas o disparo de uma transição desabilitará as demais. Os conflitos no modelo ocorrem sempre que houver uma marca em um dos lugares: P0, P2, P4, P5, P7, P12, P14 e P16 e são representados na árvore de alcançabilidade pela elipses com cores lilás.

Considerando, por exemplo, que um motorista deseja viajar do lugar P0 até o lugar P18, as possíveis rotas encontradas pela análise do espaço de estados são  $R_1 = \{P0, P1, P2, P3, P18\}$ ,  $R_2 = \{P0, P10, P11, P7, P12, P13, P2, P3, P18\}$ ,  $R_3 = \{P10, P11, P7, P8, P9, P16, P17, P18\}$  e

$R_4 = \{P0, P10, P11, P7, P8, P9, P16, P4, P5, P13, P2, P3, P18\}$ . As outras rotas são ignoradas pelo sistema por conterem laços.

O *deadlock*, representado pelo estado na cor vermelha neste modelo, ocorrerá sempre que houver um marca no lugar P18, visto que não há transições que consumam esta marca. Os retângulos na árvore de alcançabilidade representam um link para um estado já criado e são utilizados apenas para simplificação da árvore.

### 3.2.2 MATRIZES DE INCIDÊNCIA E EQUAÇÕES DE ESTADOS

Uma vez que as rotas de um veículo estejam definidas a partir da análise do espaço de estados, o sistema escolhe a melhor rota entre elas com base no tempo total de viagem calculado de acordo com as equações de estado feitas sobre a matriz de incidência da rede de Petri de alto nível.

As redes de Petri podem ser representadas por duas matrizes, uma indicando o conjunto dos lugares de entrada de cada transição, e outra indicando o conjunto de lugares de saída dessas transições, conforme pode ser observado nas matrizes da Figura 13 e Figura 14, as quais foram geradas a partir do modelo dado na Figura 10.

P0	P1	P10	P2	P3	P18	P11	P13	P14	P5	P6	P12	P7	P8	P15	P4	P16	P9	P17	S19	S20	S21	S22	S23	S24	S25	S26	S27	S28	S29	S30			
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T0-1		
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T0-10		
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	T1-2		
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T2-3	
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	T3-18	
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	T10-11		
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	T13-2	
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T2-14	
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	T14-5	
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	T14-15	
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	T5-13	
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	T5-6	
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	T12-6	
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	T6-11	
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T11-7	
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T7-12
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T7-8
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	T8-9
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	T17-18
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T16-17
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	T4-15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T16-4
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	T15-9
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T9-16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	T4-5
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	T12-13

Figura 13. Matriz de incidência reversa

P0	P1	P10	P2	P3	P18	P11	P13	P14	P5	P6	P12	P7	P8	P15	P4	P16	P9	P17	S19	S20	S21	S22	S23	S24	S25	S26	S27	S28	S29	S30	
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T0-1
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T0-10
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	T1-2
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T2-3
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T3-18
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	T10-11
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	T13-2
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T2-14
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	T14-5
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	T14-15
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	T5-13
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	T5-6
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	T12-6
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	T6-11
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T11-7
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T7-12
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T7-8
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	T8-9
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	T17-18
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	T16-17
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	T4-15
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T16-4
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	T15-9
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T9-16
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	T4-5
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	T12-13

Figura 14. Matriz de incidência direta

A primeira matriz corresponde à matriz de entrada, também chamada de matriz de incidência reversa. Esta matriz é descrita por  $W_{(j,i)}^- = \#(p_i, t_j)$ , indicando que cada elemento da matriz de entrada  $W^-$  corresponde a um número de arcos conectando o lugar  $p_i$  para cada transição de entrada  $t_j$ . A segunda é a matriz de saída, também chamada de matriz de incidência direta e descrita por  $W_{(j,i)}^+ = \#(t_j, p_i)$ , a qual indica que cada elemento da matriz de saída  $W^+$  corresponde ao número de arcos conectando a saída de cada transição  $t_j$  ao lugar  $p_i$ . A matriz característica  $W$  é obtida pela subtração da matriz de incidência reversa da matriz de incidência direta, conforme apresentado na Figura 15.

A transição  $t_j$  estará habilitada para qualquer marcação  $M$  desde que esta possua os elementos maiores ou iguais aos elementos do vetor de disparo da transição em questão multiplicado pela matriz de incidência reversa da rede, isto é,  $M \geq e[j] \cdot W^-$ . Além desta verificação de habilitação do disparo das transições, pelo fato de utilizar-se um modelo em rede de Petri de alto nível, condiciona-se o disparo das transições ao tempo e à rota associados às marcas dos lugares de entrada.

Uma vez definidas as possíveis rotas para cada veículo, o programa calcula, para cada conjunto de rota, as equações de estado da rede de Petri de alto nível representada pelas matrizes de incidência e característica. Neste ponto do programa são consideradas todas as marcas, com suas rotas, e o intervalo dos semáforos e seus tempos.

P0	P1	P10	P2	P3	P18	P11	P13	P14	P5	P6	P12	P7	P8	P15	P4	P16	P9	P17	S19	S20	S21	S22	S23	S24	S25	S26	S27	S28	S29	S30				
-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T0-1			
-1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T0-10			
0	-1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T1-2			
0	0	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T2-3			
0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T3-18			
0	0	-1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T10-11			
0	0	0	1	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T13-2			
0	0	0	-1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T2-14			
0	0	0	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T14-5		
0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T14-15		
0	0	0	0	0	0	0	0	1	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T5-13		
0	0	0	0	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T5-6		
0	0	0	0	0	0	0	0	0	0	0	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T12-6		
0	0	0	0	0	0	1	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T6-11		
0	0	0	0	0	0	-1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T11-7		
0	0	0	0	0	0	0	0	0	0	0	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T7-12	
0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T7-8	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T8-9	
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T17-18	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T16-17	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T4-15	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T16-4	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T15-9	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T9-16
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T4-5
0	0	0	0	0	0	0	1	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	T12-13

Figura 15. Matriz característica

A análise é completada assim que as marcas alcançam seus destinos finais. Após completar a análise, o programa obtém o tempo total que as marcas consumiram para se moverem por meio dos lugares definidos em sua rota. A marcação que resultar no menor tempo para alcançar a marcação final é adicionada à rede.

### 3.3 ALGORITMO GENÉTICO

A primeira parte do Algoritmo Genético tem a função de definir uma população inicial, a qual consiste de uma matriz de  $n$  linhas, correspondendo ao tamanho da população, e  $m$  colunas, correspondendo à quantidade de veículos na via em um instante  $t$  e tendo em cada lócus um vetor de rotas representando o caminho a ser percorrido. Neste trabalho, os alelos são representados pelos veículos e suas possíveis rotas nas vias urbanas controladas, sendo que cada gene pode assumir o valor de apenas um alelo, ou seja, uma única rota para cada veículo. A formação da população inicial, assim como a geração das próximas populações, precisa manter a estrutura do cromossomo factível, visto que, neste caso, não é permitido que a rota de um veículo num determinado lugar seja associada à de outro veículo em outro lugar.

Considerando o modelo da rede de Petri dado na Figura 10, e supondo que haja quatro veículos no lugar P0 num instante 0, que desejam viajar para o lugar P18, suas possíveis rotas

encontradas pela análise do espaço de estados são  $R_1 = \{P0, P1, P2, P3, P18\}$ ,  $R_2 = \{P0, P10, P11, P7, P12, P13, P2, P3, P18\}$ ,  $R_3 = \{P10, P11, P7, P8, P9, P16, P17, P18\}$  e  $R_4 = \{P0, P10, P11, P7, P8, P9, P16, P4, P5, P13, P2, P3, P18\}$ . Portanto, conforme apresentado na Figura 16, cada alelo representará uma rota e cada coluna representará um veículo.

R1	R1	R2	R3
R1	R1	R1	R2
R2	R1	R2	R1
R3	R2	R4	R4
R3	R2	R4	R4
R1	R2	R3	R4
R2	R1	R4	R3
R1	R2	R1	R2
R1	R2	R3	R4
R2	R1	R4	R3
R1	R2	R1	R2
R1	R2	R3	R4
R3	R2	R4	R4
R1	R2	R3	R4
R2	R1	R4	R3
R1	R2	R1	R2
R2	R1	R2	R1

Figura 16. População do Algoritmo Genético

Conforme apresentada na Equação 3, a função de *fitness* utilizada consiste no cálculo do tempo percorrido por todos ( $N$ ) os veículos ( $i$ ) considerando suas rotas ( $r$ ) do cromossomo, utilizando a simulação da rede de Petri de alto nível, ou seja, o cálculo da equação de estado sobre a matriz de incidência.

Equação 3. Cálculo do *fitness*

$$fitness = \sum_{i=0, r=0}^N [V_{(i,r)}]$$

A seleção dos indivíduos foi feita utilizando o algoritmo da Roleta, dada sua característica probabilística de aumentar as chances dos indivíduos com a melhor função de *fitness*, mas sem desconsiderar também os demais indivíduos da população.

A cada iteração, os indivíduos selecionados da população sofrem recombinação para que possam gerar descendentes, o que é feito pela definição aleatória de um pedaço do primeiro

cromossomo para ser passado ao filho, e preenchimento dos demais genes com parte do segundo cromossomo, sem que os elementos sejam repetidos. A recombinação de dois indivíduos selecionados pelo método de seleção de Roleta é apresentada na Figura 17.

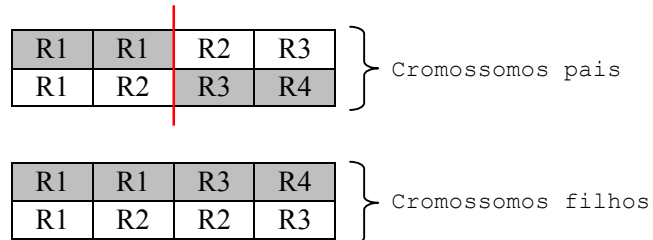


Figura 17. Processo de recombinação

Gerada a próxima população, os indivíduos sofrem novamente o processo de seleção natural com base na função de *fitness* e os indivíduos selecionados são recombinados de maneira a formar as próximas gerações. Este algoritmo pode ser executado até que se encontre uma solução ótima; entretanto, como a execução deverá ser feita a cada transição de um veículo de uma via para outra, quando identificado um novo veículo na área controlada, o critério de parada consiste na execução do algoritmo por no máximo 10 segundos, permitindo o sensoriamento das informações e a distribuição da solução encontrada para os veículos.

## 4 RESULTADOS E DISCUSSÕES

Para validar o algoritmo proposto, foram elaborados dois conjuntos de teste, e cada resultado obtido nestes conjuntos foi comparado com a execução do algoritmo de menor caminho de Dijkstra, o qual é muito utilizado na definição de rotas em diversos ambientes, conforme descrito nos trabalhos de DongKai (2010) e Randria (2007).

O algoritmo de menor caminho de Dijkstra foi desenvolvido para determinar o caminho mais curto entre o nó origem e qualquer outro nó de um grafo. Seja  $u_i$  a distância mais curta do nó de origem 1 ao nó  $i$ , e define-se  $d_{ij} (\geq 0)$  como o comprimento do arco  $(i,j)$ . Então o algoritmo define o rótulo para um nó imediatamente posterior,  $j$ , como  $[u_j, j] = [u_i + d_{ij}, j]$ ,  $d_{ij} \geq 0$ . O rótulo para o nó inicial é  $[0, -]$ , o que indica que o nó não tem nenhum predecessor. Os rótulos do nó são de dois tipos: temporários e permanentes, sendo o primeiro modificado caso haja uma rota mais curta até um nó, e o segundo caso todas as opções tenham sido verificadas. [Taha, 2008].

Para trabalhar com o algoritmo de Dijkstra, utilizou-se o *framework* JUNG (Java Universal Network/Graph), o qual consiste em uma biblioteca de código aberto para manipulação de grafos, permitindo manipulá-los e aplicar alguns algoritmos já conhecidos [Kelley, 2009].

### 4.1 REDE DE PETRI COM 19 LUGARES

Todos os testes foram executados usando um computador com processador Core i5 de 1.8 GHz e 4GB de memória RAM. O primeiro conjunto de teste utilizou-se de uma matriz de incidência de 25 linhas e 31 colunas, apresentada na Figura 10, e que representa o modelo do tráfego urbano apresentado na Figura 9. A marcação consiste de dez marcas no lugar P10 e dez marcas no lugar P12. Cada marca tem um tempo de espera associado, iniciando em 2 segundos, para o primeiro veículo, e com uma diferença de 2 segundos para os demais veículos, ou seja, a primeira marca estará disponível para ser consumida pela transição após 2 segundos, a segunda marca após 4 segundos e a última marca após 20 segundos. Cada marca adicionada no lugar de saída da transição disparada terá seu tempo de consumo incrementado em 12 segundos, de maneira a representar o tempo mínimo necessário para percorrer uma via de aproximadamente 100 metros com uma velocidade média de 40 Km/h. Como as vias urbanas são representadas por



filas de veículos, mesmo que a marca atinja seu tempo permitido para ser consumida, o sistema considerará esta marca apenas no disparo da transição se ela for a primeira da fila.

#### 4.1.1 ENTRADA DE APENAS DE UM VEÍCULO

O primeiro teste consiste na comparação do tempo gasto por uma única marca usando as rotas encontradas na análise do espaço de estado e usando o algoritmo de menor caminho de Dijkstra. Para realizar o teste, foi adicionada uma marca no lugar P0, tendo como seu destino o lugar P13. Ao gerar o espaço de estados e analisá-lo, encontraram-se as seguintes possíveis rotas:  $R_1 = \{P0, P1, P2, P14, P5, P13\}$ ,  $R_2 = \{P0, P10, P11, P7, P12, P13\}$ ,  $R_3 = \{P0, P10, P11, P7, P8, P9, P16, P4, P5, P13\}$  e  $R_4 = \{P0, P1, P2, P14, P15, P9, P16, P4, P5, P13\}$ , as quais correspondem ao vetor de disparos:  $f(R_1) = \{T0-1, T1-2, T2-14, T14-5, T5-13\}$ ,  $f(R_2) = \{T0-10, T10-11, T11-7, T7-12, T12-13\}$ ,  $f(R_3) = \{T0-10, T10-11, T11-7, T7-8, T8-9, T9-16, T16-4, T4-5, T5-13\}$  e  $f(R_4) = \{T0-1, T1-2, T2-14, T14-15, T15-9, T9-16, T16-4, T4-5, T5-13\}$ .

A menor rota encontrada pelo algoritmo de menor caminho de Dijkstra foi  $R_D = \{P0, P10, P11, P7, P12, P13\}$ , a qual é equivalente à rota  $R_2$ . Na Figura 18 é apresentado o gráfico com os lugares em que a marca foi adicionada durante seu percurso usando a rota  $R_D$ . O tempo total gasto pela marca para alcançar seu destino foi 82 segundos.

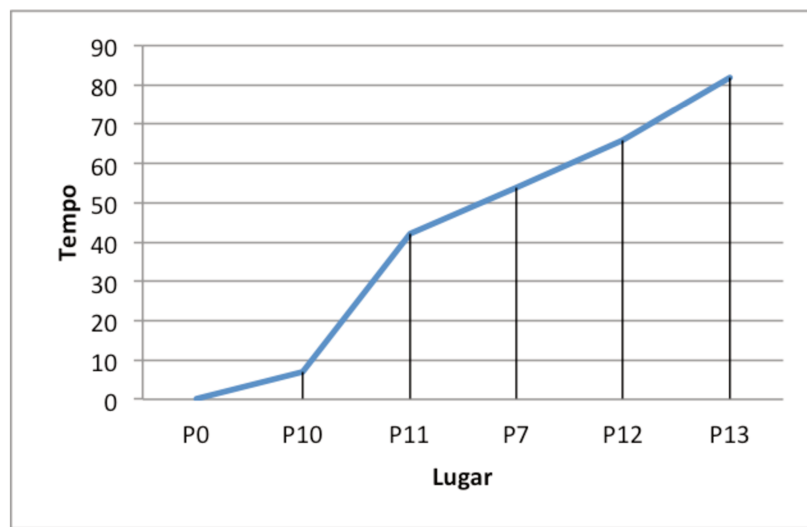


Figura 18. Lugares percorridos pela marca no decorrer do tempo usando  $R_D$

Utilizando o algoritmo proposto neste trabalho, as equações de estados para as quatro rotas  $R_1$ ,  $R_2$ ,  $R_3$  e  $R_4$  foi calculada. É possível notar que as rotas  $R_1$  e  $R_2$  têm a mesma quantidade de lugares, significando que possuem a mesma quantidade de transições a serem disparadas; entretanto, calculando-se as equações de estado na matriz de incidência, a rota  $R_1$  teve um tempo de viagem mais curto (61 segundos) que a rota definida pelo algoritmo de menor caminho de Dijkstra ( $R_2$ ), como é possível ver na Figura 19.

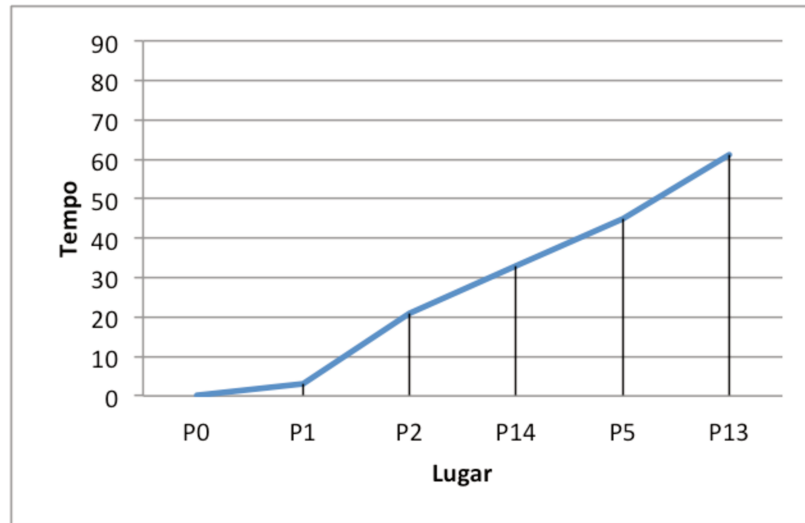


Figura 19. Lugares percorridos pela marca no decorrer do tempo usando  $R_1$

Conclui-se, a partir deste primeiro teste, que rotas diferentes com a mesma quantidade de lugares podem prover tempos de viagem diferentes quando considerados outros veículos nas vias ou interrupções causadas pelo intervalo vermelho do semáforo.

#### 4.1.2 ENTRADA UNIFORME DE VEÍCULOS

Este teste consiste na entrada de veículos em um tempo uniforme no lugar P0 e na comparação do tempo gasto por todos os veículos para alcançarem o lugar P13. As rotas encontradas na análise do espaço de estados e pelo algoritmo de menor caminho de Dijkstra são as mesmas dadas no tópico 4.1.1,  $R = \{R_1, R_2=R_D, R_3, R_4\}$ . A cada dois segundos é adicionada uma marca no lugar P0. Tendo a rota sido definida pelo algoritmo de menor caminho de Dijkstra, na Figura 20 é apresentado o gráfico com a quantidade de marcas nos lugares no decorrer do tempo. No mesmo gráfico existe uma linha pontilhada indicando a quantidade média de marcas

nos lugares durante o mesmo tempo. Observa-se que a quantidade de marcas nos lugares cresce, devido à interrupção causada pelo intervalo vermelho do semáforo, e então decresce durante o intervalo verde do semáforo. Além disso, os lugares utilizados são sempre os mesmos. O tempo total gasto para que todas as marcas alcancem o lugar P3, como observado no gráfico, é de 188 segundos.

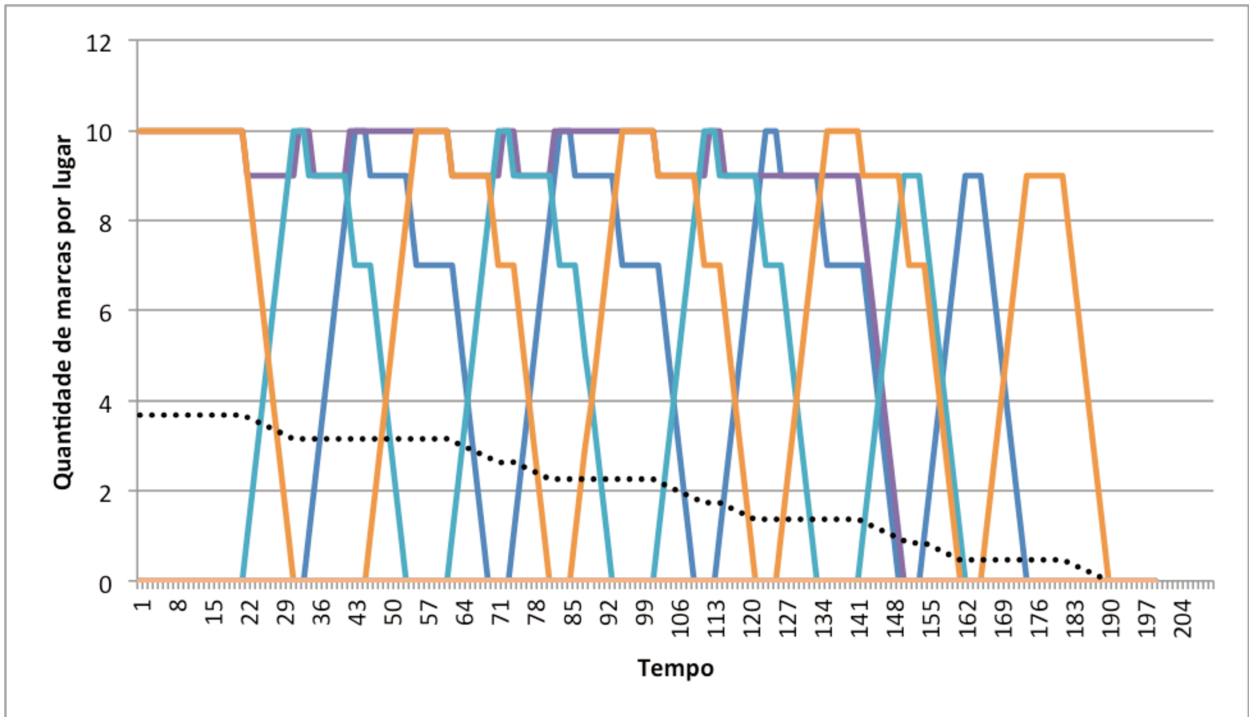


Figura 20. Quantidade de marcas nos lugares no decorrer do tempo usando a rota  $R_D$

Considerando-se as outras rotas encontradas pela análise do espaço de estado e a quantidade de marcas nos lugares em relação ao tempo, o algoritmo proposto neste trabalho procura pelas rotas que forneçam um tempo de viagem mais curto levando em conta a rota das outras marcas da rede, isto é, se uma grande quantidade de marcas assume a segunda rota ( $R_2$ ), causando um crescimento na fila de espera, por exemplo, no lugar P10, o algoritmo proposto indica uma rota alternativa sempre que esta proveja um tempo menor de viagem.

Na Figura 21 é apresentado um gráfico com a quantidade total de marcas em todos os lugares da rede no decorrer do tempo. As marcas possuem as melhores rotas encontradas na análise da rede de Petri de alto nível. A quantidade média de marcas nos lugares durante o tempo é representada pela linha pontilhada.

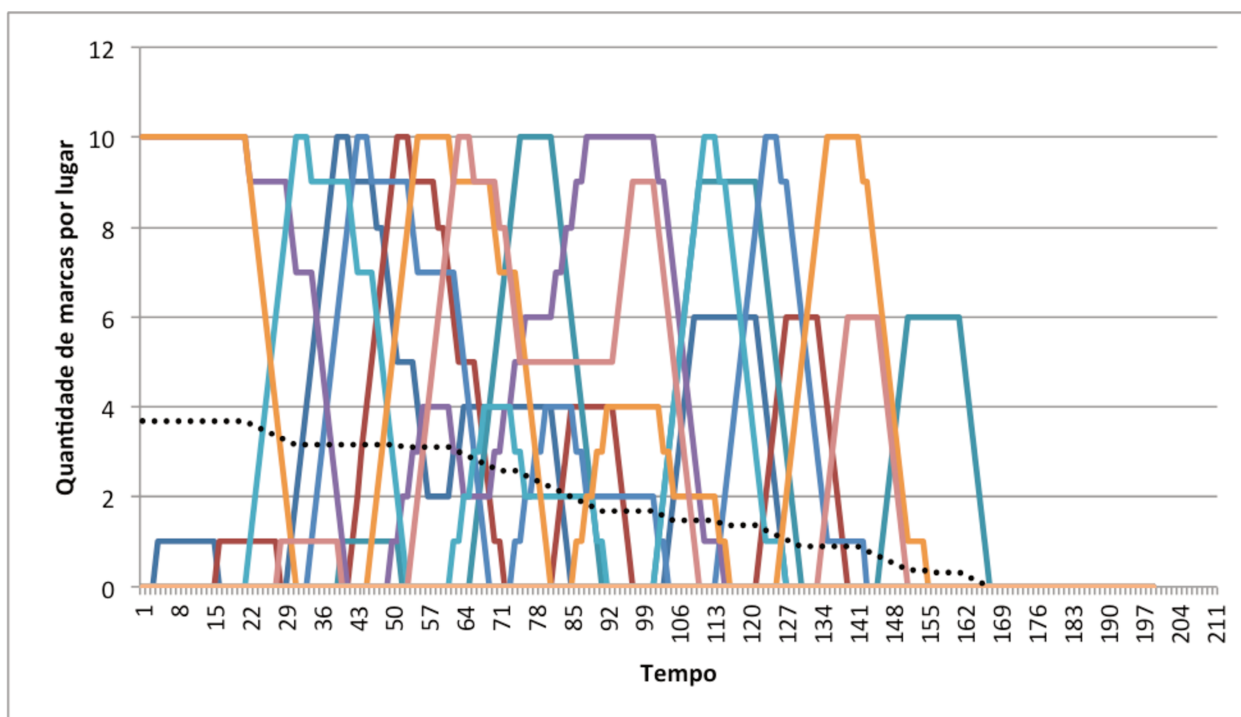


Figura 21. Quantidade de marcas nos lugares no decorrer do tempo usando as combinações de rotas.

Observa-se que alguns lugares, antes não utilizados nas rotas definidas pelo algoritmo de menor caminho de Dijkstra, por terem uma distância equivalente ou um pouco maior do que caminho mais curto, auxiliaram na otimização do tempo total de viagem, o qual caiu de 188 para 165 segundos. Se consideradas as vias urbanas, esta otimização pode reduzir não só o tempo gasto pelos veículos particulares, mas também o tempo gasto pelos transportes públicos, que, em algumas cidades, não possuem linhas exclusivas e disputam as vias com os demais veículos.

## 4.2 REDE DE PETRI COM 50 LUGARES

De maneira a validar e apresentar a eficiência e a inovação do algoritmo proposto, utilizou-se uma área de tráfego urbano maior, contendo 50 vias, conforme apresentado na Figura 22. O mapa representa um trecho da área central da cidade de São José do Rio Preto, São Paulo, Brasil, a qual possui um grande fluxo de veículos no decorrer do dia. Os testes foram desenvolvidos com base neste modelo e, para cada resultado obtido, comparou-se com a utilização do algoritmo de menor caminho de Dijkstra.

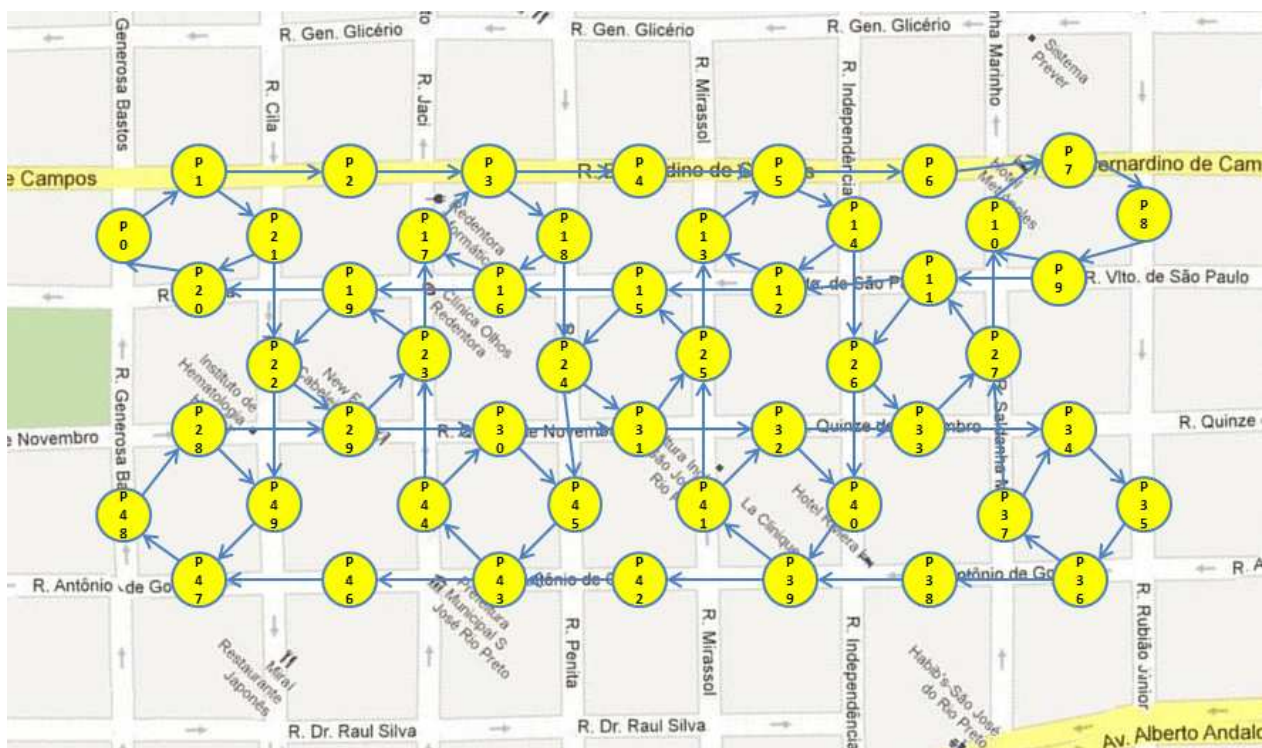


Figura 22. Mapa das vias urbanas controladas pelo sistema

Nesta seção utilizou-se um modelo de rede de Petri de alto nível, apresentado na Figura 23, e que contém 50 lugares, representando a fila de veículos, e 80 transições, utilizadas para realizar o cruzamento de uma marca de um lugar de entrada para um lugar de saída, ou seja, de uma via para outra, de acordo com sua rota.

Cada rota é definida a partir da análise do espaço de estados gerado utilizando a ferramenta IOPT Tool, tendo uma marca no lugar representado pela posição atual do veículo e zero marcas nos demais lugares.

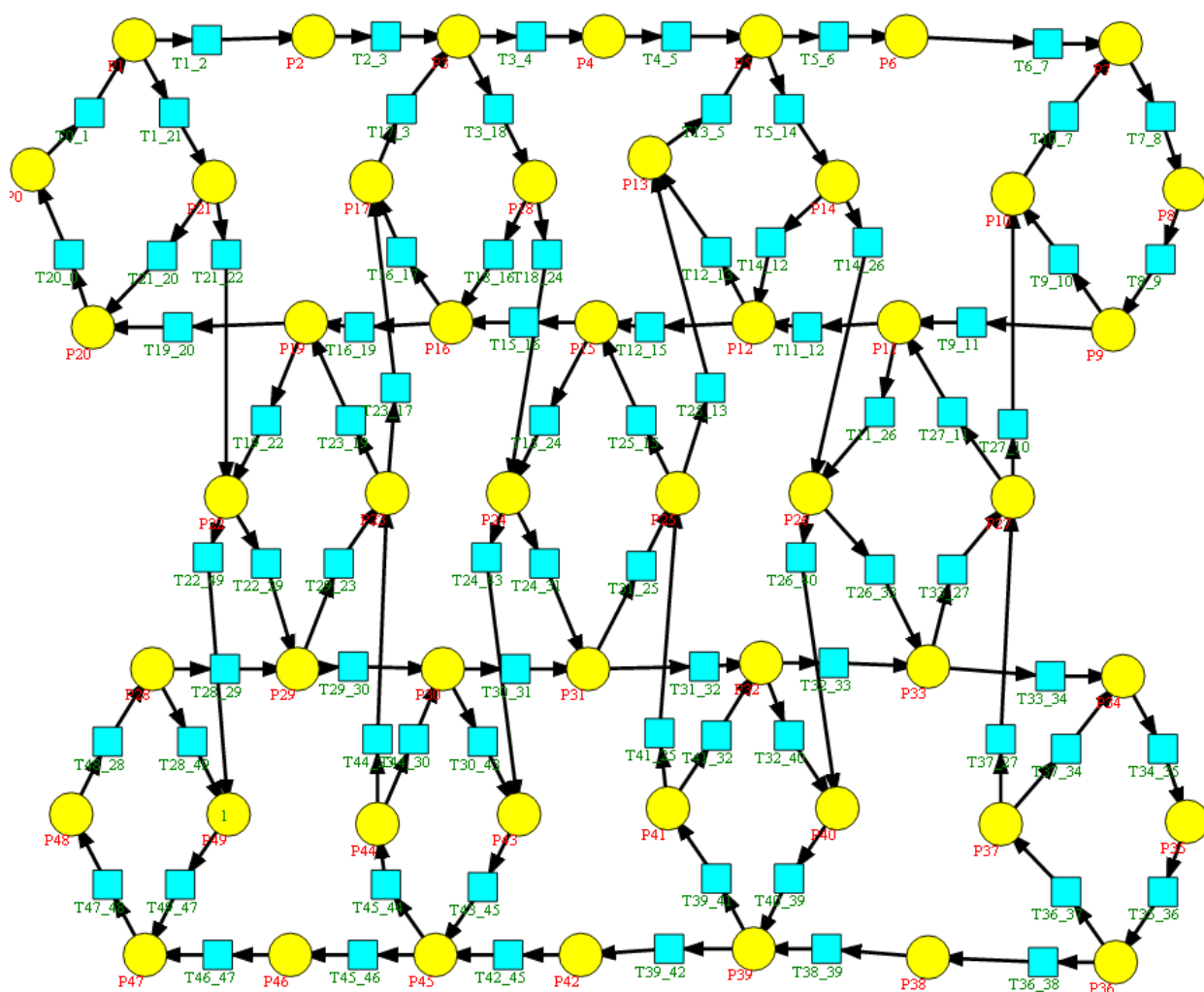


Figura 23. Modelo da rede de Petri utilizada como modelo das vias urbanas controladas

#### 4.2.1 ENTRADA UNIFORME DE VEÍCULOS

Neste cenário assume-se que uma marca inicia sua trajetória no lugar P0 em direção ao lugar P34 a cada 2 segundos. A partir da definição das rotas utilizando o algoritmo de menor caminho de Dijkstra, tem-se que o menor caminho consiste nos lugares  $R_D = \{P0, P1, P2, P3, P4, P5, P14, P26, P33, P34\}$ . Na Figura 24 é apresentado um gráfico com a quantidade de veículos em cada lugar da rede no decorrer do tempo. O tempo gasto para que todas as marcas atinjam seus lugares de destino é de 336 segundos.



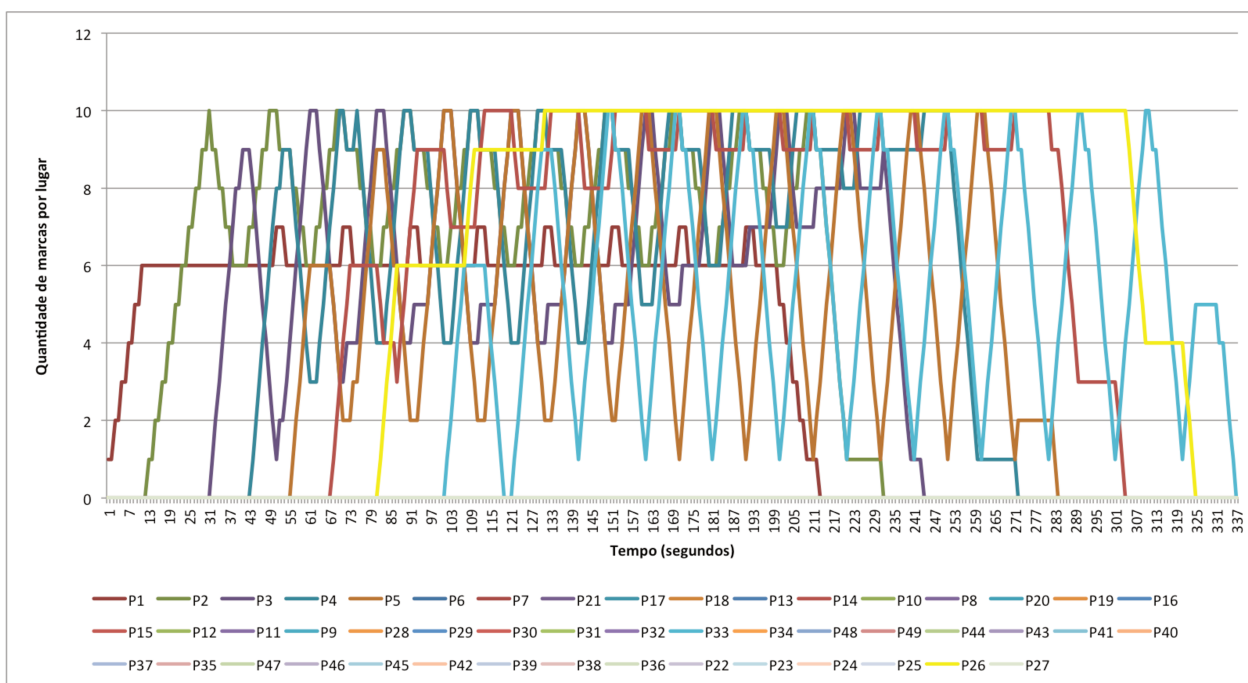


Figura 24. Quantidade de veículos nos lugares no decorrer do tempo usando a rota fornecida pelo algoritmo de menor caminho de Dijkstra

É possível observar no gráfico que a maioria dos lugares da rede atingiu a capacidade máxima de veículos durante um determinado tempo, visto que o algoritmo de menor caminho de Dijkstra determina para cada veículo, com mesma origem e destino, a mesma rota. Na Figura 24, a linha amarela representa a quantidade de marcas no lugar P26, o qual precede o penúltimo lugar de destino das marcas (P33). Verifica-se que, entre 130 e 330 segundos, este lugar contém a capacidade máxima de marcas preenchida, resultando em um longo tempo de espera para que estas possam ser consumidas pelas transições, o que resulta, se consideradas como veículos nas vias urbanas, que estes veículos aguardarão um longo tempo na via para poder chegar ao seu destino. Se considerada a via urbana, o tempo que o veículo demorará para cruzar a via se torna ainda maior, visto que as interrupções causadas pelos semáforos fazem com que os motoristas tenham que parar, e, para retomar o percurso, perdem alguns segundos para verificar o intervalo verde e atuar no aceleração do veículo. Outro problema que pode acontecer, se consideradas as vias urbanas, é a ocorrência de incidentes, tais como colisões, quebra de veículos, chuva etc., que favorecem a criação de congestionamentos, visto que a via em questão já possui sua capacidade máxima preenchida.

Na Figura 25 é mostrado um gráfico com a quantidade de marcas nos lugares da Figura 23 em relação ao tempo, tendo suas rotas definidas pelo algoritmo proposto neste trabalho. Neste

caso, sempre que um veículo começa a trafegar numa área controlada, o sistema calcula suas possíveis rotas, a partir da análise do espaço de estados, e o Algoritmo Genético define, utilizando as informações já processadas de cada veículo (rota já definida, tempo de permanência na via e posição atual), um novo conjunto de rotas. Esta definição é guiada pela minimização do resultado fornecido pela função de *fitness*, a qual consiste no retorno do tempo gasto pelas marcas, dada sua rota definida pelo Algoritmo Genético, durante a simulação da rede de Petri de alto nível.

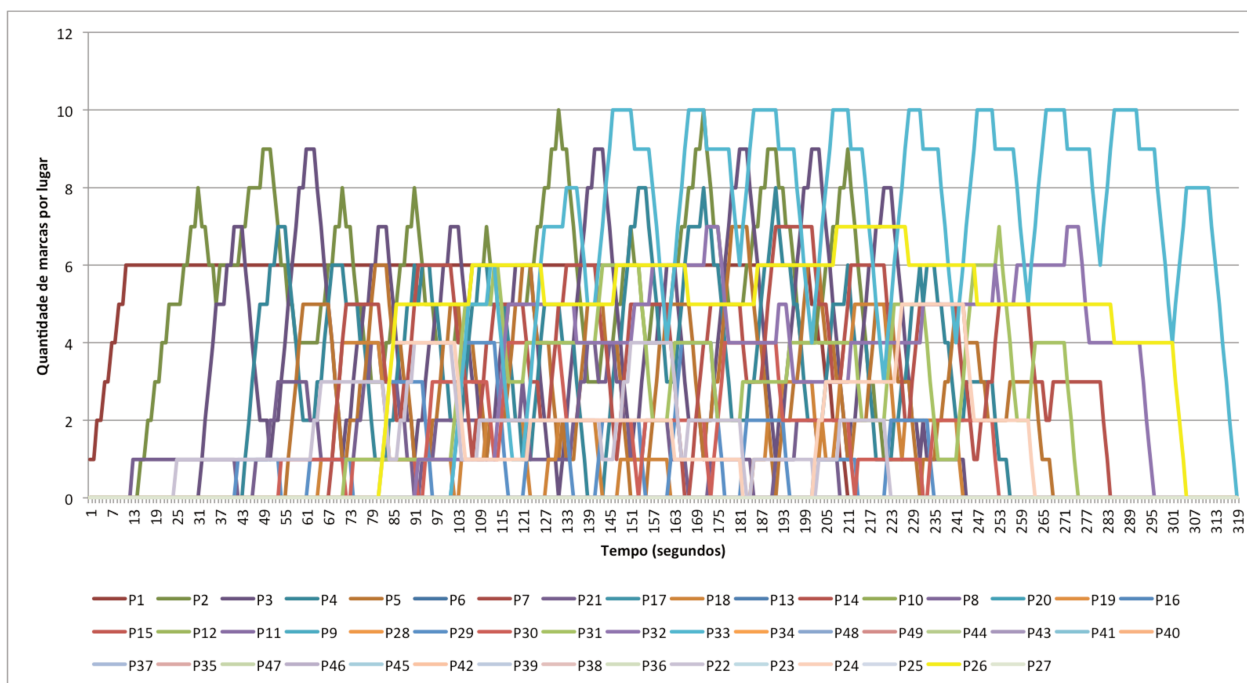


Figura 25. Quantidade de veículos (4 rotas) no decorrer do tempo usando Algoritmo Genético

Esta abordagem garante a utilização de lugares não utilizados anteriormente e a não sobrecarga da quantidade de marcas por lugar, otimizando assim o tempo total de percurso de uma marca e, conseqüentemente, de todas as demais marcas na rede, conforme pode ser visto na Figura 18, que mostra uma diminuição do tempo total de 336 segundos para 318 segundos. Outro ponto a ser considerado é que uma das rotas será sempre a rota encontrada pelo algoritmo de menor caminho de Dijkstra, garantindo assim que o Algoritmo Genético retorne, no mínimo, a mesma solução apresentada por este primeiro algoritmo.

Em comparação com o gráfico apresentado na Figura 24, o algoritmo proposto teve um decremento na quantidade total de marcas em todos os lugares da rede no decorrer do tempo pelo fato de considerar outros lugares, e suas quantidades de marcas, e não apenas o menor caminho.



Assim, como é possível ver na Figura 25, a linha amarela que representa o lugar (P26) que precede o penúltimo lugar de destino de todas as marcas não atingiu sua capacidade, mesmo se mantendo com certa quantidade de marcas no decorrer do tempo.

Na Figura 26 é mostrado um gráfico com a quantidade de veículos no decorrer do tempo, o qual foi gerado a partir do algoritmo proposto considerando apenas 2 rotas possíveis para cada marca. É possível verificar que, ainda otimizando o tempo total de percurso e diminuindo a quantidade de marcas por lugar, alguns lugares atingem a capacidade de maneira mais constante, visto que neste caso o Algoritmo Genético trabalha com um espaço de busca menor e, portanto, o resultado aproxima-se do resultado obtido utilizando-se apenas o algoritmo de menor caminho de Dijkstra.

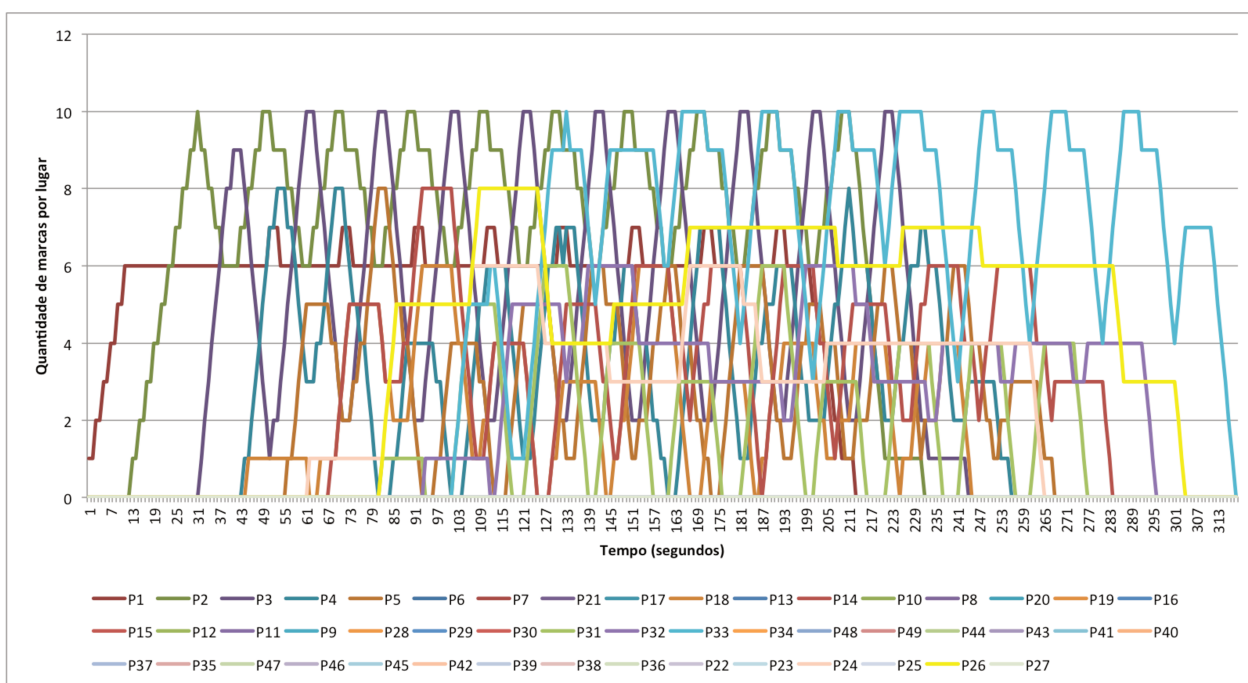


Figura 26. Quantidade de veículos (2 rotas) no decorrer do tempo usando Algoritmo Genético

É importante relatar que os Algoritmos Genéticos utilizados até este teste tiveram uma população de 2000 indivíduos e 20 gerações, diferenciando-se apenas as quantidades de rotas disponíveis para cada veículo. A variação no tamanho da população e na quantidade de gerações para mais não influenciou a solução dada; entretanto, a diminuição do tamanho da população, em alguns casos, retornou soluções não satisfatórias quando os veículos possuíam uma quantidade maior de rotas.

#### 4.2.2 ENTRADA UNIFORME DE VEÍCULOS EM VIAS DIFERENTES

Neste cenário assume-se que a cada 2 segundos uma marca é criada em cada um dos lugares da rede de Petri dada na Figura 23, representando a entrada de um novo veículo em cada uma das vias a cada 2 segundos. Considera-se que o veículo não estava trafegando anteriormente pelas vias, ou seja, este veículo iniciou sua jornada, pois estava parado em um estacionamento, por exemplo.

No gráfico da Figura 27 é apresentada a quantidade de marcas por lugar no decorrer do tempo utilizando o algoritmo de menor caminho de Dijkstra como única fonte de definição de rotas. É possível observar que, tendo todos os veículos um único destino (lugar P20), alguns lugares mantêm, inicialmente, uma quantidade baixa de marcas; entretanto, no decorrer do tempo, próximo ao destino, estas marcas são aglomeradas, conforme indicado pela linha verde do gráfico. Isso acarreta os problemas de atrasos causados pelo tempo inicial perdido ou por eventuais incidentes que podem ocorrer, já relatados anteriormente.

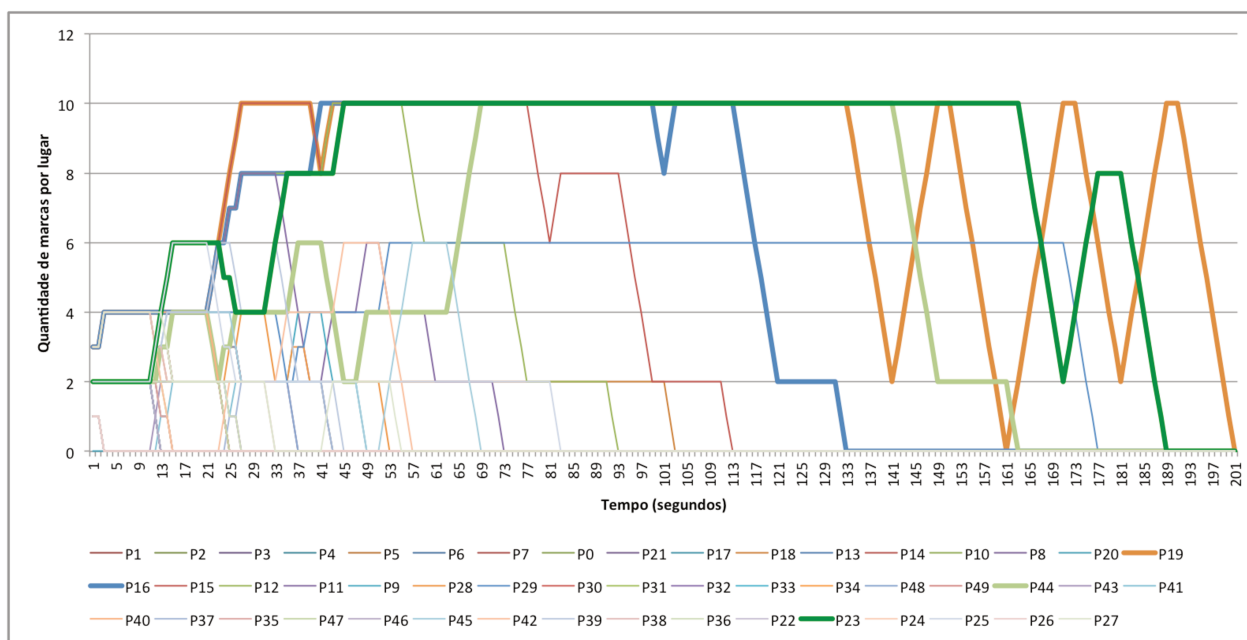


Figura 27. Quantidade de veículos no decorrer do tempo utilizando algoritmo de menor caminho de Dijkstra

No gráfico da Figura 28 é apresentada a quantidade de marcas por lugar no decorrer do tempo utilizando o algoritmo proposto neste trabalho, tendo uma população de 4000 indivíduos,

40 gerações de descendentes e um máximo de 8 possíveis rotas para cada veículo. É possível observar que, tendo todos os veículos um único destino (lugar P20), neste caso é mantida uma quantidade baixa de marcas em cada lugar por um tempo maior, e, apenas após um determinado tempo de percurso, pelo fato de as marcas se aproximarem do lugar de destino, há uma aglomeração de marcas, conforme indicado pela linha verde do gráfico. Considerando que o tempo de percurso é equivalente, tendo um ganho de apenas 1 segundo, a distribuição das marcas pelos lugares já proporciona uma eficiência no fluxo do tráfego urbano.

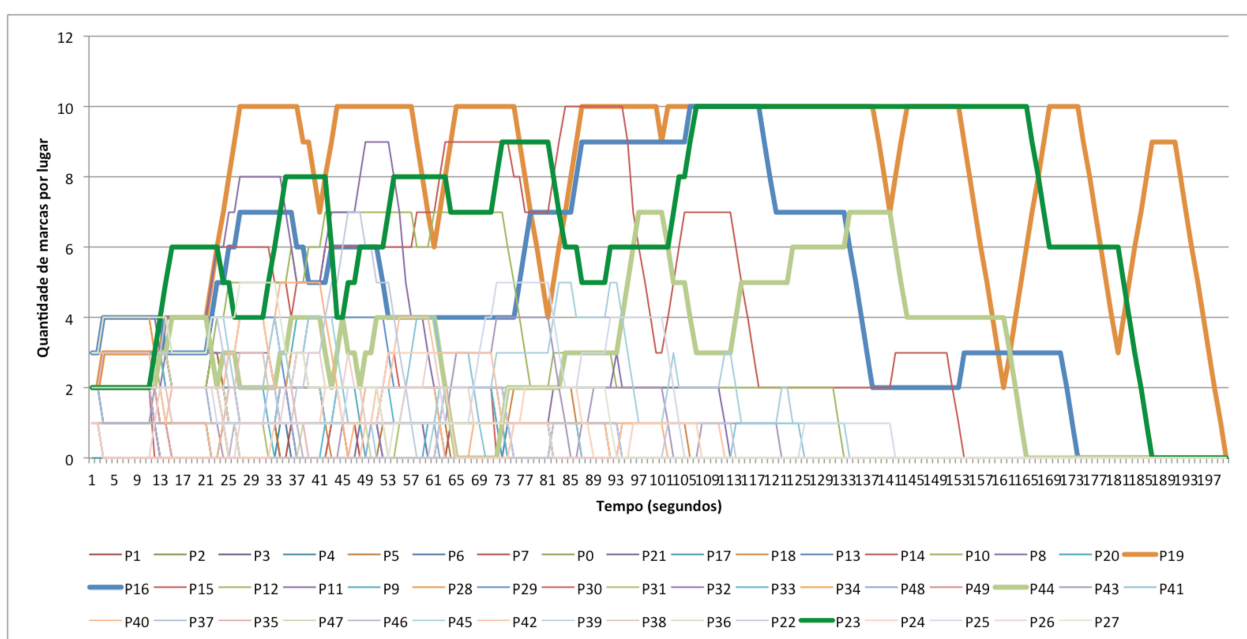


Figura 28. Quantidade de veículos no decorrer do tempo utilizando algoritmo de menor caminho de Algoritmo Genético

Em relação ao tempo individual gasto para percorrer seus trajetos, o menor tempo foi de 12 segundos, sendo igual tanto ao se aplicar o algoritmo de menor caminho de Dijkstra quanto ao se aplicar o algoritmo proposto. Em relação ao maior tempo gasto, o algoritmo proposto teve um ganho de 1 segundo em relação ao de menor caminho de Dijkstra, tendo aquele gasto 211 segundos e este, 212 segundos. O tempo médio de percurso também foi semelhante, tendo o algoritmo proposto apontado um total aproximado de 116 segundos, e o algoritmo de menor caminho de Dijkstra, um total aproximado de 117 segundos.

Portanto, conclui-se que o tempo de viagem, neste caso, manteve-se equivalente ao tempo consumido quando utilizado o algoritmo de menor caminho de Dijkstra, mas, se considerado o

fluxo do tráfego urbano e os tempos perdidos iniciais, o algoritmo proposto neste trabalho indica uma melhora significativa no tráfego urbano, visto que ele distribui os veículos pelas vias não consideradas pelo outro algoritmo, diminuindo assim a probabilidade de congestionamentos no caso de incidentes.

## 5 CONCLUSÕES

Conforme apresentado neste trabalho, a cada dia que passa aumenta a quantidade de veículos trafegando nas vias urbanas. Os veículos e a infraestrutura utilizam cada vez mais da tecnologia de informação e comunicação para auxiliar os motoristas e os órgãos competentes na verificação e controle do tráfego. Além disso, os dispositivos móveis se tornaram um bem de consumo indispensável na vida das pessoas e, com isso, a criação de aplicações que capturam as informações transmitidas por eles, em muitos casos em tempo real, é de grande benefício para a sociedade.

Neste trabalho foram abordados os principais tópicos da engenharia de tráfego urbano em vias interruptas e como estas e seus componentes, veículos e semáforos, podem ser modelados utilizando uma rede de Petri de alto nível. A possibilidade de modelar o sistema utilizando a rede de Petri permite não só uma melhor compreensão do sistema como um todo, mas permite a utilização das técnicas de análise incorporadas ao modelo para realizar uma série de validações a fim de otimizar o fluxo do tráfego.

Diversos trabalhos nesta área foram relatados durante o texto. A inovação sobre os trabalhos já existentes consiste não só da definição de um caminho mais curto para um veículo, mas na otimização do fluxo do tráfego urbano como um todo, considerando as rotas de todos os veículos nas vias. Para atingir este objetivo, estudamos e desenvolvemos um Algoritmo Genético que tem a simulação da rede de Petri como sua função de *fitness*, sendo o retorno desta simulação o valor a ser minimizado pelo algoritmo. Segundo pesquisas na área, não há trabalhos que utilizam esta abordagem na solução do problema, mesmo considerando apenas um veículo.

A metodologia proposta neste trabalho permitiu utilizar os principais recursos de análise das redes de Petri em conjunto com uma meta-heurística que possibilita retornar, em um tempo hábil, uma solução factível e ótima para o problema. Sendo assim, com a junção das tecnologias emergentes já sendo aplicadas no controle do tráfego urbano, esta ferramenta se torna exequível e aplicável em situações reais de controle do tráfego urbano.

A partir dos testes realizados é possível concluir que a metodologia utilizada agiliza o fluxo do tráfego urbano quando comparado ao algoritmo de menor caminho de Dijkstra e, no pior dos casos, retorna a mesma solução deste último algoritmo. Um fator importante a ser considerado nos resultados é que mesmo tendo um tempo de percurso similar ao algoritmo de

menor caminho de Dijkstra, a metodologia proposta permite a utilização de outras vias, antes não utilizadas, para a realização da trajetória dos veículos, acarretando um controle ágil, ou redirecionamento, no caso de incidentes em algumas destas vias, além de distribuir a carga sobre diversas rotas e assim diminuir os desgastes causados normalmente.

Quando implantado, este trabalho auxiliará na redistribuição dos veículos pelas vias urbanas de maneira eficiente, otimizando o fluxo do tráfego urbano, e também criará uma mudança no hábito dos motoristas.

A partir do trabalho desenvolvido foi possível realizar a publicação, até o momento, de três artigos em eventos voltados às redes de Petri, controles e veículos inteligentes. O primeiro trabalho, intitulado “Mobile Application as a tool for urban traffic data collection and generation to Advanced Traveler Information Systems using Wi-Fi networks available in urban centers” foi publicado no IEEE Intelligent Vehicle Symposium (IV’2012) [Dezani, 2012a]. O segundo artigo, chamado “Finding best urban routes based on analyses of High Level and IOPT Petri net models” [Dezani, 2012c], e o terceiro artigo, intitulado “Controlling Traffic Jams on Urban Roads Modeled in Coloured Petri Net using Genetic Algorithm” [Dezani, 2012b], foram apresentados no IEEE International Conference On Industrial Electronics Society (IECON’2012).

## REFERÊNCIAS

- Balan, G.; Luke, S. **History-based Traffic Control**. International Conference on Autonomous Agents and Multiagent Engineering, p. 616-621, 2006.
- Castro, L. N. **Fundamentals of Natural Computing: Basic Concepts, Algorithms and Applications**. Flórida: Chapman & Hall/CRC, 2006.
- David, R.; Alla, H. **On Hybrid Petri Nets**. Journal of Discrete Event Dynamic Systems, v.11, p. 9-40, Janeiro de 2001.
- DETRAN. Disponível em: <<http://www.detran.sp.gov.br>>. Acesso em: 28 out 2012.
- Dezani, H.; Damiani, F.; Viudes, U.; Parra, I. A. **Mobile application as a tool for urban traffic data collection and generation to Advanced Traveler Information Systems using Wi-Fi networks available in urban centers**. IEEE Intelligent Vehicles Symposium (IV'2012), p. 288-292, 2012a.
- Dezani, H.; Gomes, L.; Damiani, F.; Marranghello, N. **Controlling Traffic Jams on Urban Roads Modeled in Coloured Petri Net using Genetic Algorithm**. IEEE International Conference on Industrial Electronics Society (IECON'2012), p. 3025-3030, 2012b.
- Dezani, H.; Gomes, L.; Damiani, F.; Marranghello, N. **Finding best urban routes based on analyses of High Level and IOPT Petri net models**. IEEE International Conference on Industrial Electronics Society (IECON'2012), p. 6130-6135, 2012c.
- Di Febbraro, A.; Giglio, D.; Sacco N. **Urban traffic control structure based on hybrid Petri nets**. IEEE Transactions on Intelligent Transportation Systems, vol. 5, p. 224-237, 2004.
- DongKai, F.; Ping, S. **Improvement of Dijkstra's algorithm and its application in route planning**. 7th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'2010), p. 1901-1904, 2010.
- Dotoli, M.; Fanti, M. P. **An urban traffic network model via coloured timed Petri nets**. Control Engineering Practice, vol. 14, p. 1213-1229, 2006.
- Florea, D. **Advanced road transportation systems**. Editora University of Brasov, 2007.
- Ghali, M. O.; Smith, M. J. **Managing Traffic Congestion by using Traffic Control**. IEE Colloquium on Urban Congestion Management, p. 1-8, 1995.
- Goldberg, D. E. **Genetic algorithms in search, optimization and machine learning**. Addison-Wesley, 1989.
- Gomes, L.; Barro, J. P.; Costa, A.; Nunes, R. **The Input-Output Place-Transition Petri Net Class and Associated Tools**. IEEE International Conference on Industrial Informatics (INDIN'2007), vol. 1, p. 509-514, 2007.

Google. **Google Traffic**. Disponível em: <<http://support.google.com/maps/bin/topic.py?hl=en&topic=1687356&parent=1687355&ctx=topic>>. Acesso em: 18 jul 2012.

Jensen, K.; Kristensen, L. M. **Coloured Petri Nets: Modeling and Validation of Concurrent Systems**. Editora Springer Verlag, 2009.

Kelley, S.; Goldberg, M.; Magdon-Ismael, M.; Mertsalov, K.; Wallace, W.; Zaki, M. **graphOnt: An ontology based library for conversion from semantic graphs to JUNG**. IEEE International Conference on Intelligence and Security Informatics (ISI'09), p. 170-172, 2009.

Kong, Y.; Xue, A. **Urban Traffic Incident Detection Based On Fuzzy Logic**. 32nd Annual Conference on IEEE Industrial Electronics (IECON'2006), p. 772-775, 2006.

Kumar, P.; Singh, V.; Reddy, D. **Advanced Traveler Information System for Hyderabad City**. IEEE Transaction on Intelligent Transportation Systems, vol. 6, p. 26-37, 2005.

Linden, R. **Algoritmos Genéticos: Uma Importante Ferramenta de Inteligência Computacional**. 2 ed. Editora Brasport, 2008.

Liu, X.; Huang, S. **The simulation algorithm of military transportation shortest path based on petri net**. ISECS International Colloquium on Computing, Communication, Control, and Management (CCCM'09), p. 111-114, 2009.

Miller, J. **Vehicle-to-Vehicle-to-Infrastructure (V2V2I) Intelligent Transportation System Architecture**. IEEE Intelligent Vehicles Symposium (IV'2008), p. 715-720, 2008.

Moutinho, F.; Gomes, L. **State space generation algorithm for gals systems modeled by IOPT Petri nets**. 37th Annual Conference on IEEE Industrial Electronics Society (IECON'2011), p. 2839-2844, 2011.

Murata, T. **Petri Nets: Properties, Analysis and Applications**. Proceedings of the IEEE, 77(4)541-580, 1989.

Ni, D. **Determining Traffic-flow characteristics by definition for application in ITS**. IEEE Transactions on Intelligent Transportation Systems, vol. 8, p. 181-187, 2007.

Pereira, F.; Moutinho, F.; Gomes, L.; Campos-Rebelo, R. **IOPT Petri net state space generation algorithm with maximal-step execution semantics**. 9th IEEE International Conference on Industrial Informatics (INDIN'2011), p. 789-795, 2011a.

Pereira, F.; Moutinho, F.; Gomes, L.; Ribeiro, J.; Campos-Rebelo, R. **An IOPT-net state-space generator tool**. 9th IEEE International Conference on Industrial Informatics (INDIN'2011), p. 383-389, 2011b.

Qu, Y.; Li, L.; Liu, Y.; Chen, Y.; Dai, Y. **Travel routes estimation in transportation systems modeled by Petri Nets**. IEEE International Conference on Vehicular Electronics and Safety (ICVES), p. 73-77, 2010.



Randria, I.; Ben Khelifa, M. M.; Bouchouicha, M.; Abellard, P. **A Comparative Study of Six Basic Approaches for Path Planning Towards an Autonomous Navigation**. 33rd Annual Conference of the IEEE on Industrial Electronics Society (IECON'2007), p. 2730-2735, 2007.

Robertson, D. I.; Bretherton, R. D. **Optimizing networks of traffic signals in real time-the SCOOT method**. IEEE Transactions on Vehicular Technology, vol. 40, p. 11-15, 1991.

Roess, P.; Prassas, E. E.; MacShane, W. E. **Traffic Engineering**, 4 ed., Prentice Hall, 2011.

**SCOOT**. Disponível em: <<http://www.scoot-utc.com>>. Acesso em: 12 ago 2011.

Sierra, K.; Bates, B. **Head First: Java**. 2 ed., O'Reilly Media, 2005.

Senda, Y.; Tanev, I.; Shimohara, K. **On the possibility of Priority-Based Road Traffic Control**. SCIE Annual Conference, p. 1510-1513, 2008.

Taha, H. A. **Pesquisa Operacional**, 8 ed., Prentice Hall, 2008.

**Transport Research Laboratory**. Disponível em: < <http://www.trl.co.uk>>. Acesso em: 10 jul 2011.

Tzes, A.; Kim, S.; McShane, W. R. **Applications of Petri Networks to Transportation Network Modeling**. IEEE Transactions on Vehicular Technology, vol. 45, n. 02, 1996.

Vázquez, C. R.; Sutarto, H. Y.; Boel, R.; Silva, M. **Hybrid Petri net model of a traffic intersection in an urban network**. IEEE International Conference on Control Applications (CCA), p. 658-664, 2010.

Wong, S. C.; Wong, W. T.; Leung, C. M.; Tong, C. O. **Group-based optimization of a time-dependent TRANSYT traffic model for area traffic control**, Transportation Research Part B: Methodological, vol. 36, p. 291-312, 2002.