MODELAGEM E SIMULAÇÃO DO TRÁFEGO DE VEÍCULOS E CONTROLE SEMAFÓRICO EM UM AMBIENTE VIRTUAL

MARCELO PARAVISI[1]; SORAIA R. MUSSE[2]; ALESSANDRO DE L. BICHO[1]

[1]Fundação Universidade Federal do Rio Grande/FURG – Depto. de Matemática

Av. Itália, km 8, S/N – Carreiros – CEP 96201900 – Rio Grande/RS – Brasil

[2]Universidade do Vale do Rio dos Sinos/Unisinos – PIPCA

Av. Unisinos, 950 – CEP 93022000 – São Leopoldo/RS – Brasil

e-mail para contato: dmtbicho@furg.br

RESUMO

Simular o trânsito de áreas urbanas de maneira realista em um ambiente virtual é uma tarefa complexa de ser implementada, pois muitas características precisam ser desenvolvidas. Neste trabalho, propomos a utilização de diversas técnicas para modelar e controlar a interação dos elementos constituintes de uma simulação. Em particular, apresentamos um modelo baseado em Física para que os veículos evitem colisões, uma política de controle de acesso a cruzamentos baseada em reservas e um modelo de sincronização de semáforos utilizando rede de Petri. Os resultados apresentados demonstram que os modelos propostos reproduzem os comportamentos esperados para os contextos simulados.

PALAVRAS-CHAVE: Engenharia de tráfego; controle semafórico; computação gráfica; redes de Petri.

ABSTRACT

Simulate urban traffic realistically in a virtual environment is a complex task to implement, since many characteristics need to be developed. In this paper, we propose the use of different techniques to model and control the interaction of the elements of a simulation. In particular, we present a model physics-based for the vehicles to collisions avoid, an access control policy based on reserves and a model of synchronization of traffic signals using Petri net. The results show that the proposed model reproduces the behavior expected for the simulated contexts.

KEYWORDS: traffic engineering, traffic signals control, computer graphics, Petri nets.

1. Introdução

O estudo do tráfego de veículos em grandes centros urbanos é uma área de pesquisa de extrema importância, uma vez que os congestionamentos nas vias têm se tornado uma grande preocupação para a gestão pública, por se tratar de um dos principais fatores da diminuição na produtividade e na qualidade de vida nas cidades. Neste contexto, simuladores computacionais estão sendo intensivamente empregados por engenheiros para auxiliar no planejamento dos sistemas de tráfego.

Neste contexto, o objetivo deste trabalho é a proposição de modelos computacionais para o desenvolvimento de um micro simulador de trânsito, que permita simular padrões de comportamento de motoristas e de visualizar diversos fenômenos relacionados. Para tanto, apresentamos um modelo baseado em Física para que os veículos evitem colisões e uma política de controle de acesso a cruzamentos baseada em reservas. Também é proposta uma solução para um problema comum referente ao tempo na sincronização dos semáforos: erros cumulativos de temporização que podem ocasionar congestionamentos. Para este problema, é proposta a utilização de redes de Petri [6][7] para modelar o controle dos semáforos, uma vez que esta ferramenta garante concorrência e sincronização dos sistemas modelados.

Este trabalho apresenta a seguinte organização. Na Seção 2 definimos os elementos chave que compõem um simulador e descrevemos as principais características para modelá-los, além de oferecer uma revisão das principais técnicas existentes para esta finalidade. A Seção 3 apresenta um modelo baseado em Física para evitar colisões entre veículos, um modelo de controle de acessos a cruzamentos baseado em reservas e um modelo de sincronização de semáforos utilizando redes de Petri. Por fim, na Seção 4 encontram-se as conclusões e as perspectivas para trabalhos futuros.

2. Fundamentação e trabalhos relacionados

Nesta seção abordamos os principais elementos de um sistema para simulação de trânsito, além de uma revisão de alguns trabalhos relacionados à área de conhecimento.

2.1 Simuladores de trânsito

Basicamente, há três tipos de modelos de simulação de trânsito possíveis de serem implementados. Os **modelos microscópicos** simulam as características de deslocamento dos veículos, as interações baseadas no método "*car-following*" e no método de troca de via. Estes modelos são eficientes para validar condições de tráfego muito congestionadas, complexas configurações espaciais das vias e conseqüentes impactos das alterações na estrutura de transporte. Os **modelos macroscópicos** simulam fluxos de tráfego levando em consideração as características de um conjunto de indivíduos (por exemplo, a velocidade, o fluxo – direção e sentido – e a densidade). São utilizados equacionamentos para definir a conservação de fluxo e como possíveis distúrbios são propagados como "ondas" no sistema. Por fim, os **modelos mesoscópicos** combinam propriedades de ambos os modelos citados anteriormente. Tais modelos são menos consistentes que as ferramentas de micro simulação, mas são superiores que as técnicas restantes de análise de tráfego.

A definição de qual modelo adotar dependerá da aplicação. Em geral, a utilização de um modelo diferente do modelo microscópico implica em perder a simulação de diferentes padrões de comportamento de motoristas, um aspecto importante na análise deste tipo de sistema. Assim sendo, no presente trabalho, os modelos propostos são microscópicos, permitindo simular e analisar os padrões de comportamento característicos no tráfego urbano.

Por fim, os elementos básicos de qualquer simulador são as **vias**, que são segmentos de ruas onde os carros compartilham propriedades como velocidade permitida, sentido, direção de fluxo e comprimento, os **cruzamentos**, local onde três ou mais vias se encontram, compartilhando, da mesma forma que as vias, as propriedades dos carros, e os **veículos**, que podem ser modelados como agentes ou como simples estruturas de dados, dependendo do tipo de projeto de simulador.

2.1.1 Vias

Alguns simuladores se baseiam no modelo de autômato celular de Nagel-Schreckenberg [2], cuja proposta é a divisão das vias em células, onde o segmento pode ser ocupado por apenas um carro. Isto restringe as possibilidades de

simulação como, por exemplo, o caso de colisões de veículos, além de diminuir a densidade veícular na via e aumentar o controle sobre os agentes. Já outros projetistas preferem apenas relacionar um conjunto de veículos, sem restringir suas possíveis ocupações na via.



Figura 2-1 – Representação de via no modelo de Nagel-Schreckenberg [2].

Em geral, as vias são utilizadas como meio de comunicação de informações entre os veículos, como a velocidade e a posição. Uma dessas formas de comunicação é a utilização do conceito de *blackboard* [3]. Nesta abordagem, a via funcionaria como o *blackboard* para os agentes. Outra forma de troca de informações é através do conceito de feromônios, como utilizado no simulador SuRJE, de Joanne Penner [4].

Na técnica de *blackboard*, os agentes não precisam se comunicar diretamente. A interação entre agentes é realizada por meio de um agente intermediador, que oferece tanto transparência de localização como de tempo para os agentes. Desta maneira, a transparência de tempo é disponibilizada para um agente, permitindo que o mesmo possa receber um dado independente do momento da sua geração.

No simulador SuRJE, toda a comunicação entre os agentes é feita através da liberação de feromônios na via que são identificados pelos carros contidos no caminho. Com o passar do tempo, os feromônios perdem a qualidade de serem percebidos. Alem disso, como a velocidade dos carros varia, a distância entre os feromônios também varia. Portanto, carros com alta velocidade liberam feromônios mais distanciados uns dos outros, enquanto carros com baixa velocidade liberam feromônios próximos. Carros com velocidade igual a zero ou com alta desaceleração precisam liberar feromônios extras, de modo a indicar estes comportamentos. Muitas vezes a técnica de feromônios é comparada com as marcas deixadas pelos pneus dos carros.



Figura 2-2 – Carros liberando feromônios em via do simulador SuRJE [4].

É importante ressaltar que tanto a técnica de *blackboard* como a técnica de feromônios conseguem transmitir informação com custos computacionais relativamente baixos.

2.1.2 Cruzamentos

Em situações reais, os cruzamentos são os locais com maior risco de acidentes, por serem disputados como "recursos" pelos motoristas. Conseqüentemente, impedir que colisões ocorram nestas áreas não é uma tarefa fácil.

Existem diversas maneiras de controlar a utilização dos cruzamentos. Uma das formas mais simples é através da técnica de *overpass*. Esta técnica simula os carros como fantasmas, ou seja, um carro pode ocupar o mesmo espaço de outro carro. Uma alternativa é controlar o cruzamento através de uma rotatória, ou de um método similar, tal como o implementado no simulador SimUrban, projetado por Bem Woodhouse [5]. Neste simulador, se um cruzamento possuir mais de um veículo, aquele que chegar primeiro obtém a prioridade de uso, enquanto os demais serão orientados a parar.

Além das técnicas abordadas, também é possível controlar o uso dos cruzamentos através de semáforos ou, na ausência destes, através da prioridade de uso, como descrito nas regras de trânsito, ou através de um sistema de reservas, como foi proposto por Kurt Dresner e Peter Stone [1].

O método proposto por Kurt Dresner e Peter Stone divide os cruzamentos em uma matriz de quadrantes *n x n*, sendo *n* um número inteiro par conhecido como granularidade. Em tal método, todo carro é controlado por um agente que requisita e recebe espaços necessários para atravessar a intersecção, de acordo com um tempo preestabelecido para realização deste movimento. Na realidade, inúmeros

carros podem atravessar simultaneamente o cruzamento, devido a sua subdivisão em quadrantes. Portanto, somente os quadrantes necessários à travessia de um determinado carro serão reservados a ele.

2.1.3 Veículos

A implementação do comportamento dos agentes pode ser uma tarefa extremamente complexa dependendo, principalmente, das características desejadas para um sistema que visa retratar o comportamento de um ambiente dinâmico. Buscando solucionar tais dificuldades, alguns projetistas preferem dividir o controle do veículo entre o motorista (agente) e um módulo central de controle. Por exemplo, no simulador de Praveen Paruchuri [9], os agentes do simulador de trânsito tentam evitar situações de *deadlock* [10][11] como a apresentada na Figura 2-3. No entanto, sempre que estas ocorrerem, um controle central intervém para solucionar a questão.

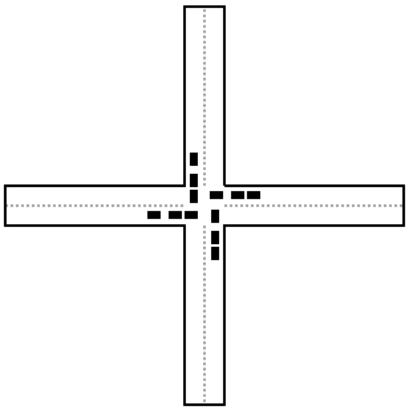


Figura 2-3 – Situação típica de *deadlock* [10].

Todo motorista possui um tempo de resposta aos eventos que o cercam. Da mesma forma que a realidade, os simuladores inserem tal característica. A diferença entre o instante inicial do evento e a reação do mesmo é conhecida por “tempo reação”. No geral, um motorista toma suas decisões baseado na interação de seu veículo com os demais. Após o agente possuir informações básicas como posição e velocidade, ele decidirá qual atitude assumir para que seus objetivos possam ser atingidos e suas regras respeitadas. Por exemplo, dada a seguinte situação: “todos os motoristas possuem como principal regra evitar colisões”. Neste contexto, se um motorista possui como objetivo específico andar a uma velocidade de 40 km/h, este

ultrapassará o carro a sua frente, caso o mesmo esteja andando a uma velocidade de 30 km/h. Caso não possa realizar tal manobra, aquele com maior velocidade deverá reduzir a sua velocidade, de modo a se locomover a 30 km/h, garantindo que a distância entre os veículos seja suficiente para evitar colisões.

Um motorista sempre tomará atitudes para evitar colisões com o carro a sua frente. No entanto, ele nunca agirá para evitar que o carro anterior colida com ele, pois esta, por princípio, não é sua responsabilidade. Conseqüentemente, em um simulador simples, um agente não precisará ter informações sobre os carros anteriores a ele. Obviamente, esta regra não será aplicada caso queiramos implementar comportamentos mais complexos como, por exemplo, o caso das ultrapassagens. Um motorista pode sinalizar a sua intenção em ultrapassar em um ambiente com tráfego intenso, sinalizando com os faróis para o carro a sua frente. O mesmo caso seria importante para implantação de ambulâncias no simulador, na qual a sinalização seria realizada com o uso de sirenes.

Para evitarmos colisões, existem duas possíveis formas: desacelerar ou mudar de direção. Em geral, os simuladores apresentam apenas o método de desaceleração, devido à alta complexidade de implementação do método de mudança de direção. Neste trabalho, utilizaremos apenas o método de desaceleração para evitar colisões. Além disso, os agentes são modelados com dois tipos de objetivos: micros e macros. O macro é composto pelo destino e pelo caminho para atingir tal destino. O objetivo micro envolve a decisão a cada instante no tempo para que o objetivo macro possa ser alcançado.

3. Metodologia e implementação

Como já apresentado na seção anterior, é possível encontrar três tipos de elementos em um simulador de trânsito: as vias, os cruzamentos e os veículos (agentes). Para validar os modelos propostos e/ou implementados, foi desenvolvido um micro simulador. Para facilitar a sua implementação, simplificações foram realizadas em alguns tipos de elementos como, por exemplo, impossibilitar que os agentes façam ultrapassagens.

3.1 Vias

O modelo proposto utiliza uma lista de agentes associada à via, ordenados de forma decrescente de posicionamento para auxiliar no controle. Foi utilizado o conceito de *blackboard* para comunicação entre agentes, onde as vias disponibilizam aos motoristas informações como posição e velocidade do carro a sua frente, estado do semáforo, se a rua é preferencial ou secundária e se há a necessidade de dobrar para atingir o próximo destino.

Definiu-se uma área ao final da via (realçada em amarelo como na Figura 3-1), sendo o seu tamanho dependente da velocidade nela permitida. Esta área em destaque representa o segmento que um condutor parado no cruzamento enxerga da via perpendicular à via aonde se encontra.



Figura 3-1 – Imagem de um cruzamento com acesso controlado sem uso de semáforos.

3.1.1 Controle de atualização

O controle de atualização deve garantir que os agentes sejam supridos de toda a informação necessária para governarem seu comportamento. Para isto, muitas vezes este controle precisa buscar informações em outras estruturas. Na Figura 3-2, tanto o veículo identificado pela seta 6 quanto o identificado pela seta 7 precisam de informações básicas, tais como, posição e velocidade do carro a sua frente. Supondo que o motorista deseja dobrar a direita no cruzamento indicado pela

seta 5, a via (*backbone*) deve ser capaz de requisitar informações sobre a existência de algum veículo no caminho entre o identificado com a seta número 7 e o fim da via indicada pela seta 4.

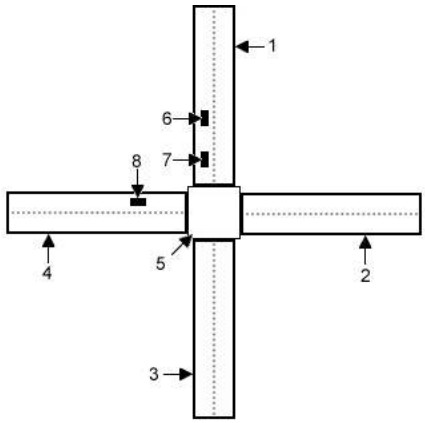


Figura 3-2 – Ilustração de intersecção [10].

Para melhorar o desempenho, o *backbone* deve coletar apenas as informações que serão efetivamente utilizadas. O caso acima da busca por informações para o veículo com seta número 7 é um bom exemplo disto. Se o *backbone* não soubesse da sua intenção em ir para a via número 4, seriam requisitadas informações de outras duas vias que jamais seriam utilizadas. Para sanar esse problema, toda vez que um agente entrar numa via, esta requisita qual a próxima via do destino do agente; caso ele não tenha uma escolha específica, um destino será sorteado. De tempos em tempos uma nova decisão deve ser tomada por parte do agente. Esta diferença de tempo, citada na seção anterior por “tempo de reação”, é administrada pelo controle de atualização da via.

Como no mundo real, existem comportamentos esperados para cada tipo de via. Naquelas em que o acesso aos cruzamentos é controlado por semáforos, espera-se que os veículos parem, caso o sinal esteja vermelho. No entanto, uma alteração no comportamento esperado das vias sem semáforos foi realizada para facilitar a implementação do controle de utilização de cruzamentos. Todo veículo na preferencial que deseja realizar o movimento de dobrar a esquerda, mas que não pode fazê-lo de imediato por existir outros veículos no sentido contrário, deverá esperar no fim da via, sem entrar no cruzamento, até conseguir um espaço e tempo suficientes para realizar a manobra.

3.2 Cruzamentos

Nesta seção são apresentadas técnicas para modelar cruzamentos com e sem semáforos.

3.2.1 Cruzamentos com semáforos

Atualmente, a utilização de semáforos é uma forma eficiente e de baixo custo para controlar cruzamentos com alto fluxo de carros. Na Figura 3-3 é apresentado um instante no tempo da execução do simulador desenvolvido. Nessa imagem vários veículos atravessam o cruzamento durante o sinal verde, enquanto alguns são obrigados a permanecerem parados, aguardando o término do sinal vermelho.

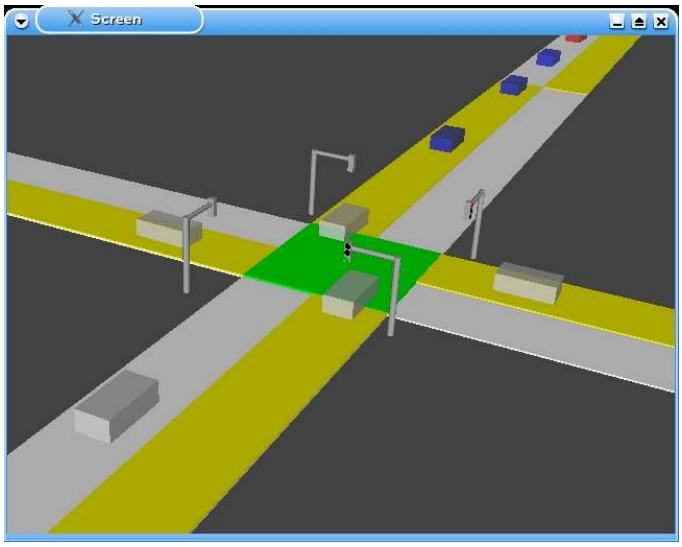


Figura 3-3 – Imagem de cruzamento com acesso controlado por semáforos.

No decorrer desta seção, apresentaremos a utilização de redes de Petri temporizadas [6][7][8] para modelar e controlar semáforos de maneira realista e precisa. Devido à capacidade desta ferramenta em proporcionar sincronização, facilmente garantiremos que erros dos sistemas de temporização não se acumulem, evitando efeitos não desejados como congestionamentos.

3.2.1.1 Modelo semáforo simples

Na Figura 3-4, apresentamos um modelo simples de semáforo com três estados: verde, amarelo e vermelho. Essa representação semafórica é composta

por três lugares com tempos associados e três transições, nomeados através da seguinte sintaxe:

- Lugares: a primeira letra, “p”, é proveniente da palavra “lugar”, em inglês; a segunda representa a primeira letra dos possíveis estados escritos em inglês (*green*, *yellow* e *red*); o restante da sintaxe indica o número do semáforo.

- Transições: a primeira letra, “t”, provém da palavra transição, em inglês; a segunda representa a primeira letra dos possíveis estados (em inglês, conforme os acima citados) do lugar de entrada. A terceira representa a primeira letra dos possíveis estados, em inglês, do lugar de saída da transição. O restante da sintaxe indica o número do semáforo.

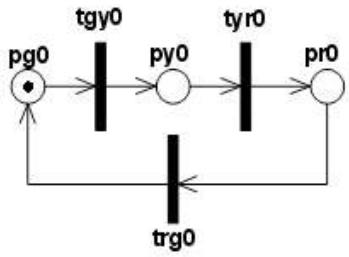


Figura 3-4 – Modelo de semáforo com três estados: verde, amarelo e vermelho, estando inicialmente no estado verde.

O estado do semáforo é indicado pela existência de uma marcação (*token*) em um dos lugares do modelo semafórico. Sempre que tiver transcorrido o tempo relacionado ao lugar com a marcação, ocorrerá a troca de estado.

3.2.1.2 Modelo de sincronização de semáforos no cruzamento

Há o interesse em modelos de sincronização mais complexos do que os apresentados até o momento. Portanto, na Figura 3-6, apresentamos um exemplo de uma modelagem capaz de sincronizar semáforos de um cruzamento compostos de “n” semáforos e no máximo “n” tempos. Entenda-se por quantidade de tempos, o número de alocações do cruzamento feito aos semáforos em um ciclo e, por um tempo, a alocação do cruzamento a um ou mais semáforos. Exemplificando: se um cruzamento é composto por três vias de entrada e três vias de saída como na Figura 3-5, e desejamos que, durante o ciclo semafórico, cada via de entrada permaneça

com total direito de manobra sobre o cruzamento e com 1/3 do período do ciclo, teremos um cruzamento com três tempos. No entanto, se as vias de entrada, representadas na horizontal na Figura 3-5, receberem, simultaneamente, o direito de uso do cruzamento, o cruzamento passa a ser de dois tempos.

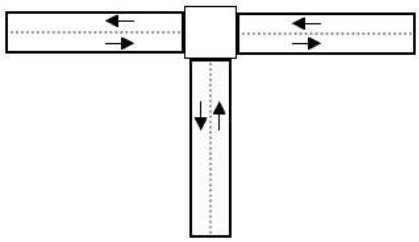


Figura 3-5 – Cruzamento com três vias de entrada e três vias de saída.

O exemplo da Figura 3-6 sincroniza quatro semáforos, numerados de zero a três, de um cruzamento com dois tempos, numerados de zero a um. No modelo, os tempos são vistos como recursos seqüenciais, os quais são compostos cada um por um lugar e duas transições. A ligação de cada semáforo com seu respectivo tempo de cruzamento é feito por um lugar nomeado com o prefixo “po”, seguido do número do semáforo, da letra r de *recurse* e do número do recurso. Um recurso “x” sempre se liga aos cruzamentos membros do recurso “x+1”, com exceção do último recurso, que se liga com o recurso zero através de um lugar nomeado com o prefixo “po” seguido do número do semáforo, da letra r de *recurse* e do número “x”.

3.2.1.3 Modelo para criação de onda verde

Segundo [9], “onda verde” é um fenômeno intencionalmente induzido, no qual uma série de semáforos (normalmente três ou mais) é coordenada a permitir um fluxo de tráfego contínuo através de várias intersecções em uma direção principal. Na Figura 3-7, apresentamos um exemplo de implementação de sincronização de onda verde com três semáforos numerados de zero a dois, sendo que o semáforo de número 0 (zero) está mais próximo que os demais do início da rua. Para criarmos uma onda-verde em “n” semáforos, são necessários 2*(n-1) lugares temporizados.

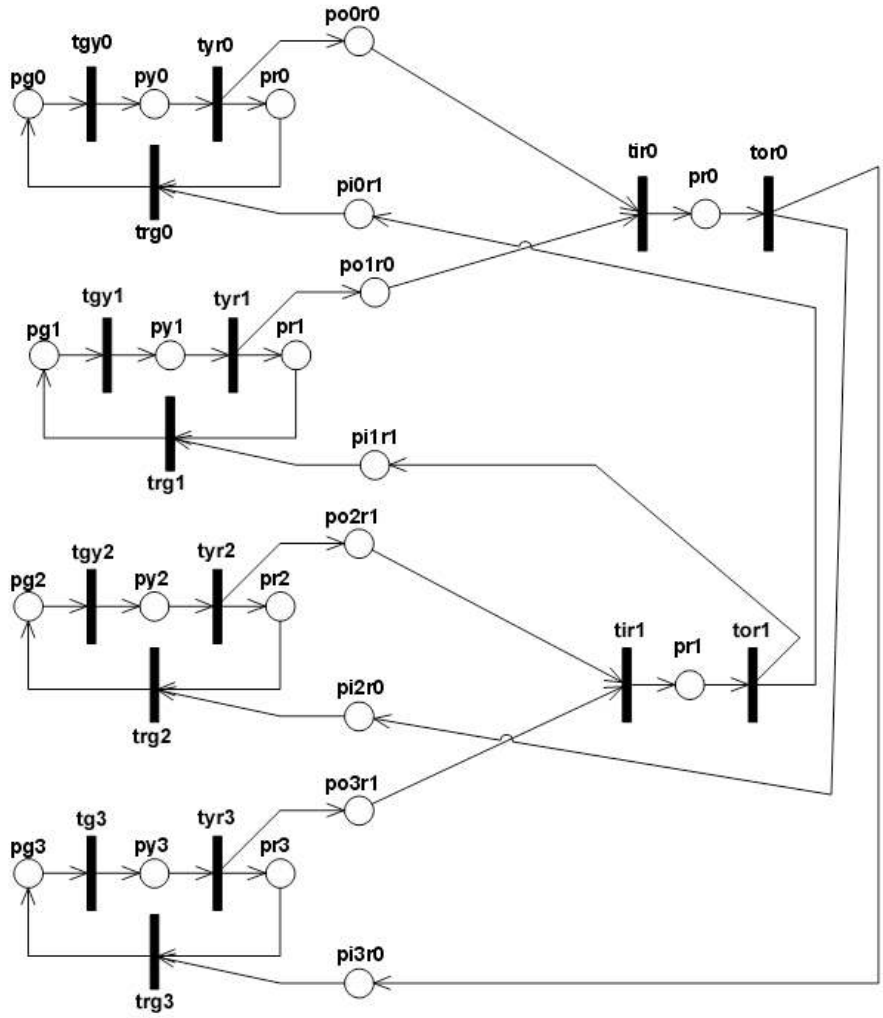


Figura 3-6 – Exemplo de sincronização de cruzamento com dois tempos e quatro semáforos.

3.2.2 Cruzamentos sem semáforos

O modelo proposto divide os cruzamentos sem semáforos em quatro quadrantes, numerados como na Figura 3-8. O controle de utilização de quadrantes é feito através de um sistema de reservas de quadrantes similar ao desenvolvido por Kurt Dresner e Peter Stone [1]. Diferentemente do proposto por eles, as reservas foram modeladas com níveis de prioridade, sendo renovadas com inicio tardio ao transcorrer um passo no tempo. Desta maneira, tal controle se comporta como um guarda de trânsito, que sinaliza o melhor momento de entrada no cruzamento. Na Tabela 3-1 consta a listagem dos níveis de prioridade das reservas.

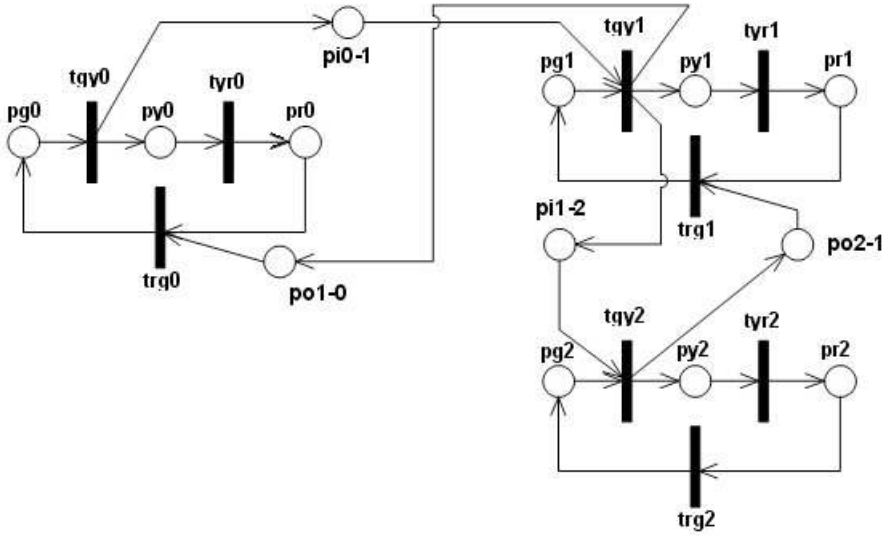


Figura 3-7 – Exemplo de sincronização de onda verde entre três semáforos de cruzamentos diferentes.

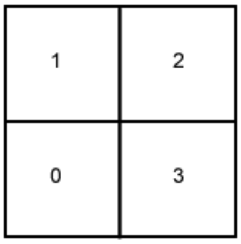


Figura 3-8 – Quadrantes numerados de um cruzamento.

	Prioridade (via preferencial)	Prioridade (via secundária)
Conversão à direita	1	3
Seguir em frente	1	3
Conversão à esquerda	2	4

Tabela 3-1 – Tabela indicativa das prioridades para os tipos de utilização dos cruzamentos.

3.2.2.1 Controle de intersecção baseado em reservas

Esse controle de intersecção permite aos agentes requisitar o uso de determinados quadrantes do cruzamento durante um período de tempo. Como somente é possível alocar um instante de um quadrante a um único agente, reservas com menor prioridade têm seu início postergado. A requisição é criada estimando-se o tempo de chegada e de permanência, no cruzamento, do agente que se enquadra em uma das seguintes opções:

- Estar em uma via preferencial, e estar com seu centro de massa sobre a área final da via destacada em amarelo como na Figura 3-1;

- Estar em uma via secundária, ser o primeiro carro da via e já ter parado próximo ao cruzamento.

Para os veículos que já contemplaram tais pré-requisitos, seu algoritmo de decisão não mais receberá o valor “-1” no parâmetro indicador de tempo ideal de entrada no cruzamento. Com isso, todo agente que recebe um valor maior ou igual a zero em tal parâmetro deve tomar providências para respeitar o instante de entrada no cruzamento. As principais ações possíveis são:

- Parar o veículo para posteriormente entrar no cruzamento, quando o tempo de entrada for muito elevado;

- Diminuir a velocidade e entrar no cruzamento para tempos de entrada superiores à estimativa atual de entrada do agente;

- Andar livremente para tempos de entrada inferiores ou iguais à estimativa atual de entrada do agente.

O controle reúne todas as requisições de uso dos quadrantes em uma lista e então executa o algoritmo de renovação de reservas. Na Figura 3-9, é possível visualizar o fluxograma deste algoritmo. Algumas melhorias foram implantadas de forma a otimizar o desempenho de utilização dos cruzamentos, a saber, sempre que um veículo abandona um quadrante, o tempo restante da reserva não será mais necessário e o controle cancela o restante da reserva daquele quadrante pertencente ao veículo.

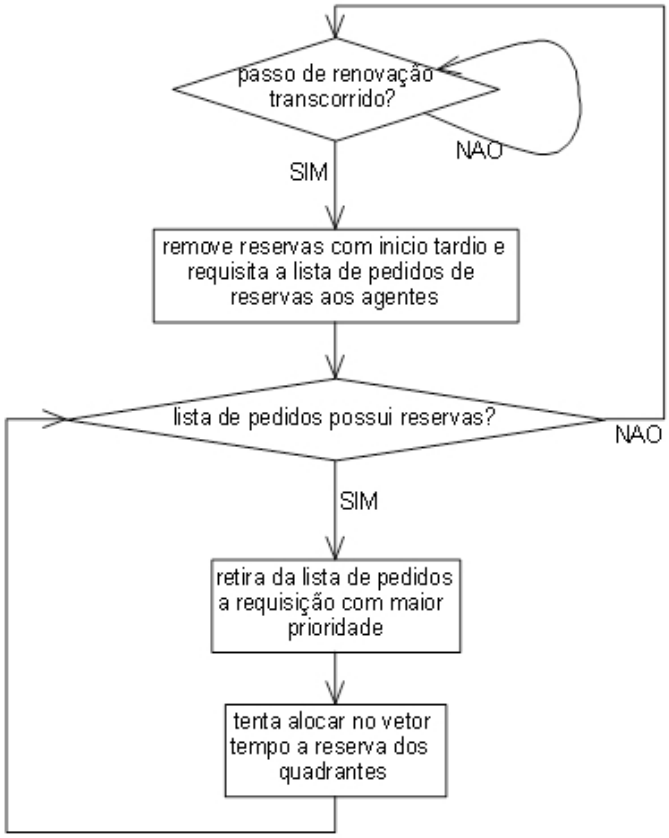


Figura 3-9 – Fluxograma simplificado do algoritmo de renovação de reservas.

3.3 Agentes

Optou-se por implementar o comportamento ideal de um motorista, para após validar expansões de comportamento anômalos dos seres humanos. A Figura 3-10, a seguir, apresenta o fluxograma do algoritmo de controle de aceleração, responsável por impedir o veículo de colidir com o carro à frente.

3.3.1 Calculo de posição ideal

Deseja-se calcular a posição ideal, Pi, que garante que o carro1 nunca colidirá com o carro2. Para calculá-la, adotou-se alguns conceitos físicos, tais como: função horária e função velocidade, pertencentes ao Movimento Retilíneo Uniformemente Variado (MRUV). Portanto, tem-se:

$$s = s_o + v_o t + \frac{at^2}{2}, \text{ a função horária e} \qquad (1)$$

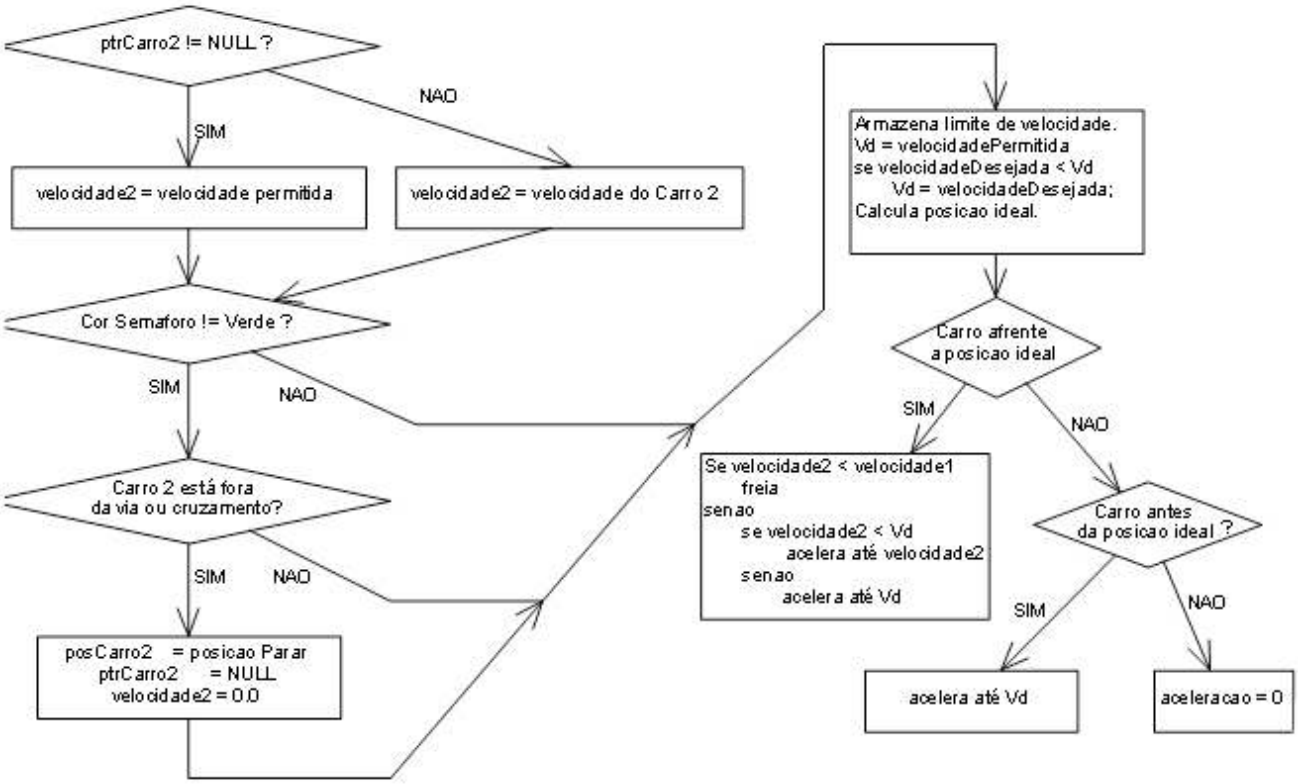


Figura 3-10 – Fluxograma do algoritmo de controle de aceleração.

$v = v_o + at$, a função velocidade. (2)

Na Figura 3-11, é possível ver uma situação hipotética, onde se deseja calcular a posição ideal de um veículo.

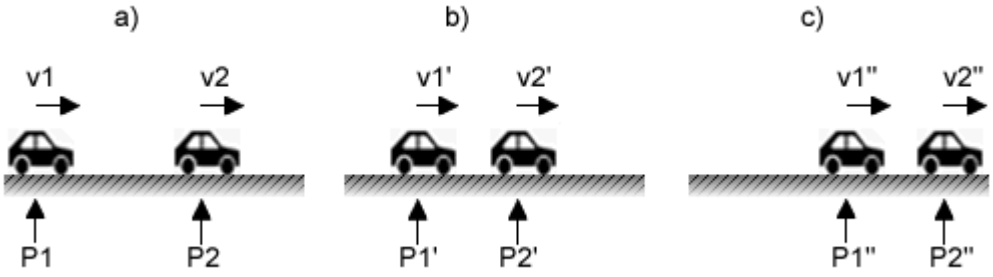


Figura 3-11 – Situação hipotética para cálculo da posição ideal. Em (a) temos a situação atual (carro2 começa a frear); em (b) temos o instante em que o carro1 percebe a frenagem do carro2; em (c) vemos a situação desejada, na qual as velocidades v1'' e v2'' são iguais a zero e a distância P2''-P1'' é igual a distância segura.

Na Figura 3-11-a, temos dois veículos nomeados carro1 e carro2 com velocidades conhecidas, respectivamente v1 e v2, e com posições igualmente conhecidas, respectivamente P1 e P2. A Figura 3-11-b mostra o instante em que o carro1 percebe a frenagem do carro2. A diferença de tempo entre os momentos de (a) para (b) na figura é igual ao tempo de reação, Δt, do carro1. A Figura 3-11-c apresenta os mesmos veículos com velocidade v1'' e v2'' iguais a zero.

Deve-se estipular que P2'' menos P1'' é igual a uma distância preestabelecida chamada "distância segura" ou ΔS, e que o carro1 somente irá frear após ter identificado a frenagem do carro2, ou seja, após ter transcorrido o tempo de reação, Δt, para o carro1.

$$P2'' - P1'' = \Delta S \tag{3}$$

Para calcular o valor de Pi, utiliza-se como hipótese o pior caso para garantir que a colisão não ocorrera. O pior caso ocorre quando o carro2 resolve parar e o carro1 decide acelerar o máximo possível. Nessas condições, o primeiro passo é identificar a aceleração máxima possível de ser aplicada no instante atual para conhecer o maior valor de v1'. Encontra-se o valor dessa aceleração por meio da equação:

$$a_{\max} = \frac{(v_d - v1)}{\Delta t}, \tag{4}$$

sendo v_d o menor valor entre dois possíveis, a velocidade permitida no nodo (via ou cruzamento) e a velocidade desejada do agente, e Δt o tempo de reação do carro1. Ainda é necessário garantir que a_{max} não seja maior que a aceleração máxima positiva do carro1. Conhecendo-se o valor de a_{max}, calcula-se v1' através da equação abaixo.

$$v1' = v1 + a_{\max}\Delta t \tag{5}$$

Com as informações acima é possível afirmar que a_{mn}, aceleração máxima negativa do veículo, pode ser calculada em:

$$v1'' = v1' + a_{mn}t_1, \tag{6}$$

e como $v1'' = 0$,

$$t_1 = \frac{-v1'}{a_{mn}}. \quad (7)$$

Temos então que:

$$P1' = Pi + v1\Delta t + \frac{a_{\max}\Delta t^2}{2} \text{ e} \quad (8)$$

$$P1'' = P1' + v1' t_1 + \frac{a_{mn} t_1^2}{2}. \quad (9)$$

Conhecendo P1' e t_1 substitui-se as equações (7) e (8) em (9), obtendo:

$$P1'' = Pi + v1\Delta t + \frac{a_{\max}\Delta t^2}{2} - \frac{(v1')^2}{2a_{mn}}. \quad (10)$$

Agora se calcula P2'' da seguinte maneira:

$$P2'' = P2 + v2t_2 + \frac{a_{mn} t_2^2}{2}. \quad (11)$$

Sabe-se que $v2'' = 0$, então em

$$v2'' = v2 + a_{mn} t_2 \quad (12)$$

é possível isolar t_2 e a equação (12) torna-se:

$$t_2 = \frac{-v2}{a_{mn}}. \quad (13)$$

Substituindo (13) em (11), encontra-se P2'' independente do valor de t_2:

$$P2'' = P2 - \frac{v2^2}{2a_{mn}}, \quad (14)$$

e finalmente obtêm-se o valor de Pi, substituindo (10) e (14) em (3):

$$Pi = P2 + \frac{[(v1')^2 - v2^2]}{2a_{mn}} - \Delta S - v1\Delta t - \frac{a_{\max}\Delta t^2}{2} \qquad (15)$$

Na Figura 3-12 é apresentado um gráfico, distância *versus* tempo, no qual as marcações em verde, azul e vermelho representam, respectivamente, a posição do centro de massa do primeiro carro na via, a posição do centro de massa do segundo carro e a distância entre estes carros. Cabe salientar que a ordem dos carros leva em consideração a proximidade dos mesmos com o cruzamento localizado ao final da via. Os carros estão inseridos em uma via secundária de comprimento igual a 100m. Próximo ao término dessa distância, o carro deverá frear para entrar no cruzamento. Sempre que um carro chegar ao final do trajeto, a análise gráfica prossegue com os demais carros que permanecerem na via. Por isto, no instante 15s, o primeiro carro está na iminência de entrar no cruzamento. Por conseqüência, o segundo carro passa a ser o primeiro a ser analisado e o terceiro passa a ser o segundo carro. As marcações em vermelho mostram a diferença entre os dois primeiros carros na via. Nesta análise gráfica, os agentes foram configurados para a distância segura (ΔS) ser de 12m. Comparando com os resultados obtidos nas marcações vermelhas, podemos verificar que o algoritmo de controle de desaceleração atingiu seu objetivo.

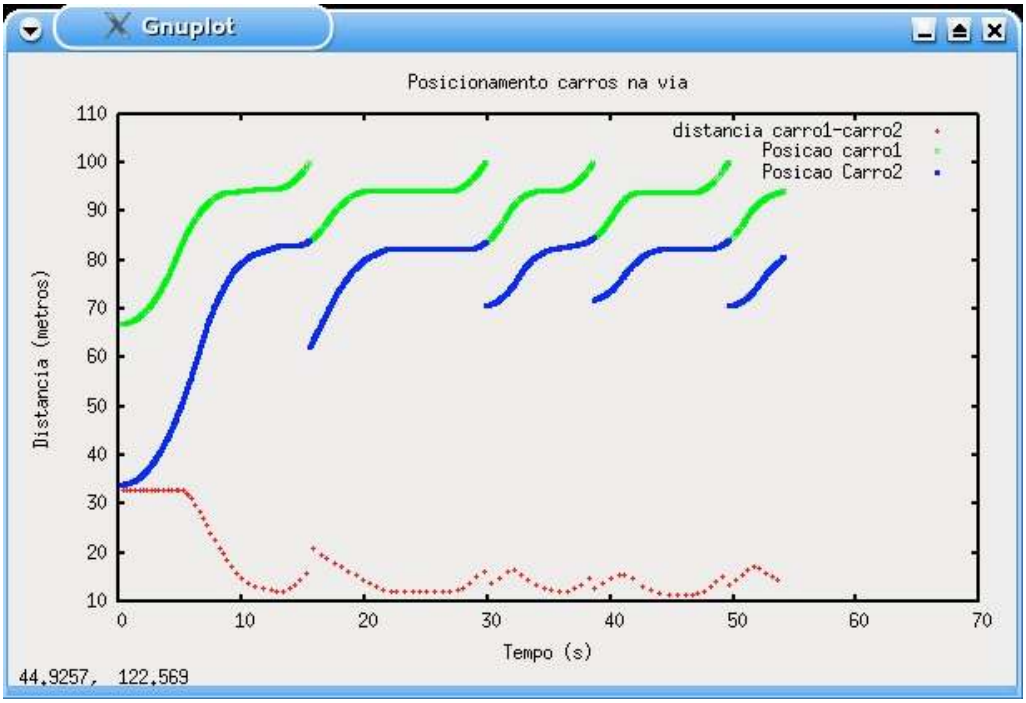


Figura 3-12 – Gráfico de análise do algoritmo de aceleração.

Todo agente possui as seguintes regras e objetivos, que devem ser respeitadas: não colidir, andar o mais próximo da velocidade desejada, não ultrapassar a velocidade máxima permitida do nodo em que o carro está inserido e respeitar as regras de trânsito.

No simulador, foi adotada uma representação gráfica para o comportamento do agente, onde a cor do veículo pode assumir: vermelha, para aceleração negativa, branco, para aceleração nula, e azul, para aceleração positiva. Essa representação facilita a compreensão e a validação dos comportamentos reproduzidos.

O controle de tempo de reação está implementado externamente ao agente, no controle de atualização dos nodos (vias e cruzamentos). Com isto, obteve-se um ganho de desempenho, por realizar apenas uma vez o cálculo de passagem de tempo de decisão por nodo, mas restringe os agentes a possuir o mesmo tempo de decisão.

4 Conclusões

O objetivo deste trabalho foi propor métodos computacionais que permitam o desenvolvimento de um micro simulador de trânsito que apresente mecanismos para sincronização de semáforos e reprodução de padrões de comportamento de motoristas.

Propôs-se um mecanismo de controle de veículos para evitar colisões baseado em física, uma política de controle de acesso a cruzamentos baseado em reservas e um modelo de sincronização de semáforos utilizando redes de Petri. Verificou-se a eficiência do algoritmo proposto para controlar a aceleração de veículos e a utilidade de redes de Petri como mecanismo de controle de semáforos, permitindo projetar modelos simples para sincronização e simulação da “onda verde”, além de resolver, também, o acúmulo de erros de temporização.

Um ponto a melhorar é a necessidade das ruas e dos cruzamentos apresentarem maior flexibilidade quanto à quantidade e à variedade dos sentidos nas vias, sendo possível propor, por exemplo, técnicas de troca de vias. Além disso, seria interessante oferecer aos agentes meios para realizar ultrapassagens, para

solucionar situações de *deadlocks* em cruzamentos e para realizar conversões à esquerda, reproduzindo, desta forma, as orientações das normas de trânsito.

5 Referências bibliográficas

[1] Kurt Dresner e Peter Stone. Multiagent Traffic Management: A Reservation-Based Intersection Control Mechanism. *Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 04)*, 2004, p. 530-537.

[2] Kai Nagel e Michael Schreckenberg. A cellular automaton model for freeway traffic. *Journal Physique I*, v. 2, n. 12, p. 2221-2229, 1992.

[3] H. Penny Nii. Blackboard Systems. *The Handbook of Artificial Intelligence*, v. 4. Editores: A. Barr, P. R. Cohen e E. A. Feigenbaum, Addison-Wesley, 1989.

[4] Joanne Penner, Ricardo Hoar and Christian Jacob. Swarm-based Traffic Simulation with Evolutionary Traffic Light Adaptation. *First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 02)*, 2002.

[5] Ben Woodhouse, SimUrban - Populating and Rendering an Urban Environment, April 2004.

[6] Tadao Murata. Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, v. 77, n. 4, p. 541-580, 1989.

[7] C. A. Petri. *Kommunikation mit Automaten*, Schriften des IIM Nr. 3, Bonn: Institute für Instrumentelle Mathematik. 1962.

[8] A. L. Bicho, A. B. Raposo e L. P. Magalhães, Control of Articulated Figures Animations Using Petri Nets. *XIV Brazilian Symposium on Computer Graphics and Image Processing*, 2001, p. 200-207.

[9] Green Wave. In Wikipedia, the free encyclopedia. Acessado em Janeiro 2006.

[10] Praveen Paruchuri, Alok Reddy Pullalarevu and Kamalakar Karlapalem. Multi Agent Simulation of Unorganized Traffic. *First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 02)*, 2002, p. 176-183.

[11] Chia-Shiang Shih e John A. Stankovic, Survey of Deadlock Detection in Distributed Concurrent Programming Environments and Its Application to Real-time Systems. COINS Technical Report 90-69. University of Massachusetts at Amherst, 1990.