

# **Aplicação de algoritmos de escalonamento de processos como proposta de solução para resolução de conflitos em interseções de múltiplas vias de trânsito**

Thiago Lopes Trugillo da Silveira<sup>1</sup> (Apresentador) e Marcia Pasin<sup>1</sup> (Orientadora)

<sup>1</sup>Centro de Tecnologia – Laboratório de Sistemas de Computação

Universidade Federal de Santa Maria

Av. Roraima 1000 – Cidade Universitária – Santa Maria – Rio Grande do Sul - Brasil

## **1. Introdução**

Diariamente, a movimentação dos cidadãos em várias cidades brasileiras é intensa e, para isso, muitos destes se utilizam de veículos automotores como carros e motocicletas. O gerenciamento do trânsito destas cidades está se tornando cada vez mais complexo visto que a frota vem aumentando consideravelmente nos últimos anos [1]. Comumente, em horários de pico, em ruas e avenidas com muitas intersecções, o trânsito praticamente não flui pois é necessário haver um modo de veículos que não estão em via preferencial, ingressar ou atravessar a mesma. Para a seleção de qual fluxo poderá trafegar em uma via, em determinado momento, pode-se usar semáforos.

A utilização de semáforos é, ainda, o mecanismo automático mais eficiente para sincronizar fluxos, realmente implantado em escala nas cidades brasileiras. Porém, os semáforos operam sobre janelas de tempo fixas para determinar o fluxo que terá acesso a uma via, e podem não ser muito efetivos visto que não levam em conta o contexto (fluxo de veículos) naquele momento. Por exemplo, um semáforo pode oferecer passagem a um único veículo que está em uma direção enquanto na outra direção há dezenas de veículos aguardando. Para contornar este problema, como contribuição, uma proposta de algoritmos de decisão sensíveis ao contexto é apresentada neste trabalho, visando um melhor escalonamento dos veículos. Neste trabalho, algoritmos de sincronização de processos são usados para sincronizar fluxos de veículos em alternativa ao uso de semáforos.

## **2. Objetivos**

Mais precisamente, este trabalho tem como objetivo identificar e avaliar potenciais algoritmos de escalonamento de processos como, por exemplo, *First In – First Out (FIFO)* e *Shortest Job First (SJF)* [2] e compará-los com os métodos tradicionais de controle de intersecção de vias como passagens livre (elevações que permitem veículos transpassar em uma intersecção sem paradas) e semáforos. A adaptação destes algoritmos para o controle de intersecções de vias é feita

naturalmente, visto que o propósito dos mesmos é, justamente, coordenar (escalonar) fluxos (de processos). A proposta deste trabalho é que exista um dispositivo gerenciador em cada interseção com um receptor de informações dos veículos que por ali trafegam, como ocorre em [3]. De posse dessas informações, este dispositivo aplica os algoritmos de escalonamento e decide qual veículo pode utilizar a via e quando. Para este sistema ser implantado na prática, todos veículos precisam ter instalado um dispositivo para possibilitar comunicação veicular, que envie as informações para o gerenciador de conflitos e receba a resposta pós-processamento (isto é, a informação que indica a ordem de quando o veículo deverá passar a interseção). Como o custo de implantação experimental é alto, neste trabalho foi utilizado um simulador de contexto para permitir analisar o impacto da aplicação destes algoritmos no cenário de redes veiculares.

### **3. Metodologia**

Como mencionado anteriormente, este trabalho propõe a aplicação de algoritmos de escalonamento de processos para a resolução de conflitos em interseção de vias de trânsito urbano. Para tornar possível a realização de experimentos para a avaliação de diferentes algoritmos, foi usado um *software* simulador. O simulador de contexto Siafu [4] foi tomado como base de desenvolvimento do ambiente para a construção do cenário e aplicação dos algoritmos de decisão e comunicação.

#### **3.1 O Siafu**

O Siafu é um simulador de contexto multiagente *Open Source* totalmente construído em linguagem Java. É possível, de acordo com a necessidade, modificá-lo visto que é um *software* livre. O simulador fornece uma interface de desenvolvimento permitindo moldar cenários, trajetos e agentes (veículos) – entidades com certo grau de autonomia – de forma que a simulação possa representar um ambiente real.

Para construir uma simulação no Siafu, é necessário estender três classes abstratas: *BaseContextModel*, *BaseWorldModel* e *BaseAgentModel*. Estas classes serão responsáveis por definir o modelo de comportamento do ambiente (como, por exemplo, intensidade de sinais de telefonia, variações de temperatura, entre outros), modelo de comportamento do cenário (como alterações de características de lugares pré-definidos) e modelo de comportamento dos agentes (como definição do que cada agente deve fazer, onde deve ir, entre outros) respectivamente.

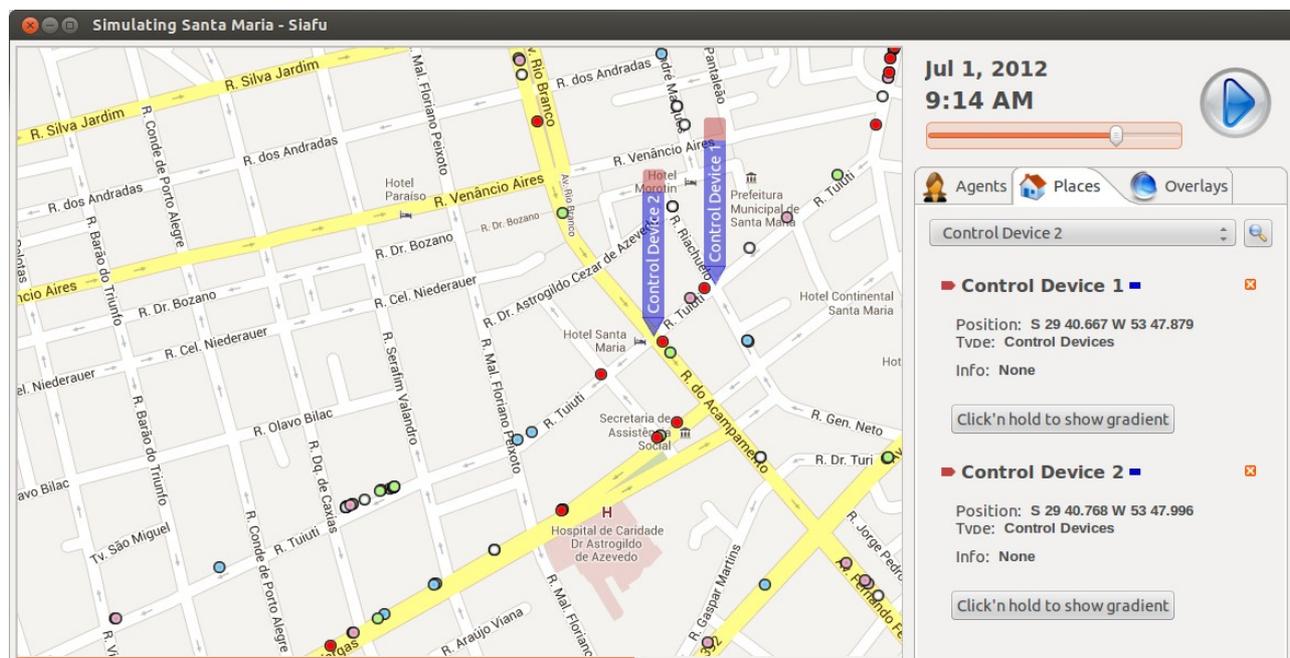
#### **3.2 Desenvolvimento da proposta**

Como o Siafu proporciona uma camada básica de controle de agentes, lugares e cenário, pode-se focar apenas no desenvolvimento da solução proposta. Assim sendo, a arquitetura deste trabalho foi

organizada tomando como base o que o Siafu implementa. O simulador de contexto Siafu oferece uma classe chamada *Place* que representa, em termos de *pixels*, um lugar que pode assumir algum tipo de comportamento. Aproveitando esta característica, neste trabalho foram criadas duas novas classes sobrepostas de *Place*: *Control Device* e *Any*. A classe *Control Device* é a entidade responsável por gerenciar a interseção de vias (no cenário atual, esta tarefa é feita manualmente ou por semáforos). *Any* descreve os pontos de partida e de chegada de uma rota de um veículo.

Além disto, o Siafu oferece uma classe chamada *Agent*. Esta classe representa todo o tipo de agente móvel de uma simulação. Para melhor organizar a simulação, foi adicionada a classe *Vehicle* que estende *Agent* e implementa particularidades dos veículos, bem como o suporte à comunicação através do *Control Device*. A classe *Vehicle* representa todo o tipo de veículo da simulação. Além destas três novas classes, foi implementada a classe *Route* que é definida por duas entidades do tipo *Any* – ou seja – por dois lugares. Uma instância da classe *Vehicle* pode obedecer uma rota pré-definida ou, alternativamente, deslocar-se livremente.

Na simulação, assim como acontece na realidade, podem existir diversas interseções com maior necessidade de controle. Neste trabalho, foi utilizado como cenário um conjunto de ruas da região central da cidade de Santa Maria – Rio Grande do Sul – e, a cada interseção onde há um semáforo, foi associado um *Control Device*. A Figura 1 apresenta a interface para visualização da simulação com duas áreas: mapa (lado esquerdo) e informações (lado direito).



**Figura 1. Interface da simulação apontando dois *Control Devices* em interseções reais da cidade de Santa Maria - RS**

Na Figura 1, o mapa indica dois *Control Devices* (1 e 2) que seriam instalados nas interseções das ruas Acampamento com Tuiuti, e Tuiuti com Riachuelo, respectivamente. Na simulação, os veículos são representados por círculos que trafegam pelas ruas. A área de informações mostra dados sobre ambos *Control Devices* na aba *Places*.

As instâncias da classe *Control Device* devem implementar um dos algoritmos de escalonamento existentes. A implementação do algoritmo *FIFO*, por exemplo, é simples: existe uma fila, onde o primeiro veículo que chegou é o primeiro a sair desta fila. Desta forma, todos os veículos sempre param nas interseções e são liberados a medida que os veículos que estavam a sua frente na fila também são liberados. O algoritmo *SJF* mantém duas filas e seleciona o primeiro elemento da fila que está mais próximo de chegar ao seu objetivo. Neste âmbito, pode-se considerar que o sistema do dispositivo gerenciador conhece onde cada veículo quer chegar (isto é, na simulação a matriz origem-destino ou matriz OD é conhecida) ou, ainda, considera-se algum tipo de premissa como, por exemplo, “o veículo que está a mais tempo circulando deve ser o que está mais próximo do seu objetivo”. De qualquer forma, estes dados devem ser enviados pelos dispositivos de comunicação presentes nos veículos para o dispositivo de gerenciamento de interseções.

#### **4. Resultados e discussões**

Testes com diversos *Control Devices*, implementando diferentes algoritmos de escalonamento, devem ser feitos de para que se possa avaliar o verdadeiro impacto da aplicação destes em contraste com algoritmos de escalonamento que utilizam janelas de tempo – como os semáforos tradicionais. Em escalonamento de processos, os algoritmos *FIFO* e *SJF* mostram-se interessantes justamente por levar em consideração o contexto (quantidade de veículos aguardando em uma interseção).

No cenário da resolução de conflitos em interseções, com, por exemplo, o *First In-First Out*, se não houverem veículos em um fluxo de uma via, o fluxo contrário estará sempre com a preferência da mesma – sem que ocorram paradas desnecessárias. É visto, também, que não há problemas como espera indefinida (*starvation*) [2], por exemplo, onde um veículo – neste caso – ficaria esperando por tempo indeterminado. Adaptações poderiam ser propostas, como, por exemplo, ao invés de controlar a passagem de um veículo por vez, poderiam passar de N em N. Neste caso, existe a necessidade de um estudo mais aprofundado sobre qual seria o tamanho ideal das filas dado determinado fluxo e, possivelmente, seria interessante a avaliação da necessidade de propor ajuste dinâmico para controlar passagem de veículos.

## 5. Conclusões

O sistema de controle do fluxo do trânsito – através de semáforos – atualmente amplamente adotado no Brasil oferece, ainda, uma solução razoável. Porém, o número de veículos em atividade tem crescido bastante e, então, a aplicação de outras técnicas de escalonamento, que não a por janelas de tempo fixas, devem ser exploradas pois podem ser uma interessante opção. A dinamização de escolhas e a sensibilidade ao contexto quando empregadas ao controle de fluxo de vias, como visto, podem oferecer resultados satisfatórios. Em contrapartida, precisa ser considerado o custo da adaptação deste novo sistema que inclui instalação de dispositivos em veículos como para a localização automática (*Automatic Vehicle Location* ou AVL), além de suporte ao processamento e à comunicação com os dispositivos gerenciadores.

## Referências

1. Departamento Nacional de Trânsito, disponível em <<http://www.denatran.gov.br/frota.htm>>, acessado em 17 de Maio 2013.
2. Tanenbaum, A. Modern Operating Systems, Editora Pearson, 3ª edição, Dezembro de 2007.
3. Dresner, K., Stone, P. Multiagent traffic management: a reservation-based intersection control mechanism. In: Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004. Proceedings of the Third International Joint Conference, 530 – 532 , Julho de 2004.
4. Martin, M., Nurmi, P. A generic large scale simulator for ubiquitous computing. In: III Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services 2006, 1 – 3, Julho de 2006.