SEMÁFORO INTELIGENTE – UMA APLICAÇÃO DE APRENDIZAGEM POR REFORÇO

GABRIEL M. COSTA, GUILHERME S. BASTOS

Centro de Referência em Tecnologias da Informação, Instituto de Engenharia de Sistemas e Tecnologias da Informação, Universidade Federal de Itajubá
Av. BPS, 1303. Itajubá-MG
E-mails: gabrielmelo12@yahoo.com, sousa@unifei.edu.br

Abstract- Considering the great development and growth of the country, a major problem that can be denoted in the cities of medium and large is the chaotic traffic of vehicles inside. This problem is often intensified by the poor system of operation of existing traffic lights. This proposal presents the development of intelligent traffic lights, which aims to control, with Reinforcement Learning techniques, time to open and close the traffic lights at intersections, as well as the relationship between the percentage of time involved traffic lights, maximizing the flow of vehicles. For this purpose, several simulations were performed in software SimEvents® (MatLab® package), initially considering an intersection with two interdependent traffic lights. We performed a modeling of the junction by Reinforcement Learning methods, seeking to adapt the time involved in the problem, maximizing the flow of vehicles.

Keywords: Reinforcement Learning, simulation, traffic flow, intersection.

Resumo- Com o grande desenvolvimento e crescimento do país, pode-se notar que um dos principais problemas nas cidades de médio e grande porte é a situação caótica do tráfego de veículos em seu interior. Problema este que muitas das vezes é intensificado com o sistema precário de funcionamento dos atuais semáforos. Este projeto tem como proposta o desenvolvimento de um Semáforo Inteligente, que visa controlar, com técnicas de Aprendizagem por Reforço, o tempo de aberto e fechamento dos semáforos em cruzamentos, bem como a relação de porcentagem deste tempo entre os semáforos envolvidos, de modo a maximizar o fluxo de veículos. Para isso, diversas simulações foram realizadas no software SimEvents® (pacote Matlab®) considerando inicialmente um cruzamento com dois semáforos interdependentes. Foi feita a modelagem do cruzamento, por métodos de Aprendizagem por Reforço, buscando adequar o tempo envolvido no problema, maximizando o fluxo de veículos.

Palavras-chave- Aprendizagem por Reforço, simulação, fluxo de veículos, cruzamento.

1 Introdução

Atualmente, com enorme crescimento do país e apelo comercial automobilístico, nota-se a imensa frota de veículos existente nas ruas das grandes cidades de todo país. Isto faz vivenciar-se com sério problema de mobilidade urbana, problema este que pode ser notado principalmente nos horários de pico, ou seja, ida e volta do trabalho, em que quilômetros de congestionamentos são formados, dado principalmente, por ineficiência no sistema de controle e distribuição do tráfego de veículos.

Toma-se como exemplo o fato de ficar parado em um semáforo sendo que na outra via não passa nenhum veículo. Esta ocorrência faz com que o semáforo deixe de atuar como um controlador de tráfego e atue como um intensificador de congestionamento. Foi feito neste projeto um estudo para suprir essa deficiência no controlador de tráfego, com o intuito de projetar um semáforo capaz de tomar decisão de acordo com a situação atual do trânsito, ou seja, semáforo que tome devida decisão em tempo real.

O problema da mobilidade urbana que as grandes cidades vêm enfrentando é citado por Scaringella (2001) em que é enfatizado o uso de tecnologia para controle do tráfego. Para isto estudos da técnica de Aprendizagem por Reforço (AR), sistema estocásticos Markovianos, foram realizados concomitantes com estudos na área de Processo de Decisão Markov (MDP). O estudo para desenvolvimento de semáforo inteligente é tema de ordem científica como demonstrado por Wiering et al (2003), Wiering (2000) e Thorpe (1997), os quais fazem uso do AR.

De acordo com Sutton e Barto (1998) a AR é um formalismo da Inteligência Artificial que permite a um indivíduo aprender a partir da sua interação com o ambiente no qual ele está inserido. A aprendizagem se dá através do conhecimento sobre o estado do indivíduo no ambiente, das ações efetuadas e das mudanças de estado decorrentes das ações que são elementos essenciais na área de AR, a qual vem sendo muito utilizada com sucesso em problemas reais nos últimos anos (Kaelbling, Littman, e Moore, 1996). Este trabalho apresenta um problema de aprendizado dos tempos envolvidos em um cruzamento contendo dois semáforos, de modo que o haja a maximização do somatório de carros em ambas as vias. Foram utilizados, o software matemático MatLab® e seu pacote de simulação SimEvents®, que produz resultados através de equações matemáticas e que tem como objetivo proporcionar melhores soluções para o desenvolvimento do processo. O algoritmo de aprendizagem por reforço utilizado neste problema foi o SARSA (Singh, Jaakkola, Littman e Szepesvári, 2000), o qual é bem aplicado em sistemas com aprendizagem em tempo real.

O artigo está organizado como se segue: primeiramente é apresentado a teoria de Processo Decisório de Markov (MDP), o qual é a base teórica para AR, a qual é apresentada logo em seguida; apresentasse-se então a modelagem do problema com o resultados na seção a seguir; finalizando o artigo são apresentadas conclusões e as perspectivas de trabalhos futuros.

2 Processo Decisório de Markov

Um Processo Decisório de Markov (MDP) é uma forma de modelar processos onde as transições entre estados são probabilísticas, os estados são observáveis e é possível interferir com o sistema dinâmico através de ações que produzem mudanças de estado e recompensas. Cada ação tem uma recompensa (ou custo), que depende do estado em que o processo se encontra.
Este processo é ditos "de Markov" (ou "Markovianos") porque os processos modelados obedecem á *propriedade de Markov*: o efeito de uma ação em um estado depende apenas da ação e do estado atual do sistema (e não de como o processo chegou a tal estado); e são chamados de processos "de decisão" porque modelam a possibilidade de um agente (ou "tomador de decisões") interferir periodicamente no sistema executando ações. MDPs podem ser aplicados em um grande número de áreas diferentes, por exemplo, finanças e investimentos, inspeção, manutenção e reparação, recursos hídricos, como mostra White (1993).

2.1 Definição MDP

Um Processo de Decisão de Markov (MDP) é uma tupla (S, A, T, R), onde:

• S é um conjunto de estados em que o processo pode estar;

• A é um conjunto de ações que podem ser executadas em diferentes épocas de decisão;

• T: S x A x S → [0, 1] é uma função que dá a probabilidade de o sistema passar para um estado s' S, dado que o processo estava em um estado s S e o agente decidiu executar uma ação a A (denotada T(s'| s, a)); e

• R: S x A → R é uma função que dá o custo (ou recompensa) por tomar uma decisão A quando o processo está em um estado S.

Considerando que o sistema está em algum estado *s* e em dada *época decisão k*, é necessário selecionar qual ação *a* deve ser executada. A ação é selecionada seguindo uma *regra de decisão*, sendo uma forma simples de regra de decisão o mapeamento direto de estados em ações. O conjunto de todas as regras de decisão é chamado de política (π).
Ao executar uma política, o tomador de decisões receberá recompensas em cada época de decisão. Para comparar duas políticas, é necessário um critério de desempenho que pode ser definido por diversos critérios de desempenho (ou "de otimalidade") para MDPs, e entre os mais conhecidos pode-se citar:

• A recompensa média por época de decisão:

$$\frac{1}{z}\sum_{k=0}^{z-1} r \quad (1)$$

• A recompensa esperada total, E :

$$\sum_{k=0}^{z-1} r \quad (2)$$

• A recompensa esperada descontada, E:

$$\sum_{k=0}^{z-1} \gamma^k r_k \quad (3)$$

Sendo z o horizonte de um MDP.
Neste trabalho foi determinado que a recompensa fosse a diferença da quantidade de carros de passaram pelo semáforo e os que ficaram na fila. Para isto foi utilizada a recompensa média por época das amostras coletadas em tempos pré-definidos. Portanto um estado ótimo para este modelo é quando se tem a máxima recompensa, ou seja, quantidade máxima de carros que passaram pelo menos a quantidade de carros que ficaram na fila.
O fator de desconto γ]0,1[é usado com horizonte finito para garantir a convergência do valor da recompensa total esperada.
Uma política é ótima (π*) quando a recompensa total esperada para todo estado é maximizada. A função valor V*(s) dá a recompensa total esperada para uma política ótima π*:

$$V^*(s) = max_{a \in A}[R(s,a) + \gamma \sum_{s' \in S} T(s'|s,a)V^*(s')] \quad (4)$$

Dado um estado $s \in S$, uma ação $a \in A$ e uma política π para um MDP, pode-se definir o valor da ação *a* no estado *s*, considerando a recompensa imediata de *a* e a recompensa esperada após *a*, nas outras épocas de decisão, desde que as ações tomadas após *a* sejam determinadas pela política π. A função que dá este valor é denotada por Q. Para a esperança da recompensa total descontada, Q é definida como:

$$Q^{\pi}(s,a) = R(s,a) + \gamma \sum_{s \in S} T(s'|s,a)V^{\pi}(s') \quad (5)$$

Para uma política ótima π*, tem-se:

$$Q^*(s,a) = R(s,a) + \gamma \sum_{s \in S} T(s'|s,a)V^*(s')$$

$$\pi^*(s) = argmax_{a \in A} Q^*(s,a)$$

$$V^*(s) = max_{a \in A} Q^*(s,a) \quad (6)$$

3 Aprendizagem por Reforço

Aprendizado por Reforço (AR) é uma técnica de aprendizado de máquina, bastante usada em controle de processos industriais, em que um agente aprende por sucessivas interações com o seu ambiente e escolhe as ações que proporcionam os melhores resultados/ganhos. O ambiente apresenta, a cada interação,

um novo estado (situação) e um valor numérico chamado reforço para avaliar a ação (conforme Figura 1).

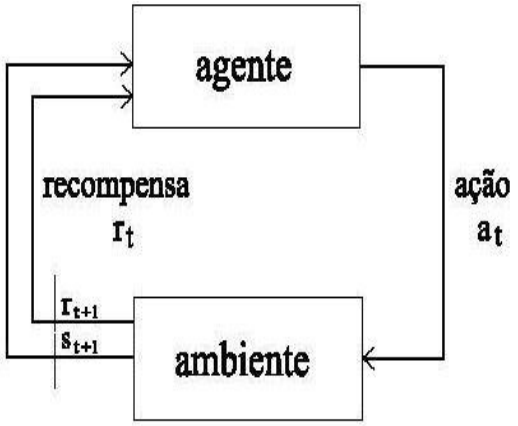

Figura 1 - Interação entre agente e ambiente

O domínio deve ser modelado como um MDP, onde o agente e o ambiente interagem em uma seqüência discreta de passos no tempo, o estado e a ação em dado instante determinam a distribuição de probabilidades para o estado seguinte e o reforço. O objetivo do agente normalmente é escolher ações de modo a maximizar uma soma descontada dos reforços subseqüentes.
A principal diferença entre o Aprendizado por Reforço e outras técnicas de aprendizado de máquina é a utilização da avaliação das ações tomadas (Sutton e Barto, 1998). Em outros métodos, são utilizados instruções ou exemplos, informando as situações e as ações corretas que devem ser tomadas. Na AR, o agente tenta descobrir, dentre as possíveis ações, quais delas promovem melhores resultados, utilizando apenas sua própria experiência. A Função de Recompensa define, para o estado atual, qual a melhor ação imediata enquanto a Função de Valor permite a avaliação das possíveis ações em longo prazo. Estas funções não são alteradas durante as interações enquanto são utilizadas para atualizar a Política, definindo os melhores mapeamentos estado-ação.

3.1 Características da aprendizagem por reforço

Os elementos que caracterizam Aprendizagem por Reforço são:

• Aprendizado por interação: característica principal que define AR. O agente AR age no ambiente e aguarda pelo valor do reforço/recompensa que o ambiente retorna em resposta perante á ação tomada, assimilando através do aprendizado o valor obtido para tomar decisões posteriores.

• Retorno atrasado: um valor de reforço alto enviado pelo ambiente ao agente não significa necessariamente que a ação tomada por este é recomendada. Uma ação é produto de uma decisão local no ambiente, sendo seu efeito imediato de natureza local, enquanto que em um sistema de AR o intuito é alcançar objetivos globais. Ou seja, a qualidade das ações é vista pelas soluções em longo prazo.

• Orientado pelo objetivo: em AR não é necessário conhecer detalhes da modelagem do ambiente. Simplesmente existe uma gente que age neste ambiente desconhecido tentando alcançar um objetivo, que geralmente é aperfeiçoar algum dentro do ambiente.

• Investigação *versus* exploração: esta questão consiste em decidir quando se deve aprender e quando não se deve aprender sobre o ambiente, mas usar a informação já obtida até o momento. Para que um sistema seja realmente autônomo, esta decisão deve ser tomada por ele próprio. A decisão é fundamentalmente uma escolha entre agir baseado na melhor informação que o agente dispõe no momento ou agir para obter novas informações sobre o ambiente que possam permitir níveis de desempenho ainda maiores no futuro. Em suma o agente deve aprender quais ações maximizam os ganhos obtidos, mas também deve agir de forma a atingir esta maximização explorando ações ainda não executadas ou regiões pouco visitadas no espaço de estados. Uma boa estratégia então é mesclar os modos de investigação e exploração.
No algoritmo SARSA há possibilidade de se utilizar vários tipos de políticas de aprendizagem. A utilizada neste modelo foi o GLIE (Greedy in the Limit with Infinite Exploration).
Estas políticas de aprendizado podem ser expressas a partir de probabilidades e um exemplo da política GLIE é uma das formas de exploração de Boltzman.

$$\Pr(a|s,t,Q) = \frac{e^{\beta t(s) Q(s,a)}}{\sum_{b \in A} e^{\beta t(s)\, Q(s,b)}} \quad (9)$$

Onde β(t) é o coeficiente de exploração do tempo *t* que controla a taxa de exploração na política de aprendizado, isto é, quanto mais β(t) tende ao infinito, mais guloso o sistema se torna, convergindo mais rapidamente, explorando mais e investigando menos. Para β(t) com valores baixos o sistema permanece em maior fase de investigação do que exploração, isto gera uma convergência mais próxima do ponto ótimo, mas com tempo de convergência maior.

4 Modelagem do Problema

O cruzamento que se deseja modelar pode ser representado pela Figura 2, ou seja, um cruzamento e dois semáforos.

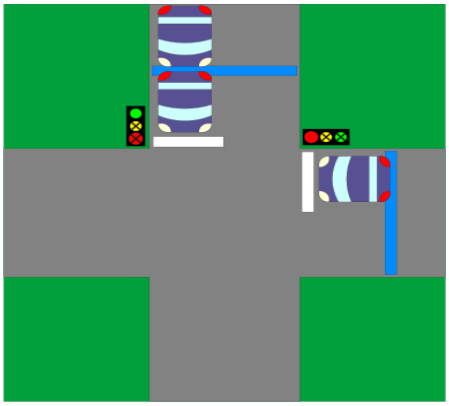

Figura 2 – Cruzamento com dois semáforos.

Neste trabalho o estado foi definido como a tupla: (Período, Largura do Pulso). Cada estado possui um período e uma porcentagem (largura de pulso) que corresponde à porcentagem do período em que um semáforo fica aberto e o outro fechado. Esta variação da largura de pulso pode ser observada nas Figuras 3 e 4:

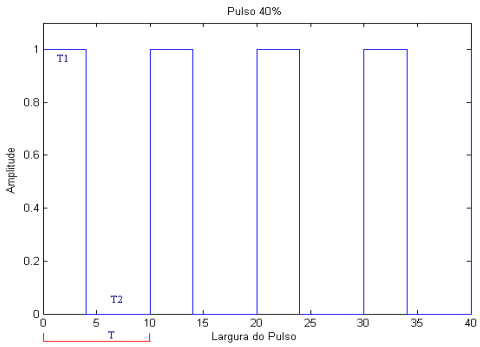

Figura 3 – Largura de pulso 40%

Onde T = T1+T2;

Por exemplo, para T=10s e uma largura de pulso de 40% o tempo que o semáforo #1 fica aberto é dado pelo tempo T1 e o tempo que o semáforo #2 fica aberto é dado por T2, como pode ser observado na figura 3.
Na figura 4 é exemplificada a largura de pulso de 80%:

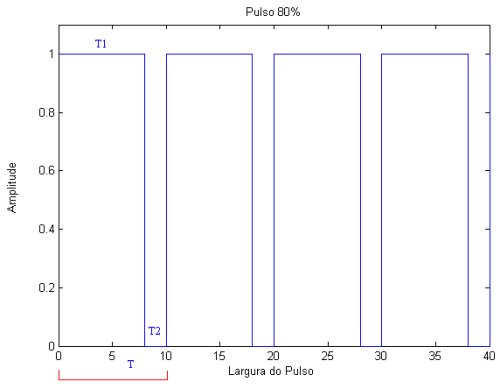

Figura 4 – Largura de pulso 80%

Nota-se que o tempo que o semáforo #1 fica aberto é maior que o tempo em que o semáforo 2 fica aberto, por ter usado uma largura de pulso de 80%.
O algoritmo de aprendizagem por reforço tem a função de encontrar, a partir de iterações, este estado ótimo, isto é, o melhor período e respectiva largura de pulso.
A recompensa foi determinada como sendo a diferença da quantidade de carros de passaram pelo semáforo e os que ficaram na fila, como descrito na equação 10.

$$RM = (X(tamanho(X,1)-X(1)))/tamanho(X,1)-(média(X1)+ média(X2)) \quad (10)$$

Onde X é a amostra da quantidade de carros que passaram pelo cruzamento no tempo de simulação estipulado, X1 é a quantidade de carros da fila1 que passaram pelo cruzamento, e X2 é a quantidade de carros da fila 2 que passaram pelo cruzamento.
Para isto foi utilizada a recompensa média por época das amostras coletadas em tempos pré-definidos. Portanto um estado ótimo para este modelo é quando se tem a máxima recompensa, ou seja, quantidade máxima de carros que passaram pelo cruzamento menos a quantidade de carros que ficaram na fila.
Fazendo análise dos métodos de convergências, concluímos que para o projeto em desenvolvimento, o melhor método a ser usado é o SARSA. Pois se trata da simulação de um cruzamento que pode apresentar diversos estados, isto é, diferentes períodos e largura de pulso. Com esta busca constante pela melhor ação a ser tomada a cada estado, este método permite encontrar um funcionamento ótimo dos semáforos nos cruzamentos, evitando geração de congestionamento muitas vezes desnecessários.
A modelagem foi feita a partir do esquemático feito no SimEvents®, pacote MatLab®, a qual é apresentada na Figura 5.

5 Resultados

Os resultados a seguir foram coletados a partir da simulação do modelo desenvolvido. Foram feitas varias simulações que poderão ser observadas nos gráficos abaixo.
Primeiramente, nas Figuras 6 e 7 pode ser observado um exemplo de fila gerada por um bloco específico do software utilizado, que pode representar um fila de trânsito real.

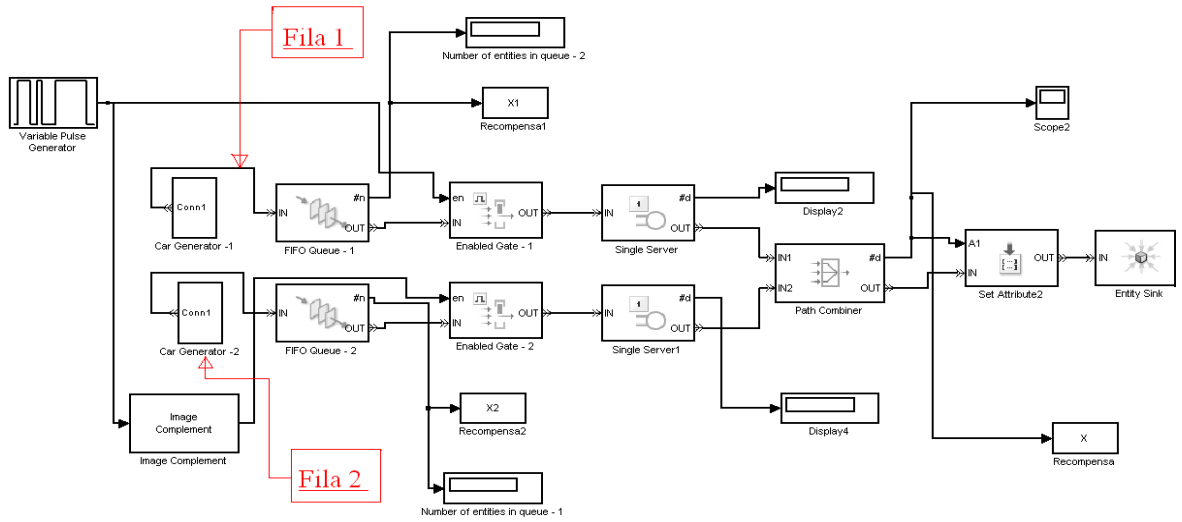

Figura 5 – Modelo de um cruzamento SimEvents®

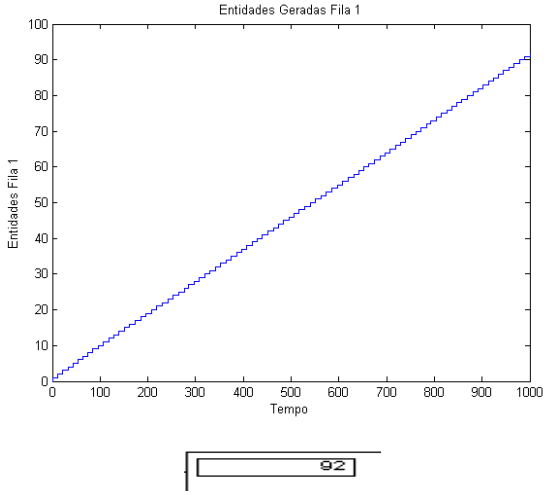

Figura 6 – Número de entidades na Fila 1

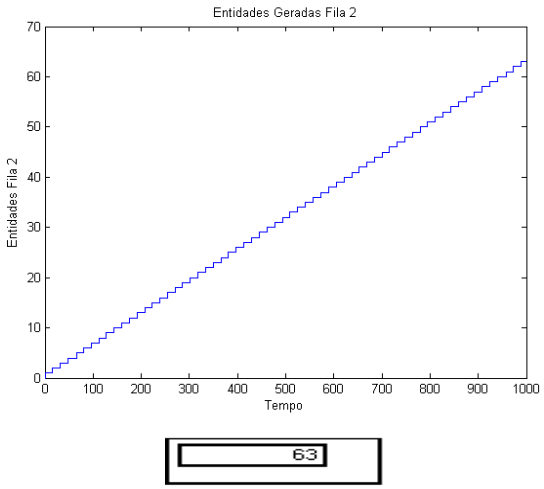

Figura 7 – Número de entidades na Fila 2

Nas próximas Figuras serão apresentadas as recompensas para Períodos e Largura de pulso aleatório, ou seja, tempo em que um semáforo fica aberto e o outro fechado, sem a aplicação do algoritmo de Aprendizagem por Reforço.

Inicialmente será simulado um Período de 50 segundos e largura de pulso igual 20%, isto é, o semáforo #1 fica aberto 10 segundos enquanto que o semáforo #2 fica aberto 40 segundos. Os gráficos das entidades nas filas podem ser observados nas figuras seguintes, bem como o gráfico que representa as entidades que passaram pelo bloco, para o estado específico.

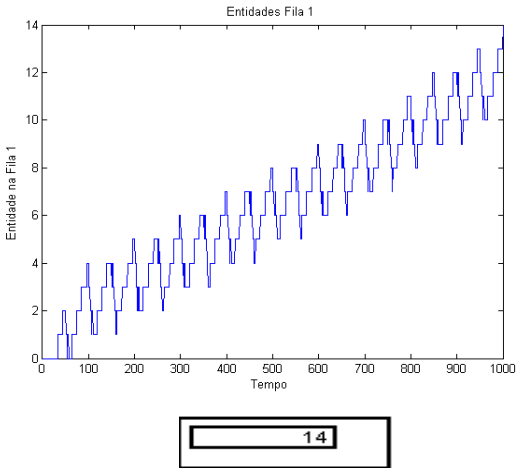

Figura 8 – Número entidades na Fila 1

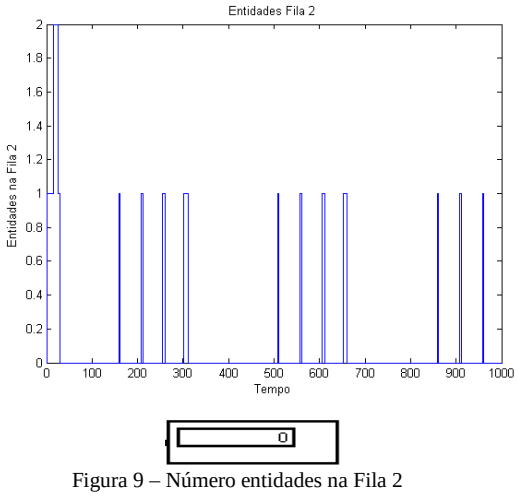

Figura 9 – Número entidades na Fila 2

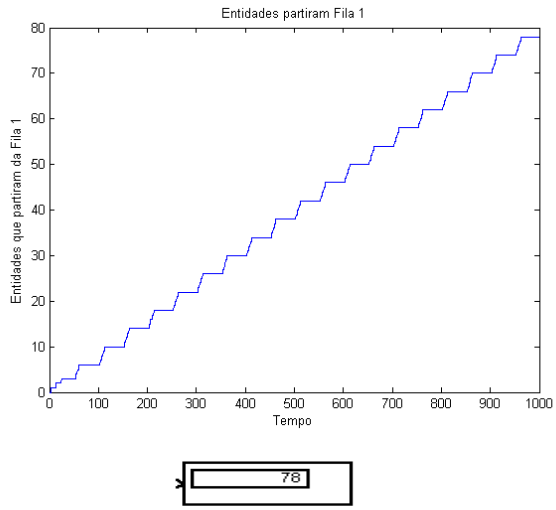

Figura 10 – Número de entidades da fila 1 que já passaram o cruzamento

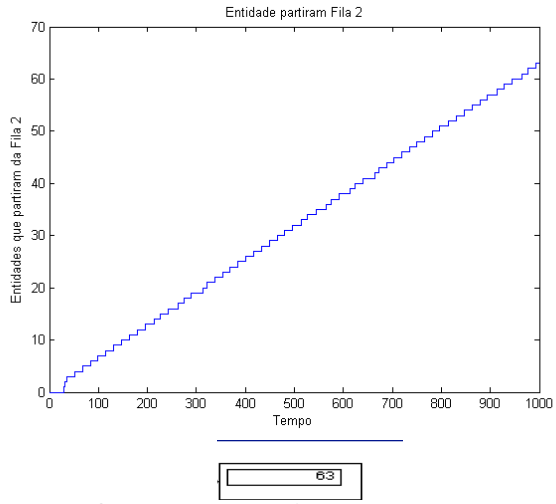

Figura 11 – Número de entidades da fila 2 que passaram o cruzamento

Como descrito anteriormente, a recompensa média será calculada a partir da diferença da quantidade de carro que passaram pelo semáforo e a quantidade de carros que ficaram na fila, equação 10.

Fazendo este cálculo para este estado aleatório, obteve-se RM= 1,9412.

Pode-se observar que o cálculo do tempo ótimo envolvido na abertura e fechamento dos semáforos e a máxima recompensam não são triviais, pois devem ser levados em considerações diversos estados possíveis de atuação. Para isso será utilizado como princípio técnica de Aprendizagem por Reforço, onde se espera que o agente encontre a melhor ação a ser tomada em um estado qualquer. Esta integração do AR com o cálculo dos tempos e porcentagem foi desenvolvido utilizando a área de programação do MatLab® e fazendo a conexão com o modelo desenvolvido no Simulink.

Para se ter como base de tempo o período foi limitado para variar entre 30 e 90 segundos, já a largura do pulso pode ser de 1% a 99% do período.

A execução de programas de Aprendizagem por Reforço requer certo número de episódios, sendo que em cada episódio é sorteado um estado qualquer e partir deste busca um estado ótimo. O episódio evita também que o programa fique 'preso' em 2 ou mais estados, e ainda cada episódio contêm vários passos que o programa executa para encontrar um estado ótimo para aquele episódio.

Neste programa foram feitas simulações usando 10 episódios e cada episódio contendo 200 passos.

Como explicado anteriormente, o valor de β(t) utilizado inicialmente foi de 1, ou seja, um valor baixo considerado que a cada passo é acrescido de um valor pré-definido. Com este valor inicial de β(t) o sistema estava em época de maior investigação do que exploração, de acordo que as iterações foram passando β(t) aumentava e então o sistema passava a fase de exploração.

Será representado inicialmente cada episódio e seu estado ótimo, posteriormente serão mostrados todos os episódios em um só gráfico. O 1º Episódio, 200 passos, está representado abaixo, o estado inicial foi sorteado aleatoriamente e a partir dele se encontrou o estado ótimo para este episódio.

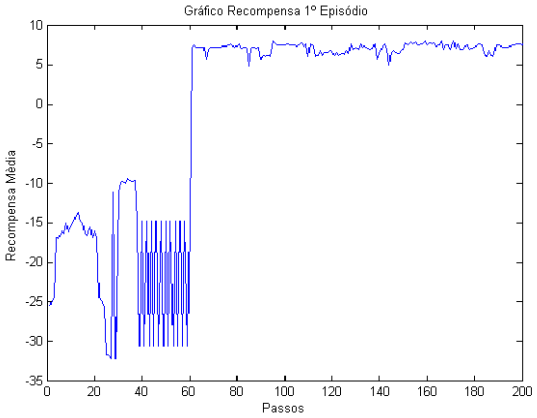

Figura 12 - Gráfico de Recompensa Média - 1º. Episódio

Este 1º. episódio começou suas iterações a partir do estado 50 segundos e 45% de largura de pulso. A partir deste estado, teve-se 200 passos e conclui-se que o estado ótimo para este episódio é 41 segundos e 39% de largura de pulso. Fazendo este cálculo para este estado, utilizando a equação 10, obteve-se RM = 8,0588.

Este estado ótimo representa uma passagem máxima de entidades pelo semáforo, tendo consequentemente fila mínima. Isto pode ser comprovado comparando os gráfico das Figuras 13 e 14 de entidades que passaram pelo cruzamento e o gráfico das Figuras 6 e 7 que representa a quantidade de entidades geradas.

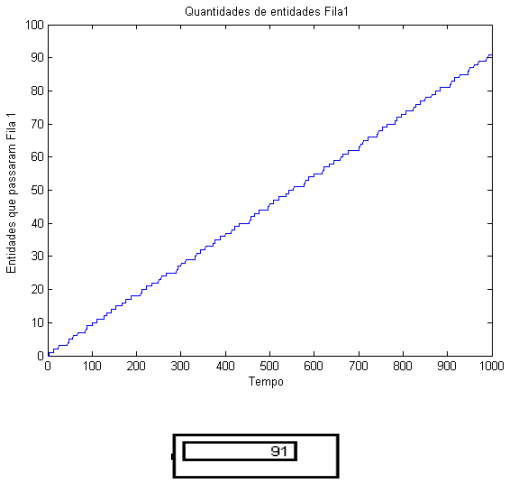

Figura 13- Quantidade entidades da fila 1 que passaram o cruzamento.

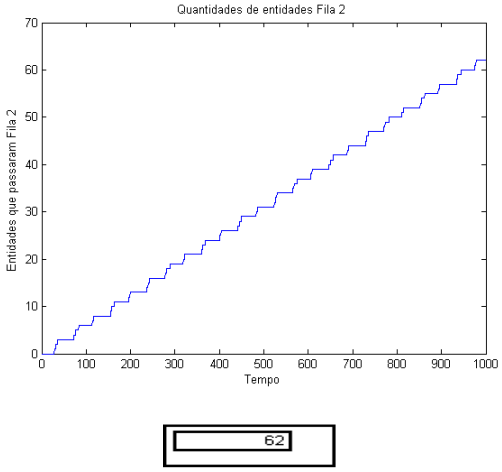

Figura 14- Quantidade entidades da fila 2 que passaram o cruzamento.

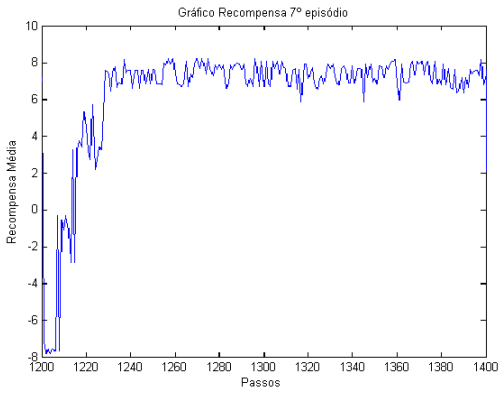

Figura 15 - Gráfico de Recompensa Média- 7º Episódio

No 7º. episódio as iterações começaram a partir do estado 58 segundos e 33% de largura de pulso. A partir deste estado, teve-se 200 passos e conclui-se que o estado ótimo para este episódio é 38 segundos e 68% de largura de pulso. Fazendo o cálculo para este estado, utilizando a equação 10, obteve-se RM = 7.

Podem-se observar os gráfico das figuras 16 e 17 de entidades que passaram pelo cruzamento e o gráfico das figuras 6 e 7 que representa a quantidade de entidades geradas.

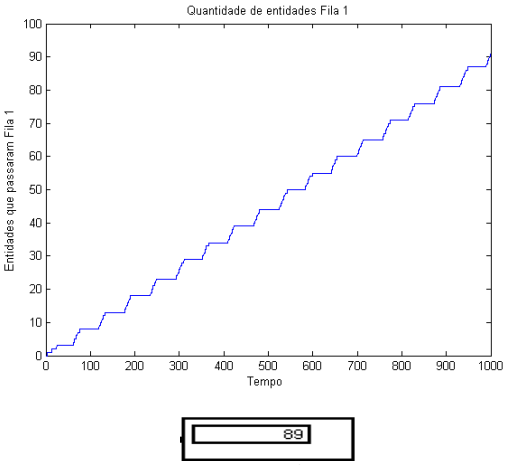

Figura 16- Quantidade entidades da fila 1 que passaram o cruzamento.

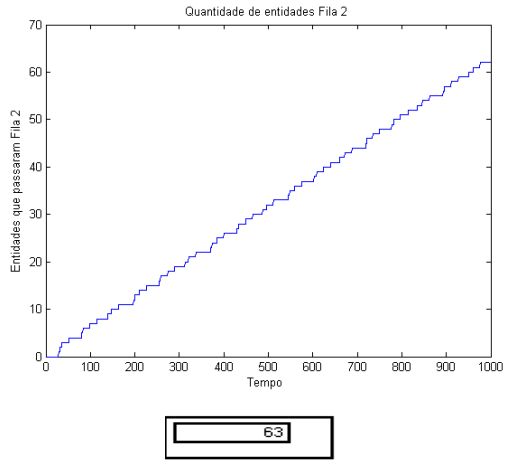

Figura 17- Quantidade entidades da fila 2 que passaram o cruzamento.

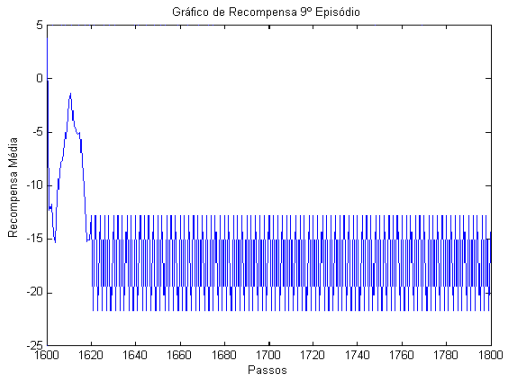

Figura 18 - Gráfico de Recompensa Média - 9º Episódio

Neste 9º. episódio, como pode ser observado na Figura 18, ocorreu de o sistema ficar preso entre os estados 53 segundos, 99% e 30 segundos, 2%, não visitando outros estados. Isto se dá por ser um sistema de aprendizagem, no decorrer das iterações isso se torna mais difícil de acontecer.

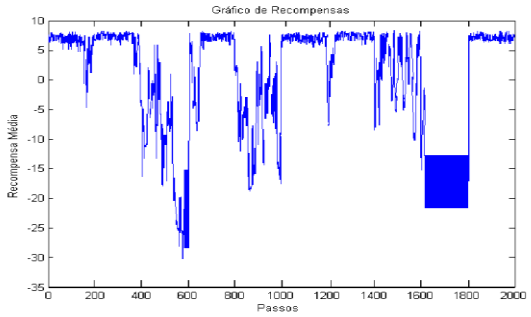

Figura 19 - Gráfico de Recompensa Média

Neste gráfico estão representados todos os episódios e passos desta simulação. Pode-se notar que a cada novo episódio o sistema tende a encontrar um novo estado ótimo até que a recompensa seja máxima. Para este sistema o estado ótimo é com um período de 30 segundos e largura de pulso 88%. Neste estado, pode-se notar nas figuras 20 e 21 que a todos os carros passaram pelo cruzamento, tendo assim fila mínima. Calculando a recompensa para este estado, utilizando a equação 10, obteve-se RM= 8,4118.

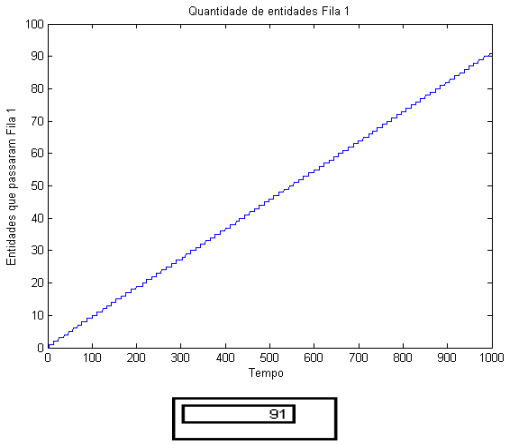

Figura 20- Quantidade entidades da fila 1 que passaram o cruzamento.

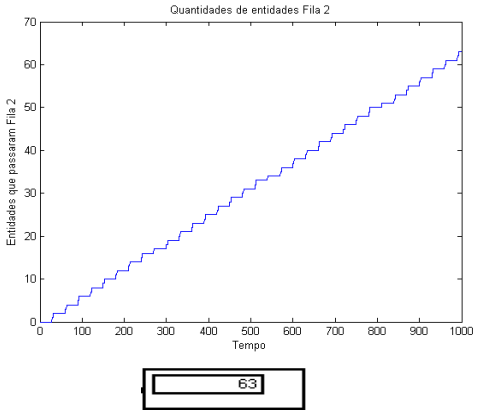

Figura 21- Quantidade entidades da fila 2 que passaram o cruzamento.

6 Conclusões e Trabalhos Futuros

Como pode ser visto nos gráficos apresentados, quando foi feita a simulação do sistema considerando apenas um estado qualquer, sem aplicação da Aprendizagem por Reforço, obteve-se uma recompensa média muito inferior ao estado obtido pela aplicação da técnica Aprendizagem por Reforço.

Trabalhos como esse são válidos para o desenvolvimento tecnológico, uma vez que relacionam os conceitos presentes no meio acadêmico com processos reais presentes em indústrias, ou seja, essa é uma forma de se aplicar a teoria à prática obtendo resultados proveitosos.

Nesse projeto foi elaborada uma forma de integrar o software SimEvents®, com a técnica de Aprendizagem por Reforço a fim de gerar um sistema que opere continuamente em estado ótimo.

Este projeto servirá como base de próximos trabalhos que estão sendo desenvolvido, considerando neste, quatro cruzamentos e oito semáforos interdependentes e dependência no tempo, isto é, o fluxo de carros depende da instante atual e tem picos, como por exemplo na hora do *rush*.

Agradecimentos

Os autores agradecem à Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) pelo apoio prestado neste projeto.

Referências Bibliográficas

Scaringella, R.S.(2001) A crise da mobilidade urbana em São Paulo. São Paulo Perpec. [online], vol.15(1).

Wiering, M., Van Veenen , J., Vreeken, J. & Koopman, A. (2003) Intelligent Trafic Light Control. European Research Consortium for Informatics and Mathematics, vol. 53, pp. 40-41.

Wiering, MA. (2000) Multi-agent reinforcement learning for traffic light control. Proceedings of the Seventeenth International Conference on Machine Learning (ICML2000), pp. 1151-1158.

Thorpe, T.L. (1997) Vehicle Traffic Light Control Using Sarsa. Master's thesis. Department of Computer Science, Colorado State University.

Sutton, R. S. & Barto, A.G. (1998) Reinforcement Learning: An Introduction. The MIT press.

Kaelbling, L. P., Littman, M. L. & Moore, A. W. (1996) Reinforcement Learning: A survey. Arxiv preprint

Singh, S.; Jaakkola, T.; Littman, M. L and Szepesvári, C (2000). Convergence results for single-step on-policy reinforcement learning algorithms. Machine Learning, Vol. 38, No. 3, pp. 287-308

White, D. J. (1993) A survey of applications of Markov decision processes. The Journal of the Operational Research Society, v. 44, n. 11, p. 1073–1096.