ε-Greedy Adaptativo

A. S. Mignon; R. L. A. Rocha

***Resumo*—Um dos grandes desafios em aprendizagem por reforço é o balanceamento entre *exploration* e *exploitation*. Neste trabalho apresenta-se um método para balanceamento denominado *ε-greedy* adaptativo. Este método é baseado no *ε-greedy* tradicional, que mantém o valor de ε estático. Na solução apresentada são utilizados os conceitos e as técnicas da tecnologia adaptativa para permitir que o valor do ε seja alterado ao longo da execução. Apresenta-se experimentos comparando numericamente os dois métodos para o problema *n-armed bandit*. No caso estacionário o método *ε-greedy* adaptativo apresentou melhor desempenho em relação ao método tradicional, enquanto, no caso não estacionário, o desempenho dos dois métodos foi similar.**

***Keywords*—aprendizagem por reforço, *ε-greedy*, adaptatividade.**

I. Introdução

Aprendizagem por reforço (*reinforcement learning*) é uma forma de aprendizagem de máquina (*machine learning*) não supervisionada na qual um agente aprende através de sua interação com o ambiente para atingir um determinado objetivo [1]. Ao interagir com o ambiente através de ações tomadas, o agente recebe recompensas numéricas. A meta do agente é maximizar o total de recompensas numéricas recebidas.

Um dos grandes desafios em relação à aprendizagem por reforço está no balanceamento entre *exploration* e *exploitation*[1]. *Exploitation* significa que o agente seleciona a ação que obteve maior média de recompensas. *Exploration* significa que o agente seleciona uma ação aleatoriamente, independente dos valores de recompensa obtidos anteriormente. *Exploitation* é a melhor coisa a se fazer para receber uma boa recompensa imediatamente. Entretanto, para descobrir quais ações são as melhores, é necessário executar o modo de *exploration*. Neste modo, pode-se encontrar uma ação melhor e se obter melhores resultados a longo prazo. Como não é possível executar o modo de *exploration* e de *exploitation* ao mesmo tempo, deve-se decidir por qual modo usar para realizar uma ação em um determinado instante de tempo.

Um método simples para realização do balanceamento entre *exploration* e *exploitation* é o método denominado *ε-greedy*. Este método comporta-se gulosamente (*greedily*) a maior parte do tempo, porém, de vez em quando, com uma pequena probabilidade ε, seleciona uma ação aleatoriamente entre todas as ações. Entretanto, o valor dessa probabilidade ε é estática.

A tecnologia adaptativa trata de técnicas e dispositivos que permitem a um sistema modificar seu próprio comportamento, em resposta a algum estímulo de entrada ou ao seu histórico de operações, sem a interferência de qualquer agente externo [2]. A tecnologia adaptativa permite que um sistema com regras estáticas torne-se um sistema com regras dinâmicas.

Neste trabalho apresenta-se o método *ε-greedy* adaptativo. Esse método tem como base o método *ε-greedy* tradicional, porém permite que o valor de ε seja alterado de acordo com as recompensas recebidas do ambiente. Comparou-se numericamente os dois métodos para o problema *n-armed bandit*. O *ε-greedy* adaptativo apresentou desempenho superior ao *ε-greedy* no caso estacionário. Já no caso não estacionário estudado, os dois métodos apresentaram resultados similares.

Este trabalho está dividido da seguinte forma: na seção II apresenta-se os principais conceito e elementos da aprendizagem por reforço. Na seção III descreve-se o problema *n-armed bandit* utilizado para a comparação dos métodos. Na seção IV apresenta-se o método *ε-greedy*. Na seção V descreve-se brevemente os conceito da tecnologia adaptativa. Na seção VI descreve-se o método e o algoritmo do *ε-greedy* adaptativo. Na seção VII apresenta-se os resultados dos experimentos realizados para a comparação dos métodos. Finalmente, na seção VIII apresenta-se as conclusões e trabalhos futuros.

II. Aprendizagem por Reforço

Aprendizagem por reforço é uma forma de aprendizagem de máquina com foco em aprender o que fazer. O aprendizado é realizado por um *agente* através de sua interação com um *ambiente* [1]. O agente deve possuir uma ou mais metas relativas ao ambiente. O ambiente fornece ao agente comentários (*feedback*) em relação a ações tomadas, na forma de recompensas numéricas. A meta do agente é maximizar o total de recompensas numéricas.

As recompensas servem para definir *políticas* ótimas em processos de decisão de Markov (*Markov Decision Processes - MDPs*). Uma política ótima é uma política que maximiza o total de recompensa esperada. A tarefa da aprendizagem por reforço é usar as recompensas observadas para aprender uma política ótima (ou quase ótima) para o ambiente [3].

Diferentemente de outras formas de aprendizagem de máquina, ao agente aprendiz não é dito que ações tomar, ele deve descobrir, através de um processo de tentativa e erro, quais ações produzem maior recompensa. Na maioria dos casos, as ações podem não somente afetar a recompensa imediata, mas também a próxima situação e, por conseguinte, todas as recompensas posteriores [1].

Aprendizagem por reforço, uma forma de aprendizagem não supervisionada, é diferente das formas de aprendizagem supervisionada. Na aprendizagem supervisionada o aprendizado é feito através de exemplos fornecidos por algum supervisor externo [3] [4]. É um importante tipo de aprendizado, porém ele somente não é adequado em situações de aprendizagem por interação. Em problemas que requerem interação é difícil obter-se exemplos do comportamento desejado que são

[1]Neste trabalho optou-se em manter em inglês os termos relacionados à aprendizagem por reforço.

corretos e representativos para todas as situações em que o agente tem que agir. Neste tipo de problema o agente deve ser capaz de aprender a partir de sua própria experiência.

Um dos desafios na área de aprendizagem por reforço está no dilema entre *exploration* e *exploitation*. Para obter uma maior recompensa, um agente deve preferir tomar ações que obtiveram melhor recompensa no passado. Mas para descobrir tais ações ele tem que tentar ações que ainda não foram selecionas anteriormente. O agente deve executar o modo de *exploitation*, isto é, selecionar a melhor ação conhecida até o momento, para obter imediatamente uma maior recompensa, porém ele também deve executar o modo de *exploration*, isto é, selecionar outras ações que não a melhor, para poder realizar melhores ações no futuro. Em aprendizagem por reforço são propostos diversos métodos para tentar o balanceamento entre *exploration* e *exploitation* [1].

A. Elementos da Aprendizagem por Reforço

Além do agente e do ambiente, pode-se identificar quatro sub-elementos principais de um sistema de aprendizagem por reforço: uma *política*, uma *função de recompensa*, uma *função valor*, e, opcionalmente, um *modelo* do ambiente [1].

Uma política define a forma como o agente aprendiz comporta-se em um dado momento. Ela é um mapeamento dos estados percebidos do ambiente para ações a serem tomadas quando se está naquele estado. O agente precisa aprender uma política ótima (ou próximo da ótima) de modo a realizar o seu objetivo.

Uma função de recompensa define o objetivo em um problema de aprendizagem por reforço. Ela mapeia cada estado percebido (ou o par estado-ação) do ambiente a um único número, uma recompensa. A tarefa do agente é maximizar o total de recompensas recebidas a longo prazo. A função de recompensa define quais os eventos são bons ou ruins para o agente.

Uma função valor especifica o que é bom a longo prazo. O *valor* de um estado é o montante total de recompensa que um agente pode esperar acumular em um futuro, iniciando daquele estado.

Um modelo do ambiente é algo que simula o comportamento daquele ambiente. Por exemplo, dado um estado e uma ação o modelo pode prever o próximo estado e a próxima recompensa. Ele é usado para planejamento, através do qual temos um meio de decidir o curso de uma ação considerando situações futuras antes de realmente experimentá-las.

III. O Problema *n-Armed Bandit*

Imagine que um agente depara-se constantemente com uma escolha entre *n* diferente opções, ou ações, a cada escolha recebe uma recompensa numérica do ambiente, escolhida a partir de uma distribuição de probabilidades estacionária que depende da ação selecionada. O objetivo é maximizar o total de recompensas esperadas sobre algum período de tempo, por exemplo, 1000 opções de ações ou passos de tempo. Cada ação de seleção é denominada de *jogada* (*play*).

Esta é a forma original do problema *n-armed bandit*, nomeado pela analogia a uma máquina caça-níqueis. Cada ação selecionada é como puxar uma alavanca da máquina caça-níqueis, e as recompensas são os pagamentos obtidos. Através de repetidas jogadas você tem que maximizar seus ganhos concentrando-se em puxar as melhores alavancas. Outra analogia é a de um médico escolher tratamentos experimentais para uma série de pacientes doentes. Cada jogada é a seleção de um tratamento, e cada recompensa é o bem estar do paciente.

No problema *n-armed bandit* cada ação tem uma recompensa esperada. Denomina-se essa recompensa esperada de *valor* daquela ação. Se o valor de cada ação é conhecido, então, é trivial a solução do problema. Entretanto, assume-se que não se conhece os valores das ações com certeza, tem-se apenas uma estimativa de cada valor.

Se mantivermos as estimativas dos valores de cada ação, então, em algum espaço de tempo, existe pelo menos uma ação cujo valor estimado é maior. Denomina-se essa ação com maior valor de *greedy* (gulosa). Se a ação *greedy* é selecionada, diz-se que se está *exploiting* o conhecimento corrente dos valores das ações. Se, ao invés disto, seleciona-se uma ação não *greedy*, diz-se que se está *exploring*. O modo de *exploration* permite melhorar as estimativas das ações com valores menores.

Exploitation é a melhor coisa a se fazer para maximizar a recompensa esperada em um passo, mas *exploration* pode produzir uma maior recompensa total a longo prazo. Por exemplo, suponha que os valores das ações *greedy* são conhecidos com certeza, enquanto diversas outras ações são estimadas como boas, mas com uma incerteza substancial. A incerteza é tal que pelo menos uma destas outras ações provavelmente são realmente melhores que a ação *greedy*, porém não se sabe qual é. Se existem muitos passos à frente para realizar seleções de ações, então pode ser melhor executar o modo de *exploration* para as ações não *greedy* e descobrir quais delas são melhores que a ação *greedy*. Com isso, a recompensa é baixa a curto prazo, durante o modo de *exploration*, porém pode ser alta a longo prazo porque depois de descoberta as melhores ações pode-se executar o modo de *exploitation* com elas muitas vezes. Por não ser possível executar o modo de *exploration* e o modo de *exploitation* ao mesmo tempo com uma única seleção de ações, refere-se a isso como o conflito entre *exploration* e *exploitation*.

O problema *n-armed bandit* tem sido usado para modelar o dilema do balanceamento entre *exploration* e *exploitation*. Na próxima seção, apresenta-se um método simples para tal balanceamento. Esse método, descrito em [1], é utilizado no restante deste trabalho.

IV. ε-Greedy

Nesta seção apresenta-se um método simples, denominado *ε-greedy*, para balanceamento entre *exploration* e *exploitation* que utiliza a estimativa de recompensa esperada para cada ação como forma de seleção de uma ação. A descrição deste método foi baseada em [1].

Denota-se o real valor de uma ação a como $q_*(a)$, e o valor estimado no t-ésimo passo de tempo como $Q_t(a)$. Uma forma de calcular a estimativa da ação a é através da média de recompensas recebidas quando a ação foi selecionada. Em outras palavras, se pelo t-ésimo passo de tempo a ação a foi selecionada K_a vezes antes de t, recebendo recompensas R_1, R_2,..., R_{K_a}, então seu valor estimado é:

$$Q_t(a) = \frac{R_1 + R_2 + ... + R_{K_a}}{K_a} \quad (1)$$

Se $K_a = 0$, então define-se para $Q_t(a)$ um valor padrão como, por exemplo, $Q_1(a) = 0$. Como $K_a \to \infty$, pela lei dos grandes números, então $Q_t(a)$ converge para $q_*(a)$ [1]. Esta técnica é denominada de média das amostras (*sample-average*) para a estimativa dos valores das ações porque cada estimativa é uma média simples das recompensas.

A regra mais simples de seleção de ações é selecionar a ação com maior valor estimado. Isto é, selecionar no passo t uma das ações *greedy*, A_t^*, para o qual $Q_t(A_t^*) = max_a Q_t(a)$. Este método sempre executa o modo de *exploitation* utilizando o conhecimento corrente das estimativas para maximizar a recompensa imediata. Entretanto, como visto na seção anterior, pode haver ações melhores a serem tomadas. Neste caso, é necessário que se execute o modo de *exploration* para descobrir tais ações.

Uma alternativa simples é comportar-se *greedily* (gulosamente) a maior parte do tempo, porém, de vez em quando, com uma pequena probabilidade ε, seleciona-se uma ação aleatoriamente entre todas as ações, com probabilidade igual, e independente do valor estimado para cada ação. Denomina-se esse método de seleção ação de ε-*greedy*.

Uma vantagem deste método é que, à medida que o número de jogadas aumenta, cada ação será recompensada um número infinito de vezes, garantindo que $K_a \to \infty$ para todo a, garantindo assim que todos os $Q_t(a)$ convergem para $q_*(a)$. Isto implica que a probabilidade de seleção da ação ótima converge para um valor maior que 1 - ε, isto é, para próximo da certeza.

A seguir, comparam-se numericamente os métodos *greedy* e ε-*greedy* em um conjunto de testes. Os testes foram realizados com um conjunto de 2000 tarefas geradas aleatoriamente para o problema *n-armed bandit* com $n = 10$. Para cada alavanca, os valores das ações, $q_*(a)$, $a = 1, ..., 10$, foram selecionados de acordo com uma distribuição normal (Gaussiana) com variância entre 0 e 1.

No $t-$ésimo passo de tempo com uma dada alavanca, a recompensa real R_t é o $q_*(A_t)$ para a alavanca (onde A_t é a ação selecionada) mais um termo de ruído distribuído normalmente com variância entre 0 e 1. Calculando a média sobre as alavancas, podemos traçar o desempenho e o comportamento dos métodos e verificar como eles se comportam com a experiência de 1000 passos (jogadas).

As Figuras 1 e 2 apresentam os gráficos de comparação do método *greedy* com dois métodos ε-*greedy* ($\varepsilon = 0.01$ e $\varepsilon = 0.1$), conforme descrito acima. Os dois métodos calculam as estimativas dos valores das ações usando a técnica de média das amostras.

O gráfico da Figura 1 apresenta o aumento da recompensa esperada com a experiência. O método *greedy* melhorou ligeiramente mais rápido no início em relação aos outros métodos, mas depois estabilizou-se em um nível bem abaixo. Ele conseguiu uma recompensa por passo de apenas 1, em comparação com o melhor valor obtido neste teste que foi de

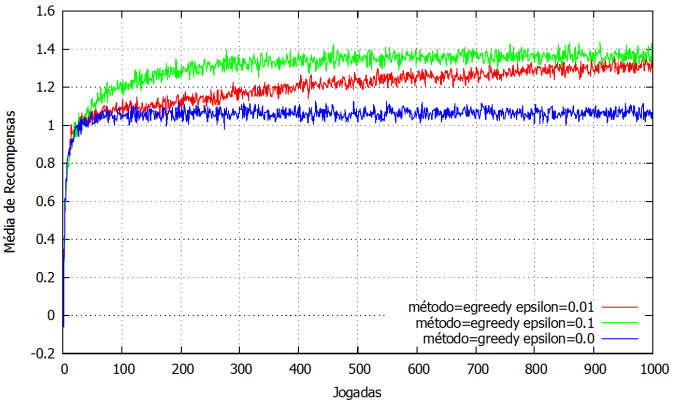

Figura 1. Desempenho médio dos métodos *greedy* e ε-*greedy* para o aumento da recompensa esperada.

cerca de 1.45. O método *greedy* executa significantemente pior a longo prazo porque fica preso a uma ação sub-ótima.

O gráfico da Figura 2 mostra que o método *greedy* encontra a ação ótima em aproximadamente um terço das tarefas. O método ε-*greedy* executa melhor porque continua a executar o modo de *exploration*, e melhora suas chances de descobrir a ação ótima. O método com $\varepsilon = 0.1$ executa mais o modo de *exploration*, e geralmente encontra a ação ótima mais cedo, mas nunca a seleciona mais de 91% das vezes. O método com $\varepsilon = 0.01$ melhora mais lentamente, mas eventualmente tem desempenho melhor que o $\varepsilon = 0.1$ a longo prazo.

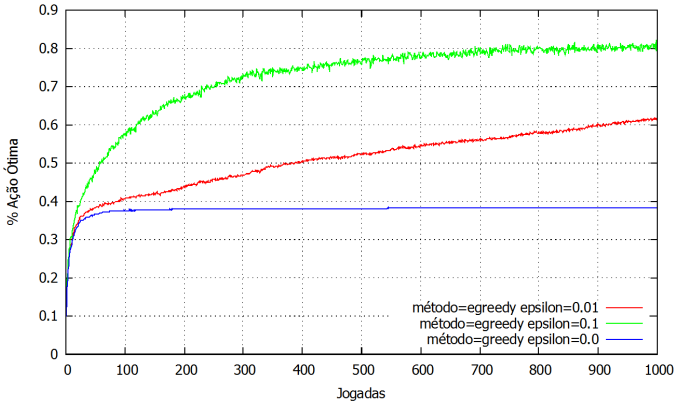

Figura 2. Desempenho médio dos métodos *greedy* e ε-*greedy* em selecionar uma ação ótima.

O método discutido utilizando a média das amostras é apropriado em um ambiente estacionário, isto é, quando o ambiente não sofre alterações. Entretanto, frequentemente encontra-se problemas de aprendizagem por reforço que são não estacionários, isto é, quando o ambiente sofre alterações ao longo do tempo. Nesses casos, é necessária uma técnica diferente da média das amostras para calcular o valor estimado de uma ação. Tais técnicas não são aqui discutidas por estarem fora do escopo deste trabalho.

V. Tecnologia Adaptativa

Adaptatividade é a capacidade que um sistema tem de modificar seu próprio comportamento, em resposta a algum

estímulo de entrada ou ao seu histórico de operações, sem a interferência de qualquer agente externo [2]. A tecnologia adaptativa trata de técnicas e dispositivos que tem característica adaptativa. Dispositivos adaptativos são descrições abstratas de problemas que tem comportamento dinâmico. Essas descrições são associadas a dispositivos não adaptativos subjacentes que representam problemas com comportamento estático.

Dispositivos não adaptativos têm seu comportamento definido por um conjunto de regras estáticas. Um dispositivo não adaptativo subjacente é melhorado adicionando-se um conjunto de ações adaptativas. Essas ações caracterizam as operações necessárias para fazer o sistema comportar-se adaptativamente [2].

Um algoritmo é dito adaptativo quando é capaz de modificar espontaneamente o seu comportamento em resposta a uma condição especial de entrada [5]. Na próxima seção, apresenta-se um algoritmo adaptativo para o método ε-*greedy*. Esse algoritmo tem por objetivo permitir que o valor do ε seja modificação ao longo de sua execução, de acordo com as recompensas obtidas do ambiente.

VI. ε-Greedy Adaptativo

Utilizando os conceitos e técnicas da tecnologia adaptativa, criou-se o método denominado ε-*greedy* adaptativo. Este método é baseado no ε-*greedy*, porém permite que o valor do ε varie ao longo da execução. A variância do valor do ε depende das recompensas retornadas pelo ambiente.

O Algoritmo 1 apresenta o algoritmo usado no método ε-*greedy* adaptativo. Esse algoritmo é usado para a seleção de uma ação a quando se está em um estado s. Ele decide se deve executar o modo de *exploration* ou de *exploitation*. Se decidir pelo modo de *exploitation* seleciona a ação com maior média de recompensa (A_t^*). Se decidir pelo modo de *exploration* seleciona uma ação aleatoriamente, conforme descrito na seção IV. Quando o algoritmo estiver no modo de *exploration* ele pode executar uma ação adaptativa, dependendo de sua configuração, que modifica o valor do ε.

As variáveis max_{ant} e k são globais, sendo utilizadas para armazenar o estado do algoritmo. Quando o algoritmo é executado para a seleção de uma ação ele deve decidir se executa o modo de *exploration* ou de *exploitation*. Ao modo de *exploration* adicionou-se uma ação adaptativa que pode alterar o valor de ε. São contadas quantas vezes o algoritmo entrou no modo de *exploration*, utilizando a variável k, após a última vez que ele alterou o valor de ε. Antes da variável k atingir um limite especificado, se houver uma grande diferença entre as variáveis max_{ant} e max_{atual}, significa que deve-se ajustar o valor de ε, pois a alteração no ambiente foi significativa. Caso o limite seja atingido, então altera-se o valor de ε independente da diferença obtida. Após a alteração do valor de ε a variável k é zerada e a variável max_{ant} recebe o valor da variável max_{atual}.

O algoritmo contém alguns valores constantes que foram definidos empiricamente como melhores após experimentos realizados entre uma faixa de valores. O valor da variável *LIMIT* é de 12, porém testaram-se valores na faixa entre 3 e 18. O fator de multiplicação 10 para o cálculo do Δ foi usado para acentuar a diferença entre os valores de max_{ant} e max_{atual}.

Algoritmo 1 ε-greedy adaptativo

```
max_ant ← 0
k ← 0
se distribuição normal < ε então
  max_atual ← Q_t(A*_t)
  k ← k + 1
  Δ ← |max_ant − max_atual| * 10
  se k < LIMIT então
    se Δ >= 0.8 então
      ε ← sigmoid(Δ)
      max_ant ← max_atual
      k ← 0
    fim se
  senão
    ε ← sigmoid(Δ)
    max_ant ← max_atual
    k ← 0
  fim se
  seleciona uma ação aleatoriamente
senão
  seleciona A*_t
fim se
```

Isso foi necessário para obter-se um valor mais adequado para ε retornado pela função sigmóide. Para definirmos o valor do fator de multiplicação testaram-se valores na faixa entre 5 e 30. Para definirmos o que seria um valor grande para a diferença Δ testou-se valores na faixa entre 0.3 e 2.0.

A definição de um novo valor para o ε é feita utilizando-se uma função sigmóide, conforme apresentada abaixo. O gráfico da função é apresentado na Figura 3. Nele pode-se verificar que o valor de ε pode variar entre 0 e 0.5. Empiricamente verificou-se que não há ganhos significativos se o valor de ε for superior a 0.5.

$$sigmoid(x) = \frac{1.0}{1.0 + \exp(-x)} - 0.5 \qquad (2)$$

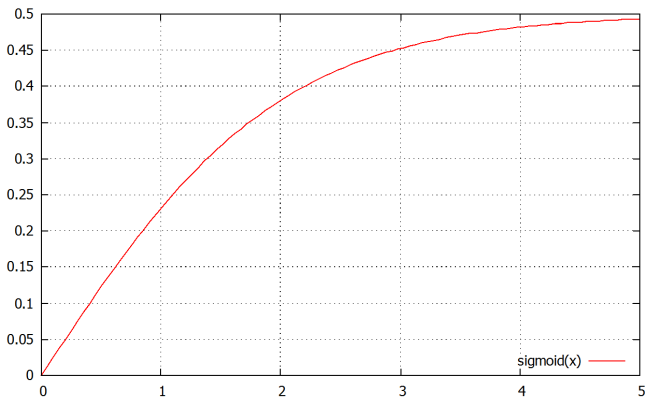

Figura 3. Gráfico da função sigmóide utilizada para definir novos valores para ε.

VII. Experimentos e Resultados

Esta seção apresenta os experimentos realizados para comparar numericamente os métodos ε-*greedy* e ε-*greedy*

adaptativo para o problema *n-armed bandit*. Implementou-se os experimentos em linguagem Java, utilizando o arcabouço *TeachingBox*[2]. Este arcabouço oferece diversas ferramentas e algoritmos de aprendizagem por reforço, inclusive o método ε-*greedy* utilizado no experimento.

Para a implementação do algoritmo ε-*greedy* adaptativo criou-se a classe *EpsilonGreedyPolicyAdapt* no ambiente do *TeachingBox*. Esta classe estende a classe *GreedyPolicy*. A execução dos experimentos utilizou os recursos disponíveis no ambiente.

Nos experimentos comparou-se dois métodos ε-*greedy* ($\varepsilon = 0.1$ e $\varepsilon = 0.5$) com o método ε-*greedy* adaptativo ($\varepsilon = 0.5$). O valor de $\varepsilon = 0.5$ na versão adaptativa é apenas um valor inicial para ε, já que ele será modificado ao longo da execução. Foram testados valores iniciais para ε variando entre 0.1 e 0.9, e os resultados foram similares. Definiu-se adotar o valor inicial de 0.5 por ser o maior valor retornado pela função sigmóide, utilizada para a modificação do ε. Os métodos calculam as estimativas dos valores das ações usando a técnica de média das amostras.

Os experimentos foram realizados com um conjunto de 2000 tarefas geradas aleatoriamente para o problema *n-armed bandit* com $n = 10$. Verificou-se o desempenho e o comportamento dos métodos com a execução de 1000 passos (jogadas). Dois tipos de experimentos são apresentados: uma para o caso estacionário e outro para o caso não estacionário.

A. Caso Estacionário

As Figuras 4 e 5 apresentam os gráficos de comparação dos métodos, conforme descritos acima.

O gráfico da Figura 4 apresenta o aumento da recompensa esperada com a experiência. O método ε-*greedy* adaptativo apresentou desempenho superior aos outros dois métodos. Pode-se verificar que demorou um pouco mais para melhorar no início em relação ao ε-*greedy* com $\varepsilon = 0.1$, porém ao final foi superior a esse método atingindo uma recompensa de cerca de 1.55, enquanto o método ε-*greedy* com $\varepsilon = 0.1$ obteve, ao final, uma recompensa de cerca de 1.45. O método ε-*greedy* com $\varepsilon = 0.5$ obteve uma recompensa em cerca de apenas 0.85.

O gráfico da Figura 5 apresenta que o método ε-*greedy* adaptativo encontra ações ótimas mais cedo e no final atinge um percentual de cerca de 85% de seleção da ação ótima. Enquanto, o método ε-*greedy* com melhor desempenho ($\varepsilon = 0.1$) atingiu, ao final, um percentual de cerca de 80% de seleção da ação ótima.

B. Caso Não Estacionário

Para a realização dos experimentos no caso não estacionário, definiu-se que o ambiente altera a sua configuração após 500 passos (jogadas) de execução. As Figuras 6 e 7 apresentam os gráficos de comparação dos métodos. Logo após a mudança de configuração do ambiente no passo 500, os dois métodos apresentam um desempenho muito baixo, uma vez que o conhecimento anterior obtido está em discordância com

[2]Disponível em: http://servicerobotik.hs-weingarten.de/en/teachingbox.php. Acesso em 25/11/2013.

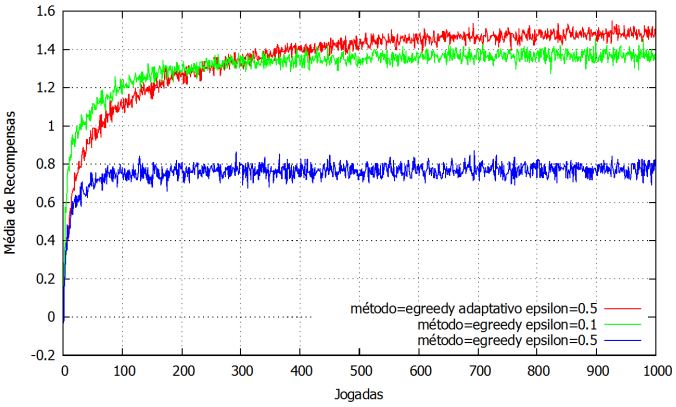

Figura 4. Desempenho médio, no caso estacionário, dos métodos ε-*greedy* e ε-*greedy* adaptativo para o aumento da recompensa esperada.

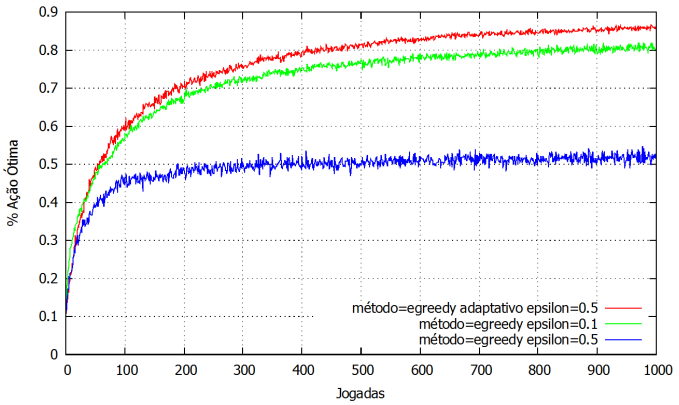

Figura 5. Desempenho médio, no caso estacionário, dos métodos ε-*greedy* e ε-*greedy* adaptativo em selecionar uma ação ótima.

a nova configuração do ambiente. Eles têm que re-aprender essa nova configuração.

O gráfico da Figura 6 apresenta o aumento da recompensa esperada com a experiência. Após a mudança de configuração do ambiente no passo 500, o método ε-*greedy* apresentou desempenho similar ao método ε-*greedy* com $\varepsilon = 0.1$, atingindo uma recompensa de cerca de 1.0. O método ε-*greedy* com $\varepsilon = 0.5$ obteve uma recompensa em cerca de apenas 0.55.

O gráfico da Figura 7 mostra que os métodos ε-*greedy* adaptativo e ε-*greedy* com $\varepsilon = 0.1$ apresentam resultados similares após a alteração da configuração do ambiente. Ambos os métodos, ao final dos 1000 passos, seleciona-se cerca de 40% das vezes a ação ótima.

VIII. Conclusão

Um dos grandes desafios da área de aprendizagem por reforço é o balanceamento entre *exploration* e *exploitation*. O ε-*greedy* é um método simples para tal balanceamento, entretanto o valor de ε é estático. Neste trabalho apresentou-se o método ε-*greedy* adaptativo. Ele possibilita que o valor de ε seja dinâmico, adaptando-se ao comportamento do ambiente.

Os experimentos realizados mostraram que no caso estacionário o método ε-*greedy* adaptativo teve melhor desempenho em relação ao método ε-*greedy*. A versão adaptativa

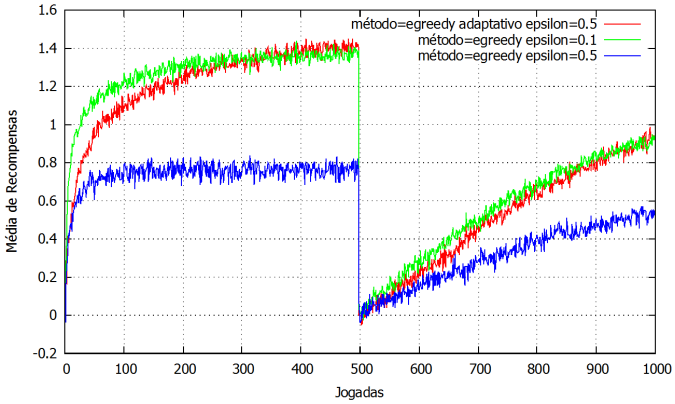

Figura 6. Desempenho médio, no caso não estacionário, dos métodos ε-*greedy* e ε-*greedy* adaptativo para o aumento da recompensa esperada.

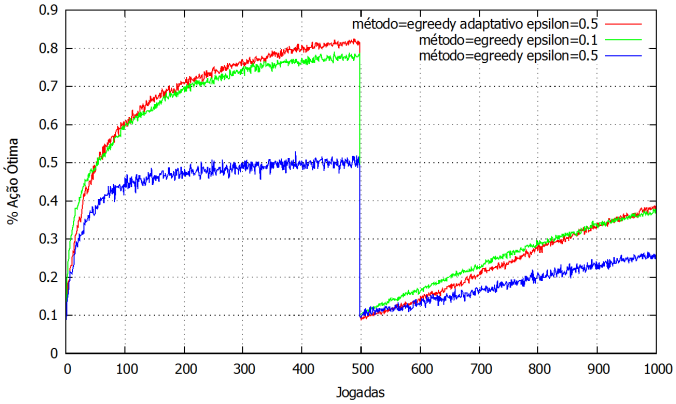

Figura 7. Desempenho médio, no caso não estacionário, dos métodos ε-*greedy* e ε-*greedy* adaptativo em selecionar uma ação ótima.

encontra mais cedo uma ação próxima da ótima e também seleciona a ação ótima uma porcentagem maior de vezes, em torno de 5% a mais. No caso não-estacionário o método ε-*greedy* adaptativo obteve resultados similares ao método ε-*greedy* após a mudança de configuração do ambiente.

Como trabalhos futuros pretende-se estudar novas políticas adaptativas de aprendizagem para melhorar o desempenho do método ε-*greedy* adaptativo para o caso não estacionário.

REFERÊNCIAS

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[2] J. J. Neto, "A small survey of the evolution of adaptivity and adaptive technology," *Revista IEEE América Latina*, vol. 5, no. 7, pp. 496–505, 2007, (in Portuguese).

[3] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Prentice Hall, 2002.

[4] T. M. Mitchell, *Machine Learning*. McGraw-Hill, 1997.

[5] J. C. Luz, "Tecnologia adaptativa aplicada à otimização de código em compiladores," Master's thesis, Escola Politécnica, Universidade de São Paulo, 2004.