ANÁLISE DO DESEMPENHO DO APRENDIZADO POR REFORÇO NA SOLUÇÃO DO PROBLEMA DO CAIXEIRO VIAJANTE

André Luiz Carvalho Ottoni*, Erivelton G. Nepomuceno*, Lara Toledo Cordeiro*, Rubisson Duarte Lamperti†, Marcos Santos de Oliveira*

* *Universidade Federal de São João del-Rei*
São João del-Rei, MG, Brasil

† *Universidade Tecnológica Federal do Paraná*
Medianeira, PR, Brasil

Emails: andreottoni@ymail.com, nepomuceno@ufsj.edu.br, lara1993gv@hotmail.com, duartelamperti@yahoo.com.br, mso@ufsj.edu.br

Abstract— The choice of learning rate (α) and the stock selection policy $\epsilon - greedy$ affects the performance of the reinforcement learning algorithm. Thus, this study aims to investigate the influence of α and policy $\epsilon - greedy$ in reinforcement learning performance by applying the Q-learning in optimization studies of the Traveling Salesman Problem cases. Thus, the methodology proposed in this paper allows you to challenge the Q-learning, assessing the impact generated by the initial conditions of learning rate and $\epsilon - greedy$.

Keywords— Reinforcement Learning, Traveling Salesman Problem, Q-learning.

Resumo— A escolha da taxa de aprendizado (α) e da política de seleção de ações $\epsilon - greedy$ pode afetar profundamente o desempenho do algoritmo de aprendizado por reforço. Dessa forma, este trabalho visa investigar a influência de α e da política $\epsilon - greedy$ no desempenho do aprendizado por reforço, aplicando o Q-learning na solução de estudos de casos do Problema do Caixeiro Viajante. Assim, a metodologia proposta neste trabalho permite desafiar o Q-learning, avaliando qual o impacto gerado pelas condições iniciais de taxa de aprendizado e $\epsilon - greedy$.

Palavras-chave— Aprendizado por Reforço, Problema do Caixeiro Viajante, Q-learning.

1 Introdução

No aprendizado por reforço (AR), a ideia principal se baseia na otimização da tomada de decisão via retornos de sucessos e fracassos no ambiente.

Um dos algoritmos de AR mais conhecidos é o Q-learning (Watkins, 1989; Watkins e Dayan, 1992; Sutton e Barto, 1998). No Q-learning, a taxa de aprendizado (no inglês, *step-size* ou *learning rate*), denotado por α, pode assumir qualquer valor entre 0 e 1. Além disso, α pode ser constante, ou decair ao longo do tempo. O parâmetro α controla a velocidade em que a informação nova é acumulada pelo algoritmo de AR e está entre os parâmetros ajustáveis mais difundidos na aprendizagem de máquina (Dabney, 2014). Dessa forma, os algoritmos de aprendizado por reforço são muitas vezes mais sensíveis ao valor da taxa de aprendizado do qualquer outro parâmetro. Assim, a escolha do parâmetro α pode afetar profundamente o desempenho da aprendizagem do algoritmo (Dabney, 2014).

Já quanto a política de seleção de ações adotada para os algoritmos de AR, a mais simples é a gulosa (*greedy*). A política gulosa sempre escolha a ação mais bem estimada até o momento (Sutton e Barto, 1998). Outra política é a $\epsilon - greedy$, que seleciona com probabilidade ϵ, ações de forma aleatória. O método $\epsilon - greedy$ pode ter melhor desempenho que a política (*greedy*), já que ao selecionar ações aleatórias, o AR continua a explorar o ambiente, aumentando as chances de encontrar a política ótima (Sutton e Barto, 1998).

Para a resolução de problemas que devem ter decisões inteligentes são frequentemente usados métodos de otimização combinatória, que consistem em maximizar ou minimizar uma função definida sobre certo domínio finito. Essa técnica baseada em programação matemática e na computação evolucionária se encontra em alguns algoritmos conhecidos como: Algoritmos gulosos, Algoritmo Simplex, Programação Dinâmica, Algoritmos Genéticos, Algoritmo de Colônia de Formigas, Redes Neurais e Aprendizado por Reforço (Lima Júnior, 2009; Santos, 2009; Masutti e Castro, 2007; Silveira et al., 2011; Silva et al., 2013).

Uma das aplicações mais conhecidas de otimização combinatória é o Problema do Caixeiro Viajante (PCV). Seu objetivo é definir a menor rota entre n cidades. Assim, o caixeiro deve passar por todas as cidades uma única vez, exceto aquela na qual se inicia e termina a jornada (Silveira et al., 2011).

O aprendizado por reforço vem sendo aplicado na solução do Problema Caixeiro Viajante. Em (Gambardella e Dorigo, 1995), os autores apresentaram o algoritmo Ant-Q, que integra conceitos do Algoritmo de Colônia de Formigas e ideias do Q-learning. Já nos trabalhos de (Lima Júnior, 2009; Santos, 2009), o AR é adotado em conjunto com as técnicas de Algoritmos Genéticos e GRASP. Em (Bianchi, 2004), por sua vez, téc-

nicas de aceleração por heurísticas são adotadas para acelerar o aprendizado por reforço na resolução de estudos de caso do PCV.

Dessa forma, este trabalho visa investigar a influência da taxa de aprendizado (α) e da política $\epsilon-greedy$ no desempenho do aprendizado por reforço, aplicando o Q-learning na solução de estudos de casos do problema do caixeiro viajante. Assim, o objetivo é aplicar o AR no PCV, não para encontrar a solução ótima, mas para desafiar o Q-learning, verificando como seu desempenho é afetado pelas condições iniciais de taxa de aprendizado e $\epsilon - greedy$.

Este artigo está organizado em seções. Na seção 2, são definidos conceitos teóricos do aprendizado por reforço. Na seção 3, o problema do caixeiro viajante é descrito. Já na seção 4, é apresentada a modelagem da estratégia de aprendizagem, por meio das definições das ações, dos estados, modelagem de recompensa. Uma descrição dos experimentos realizados é apresentada na seção 5. A análise dos resultados obtidos é apresentada na seção 6. Finalmente, na seção 7 são apresentadas as conclusões.

2 Aprendizado por Reforço

2.1 Algoritmo Q-learning

O algoritmo Q-learning, proposto por Watkins (Watkins, 1989), é um método de aprendizado por reforço usado para propósitos de controle. A ideia básica do Q-learning é que o algoritmo aprende uma função de avaliação ótima sobre todo o espaço de pares estado-ação $S \times A$. Quando a função ótima Q for aprendida, o agente saberá qual ação resultará na maior recompensa em uma situação particular futura (Monteiro e Ribeiro, 2004).

A função $Q(s, a)$ de recompensa futura esperada ao se escolher a ação a no estado s, é aprendida por meio de tentativas e erros segundo a equação (1):

$$Q_{t+1} = Q_t(s_t, a_t) + \alpha[r_t + \gamma V_t(s_{t+1}) - Q_t(s_t, a_t)] \quad (1)$$

em que:

- α: taxa de aprendizagem;
- r_t: é a recompensa imediata;
- γ é o fator de desconto;
- $V_t(s_{t+1}) = max_a Q(s_{t+1}, a_t)$ é a utilidade do estado s.

O Q-learning é retratado no Algoritmo 1 (Watkins e Dayan, 1992).

Para cada ***s,a*** inicialize ***Q(s,a)***=0
Observe o estado ***s***
repita
  Selecione a ação ***a*** usando a política ε-gulosa
  Execute a ação ***a***
  Receba a recompensa imediata ***r(s,a)***
  Observe o novo estado ***s'***
  Atualize o item ***Q(s,a)*** de acordo com a equação (1)
  s $\leftarrow$ ***s'***
até *o critério de parada ser satisfeito*;

Algoritmo 1: Algoritmo Q-learning.

2.1.1 Taxa de Aprendizado (α)

No algoritmo Q-learning, a taxa de aprendizado pode ser definida em qualquer valor no intervalo entre 0 e 1, $0 \leq \alpha \leq 1$ (Sutton e Barto, 1998). Vale ressaltar que, para $\alpha = 0$ não existe aprendizado, já que a atualização da Equação (1) se simplica em $Q_{t+1} = Q_t(s_t, a_t)$.

Uma condição de convergência do Q-learning (Mitchell, 1997), estabelece que cada par estado-ação (s, a) deve ser visitado infinitas vezes, com $0 \leq \alpha < 1$, e atendendo as Equações (2) e (3):

$$\sum_{n=1}^{\infty} \alpha_n(s, a) = \infty, \quad (2)$$

$$\sum_{n=1}^{\infty} [\alpha_n(s, a)]^2 < \infty. \quad (3)$$

Uma maneira de satisfazer essas duas condições, é decaindo o valor da taxa de aprendizado de acordo com o número de acessos a cada par (s, a). Para isso, (Mitchell, 1997) estabeleceu a Equação (4):

$$\alpha_n(s, a) = \frac{1}{1 + visitas_n(s, a)}, \quad (4)$$

onde, n é o número de visitas ao par estado-ação (s, a).

Em problemas não-estacionários, faz sentido ponderar recompensas recentes mais fortemente do que as do passado, já que o ambiente é dinâmico e se altera ao longo do tempo. Uma forma de fazer isso é adotar a taxa de aprendizado constante (α_k) (Sutton e Barto, 1998). Nesse caso, a Equação (3) não é satisfeita. Dessa forma, o sistema pode nunca completar a convergência, pois continua variando em reposta às recompensas recentes (Sutton e Barto, 1998).

2.1.2 Política $\epsilon - greedy$

No Aprendizado por Reforço, a política gulosa (*greedy*) seleciona a ação mais bem estimada até o momento. Esse método pode facilmente encontrar uma solução local e ficar preso nesse ponto

indefinidamente. Uma alternativa para esse problema é selecionar ações de forma aleatória com pequena probabilidade ϵ. Esse método é denomidado $\epsilon - greedy$ (ou, quase-guloso) (Sutton e Barto, 1998).

Uma vatagem do método $\epsilon - greedy$, é que com o número de episódios tendendo ao infinito, cada ação será tomada infinitas vezes, garantindo a condição de convergência para a política ótima (ou, quase ótima) (Sutton e Barto, 1998).

3 Problema do Caixeiro Viajante

Nos últimos 50 anos, o Problema do Caixeiro Viajante (PCV) tem sido amplamente estudado. Desde as técnicas clássicas a abordagens baseadas em busca tabu, redes neurais e algoritmos genéticos, o PCV tem sido adotado como estudo de caso por muitos algoritmos de otimização combinatória (Santos, 2009).

O problema é definido como sendo um conjunto de cidades (c_1, c_2, c_3, ..., c_N) e para cada par de cidades distintas (c_i; c_j), uma distância d(c_i; c_j). Com o objetivo de determinar a menor rota, passando por todas as cidades e retornando ao final da trajetória para a cidade inicial (Santos, 2009). A Equação (5) representa a quantidade que o PCV busca minimizar:

$$\sum_{i=1}^{N-1} = (d(c_{n(i)}, c_{n(i)}) + (d(c_{n(N)}, c_{n(1)}). \qquad (5)$$

Uma forma de classificar os algoritmos com base no tempo de execução e na quantidade de recursos computacionais é a "Teoria da Complexidade". Nesse aspecto, a literatura estabelece que o PCV é NP-Completo no caso geral e NP-Difícil em alguns casos especiais (Santos, 2009).

Os experimentos computacionais foram realizados utilizando a biblioteca TSPLIB[1], que possui diversas opções para aplicações de estudos de caso (instâncias) do Problema do Caixeiro Viajante.

4 Modelagem do Sistema de Aprendizado por Reforço

A metodologia adotada para o desenvolvimento da estratégia de aprendizagem é dividida em quatro etapas:

1. Definição do conjunto finito de estados do ambiente: Nesse caso, os estados são todas as localidades em que o caixeiro viajante (agente) deve acessar.

2. Definição do conjunto finito de ações que o agente pode realizar: Cada ação foi definida como sendo intenção de ir para outra localidade (estado) do problema. Vale ressaltar que, para evitar a repetição de localidades na rota, as ações que levem aos estados já visitados não devem estar disponíveis (Lima Júnior, 2009).

3. Definição dos valores dos reforços, para cada par estado (s) $\times$ ação (a): Os reforços foram definidos como as distâncias entre as localidades multiplicada por -1. Assim, quanto maior a distância, mais negativo é o reforço. Dessa forma, espera-se que o agente procure encontrar a distância mais curta entre duas localidades para diminuir a penalidade.

4. Aplicação do algoritmo de aprendizado por reforço Q-learning no simulador desenvolvido: Foi desenvolvido um simulador no software $MATLAB^{®}$ para realizar os experimentos.

5 Experimentos Realizados

Os experimentos tiveram como objetivo principal investigar a influência da taxa de aprendizado (α) e a política $\epsilon - greedy$ no desempenho do AR na solução do Problema do Caixeiro Viajante.

Dessa maneira, busca-se entender como esses parâmetros podem interferir nos resultados do aprendizado por reforço no ambiente em estudo. Para isso, o AR foi simulado com 3 distintos valores para a política $\epsilon - greedy$: 0, 0,01 e 0,1. Assim, o sistema é:

- 100% guloso para $\epsilon = 0$;
- 99% guloso para ϵ = 0,01;
- 90% guloso para ϵ = 0,1.

Além disso, a taxa de aprendizado foi analisada sobre dois paradigmas:

- α_k = 0,99, taxa de aprendizado constante;
- $\alpha_n(s, a)$, taxa de aprendizado decaindo durante o processo de AR de acordo com a Equação 4.

Quanto a fator de desconto (γ), foi fixado em 0,01 em todas as simulações.

Foram realizados testes com o caixeiro viajante adotando três instâncias da TSPLIB, conforme Tabela 1:

Tabela 1: Problemas da TSPLIB estudados.

Problema	Cidades	Solução Ótima
Berlin52	52	7542
Brazil58	58	25395
Kroa100	100	21282

Vale ressaltar que, cada simulação foi padronizada em cinco épocas com 1000 episódios. Sendo que, cada episódio teve como resposta a distância total percorrida pelo agente na rota da instância.

[1] http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/

6 Análise dos Resultados

Em seguida, a análise de desempenho do sistema de aprendizado por reforço para cada uma das três instâncias da TSPLIB estudadas: Berlin52, Brazil58 e Kroa100.

6.1 Resultados com 52 localidades - Berlin52

Para a instância Berlin52, a menor distância calculda foi 8009. Esse valor foi encontrado quando adotado $\epsilon = 0{,}01$ e $\alpha_n(s,a)$, e também para $\epsilon = 0{,}1$ e $\alpha_n(s,a)$. A Tabela 2 apresenta os resultados mínimos encontrados de acordo com a taxa de aprendizado e a política $\epsilon - greedy$.

Tabela 2: Menor distância calculada para a instância Berlin52.

$\epsilon - greedy$	α_k	$\alpha_n(s,a)$
0	9148	8146,3
0,01	8009,5	8009
0,1	8037	8009

Vale ressaltar que, a combinação entre a taxa de aprendizado constante (α_k) e o sistema 100% guloso ($\epsilon = 0$), levou a convergência para o valor 9148, distância muito superior ao valor encontrado pelas demais combinações. Pela Figura 1, é possivel visualizar essa estabilização do sistema adotando α_k e $\epsilon = 0$.

Já a Figura 2, apresenta e evolução da distância calculada, adotando $\alpha_n(s,a)$ para os três valores da política $\epsilon - greedy$.

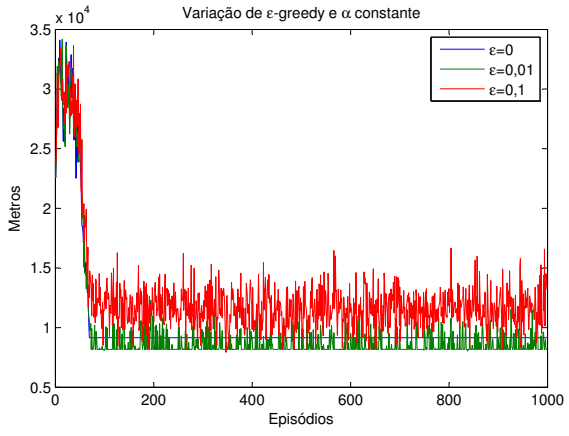

Figura 1: Distância calculada versus o episódio de aprendizado, para a instância Berlin52 do TSP, com α_k=0,99 (constante).

As Figuras 3 à 5, mostram o comportamento do AR, a partir dos caminhos encontrados para a instância Berlin52, para o ínicio do aprendizado (Figura 3), melhor solução para o sistema 100% guloso e taxa de aprendizado constante (Figura 4), e melhor solução para o sistema 100% guloso e taxa de aprendizado decaindo (Figura 5).

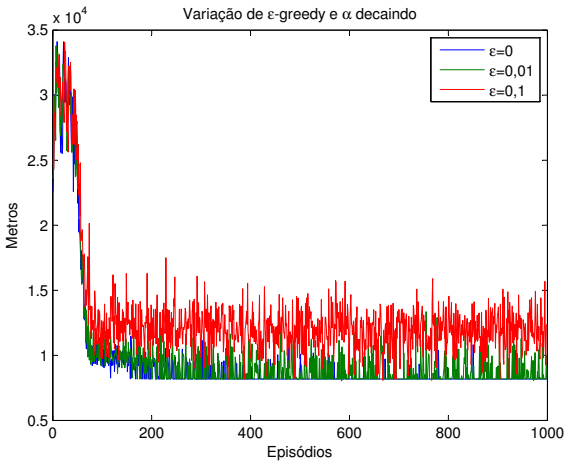

Figura 2: Distância calculada versus o episódio de aprendizado, para a instância Berlin52 do TSP, com $\alpha_n(s,a)$ (decaindo).

6.2 Resultados com 58 localidades - Brazil58

Para a instância Brazil58, a menor distância calculda foi 27753. Esse valor foi encontrado quando adotado $\epsilon = 0{,}01$ e $\alpha_n(s,a)$. A Tabela 3 apresenta os resultados mínimos encontrados de acordo com a taxa de aprendizado e a política $\epsilon - greedy$.

Tabela 3: Menor distância calculada para a instância Brazil58.

$\epsilon - greedy$	α_k	$\alpha_n(s,a)$
0	31223	27922
0,01	27922	27753
0,1	28930	28117

6.3 Resultados com 100 localidades - Kroa100

Para a instância Kroa100, a menor distância calculda foi 25748. Esse valor foi encontrado quando adotado $\epsilon = 0{,}01$ e $\alpha_n(s,a)$. A Tabela 4 apresenta os resultados mínimos encontrados de acordo com a taxa de aprendizado e a política $\epsilon - greedy$.

Tabela 4: Menor distância calculada para a instância Kroa100.

$\epsilon - greedy$	α_k	$\alpha_n(s,a)$
0	28437	26291
0,01	26268	25748
0,1	28832	28944

As Figuras 6 à 8 mostram o comportamento do AR, a partir dos caminhos encontrados para a instância Kroa100, para o ínicio do aprendizado (Figura 6), melhor solução para o sistema 100% guloso e taxa de aprendizado constante (Figura 7), e melhor solução para o sistema 100% guloso e taxa de aprendizado decaindo (Figura 8).

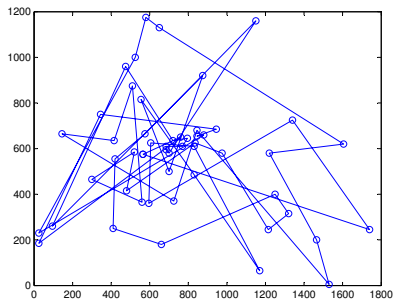

Figura 3: Caminho definido com a distância percorrida igual a 22561 para a instância Berlin52 do TSP, para o primeiro episódio do aprendizado, com ϵ=0 e α_k=0,99 (constante).

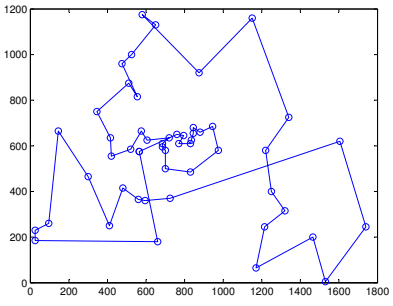

Figura 4: Caminho definido com a distância percorrida igual a 9148 para a instância Berlin52 do TSP, com ϵ=0 e α_k=0,99 (constante).

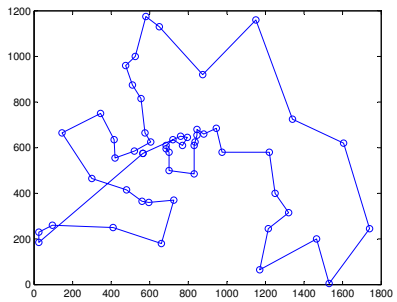

Figura 5: Caminho definido com a distância percorrida igual a 8146,3 para a instância Berlin52 do TSP, com ϵ=0 e $\alpha_n(s,a)$ (decaindo).

6.4 Resultados Gerais

A Tabela 5 resume os melhores resultados encontrados neste trabalho para cada problema.

Tabela 5: Melhor solução encontrada para cada problema estudado.

Problema	**Melhor Solução**	ϵ	α
Berlin52	8009	0,01/0,1	$\alpha_n(s,a)$
Brazil58	27753	0,01	$\alpha_n(s,a)$
Kroa100	25748	0,01	$\alpha_n(s,a)$

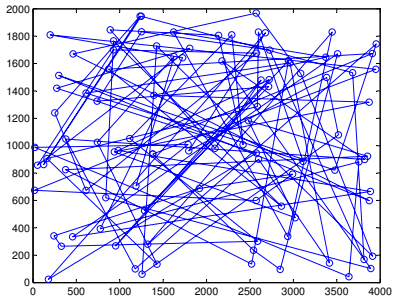

Figura 6: Caminho definido com a distância percorrida igual a 189260 para a instância Kroa100 do TSP, para o primeiro episódio do aprendizado, com ϵ=0 e α_k=0,99 (constante).

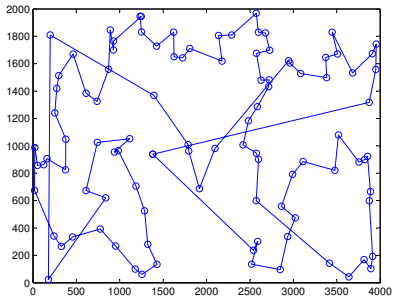

Figura 7: Caminho definido com a distância percorrida igual a 28437 para a instância Kroa100 do TSP, com ϵ=0 e α_k=0,99 (constante).

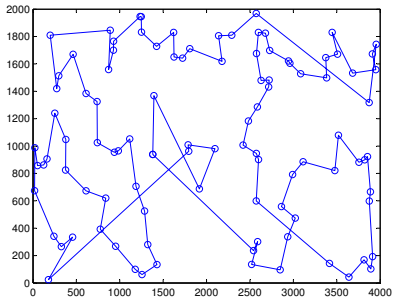

Figura 8: Caminho definido com a distância percorrida igual a 26291 para a instância Kroa100 do TSP, com ϵ=0 e $\alpha_n(s,a)$ (decaindo).

7 Conclusão

Este trabalho teve como objetivo principal estudar os efeitos da taxa de aprendizado (α) e a política $\epsilon - greedy$ sobre o desempenho do aprendizado por reforço. Para isso, o algoritmo Q-learning foi aplicado na solução do problema do caixeiro viajante. Assim, a metodologia proposta neste trabalho permite desafiar o Q-learning, avaliando qual o impacto gerado pelas condições iniciais de taxa de aprendizado e $\epsilon - greedy$.

O método guloso (ϵ=0) pode facilmente encontrar uma solução local e ficar preso nesse

ponto. O método $\epsilon = 0{,}1$ explora mais, mas nunca seleciona a ação mais bem estimada por mais de 91% do tempo. Já o método $\epsilon = 0{,}01$ teve um melhor desempenho do que os outros dois valores para $\epsilon - greedy$. Assim, a utilização da política $\epsilon - greedy$, com $\epsilon = 0{,}01$, mostrou-se o método mais eficiente para os estudos de casos.

Quanto a taxa de aprendizado, em quase todos os casos $\alpha_n(s, a)$ (decaindo ao longo do tempo) apresentou rendimento superior a α_k (constante).

Vale ressaltar, que os valores mínimos encontrados não correspondem aos valores ótimos já calculados para as instâncias TSP estudadas. No entanto, vale lembrar que o problema do caixeiro viajante é complexo, por isso é interessante adotar outras técnicas em conjunto ao algoritmo Q-learning para chegar a esses valores, como Algoritmos Genéticos (Santos, 2009; Lima Júnior, 2009), GRASP (Santos, 2009; Lima Júnior, 2009) e Aceleração por Heurísticas (Bianchi, 2004).

Em trabalhos futuros, pretende-se verificar os efeitos do método $\epsilon - greedy$ adaptativo (Lima Júnior, 2009), e também analisar a influência na definição de outros valores para a taxa de aprendizado constante. Além disso, o Q-learning será aplicado em instâncias com maior número de cidades, aumentando assim a dificuldade na resolução do problema. Pretende-se também adotar metodologias estatísticas mais elaboradas na análise de desempenho do aprendizado reforço na solução do PCV (Ottoni et al., 2012; Ottoni et al., 2015).

Agradecimentos

Agradecemos à CAPES, CNPQ, FAPEMIG, UFSJ, UTFPR e PPGEL (Associação Ampla UFSJ/CEFET-MG) pelo apoio.

Referências

Bianchi, R. A. C. (2004). *Uso de heurística para a aceleração do aprendizado por reforço.*, Master's thesis, Tese (Doutorado) Escola Politécnica da Universidade de São Paulo.

Dabney, W. (2014). *Adaptive step-sizes for reinforcement learning*, Master's thesis, University of Massachusetts Amherst.

Gambardella, L. M. e Dorigo, M. (1995). Ant-q: A reinforcement learning approach to the traveling salesman problem, *Proceedings of the 12th International Conference on Machine Learning* .

Lima Júnior, F. C. (2009). *Algoritmo q-learning como estratégia de exploração e/ou explotação para as mataheurísticas grasp e algoritmo genético*, Master's thesis, Tese (Doutorado), Programa de Pós-Graduação em Eng. Elétrica e de Computação da UFRN.

Masutti, T. A. S. e Castro, L. N. (2007). Uma abordagem neuro-imune para a solução do problema de múltiplos caixeiros viajantes, *XIII SBAI - Simpósio Brasileiro de Automação Inteligente* .

Mitchell, T. M. (1997). *Machine Learning*, McGraw-Hill Science.

Monteiro, S. T. e Ribeiro, C. H. C. (2004). Desempenho de algoritmos de aprendizagem por reforço sob condições de ambiguidade sensorial em robótica móvel, *Revista Controle & Automação* **Vol.15 no.3**.

Ottoni, A. L. C., Lamperti, R. D., Nepomuceno, E. G. e Oliveira, M. S. (2012). Desenvolvimento de um sistema de aprendizado por reforço para times de robôs - uma análise de desempenho por meio de testes estatísticos, *XIX Congresso Brasileiro de Automática, ISBN 978-85-8001-069-5* pp. 3557–3564.

Ottoni, A. L. C., Oliveira, M. S., Nepomuceno, E. G. e Lamperti, R. D. (2015). Análise do aprendizado por reforço via modelos de regressão logística: Um estudo de caso no futebol de robôs, *Revista Junior de Iniciação Científica em Ciências Exatas e Engenharia* **10**: 44–49.

Santos, J. P. Q. (2009). *Uma implementação paralela híbrida para o problema do caixeiro viajante usando algoritmos genéticos, grasp e aprendizagem por reforço*, Master's thesis, Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da UFRN.

Silva, D. J. A., Silva, J. A. L., Affonso, C. M. e Oliveira, R. C. L. (2013). Uso de algoritmo cultural com uma nova abordagem memética por meio do simulated annealing para o problema do caixeiro viajante, *XI SBAI - Simpósio Brasileiro de Automação Inteligente* .

Silveira, L. R., Tanscheit, R. e Vellasco, M. (2011). Algoritmos genéticos com inspiração quântica aplicados a problemas de otimização combinatória de ordem, *X SBAI - Simpósio Brasileiro de Automação Inteligente* .

Sutton, R. e Barto, A. (1998). *Reinforcement Learning: An Introduction*, 1st edn, Cambridge, MA: MIT Press.

Watkins, C. J. (1989). *Models of delayed reinforcement learning*, Master's thesis, PhD thesis, Psychology Department, Cambridge University, Cambridge, United Kingdom.

Watkins, C. J. e Dayan, P. (1992). Technical note q-learning, *Machine Learning* .