



## Aplicação de Técnicas *Q-Routing* no Planejamento de Caminhos de Robôs Móveis

Universidade Federal de Viçosa

Departamento de Engenharia Elétrica

Iure Rosa Lima de Oliveira (iure.oliveira@ufv.br) – Estudante de Engenharia Elétrica; Alexandre Santos Brandão (alexandre.brandao@ufv.br) – Departamento de Engenharia Elétrica

Palavras Chave: *Q-Routing*, *Reinforcement Learning*, Robótica Móvel.

Área Temática: Inteligência Artificial | Grande Área: Ciências Exatas e Tecnológicas

Projeto de Pesquisa

### Introdução

Os robôs móveis autônomos estão cada vez mais interessantes, pois podem tomar decisões por conta própria ao executar ações em um determinado ambiente. Este comportamento popularizou sua utilização em tarefas não supervisionadas, onde o robô aprenderá por conta própria e executará uma série de ações para atingir o objetivo predefinido. Desta forma, o planejamento de caminhos é crucial para a navegação de robôs móveis.

### Objetivos

Desenvolvimento de um algoritmo baseado em *Reinforcement Learning* capaz de realizar a leitura de uma matriz de dados contendo as informações extraídas do ambiente e, através dela, resolver problemas de localização de caminho.

### Material e Métodos

O *V-REP* é um software desenvolvido para criação e simulação de robôs para diversas aplicações. Neste trabalho, foi utilizado o robô *Pioneer 3DX* já disponível na plataforma. No ambiente de simulação (Figura 1) estão dispostos obstáculos, tal como, discos verdes cujo o único intuito é representar visualmente os nós (pontos livres) extraídos do mapa.

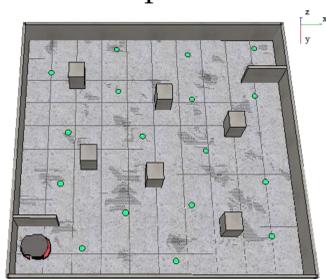


Figura 1 – Ambiente de Simulação.

Devido ao grande leque de ferramentas presentes no *Python*, ele foi escolhido para realizar a extração de dados e, posteriormente, aplicar uma política de ações estabelecida pelo algoritmo. Segue, abaixo, o código em *Python* para o *Q-Routing*.

```
def update_Q(T,Q,current_state, next_state, alpha):  
    current_t = T[current_state][next_state]  
    current_q = Q[current_state][next_state]  
    new_q = current_q + alpha * (current_t + min(Q[next_state].values()) - current_q)  
    Q[current_state][next_state] = new_q  
    return Q
```

Figura 2 – Código em *Python* para o *Q-Routing*.

### Agradecimentos

O autor agradece ao Prof. Dr. Alexandre Santos Brandão e aos membros do Núcleo de Especialização em Robótica (NERO).

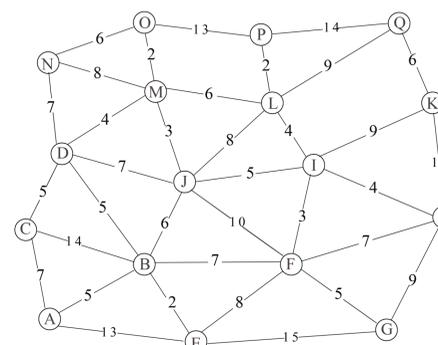


Figura 3 – Rede de Pontos Livres.

Após a extração dos dados do ambiente, as rotas livres de obstáculos são armazenadas em uma matriz de dados que compõe a rede de pontos livres (Figura 3), essa matriz é lida pelo algoritmo que, em seguida, retorna as rotas ótimas baseado no custo entre dois nós interligados.

### Resultados e Discussão

Alguns algoritmos de caminhos mais curtos são o BFS (*Breadth First Search*), DFS (*Depth First Search*) e o algoritmo de *Dijkstra*. No entanto, BFS e DFS são muito lentos e sofrem um aumento exponencial de tempo conforme a árvore do gráfico se torna cada vez mais profunda. Para o algoritmo de *Dijkstra*, é rápido fornecer um caminho mais curto, porém fornecerá apenas um, por exemplo, se houver 15 rotas ótimas em um grafo ponderado, o algoritmo de *Dijkstra* retornará apenas uma das melhores rotas. O algoritmo desenvolvido neste trabalho retorna todas as rotas ótimas encontradas, (Figura 4).

Na Figura 4 estão dispostas as rotas ótimas encontradas pelo algoritmo, tal como o tempo percorrido para encontrá-las. O algoritmo foi testado em redes maiores e ainda se mostrou eficiente.

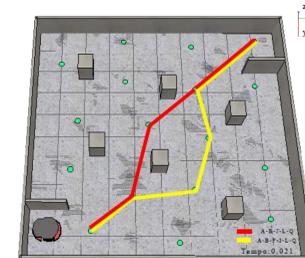


Figura 4 – Rotas Encontradas

### Conclusões

O algoritmo desenvolvido, além de fornecer as rotas ótimas em um curto intervalo de tempo, possibilita que o robô decida qual rota seguir baseado em alguma métrica previamente estabelecida, sendo essa última observação um trabalho futuro.

### Bibliografia

- [1] Khan, M. U. Mobile robot navigation using reinforcement learning in unknown environments. *Balkan Journal of Electrical and Computer Engineering*, 7(3):235–244.
- [2] Khrijji, L., Touati, F., Benhmed, K., y Al-Yahmedi, A. Mobile robot navigation based on q-learning technique. *International Journal of Advanced Robotic Systems*,8(1):4, 2011.



QR Code para o vídeo da simulação.

