

Comparando planejamento e aprendizado por reforço em ambientes com obstáculos

Daniel Orlandi Maurício Pires

Instituto de Computação
Universidade Federal Fluminense

Junho, 2018

Outline

1 Planejamento baseado em amostras

2 Aprendizado por reforço

3 Experimentos e ferramentas

4 Resultados

Planejamento baseado em amostras

- Estado totalmente observável
- Consiste em gerar amostras aleatórias e validar se essas amostras representam uma colisão
- As amostras são então combinadas de alguma forma para calcular caminhos livres de colisão
 - Grafos e árvores
 - RRT utiliza árvore enraizada na origem
 - Variações para melhor custo heurístico e uso de duas árvores

Aprendizado por reforço

- Ramo do Aprendizado de máquina que está entre o aprendizado supervisionado e o não supervisionado.
- Basicamente, tentativa e erro
 - Onde uma função de recompensa é responsável por dizer se uma ação foi boa ou ruim
- Pode ser visto como processo de decisão de Markov.

Aprendizado por reforço

- Considera-se a recompensa futura além da atual
 - Utiliza-se o processo de Markov para se calcular a recompensa total para um episódio
- Ambiente estocástico → Ações repetidas não proporcionam a mesma recompensa
 - Fator de desconto
- Quality-Learning → Tem como estratégia maximizar o Discounted Future Reward
 - Utiliza a função de Belman de forma interativa, para chegar ao resultado desejado

Implementação - RL

- A rede Q-net mapeia valores obtidos pelos sensores de proximidade.
 - Valores se tornam um vetor de recompensas
 - Cada coluna no vetor representa uma ação
 - A rede escolhe a ação com maior recompensa
- Exploration-Exploitation → Uso do fator de probabilidade ϵ

Ferramentas

- Aprendizado por reforço
 - V-rep
 - Keras e TensorFlow
 - Scripts em Python

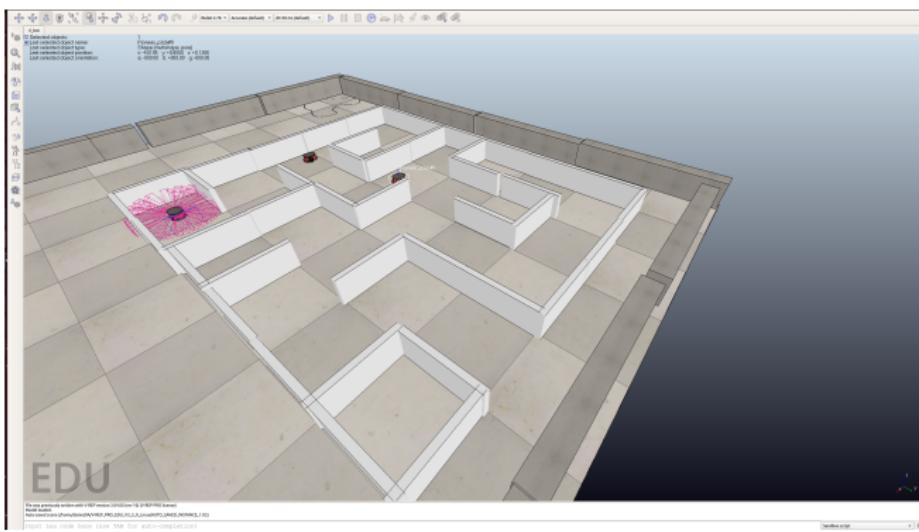
- Planejamento
 - V-rep
 - Open Motion Planning Library (OMPL)
 - Scripts em LUA

Experimentos

- Aprendizado por reforço
 - Robô Pioneer 3dx
 - Duas cenas
 - Acesso em Python via API remota do V-Rep
- Planning
 - Três cenas
 - Robôs abstraídos em blocos
 - Usando OMPL e algoritmo RRT
 - Scripts Lua embutidos nos elementos da cena

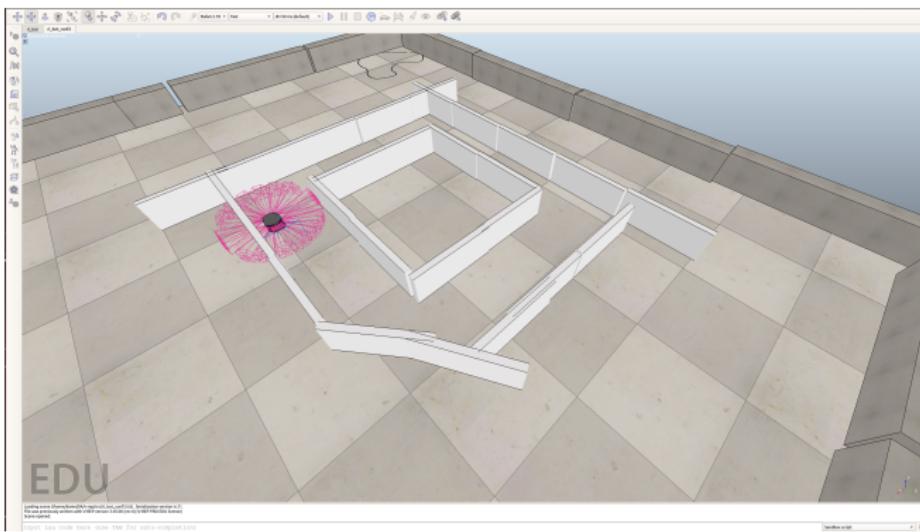
Resultados - RL

Figura: Cena1, mais complexa.



Resultados - RL

Figura: Cena 2, menos complexa



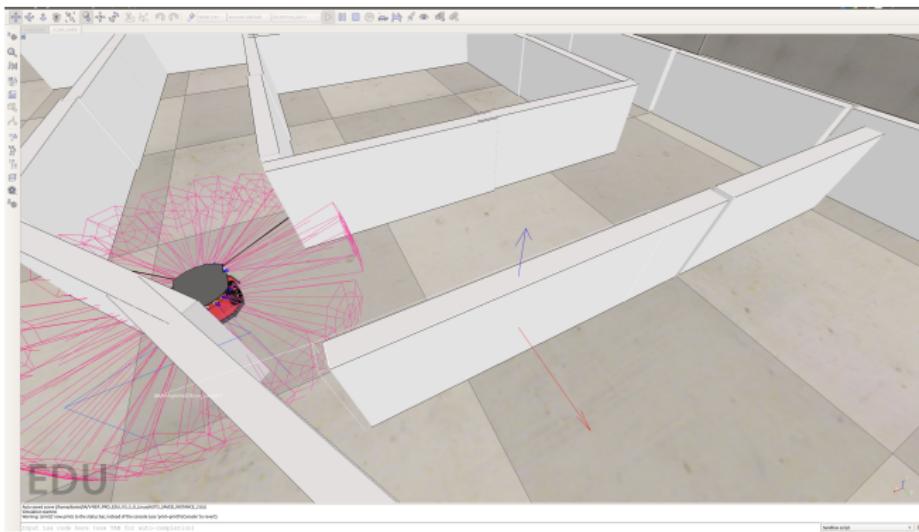
Resultados - RL

Figura: Posição final após primeiro treinamento.



Resultados - RL

Figura: Posição final após segundo treinamento.



Resultados - RL

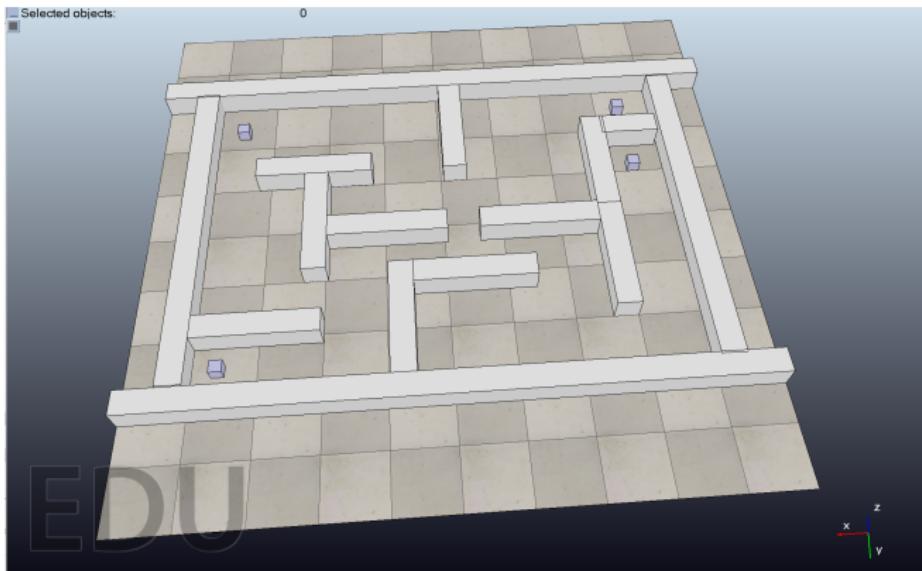
Tabela: Resultados dos experimentos com planejamento.

Treinamento	Cena	Tempo	Épocas	Passos por época	γ	ϵ	Desconto de ϵ
3	Cena 2	9h	150	300	0.99	0.1	-0.001
4	Cena 2	9h	150	300	0.99	0.3	-0.001
5	Cena 2	9h	150	300	0.99	0.2	-0.001
6	Cena 2	9h	150	300	0.99	0.2	-0.01
7	Cena 2	9h	150	300	0.99	0.2	-0.02

- Aumento gradativo do numero de passos → Evitar que o robô acabasse preso por muito tempo
- Aumento da taxa de desconto de ϵ após a Época 50 → Aumentar o conhecimento de mundo do robô.

Resultados - Planning

Figura: Cena extra para o teste do planning.



Resultados - Planning

Tabela: Resultados dos experimentos com planejamento.

Cena	Tempo	Colisões	Objetos
Cena 1	X	X	2
Cena 2	45s	0	1
Cena 3	46s	4	3

Conclusões

Robôs são do mal!



© Marvel Studios

Conclusões

- Path planning é mais rápida que o aprendizado por reforço
 - Até que ponto é eficiente?
- Apesar de sua complexidade de implementação, aprendizado por reforço exige muito menos informações que path planning
- Path planning em robôs devem ser implementados com algum mecanismo para tratar problemas de ambiente dinâmico