Disciplina		Professor		
DCC642 - Introdução à Inteligência Artificial		Luiz Chaimowicz		
Monitor	**Data Entrega**	**Semestre**	**Ano**	**Trabalho**
Thiago Meireles Grabe	02/11/2020	1	2020	TP2 - Aprendizado por Reforço

1 Introdução

Márcio novamente precisa da sua ajuda!

Nesta nova etapa, Márcio percebeu que algoritmos de busca são ferramentas importantes, mas existem técnicas que podem ser exploradas no contexto do AGV coletar a produção das prensas e levar ao almoxarifado o produto acabado para ser embalado e, enfim, faturado para o cliente.

Márcio agora precisa resolver o mesmo problema anterior, mas utilizando Aprendizado por reforço.

2 O Problema

Márcio precisa novamente planejar e definir as rotas de coletas para o AGV. Sabe-se que os veículos adquiridos pela empresa do Márcio podem realizar movimentos em quatro direções: **BAIXO, CIMA, ESQUERDA, DIREITA**. Os AGV's são extremamente rápidos, porém não podem caminhar muito tempo sem se localizar através de um sinal específico. Na prática, andam W ($W > 0$) movimentos sem precisar passar por um ponto de localização e, quando o movimento é o $W + 1$ este movimento deve ser obrigatoriamente para uma posição que contenha o ponto de localização.

O mapa é conhecido, pois a fábrica é estática e foi representado da seguinte forma:

- **'.'** espaços vazios na fábrica e sem perigo de colisões.
- **'*'** são paredes ou obstáculos.
- **'#'** são pontos de localização.
- **'$'** são os pontos de coleta da produção.

As entradas possuem uma linha inicial com as dimensões do galpão (eixos x e y) e o número de passos que o AVG pode dar antes do ponto de reabastecimento, ou seja, se $W = 3$, o terceiro passo obrigatoriamente deve ser para um ponto de localização. Uma modificação da abordagem anterior é que em cada simulação, o AGV é inserido aleatoriamente em uma posição **não terminal**. Os arquivos de entrada exemplo podem ser acessados aqui.

Neste contexto de aprendizado por reforço, Márcio sabe que o mundo precisa ser modelado por um **MDP** que apresenta algumas características:

1. $\mathcal{S}$ é um espaço de estados em que todos os possíveis estados podem estar, seja um espaço vazio, um ponto de localização ou o próprio objetivo.
2. $\mathcal{A}$ é o espaço de ações que contém todas as possíveis ações que um agente pode realizar **BAIXO, CIMA, ESQUERDA, DIREITA**.
3. $r : SxAxS \rightarrow R$, é a função de recompensa. Márcio decidiu definir essa função de uma forma que permanecer no mundo (dar um passo para um ponto vazio) a recompensa é -1. Em contrapartida, dar um passo para um ponto de localização, a recompensa é $+1$.

Contudo, bater em uma parede ou se perder no mundo a recompensa é -10. Chegar ao objetivo a recompensa é $+10$. Formalmente, $r(s, a, s')$ é a recompensa associada a uma transição de estado dada uma determinada ação a. A função recompensa pode ser alterada pensando em um parâmetro de aprendizado, então sinta-se livre para realizar experimentos com valores diferentes.

4. λ define o fator de desconto. Esta valor significa o quanto iremos "descontar" da recompensa ao tomar uma determinada decisão.

5. $T : SxA \rightarrow \Delta(S)$ por fim é função de transição. No contexto deste trabalho, iremos considerar a transição determinística. Em outras palavras, o agente irá realizar a ação desejada sem uma distribuição de probabilidade associada àquela. Formalmente, $T(s'|s, a) = 1$ que significa que a probabilidade de ir para um estado s' dado um estado atual s e uma ação a é igual a 1.

Na configuração do problema os estados podem ser representados pela tupla da posição e número de passos restantes antes de encontrar um ponto de localização: (i, j, f)

3 Tarefa

A sua tarefa nesta nova etapa é implementar o algoritmo *Q-Learning* para auxiliar o Márcio a solucionar o problema na fábrica. Você Deverá escrever um programa que lerá um arquivo com a descrição do mundo e receberá os parâmetros de aprendizado:

- $\alpha \rightarrow$ Taxa de aprendizado;
- $\epsilon \rightarrow$ Fator de exploração da estratégia $\epsilon - greedy$;
- Número N de episódios de treinamento. Um episódio inicia com o agente em um estado aleatório e termina com o agente em um estado terminal;

Um estado terminal é um ponto do mapa em que o AGV encontra uma parede, fica sem localização ou chega ao objetivo.

Lembre-se de balancear exploração e aproveitamento (*exploration vs exploitation*) usando a estratégia adequada.

O seu programa deve gerar um arquivo com os valores da política aprendida para cada estado em um arquivo *pi.txt* com a seguinte formatação: (linha, coluna, passos), ação, valor. O valor deverá conter 2 casas decimais (os valores apresentados não necessariamente representam os que valores ótimos).

Um exemplo da entrada e saída podem ser visualizados na tabela 1

4 O que deve ser entregue:

Você deverá entregar os seguintes arquivos:

- Um *script* **./busca.sh**;

Entrada	Saída
10 19 3 ******************** *..#.*#..#..#.*.#..* *#*******.********#* *..#.#..*.*..#..#..* *#..*...*#*...*...#* *..#....*.*....#...* *#....*.....*.....#* *..#....*#*....#...* *#..*...*.*...*...#* *..#....*$*....#...* ********************	(1, 1, 0), RIGHT, 0.76 (1, 1, 1), UP, 0.21 (1, 1, 2), RIGHT, 0.95 (1, 1, 3), RIGHT, 1.0 (1, 2, 0), LEFT, 0.78 (1, 2, 1), UP, 0.97 (1, 2, 2), UP, 0.96

Table 1: Exemplo de entrada e saída esperada

- Outro *script* **./compila.sh** caso o projeto seja feito em linguagem compilada;
- A documentação em PDF contendo:
  - Breve descrição da implementação;
  - Eventuais decisões de projeto;
  - Gráficos de aprendizado e tabelas comparativas com a recompensa média obtida ao longo do tempo para a sua implementação nas diferentes entradas, e variando a taxa de aprendizado, fator de exploração. Como exemplo, teste α e ϵ com valores $0.2, 0.5$ e 0.9 por exemplo.
  - Discussão dos resultados encontrados.
  - Bibliografia
- Arquivos com o código fonte utilizados no desenvolvimento.

Para a execução do seu trabalho, o *script* a ser executado considerando uma entrada *input.txt*, $\alpha = 0.3$, $\lambda = 0.1$, $\epsilon = 0.9$ e $N = 300$

./qlearning.sh input.txt 0.3 0.9 0.1 300

A sua entrega deve conter:

- *compila.sh* para compilar o seu projeto caso ele seja em linguagem compilada;
- *qlearning.sh* para executar o trabalho;
- documentação.pdf;
- os arquvios de código e, neste caso, podem haver subdiretórios se for necessário.