PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

UTILIZAÇÃO DE PESQUISA EM ÁRVORE DE MONTE-CARLO NO JOGO COLONIZADORES DE CATAN

BRUNO PAZ E GABRIEL RUBIN

Proposta de Trabalho de Conclusão apresentada como requisito parcial à obtenção do grau de Bacharel em Ciência da Computação na Pontifícia Universidade Católica do Rio Grande do Sul.

Orientador: Prof. Felipe Meneguzzi

Porto Alegre
2016

LISTA DE SIGLAS

ABC – Associação Brasileira de Computadores

XYZ – lorem ipsum dolor sit

IJK – lorem ipsum dolor sit

SUMÁRIO

1 INTRODUÇÃO **4**

2 REVISÃO BIBLIOGRÁFICA **5**

2.1 JOGOS 5

2.1.1 TEORIA DE JOGOS 5

2.1.2 MINIMAX 5

2.2 TOMADA DE DECISÃO 6

2.2.1 MARKOV DECISION PROBLEM(MDP) 6

2.2.2 PARTIALLY OBSERVABLE MARKOV DECISION PROCESSES (POMDPS) . . . 7

2.3 APRENDIZADO POR REFORÇO 7

2.3.1 FUNÇÃO DE RECOMPENSA 7

2.3.2 Q-VALUE E Q-LEARNING 7

2.3.3 EXPLORAÇÃO (DESCOBRIMENTO) VS. EXPLORAÇÃO (APROVEITAMENTO): (EXPLORATION VS EXPLOITATION - TRADUÇÃO MEIA TOSCA) 7

2.4 MÉTODOS DE MONTE-CARLO 7

2.4.1 ÁRVORE DE PESQUISA DE MONTE-CARLO 7

2.4.2 N-ARMED-BANDIT PROBLEM 8

2.4.3 UPPER-CONFIDENCE-BOUND (UCB) 8

2.4.4 VARIAÇÕES DE MCTS 8

2.5 COLONIZADORES DE CATAN 8

3 OBJETIVOS **9**

3.1 OBJETIVOS GERAIS 9

3.2 OBJETIVOS ESPECÍFICOS 9

4 ANÁLISE E PROJETO **10**

4.1 ATIVIDADES 10

4.2 CRONOGRAMA 10

REFERÊNCIAS **11**

1. INTRODUÇÃO

lorem ipsum dolor sit amet Capítulo 1 consetetur sadipscing elitr sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat sed diam voluptua at vero eos et accusam et justo duo dolores et ea rebum stet clita kasd gubergren no sea takimata sanctus est lorem ipsum dolor sit amet lorem ipsum dolor sit amet consetetur sadipscing elitr sed diam nonumy eirmod. [CGMV98]

2. REVISÃO BIBLIOGRÁFICA

2.1 Jogos

2.1.1 Teoria de Jogos

Jogos são muito utilizados como ambiente de teste para inteligências artificiais. Jogos podem ser divididos entre diversas categorias, o que ajuda à classifica-los e determinar a sua complexidade dentro da área de inteligência artificial. As categorias são:

- Determinismo: Jogos podem ser determinísticos ou estocásticos, onde existem elementos de chance que não dependem do jogador.
- Observáveis (ou de Informação Perfeita ou Imperfeita): Classifica se o jogo possui elementos ocultos aos jogadores ou não.
- Soma-zero: Se a soma dos pontos obtidos por todos os jogadores é sempre zero ou se pode haver diferenças entre a pontuação, sobrar ou faltar pontos.
- Turnos: As ações tomadas nos jogos podem ser sempre em sequencia ou acontecer simultaneamente.
- Número de Jogadores.
- Tempo de Jogo: Se as ações são discretas ou em tempo real.

Essas categorias nos ajudam à avaliar os jogos e considerar qual o melhor algoritmo para resolve-lo. Resolver um jogo é ter o comportamento ótimo neste, o que pode significar obter a vitória ou outro tipo de recompensa específico ao qual o agente deseja dentro do jogo.

2.1.2 Minimax

O Minimax é um algoritmo muito utilizado como modelo para resolver jogos deterministicos, observaveis, de soma-zero, jogados em turnos, de 2 jogadores e tempo de jogo discreto. Esse algoritmo funciona com estados, jogadas e uma função de utilidade, para calcular quanto vale cada estado final do jogo.

- De um certo estado atual de jogo, toda a arvore de jogo é criada e os estados finais da árvore tem a sua utilidade calculada utilizando a função de utilidade.
- Partindo do estado final, a utilidade de cada estado intermediário do jogo é calculado subindo pela árvore, sempre trocando entre uma etapa de maximização de utilidade e outra de minimização, para simular a jogada de ambos os jogadores, partindo do pressuposto de que ambos vão sempre buscar as jogadas ótimas, que possuem maior utilidade.
- Finalmente, quando a raiz é atingida, a maior utilidade é escolhida dentre os estados imediatamente abaixo do estado raiz e uma jogada é feita pela máquina.

Este algoritmo serve de modelo ideal para resolver diversos jogos, porém é bastante custoso e complexo, devido ao seu fator de ramificação alto, ou seja, a árvore de possibilidades de jogada cresce rapidamente e torna vários jogos intratáveis para um computador. Existem diversas maneiras de lidar com este problema, a maior parte das técnicas visam cortar a árvore de jogo afim de descartar jogadas e estados "ruins"mais rapidamente.

- Alpha-Beta-Prunning permite que estados que não seriam explorados pela regra de maximização e minimização sejam considerados durante a formação da arvore.
- Funções de Avaliação permitem que a utilidade seja calculada antes de se chegar ao fim do jogo, assim a arvore não precisa estar completa. Para avaliar jogadas não finais, o algoritmo necessita ter um conhecimento prévio das regras e estratégias do jogo e uma heurística para estimar as utilidades.

A desvantagem dessas abordagens é a necessidade de se ter conhecimento do domínio, as regras e a qualidade das jogadas do jogo, que é custoso e exige um estudo aprofundado do jogo, algo que muitos jogos não dispõem o que torna essas abordagens caras ou inviaveis.

2.2 Tomada de Decisão

2.2.1 Markov Decision Problem(MDP)

Busca encontrar a política ótima em um sistema de decisões sequenciais em um ambiente acessível e estocástico, definindo um sistema de estados, ações, um modelo de transição e uma política. Estados representam cenários possíveis do problema. Ações levam o problema de um estado para outro. O modelo de transição aponta qual a utilidade

de cada ação em relação à cada estado. A política é a estratégia que será tomada para cada estado, tendo em vista que o problema é estocástico e as transições entre estados via ações não é exata e depende de chance. Existe uma política ótima que resolve o problema com o melhor aproveitamento de utilidades.

2.2.2 Partially Observable Markov Decision Processes (POMDPs)

Se o problema de Markov não tiver o modelo de transição disponível, o processo de se descobrir as utilidades passa a ser parte do problema.

2.3 Aprendizado por Reforço

2.3.1 Função de Recompensa

2.3.2 Q-Value e Q-Learning

2.3.3 Exploração (descobrimento) vs. Exploração (aproveitamento): (exploration vs exploitation - tradução meia tosca)

2.4 Métodos de Monte-Carlo

2.4.1 Árvore de Pesquisa de Monte-Carlo

A Árvore de Pesquisa de Monte Carlo é um algoritmo moderno para resolver jogos que não exige conhecimento do domínio do jogo, diferente das técnicas de corte do Minimax e tem as qualidades dessas técnicas por não necessitar exploração total da árvore de jogo. Outra vantagem desse algoritmo em relação ao Minimax é que ele pode ser usado em jogos estocásticos, onde existem elementos de chance, em jogos não observáveis, onde existem elementos ocultos ao jogador e jogos de vários jogadores. O Minimax possui versões para resolver esses tipos de jogos, mas elas são mais caras e complexas do que este algoritmo. -(talvez podemos falar de expectimax?)- O algoritmo funciona em 4 estágios, construindo

uma árvore de jogo assim como a do Minimax, a partir de um nodo raiz são expandidos os futuros possíveis nodos da árvore, cada nodo representando um estado de jogo.

- Seleção: Do nodo raiz, é selecionado um nodo inexplorado da árvore de jogo a partir de uma politica de seleção, que busca reduzir o arrependimento do algoritmo, escolhendo a jogada mais promissora buscando lucrar ou explorar. -(conceito de aprendizado por reforço, correto?)-
- Expansão: Um ou mais nodos são adicionados como filhos do nodo selecionado a partir das jogadas válidas possíveis.
- Simulação: O jogo é simulado a partir de um ou mais nodos gerados na fase de expansão a partir de uma política de jogo padrão, que pode ser movimentos aleatórios, semi-aleatórios ou algo mais sofisticado. Quanto mais sofisticada a política padrão, mais cara e lenta fica a fase de simulação.
- Propagação Inversa: Com a recompensa -(também de aprendizdo por reforço, correto?)- obtida na fase de Simulação, a estatística de cada nodo é atualizada.

Com uma quantidade suficiente de simulações, a melhor jogada pode ser inferida a partir de uma política de jogo padrão simples, ou até mesmo aleatória, devido a experiencia obtida pela propagação dos resultados da simulação na árvore. Cada nodo guarda 2 valores que compoem as suas informações estatisticas:

- Q-Value: O valor de recompensa que foi obtido escolhendo tal nodo.
- Numero de Visitações: Quantas vezes este nodo já foi escolhido para ser explorado.

É importante para o algoritmo ter uma boa política de seleção baseada nestas duas variaveis, pois a jogada ótima pode estar "escondida"por uma jogada tentadora a curto prazo e o equilibrio entre lucro e exploração deve ser garantido na fase de seleção.

2.4.2 N-Armed-Bandit Problem

2.4.3 Upper-Confidence-Bound (UCB)

2.4.4 Variações de MCTS

2.5 Colonizadores de Catan

3. OBJETIVOS

lorem ipsum dolor sit amet Capítulo 1 consetetur sadipscing elitr sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat sed diam voluptua at vero eos et accusam et justo duo dolores et ea rebum stet clita kasd gubergren no sea takimata sanctus est lorem ipsum dolor sit amet lorem ipsum dolor sit amet consetetur sadipscing elitr sed diam nonumy eirmod.

3.1 Objetivos Gerais

lorem ipsum dolor sit amet Capítulo 1 consetetur sadipscing elitr sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat sed diam voluptua at vero eos et accusam et justo duo dolores et ea rebum stet clita kasd gubergren no sea takimata sanctus est lorem ipsum dolor sit amet lorem ipsum dolor sit amet consetetur sadipscing elitr sed diam nonumy eirmod.

3.2 Objetivos Específicos

lorem ipsum dolor sit amet Capítulo 1 consetetur sadipscing elitr sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat sed diam voluptua at vero eos et accusam et justo duo dolores et ea rebum stet clita kasd gubergren no sea takimata sanctus est lorem ipsum dolor sit amet lorem ipsum dolor sit amet consetetur sadipscing elitr sed diam nonumy eirmod.

4. ANÁLISE E PROJETO

lorem ipsum dolor sit amet Capítulo 1 consetetur sadipscing elitr sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat sed diam voluptua at vero eos et accusam et justo duo dolores et ea rebum stet clita kasd gubergren no sea takimata sanctus est lorem ipsum dolor sit amet lorem ipsum dolor sit amet consetetur sadipscing elitr sed diam nonumy eirmod.

4.1 Atividades

lorem ipsum dolor sit amet Capítulo 1 consetetur sadipscing elitr sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat sed diam voluptua at vero eos et accusam et justo duo dolores et ea rebum stet clita kasd gubergren no sea takimata sanctus est lorem ipsum dolor sit amet lorem ipsum dolor sit amet consetetur sadipscing elitr sed diam nonumy eirmod.

4.2 Cronograma

lorem ipsum dolor sit amet Capítulo 1 consetetur sadipscing elitr sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat sed diam voluptua at vero eos et accusam et justo duo dolores et ea rebum stet clita kasd gubergren no sea takimata sanctus est lorem ipsum dolor sit amet lorem ipsum dolor sit amet consetetur sadipscing elitr sed diam nonumy eirmod.

REFERÊNCIAS BIBLIOGRÁFICAS

[CGMV98] Coffman, E.; Galambos, J.; Martello, S.; Vigo, D. “Bin packing approximation algorithms: Combinatorial analysis”. Capturado em: http://citeseer.ist.psu.edu/coffman98bin.html, Dez 2007.