Sociedades de Agentes

Capítulo 9:
Costa, E. e Simões, A. (2008). Inteligência Artificial – Fundamentos e Aplicações, 2.ª edição, FCA.

Sociedades de Agentes

- O tratamento que temos dado aos agentes tem sido, no essencial, considerá-los isolados num ambiente tentando resolver uma dada tarefa
- No entanto, em muitas situações reais, os agentes têm de conviver com outros agentes, seja em competição, seja em colaboração
- Vamos analisar de seguida aspetos específicos de sociedades de agentes

Dois Agentes

- A situação mais simples de sociedade é aquela em que existem apenas dois agentes
- Existem dois tipos de sociedades de dois agentes: competição e cooperação

Dois Agentes

- Na natureza:
  - Situações tipo predador-presa
  - Situações de luta por recurso (comida, fêmea)

  são formas de competição entre dois agentes
- Mesmo em sociedades simples de 2 agentes, estes podem cooperar e não competir:
  - Exemplo clássico dos dois prisioneiros que preferem não denunciar o colega à polícia, evitando dessa maneira uma condenação pesada

Competição – Teoria de Jogos

- Sempre que dois agentes têm objetivos que colidem (o que um ganha é o que o outro perde), entram naturalmente em competição
- Os jogos como as damas, o xadrez ou o gamão são apenas alguns exemplos que envolvem dois jogadores
- Desde há muito tempo que os jogos têm vindo a ser estudados, existindo mesmo hoje um ramo da matemática conhecido por teoria dos jogos, de que *John von Neuman* foi o iniciador

Competição – Teoria de Jogos

- A teoria de jogos tem por objeto o estudo formal da decisão racional, isto é, o estudo de como um agente escolhe ou decide uma de entre várias alternativas à sua disposição
- Para tal, o agente necessita de ter um conjunto de preferências que lhe permitam ordenar as escolhas

Competição – Teoria de Jogos

- A teoria de jogos pretende responder a questões como a de saber se existe uma estratégia ótima para um determinado jogo
- Por estratégia entende-se todo o conjunto de diretivas que indica a um jogador como proceder em qualquer situação possível que se lhe depare durante um jogo

Competição – Teoria de Jogos

Segundo *Rapoport* (1966), a teoria de jogos interessa-se por situações que possuem as seguintes 6 características:

1. Existem pelo menos dois jogadores;
2. O jogo começa com um dos agentes a efetuar um movimento, ou seja, escolhendo uma jogada de entre várias em alternativa.

Os movimentos podem ser condicionados pelo acaso, como é o caso do gamão onde é usado um dado;

Competição – Teoria de Jogos

Segundo *Rapoport* (1966), a teoria de jogos interessa-se por situações que possuem as seguintes 6 características:

3. De seguida, um agente reage com outro movimento de entre os possíveis deixados pelo primeiro movimento;
4. As escolhas de cada agente podem ou não ser conhecidas.

No caso em que são conhecidas falamos de jogos de informação perfeita (na maior parte dos jogos de cartas não são conhecidas);

Competição – Teoria de Jogos

Segundo *Rapoport* (1966), a teoria de jogos interessa-se por situações que possuem as seguintes 6 características:

5. Num jogo descrito por movimentos sucessivos, existe uma regra que identifica o fim do jogo

(*xeque-mate* no xadrez, por exemplo);

Competição – Teoria de Jogos

Segundo *Rapoport* (1966), a teoria de jogos interessa-se por situações que possuem as seguintes 6 características:

6. Quando um jogo termina, é possível determinar uma recompensa para cada jogador, função do estado em que o jogo terminou.

   No xadrez, as situações finais possíveis são vitória, empate e derrota.

   Em teoria dos jogos cada um dos resultados pode ser sempre interpretado por um número, positivo ou negativo. Exemplo: podemos atribuir a pontuação 1 à vitória, 0 ao empate e -1 à derrota.

   - Os agentes têm como objetivo maximizar a sua recompensa.

   - Quando a "sorte" de um agente é o "azar" do outro, como no caso anterior, falamos de jogos de soma zero

Competição – Teoria de Jogos

- Os jogos que nos vão interessar no nosso estudo:

  - Jogos determinísticos;

  - Envolvendo 2 jogadores que jogam alternadamente;

  - de soma zero;

  - e de informação perfeita.

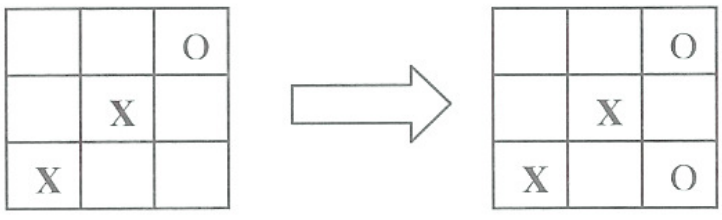
Competição – Teoria de Jogos

- Nem todos os jogos têm associadas estratégias de vitória ou, a existirem, podem ser complexas de formular
- Este facto está associado ao número de movimentos que cada jogador pode efetuar a cada momento

Competição – Teoria de Jogos

Consideremos o Jogo do Galo:

- Joga-se num tabuleiro 3 por 3 colocando cada jogador uma marca (X ou O) numa das posições livres



Jogo do galo

Competição – Teoria de Jogos

- O objetivo de um jogador é ser o primeiro a colocar 3 peças do seu tipo em linha
- Quantas estratégias existem para o jogador X ?
- Na 1.ª jogada ele tem 9 possibilidades
- Depois de o adversário jogar (tem 8 possibilidades) o jogador X tem 7^8 possibilidades
- Na jogada seguinte terá 5^6, na penúltima 3^4 e na última apenas 1

Competição – Teoria de Jogos

- Assim, o jogador X tem

$$9 \times 7^8 \times 5^6 \times 3^4 \approx 65{,}6 \times 10^{12}$$

estratégias possíveis !

Competição – Teoria de Jogos

Um exemplo de estratégia para o primeiro jogador é a seguinte

- Colocar X na posição central.
- O segundo jogador, O, pode responder de duas maneiras distintas:

Competição – Teoria de Jogos

a) Se O marcar uma posição que não corresponda a um canto, então colocar um X numa célula do canto adjacente ao O.

O 1.º jogador fica com 2 marcas em linha.

Se O não bloquear a posição que falta, marcar essa posição com X e ganhar. Se bloquear, então colocar X no canto que não é adjacente à 1.ª posição de O.

Fica-se com 2 modos de fazer 3 em linha, pelo que X ganha o jogo independentemente do que O fizer a seguir;

Competição – Teoria de Jogos

b) Se O marcar uma posição num canto, X responde marcando uma das posições adjacentes que não são cantos.
Fica-se com 2 em linha.
Se O não bloquear a 3.ª posição, jogar para lá e ganhar.
Se O bloquear essa posição, então marcar a posição do canto na intersecção da linha e coluna das duas marcas de O. Fica com 2 em linha.
Se O não bloquear essa posição, marcá-la e ganhar.
Se bloqueou, jogar para uma das posições livres que não são cantos. Fica com 2 em linha.
Se O não bloquear essa posição, jogar para lá e ganhar.
Se O bloqueou essa posição, jogar para a única posição livre e terminar num empate.

Competição – Teoria de Jogos

- Este exemplo permite mostrar que a definição de uma estratégia nem sempre é fácil.
- Como será para o xadrez ?

Competição – Teoria de Jogos

- Outra das questões que se coloca é saber se existirá alguma estratégia que seja ótima ?

- E em que sentido é ótima ?

Competição – Teoria de Jogos

- *John von Neuman* provou um teorema denominado teorema minimax que estabelece que para os jogos com as características acima enunciadas existe uma estratégia ótima
- Por ótima entende-se uma estratégia que maximiza o ganho com base em escolhas racionais
- Veremos de seguida como esse princípio foi recuperado e usado em Inteligência Artificial

Competição – Teoria de Jogos

- Atendendo à natureza destes jogos, eles podem ser vistos como um problema de procura, existindo, no entanto, alguns aspetos que os tornam distintos dos métodos de procura que estudámos anteriormente

Competição – Teoria de Jogos

Vejamos:

- Existe um estado inicial

  (tabuleiro vazio no caso do jogo do galo);
- As regras do jogo permitem determinar em qualquer momento quais os movimentos legais para o jogador, constituindo os operadores dos agentes de procura;

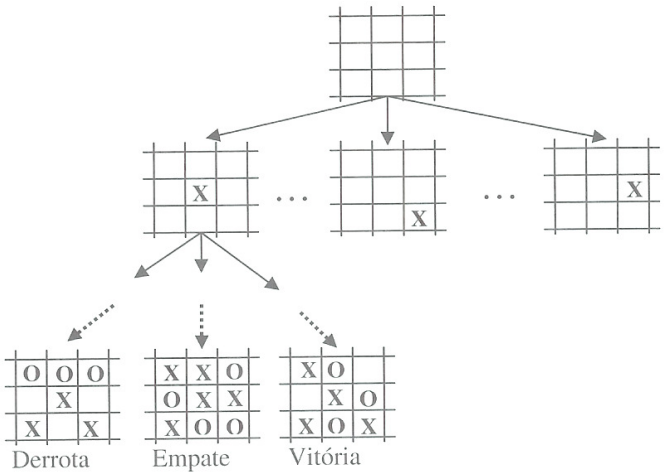
Competição – Teoria de Jogos

Vejamos:

- Existe uma regra que permite determinar quando é que o jogo chegou ao fim, identificando os estados finais;
- Finalmente, existe uma função de recompensa ou utilidade que permite atribuir um valor numérico a cada estado final.

Competição – Teoria de Jogos

- Os estados e os operadores definem implicitamente uma árvore de procura do jogo



Árvore de procura do jogo do galo

Competição – Teoria de Jogos

- Porque não usar os algoritmos de procura já conhecidos ?

- A resposta prende-se com dois aspetos:
  - Número de estados
  - Função heurística

Competição – Teoria de Jogos

- Em primeiro lugar, o número de estados é astronomicamente elevado
  - Por exemplo, estima-se que num jogo típico de xadrez o número de nós da árvore de procura seja da ordem de 10^{154}: o factor de ramificação é elevado (da ordem de 35) e a profundidade da árvore também é elevada (estimada em 100, ou seja, 50 jogadas para cada jogador)

Competição – Teoria de Jogos

- Num jogo simples como o jogo do galo, não entrando em linha de conta com as simetrias do jogo, existem 126 estados distintos, que se obtêm considerando o número de possibilidades de colocar 5 X em 9 células

Competição – Teoria de Jogos

- Em segundo lugar, é difícil definir uma função heurística
  - Não esquecer que cada jogador tem de defrontar um oponente

Competição – Teoria de Jogos

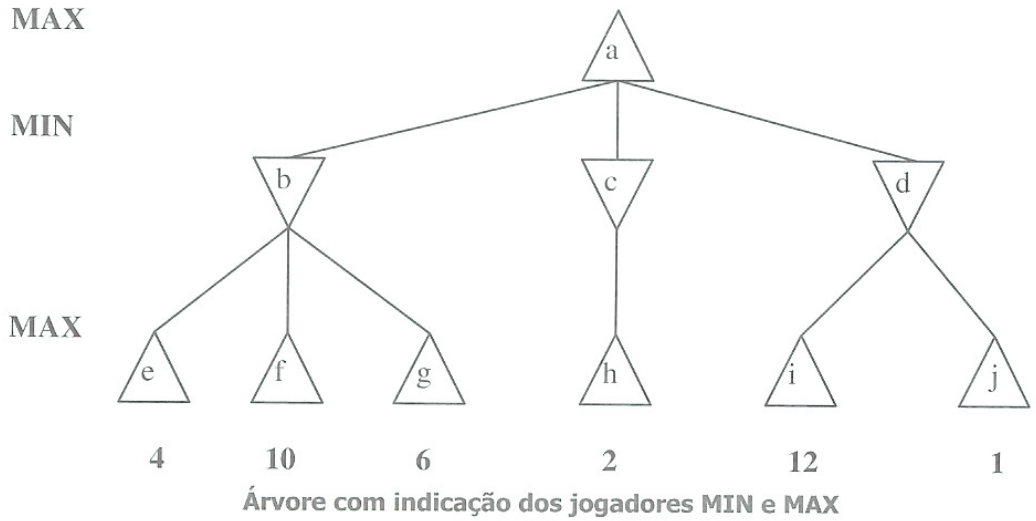
- Deste modo, é importante desenvolver métodos de procura específicos para jogos que tenham em atenção os dois aspetos referidos
- Vamos começar pelo segundo, isto é, estudar uma estratégia simples que permite a cada jogador determinar a melhor jogada
- De seguida, veremos um método que permite reduzir a dimensão do espaço de procura a explorar sem perda de informação

A Estratégia MiniMax

- Consideremos que os jogadores se chamam MAX e MIN e que um valor elevado de recompensa é bom para MAX e mau para MIN (daí os nomes)
- Consideremos também que MAX é o primeiro a jogar e que é possível explorar todo o espaço de procura

A Estratégia MiniMax

- Seja o exemplo da figura



Árvore com indicação dos jogadores MIN e MAX

A Estratégia MiniMax

Jogador MAX: triângulo com vértice para cima

Jogador MIN: triângulo com vértice para baixo

- Os valores associados aos nós terminais correspondem aos valores dados pela função de recompensa

A Estratégia MiniMax

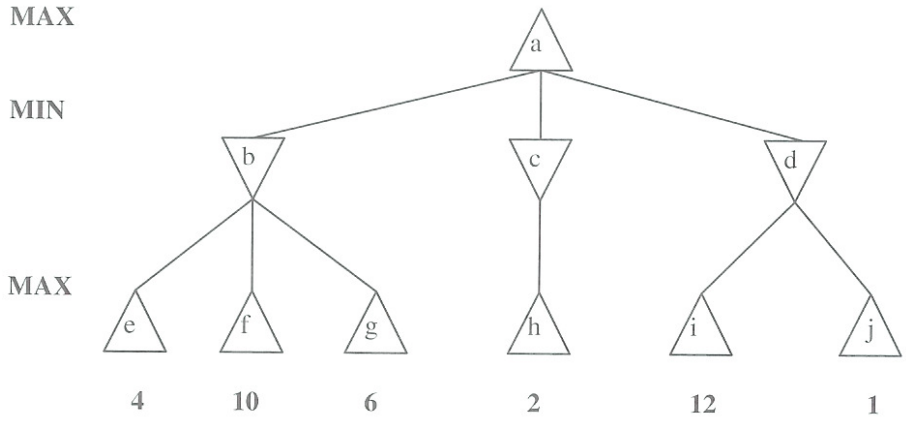
- A estratégia MiniMax supõe que os jogadores são racionais e igualmente competentes, isto é, procuram maximizar a sua recompensa e usam a mesma estratégia

A Estratégia MiniMax

- Para determinar o seu movimento, o jogador MAX começa por simular toda a árvore do jogo
- Determina os valores de recompensa e retropropaga esses valores da maneira seguinte:
- Conhecidos os valores de todos os nós sucessores de um dado nó, o valor propagado para trás é o maior desses valores, se o nó for MAX, ou o menor dos valores, se for MIN

A Estratégia MiniMax

- Assim, no exemplo anterior, o valor associado a b será 4, a c será 2 e a d será 1, uma vez que se trata de nós MIN.



Árvore com indicação dos jogadores MIN e MAX

A Estratégia MiniMax

- Por seu turno, o valor passado para a será o maior de entre os valores de b, c e d, ou seja, 4.
- Daqui se infere que o jogador MAX efetuará um movimento que o levará para a situação b.
- Esta decisão designa-se MiniMax uma vez que maximiza a recompensa pressupondo que o adversário a vai minimizar

A Estratégia MiniMax

- De um modo mais algorítmico, podemos dizer que o valor associado a um nó genérico n é dado por:

$$MiniMax(n) = \begin{cases} recompensa(n) & se \text{ n terminal} \\ \max_{s \in sucessores(n)} MiniMax(s) & se \text{ n MAX} \\ \min_{s \in sucessores(n)} MiniMax(s) & se \text{ n MIN} \end{cases}$$

A Estratégia MiniMax

- Podemos associar a esta estratégia um algoritmo recursivo que efetua uma procura em profundidade primeiro.
- Já sabemos neste caso que, para uma profundidade p e um factor de ramificação r, a complexidade temporal do algoritmo é da ordem de $O(r^p)$ e a complexidade espacial é da ordem de O(rxp).

A Estratégia MiniMax

- O algoritmo seguinte é o algoritmo principal (MiniMax) que chama a função ValMAX que obtém o valor do nó MAX (raiz da árvore)

```
Função MiniMax(estado): movimento
        1. v← ValMAX(estado)
        2. Devolve movimento associado ao sucessor de estado de valor v
Fim_de_Função
```

Algoritmo MINIMAX

A Estratégia MiniMax

- A função ValMAX pode ser definida por

```
Função ValMAX(estado): recompensa
        1. Se final(estado) Então
                1.1. Devolve recompensa(estado)
        Fim_de_Se
        2. v ← -∞
        3. Para s ∈ sucessores(estado) Faz
                3.1. v ← max(v,ValMIN(s))
                Fim_de_Para
        4. Devolve v
Fim_de_Função
```

Algoritmo para encontrar o valor do nível MAX

A Estratégia MiniMax

e do mesmo modo se define a função ValMIN

```
Função ValMIN(estado): recompensa
        1. Se final(estado) Então
              1.1. Devolve recompensa(estado)
           Fim_de_Se
        2. v ← ∞
        3. Para s ∈ sucessores(estado) Faz
              3.1. v ← min(v,ValMAX(s))
           Fim_de_Para
        4. Devolve v
Fim_de_Função
```

Algoritmo para encontrar o valor do nível MIN

Corte Alfa-Beta

- Consideremos agora a questão de reduzir o espaço de procura sem perder informação: Corte Alfa-Beta (*Alpha-Beta Pruning*)

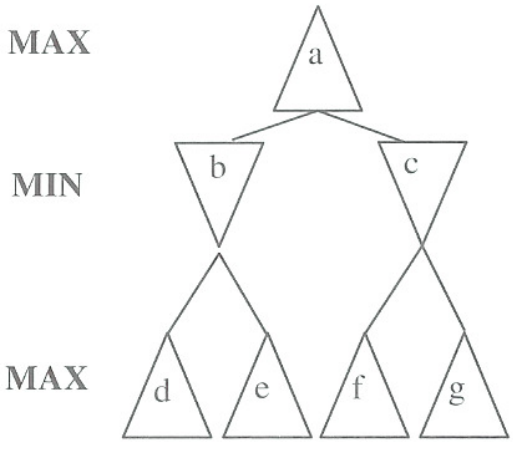
- O princípio utilizado é o seguinte: quando existem duas alternativas e uma é seguramente inferior, não interessa quantificar quanto é que é pior.

Corte Alfa-Beta

- Com base nesta ideia, vamos ver como é possível a um jogador determinar que já tem garantido um dado valor que é o melhor que alguma vez pode obter, explorando apenas uma parte da árvore de procura
- Nesse caso, a outra parte da árvore pode ser cortada

Corte Alfa-Beta

- Exemplo: consideremos a seguinte árvore de jogo



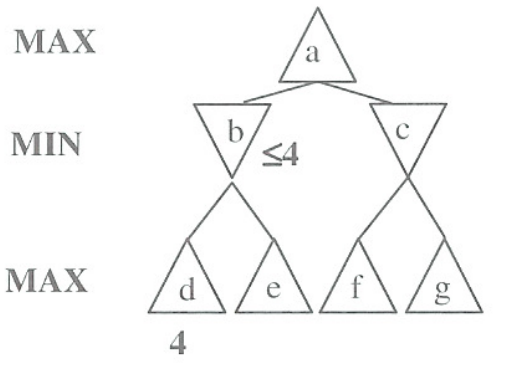
Nova árvore de procura

Corte Alfa-Beta

- Sendo a procura efetuada em profundidade primeiro, a primeira recompensa a ser determinada é a do nó d, admitamos ser 4

Corte Alfa-Beta

- Nessa altura, de acordo com a estratégia MiniMax, podemos afirmar que o valor de recompensa do nó b será no máximo 4



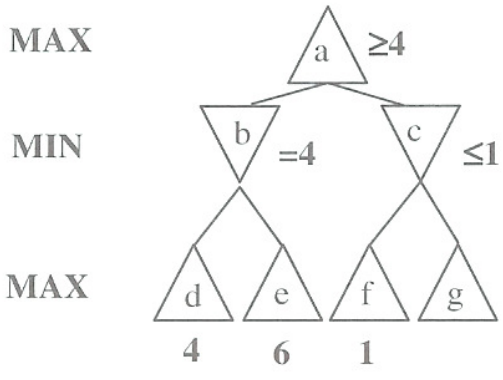
O primeiro valor e suas consequências

Corte Alfa-Beta

- Depois de calculada a recompensa do nó e, suponhamos 6, ficamos a saber a recompensa exata de b, que é 4 e a tem a expectativa de ter pelo menos 4

Corte Alfa-Beta

- Continuando a execução do algoritmo, o próximo nó a ter a sua recompensa fixa é f, que admitamos ser 1



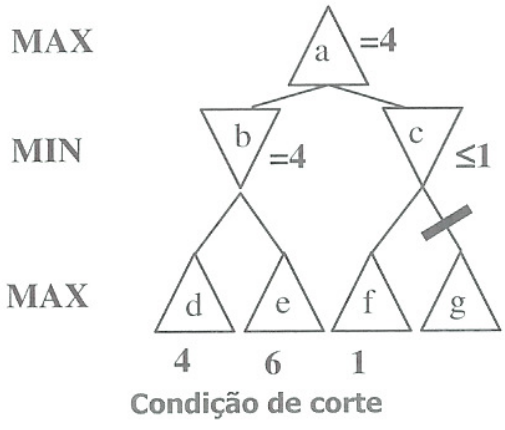
Situação mais à frente

Corte Alfa-Beta

- Ou seja, se formos analisar o valor de g, este só pode contribuir para baixar o valor de recompensa de c
- Mas como o nó a tem garantido pelo menos 4, que será sempre maior do que o valor associado a c, não altera em nada ao valor final de a este valor encontrado para g
- Assim, esta procura pode ser descartada

Corte Alfa-Beta

- A situação final será:



Corte Alfa-Beta

- O algoritmo alfa-beta, que implementa este princípio de corte, funciona com dois valores adicionais, para além do valor da recompensa de um nó terminal: o valor α e o valor β.
- α: valor mais alto encontrado no caminho de MAX
  (isto é, traduz o valor mínimo que o jogador MAX tem garantido)
- β: valor mais baixo encontrado no caminho de MIN
  (isto é, traduz o valor máximo que o jogador MIN pode alcançar)

Corte Alfa-Beta

- Quando, num dado nó do caminho a partir da raiz, o valor de α deixa de ser inferior ao valor de β (condição de corte), então a procura pode parar para os nós sucessores desse nó que ainda não foram analisados

Corte Alfa-Beta

- Se o nó em que esta situação ocorre for do tipo MAX, então o valor de α é atualizado, passando a ser o maior entre o valor antigo e a recompensa calculada para o nó
- Se for do tipo MIN, é β que é atualizado, passando a valer o menor dos valores do β atual e da recompensa já calculada para o nó

Corte Alfa-Beta

- O algoritmo Alfa-Beta:

Função AlfaBeta(estado): movimento
1. v ← ValMAX(estado,-∞, +∞)
2. Devolve *movimento* associado ao sucessor de *estado* de valor v
Fim_de_Função

Algoritmo para cálculo dos valores de α e de β

- Inicialmente o algoritmo é chamado com α=-∞ e β=+∞;
- -∞ é o valor que o jogador MAX tem sempre garantido;
- +∞ é o valor máximo que o jogador MIN poderá alcançar.

Corte Alfa-Beta

- O algoritmo da função ValMAX:

```
Função ValMAX(estado, α, β): recompensa
        1. Se final(estado) Então
                1.1. Devolve recompensa(estado)
        Fim_de_Se
        2. v ← -∞
        3. Para s ∈ sucessores(estado) Faz
                3.1. v ← max(v,ValMIN(s, α, β))
                3.2. Se v ≥β Então
                        3.2.1. Devolve v
                Fim_de_Se
                3.3. α ← max(α,v)
        Fim_de_Para
        4. Devolve v
Fim_de_Função
```

Algoritmo para cálculo do valor MAX, com corte alfa-beta

Corte Alfa-Beta

- O algoritmo da função ValMIN:

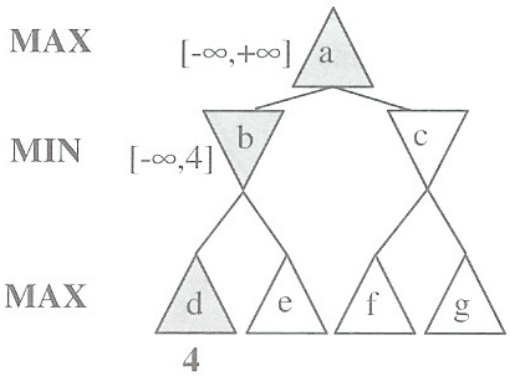
```
Função ValMIN(estado, α, β): recompensa
        1. Se final(estado) Então
                1.1. Devolve recompensa(estado)
                Fim_de_Se
        2. v ← +∞
        3. Para s ∈ sucessores(estado) Faz
                3.1. v ← min(v,ValMAX(s, , α, β))
                3.2. Se v ≤α Então
                        3.2.1. Devolve v
                Fim_de_Se
                 3.3. β ← min (β,v)
                Fim_de_Para
        4. Devolve v
Fim_de_Função
```

Algoritmo para cálculo do valor MIN, com corte alfa-beta

Corte Alfa-Beta

- Exemplo:

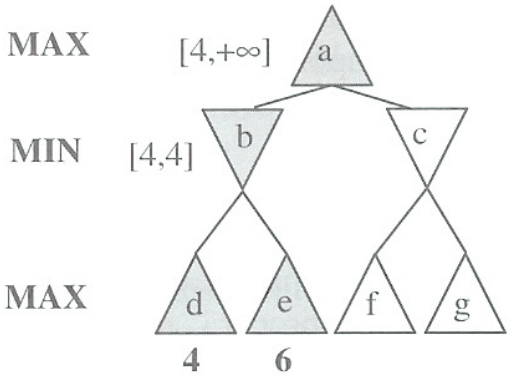
Vejamos como evolui o algoritmo para o exemplo anterior



Os valores de α e de β depois da primeira avaliação

- Os valores de α e de β aparecem como [α, β]
- A primeira vez que o algoritmo chega a um nó terminal calcula a recompensa e altera o valor de β

Corte Alfa-Beta

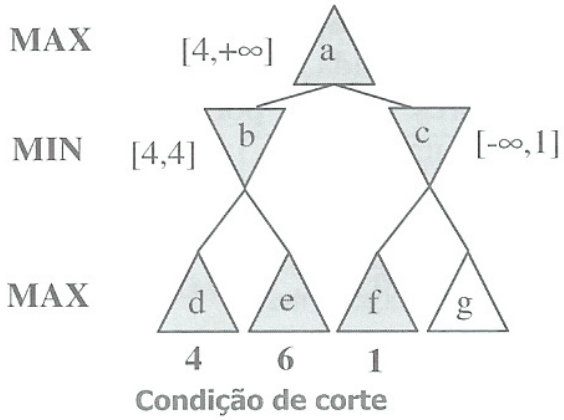


Os valores intermédios de α e de β

- Ao avaliar e, o valor de β não se altera, mas completada a chamada a partir do nó b, obriga à alteração do valor de α, que é propagado para trás para a chamada (ainda incompleta) a partir do nó a

Corte Alfa-Beta

- Passando para o ramo direito de a, temos a situação seguinte, até obter um valor de recompensa para f:

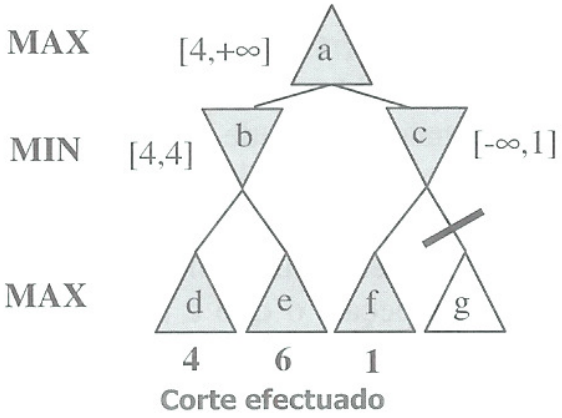


Corte Alfa-Beta

- O valor de α em a supera o valor que o nó c recebe do primeiro sucessor, pelo que temos condição de corte: o valor que a já tem garantido é maior do que o valor máximo que c alguma vez pode ter, pelo que o valor de c nunca pode contribuir para melhorar o valor de a

Corte Alfa-Beta

- Daí a situação final ilustrada na figura
- O valor do nó a será 4



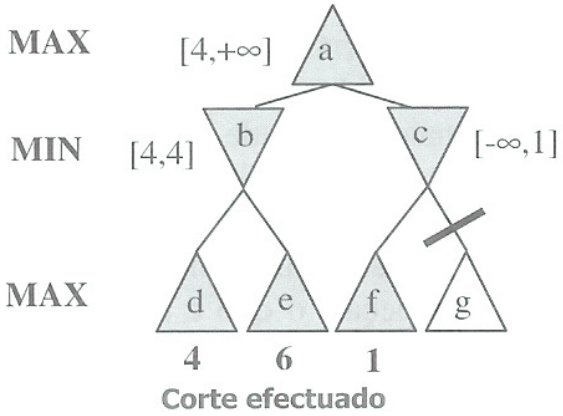
Corte Alfa-Beta: Conclusões

- Como quantificar o que se ganha com este algoritmo ?

- A resposta não é simples, até porque a capacidade de corte do algoritmo depende da ordem pela qual os nós sucessores são analisados

Corte Alfa-Beta: Conclusões

- No exemplo anterior, se o nó g fosse visitado primeiro e o seu valor fosse maior do que 4, já não haveria corte de nenhum ramo da árvore



Corte Alfa-Beta: Conclusões

- Assim, é possível que, no pior dos casos, o algoritmo visite os mesmos nós que o algoritmo MiniMax
- Se a ordem de visita for a mais favorável, então a complexidade temporal é $O(r^{p/2})$ e mesmo com uma ordem de visita aleatória a complexidade temporal é $O(r^{3p/4})$ – (*Pearl*, 1984).

Cooperação

- Em teoria de jogos é normal recorrer a uma matriz para representar as diferentes estratégias dos dois jogadores
- A matriz é preenchida com o valor do resultado (da recompensa)
- Quando se usam as matrizes para explicitar as alternativas, falamos em teoria de jogos na forma normal
- Quando os jogos são de soma zero, basta indicar a recompensa de um dos jogadores

Dilema do Prisioneiro

- Consideremos o caso seguinte:
  - Dois ladrões assaltam um banco e conseguem escapar;
  - Juram nunca denunciar-se mutuamente no caso de serem apanhados;
  - São mais tarde apanhados pela polícia que não tem provas concretas do seu envolvimento no assalto;
  - A polícia separa-os e a cada um propõe o seguinte acordo: se denunciar o colega sai em liberdade enquanto o colega apanha a pena máxima;
  - Caso não colaborem com a polícia, apanham ambos uma pena, embora menor, por outro crime de que vêm também acusados e sobre o qual existem provas;
  - Mas se ambos forem delatores também incorrem numa pena, inferior à pena máxima mas maior do que no caso de ambos recusarem cooperar com a polícia;
  - O que devem fazer ?

Dilema do Prisioneiro

- Este problema é conhecido pelo nome de dilema do prisioneiro
- Pode ser formulado pela seguinte matriz

	B RECUSA COOPERAR (B_1)	B DENUNCIA A (B_2)
A RECUSA COOPERAR (A_1)	1 ano, 1 ano	3 anos, 0 anos
A DENUNCIA B (A_2)	0 anos, 3 anos	2 anos, 2 anos

Matriz do jogo do dilema do prisioneiro

Dilema do Prisioneiro

	B RECUSA COOPERAR (B_1)	B DENUNCIA A (B_2)
A RECUSA COOPERAR (A_1)	1 ano, 1 ano	3 anos, 0 anos
A DENUNCIA B (A_2)	0 anos, 3 anos	2 anos, 2 anos

Matriz do jogo do dilema do prisioneiro

- Analisando os valores, parece que a melhor estratégia para ambos é denunciar. Porquê ?

Dilema do Prisioneiro

- Envolvendo o jogo uma escolha simultânea, se um jogador denunciar, retirará sempre 1 ano à sua pena, qualquer que seja a decisão do outro jogador!
- Isto é, o par de estratégias (denúncia, denúncia) é um ponto de equilíbrio
- Na realidade é o único, no pressuposto de que ambos os agentes atuam racionalmente

Dilema do Prisioneiro

- Esta situação é curiosa:

  Para quem está *"de fora"* a solução (recusa cooperar, recusa cooperar) seria a melhor do ponto de vista da recompensa

	B RECUSA COOPERAR (B_1)	B DENUNCIA A (B_2)
A RECUSA COOPERAR (A_1)	1 ano, 1 ano	3 anos, 0 anos
A DENUNCIA B (A_2)	0 anos, 3 anos	2 anos, 2 anos

Matriz do jogo do dilema do prisioneiro

Dilema do Prisioneiro Iterado

- Na formulação anterior o problema do dilema do prisioneiro é jogado apenas uma vez
- *Robert Axelrod* (1984) estudou o que acontece quando o jogo pode ser jogado um número indeterminado, mas finito, de vezes: esta versão é conhecida pelo dilema do prisioneiro iterado

Dilema do Prisioneiro Iterado

- Pode-se provar que se jogarmos o jogo um número indeterminado mas suficientemente grande de vezes, então a atitude racional é cooperar

Dilema do Prisioneiro Iterado

- A melhor estratégia, conhecida por TIT FOR TAT, consiste no seguinte:

  começa por não denunciar e depois joga sempre de forma idêntica à última escolha do adversário

Dilema do Prisioneiro Iterado

- O que torna esta estratégia simples tão robusta são as seguintes características (*Axelrod*):

1. Simpática, pois evita conflitos, não denunciando enquanto o outro jogador não o fizer;
2. Respondona, pois reage a uma denúncia denunciando;
3. Que perdoa, quando o adversário depois de denunciar volta a cooperar e
4. de comportamento simples, de modo que o outro jogador pode compreender facilmente o padrão de jogo.

Multiagentes

- Nas sociedades o que é normal é existirem vários agentes;
- Podem ser da mesma classe ou de classes diferentes, ter objetivos semelhantes ou opostos;
- Podem comunicar entre si, direta ou indiretamente, podem formar alianças, podem negociar, podem ter o mesmo poder ou existirem relações hierárquicas de poder, podem pois, numa frase, estabelecer diversos tipos de interação;

Multiagentes

- Dão origem a sociedades complexas, independentemente de cada agente da sociedade ser ele próprio simples, na sua arquitetura e capacidades ou ainda nas interações que estabelece com os outros agentes que povoam o seu ambiente.

Multiagentes: Vida Artificial

- A Vida Artificial (VA) é uma área recente de investigação que tem vindo a ganhar crescente importância
- Definir VA tem-se revelado tão complicado como definir Inteligência Artificial.
- Consideremos a seguinte definição (*Langton*, 1989):

Vida Artificial é o estudo de sistemas construídos pelo homem e que exibem comportamentos característicos dos sistemas vivos naturais

Multiagentes: Vida Artificial

- Ao contrário da biologia, a VA tem uma preocupação sintética e não analítica;
- Não se trata de analisar os seres vivos (para melhor os compreender), mas antes de mostrar a viabilidade da construção de entidades que possuem as características do seu contraponto natural;
- No entanto, a VA pode contribuir para uma melhor compreensão da vida natural, ao permitir criar mundos nos quais é possível explorar a vida como poderia ter sido.

Multiagentes: Vida Artificial

- Vida Artificial
  - Autómatos Celulares
  - Inteligência de Enxame
    - As colónias de formigas
    - Inteligência de Enxame baseada em partículas
  - …

Multiagentes: BDI

- Modelo e Arquitetura BDI
- Estudámos vários tipos de agentes e respetivas arquiteturas: dos mais "simples", ou reativos, até aos mais complexos, os baseados em conhecimento, que podem ser enriquecidos com a capacidade de aprender, com a capacidade de evoluir ou com ambas;
- Estes agentes podem ser aperfeiçoados quando, por exemplo, são capazes de lidar com conhecimento imperfeito ou interagem com o ambiente através do uso da linguagem natural.

Modelo e Arquitetura BDI

- Mas na sua essência todos os agentes têm uma estrutura básica: repetem ciclicamente um processo de perceção do ambiente, tomada de decisão e atuação

Modelo e Arquitetura BDI

- Até agora temos evitado referir um aspeto essencial relativamente aos nossos agentes: em que medida as suas ações são intencionais ?
  - Os Agentes Deliberativos

Modelo e Arquitetura BDI

- **Modelo e Arquitetura de um Agente Deliberativo**
- Crenças, desejos e intenções são estados mentais e formam a base de uma arquitetura de um agente deliberativo, voltada para a ação;
- Essa arquitetura é conhecida pela designação BDI
  (*Belief* – *Desire* – *Intention*).

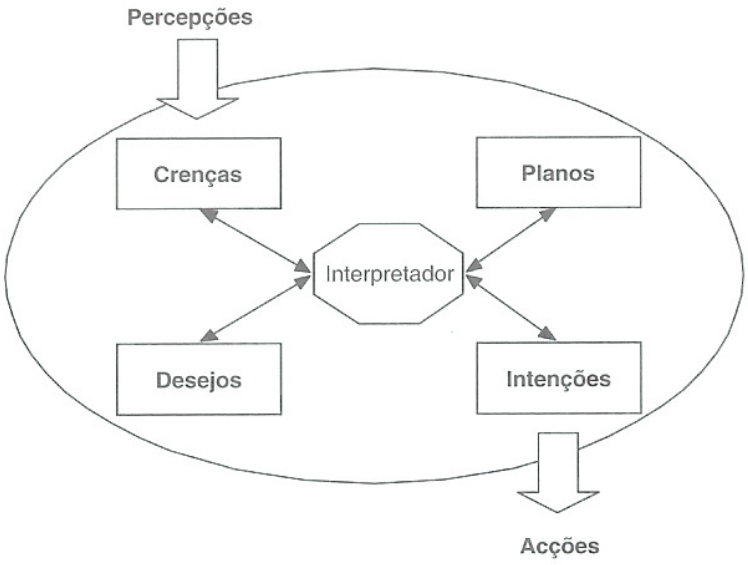
Modelo e Arquitetura BDI

- De um modo simples:
  - As crenças materializam a visão que um agente tem do mundo e de si próprio;
  - Os desejos são as suas motivações, preferências ou objetivos;
  - Enquanto as intenções são os desejos escolhidos para serem realizados, com os quais nos comprometemos.

Modelo e Arquitetura BDI

- A escolha dos desejos que queremos concretizar obriga a uma atividade que se designa por deliberação
- O seu resultado consiste precisamente nas intenções
- O modo como realizamos as nossas intenções leva o agente a construir um plano que determina como alcançamos um estado de coisas em que as nossas intenções se concretizaram.

Modelo e Arquitetura BDI



Arquitectura de um agente segundo a abordagem BDI

Modelo e Arquitetura BDI

- A implementação de um agente segundo o modelo BDI, voltado para o raciocínio prático, pode ser descrita assim

```
Função AgenteBDI(ambiente, agente): ambiente
    1.  Enquanto Verdade Fazer
        1.1 Observa o mundo
        1.2 Actualiza o modelo interno do mundo (crenças)
        1.3 Delibera sobre a próxima intenção a concretizar
        1.4 Constrói o plano para a intenção
        1.5 Executa o plano
    Fim_de_Enquanto
Fim_de_Função
```

Agente BDI