Inteligência Artificial - Lista de Exercícios 1

Prof. Fabrício Olivetti de França, Prof. Denis Fantinato

3º Quadrimestre de 2018

Inteligência Artificial

Exercícios

Baseados em:

Capítulo 6 (CSP): (6.2, 6.4b, 6.6, 6.7, 6.8, 6.9, 6.11, 6.13, 6.16)

Capítulo 5 (Competição): (5.4, 5.7, 5.9, 5.12, 5.16, 5.20)

Tradução

CSP:

1. Considere o problema de alocação de k cavalos em um tabuleiro de xadrez $n \times n$ tal que nenhum cavalo é atacado. Assume-se que k é dado e que $k \leq n^2$.
   - a. Formule o problema CSP.
   - b. Considere agora o problema de colocar no tabuleiro tantos cavalos quanto for possível, sem que nenhum seja atacado. Explique como resolver esse problema usando uma busca local a partir da definição adequada da funções AÇÃO e RESULTADO e da função objetivo.
2. Dê uma formulação CSP precisa para o seguinte problema:
   - Alocação de Disciplinas: Existe um número fixo de professores e de salas de aula, uma lista de disciplinas a serem oferecidas e uma lista de possíveis horários para as disciplinas. Cada professor tem um conjunto de disciplinas que ele(a) pode lecionar.
3. Mostre como uma única restrição ternária do tipo $A + B = C$ pode se tornar três restrições binárias através do uso de uma variável auxiliar. Você pode assumir domínios finitos (*Dica:* Considere uma nova variável que recebe valores que são pares de outros valores, e considere restrições do tipo “X é o primeiro elemento do par Y”). Em seguida, mostre como restrições com mais do que três variáveis podem ser tratadas de forma similar. Finalmente, mostre como restrições unárias podem ser eliminadas pela alteração do domínio das variáveis. Isto completa

a demonstração de que qualquer CSP pode ser transformado em um CSP com apenas restrições binárias.

4. Considere o seguinte quebra-cabeça: Em uma mesma rua, há cinco casas de diferentes cores. Em cada uma delas, vive uma pessoa de uma nacionalidade diferente. Cada uma dessas pessoas gosta de uma bebida diferente e de uma marca de doces diferente da dos demais. Além disso, cada uma possui uma espécie diferente de animal de estimação. As perguntas são: Onde a zebra mora e em qual casa se bebe água?

   Pistas:

   - O britânico mora na casa vermelha.
   - O espanhol possui um cão.
   - O norueguês vive na primeira casa à esquerda.
   - A casa verde está imediatamente à direita da casa de marfim.
   - O homem que come barras de 'Hershey' mora na casa ao lado do homem com a raposa.
   - 'Kit Kats' são comidos na casa amarela.
   - O norueguês mora ao lado da casa azul.
   - O comedor de 'Smarties' tem caracóis de estimação.
   - O comedor 'Snickers' bebe suco de laranja.
   - O ucraniano bebe chá.
   - Os japoneses comem 'Milky Ways'.
   - 'Kit Kats' são comidos em uma casa ao lado da casa onde o cavalo é mantido.
   - O café é bebido na casa verde.
   - Leite é bebido na casa do meio.

   Faça uma representação desse problema como um CSP e tente implementá-lo.

5. Considere o grafo com 8 nós: A_1, A_2, A_3, A_4, H, T, F_1, e F_2. A_i é conectado a A_{i+1} para todo i, cada A_i é conectado à H, H é conectado à T, e T é conectado à cada F_i. Com três cores diferentes, atribua cores aos nós do grafo (sem nós vizinhos com cores iguais) seguindo a estratégia: método de *backtracking*, com função `selecioneVar` escolhendo as variáveis na ordem A_1, H, A_4, F_1, A_2, F_2, A_3, T, e a função `sortValues` escolhendo os valores na ordem $verm, verd, azul$.

6. Explique por que escolher a variável *mais* restrita e o valor que *menos* restringe é uma boa heurística na busca CSP.

7. Use o algoritmo AC-3 para mostrar que a consistência de arcos pode detectar a inconsistência na atribuição parcial $\{WA = verd, V = verm\}$ para o problema mostrado na Figura 6.1 do livro.

8. O algoritmo AC-3 reinsere na fila todo arco (X_k, X_i) sempre que qualquer valor é apagado do domínio de X_i, mesmo se cada valor de X_k é consistente com vários valores restantes de X_i. Suponha que, para todo arco (X_k, X_i), é possível saber o número de valores restantes de X_i que são consistentes com cada valor de X_k. Explique como atualizar estes números de forma eficiente.

9. Define com suas próprias palavras os termos: restrição, busca backtracking e consistência de arco.

Competição:

1. Formalize e implemente um jogo (descrições de estado, ações, resultado de transição de estado, estado terminal e função utilidade) para um ou mais dos seguintes jogos estocásticos: jogo imobiliário (ou monopoly), scrabble, Texas hold'em poker.

2. Prove o seguinte: Para toda árvore de jogo, a utilidade obtida pelo jogador MAX usando decisões minimax contra um jogador MIN subótimo nunca será menor que aquela obtida contra um jogador MIN ótimo. Você conseguiria pensar em uma árvore de jogo no qual MAX pode se dar melhor usando uma estratégia subótima contra um jogador MIN subótimo?

3. No jogo da velha, seja X_n o número de linhas, colunas ou diagonais com exatamente $n\ X's$ e nenhum O. Similarmente, assuma que O_n é o número de linhas, colunas ou diagonais com apenas $n\ O's$. A função utilidade atribui $+1$ para qualquer posição com $X_3 = 1$ e -1 para qualquer posição com $O_3 = 1$. Todas as outras posições terminais tem utilidade 0. Para posições não terminais, usaremos a função de avaliação:

   $Eval(s) = 3X_2(s) + X_1(s) - (3O_2(s) + O_1(s))$

   - a. proximadamente quantos jogos diferentes existem?
   - b. Mostre uma árvore de jogo completa começando de um jogo vazio indo até a profundidade 2 (isto é, com um X e um O), levando em consideração a simetria.
   - c. Marque em sua árvore as avaliações para todas as posições na profundidade 2.
   - d. Usando o algoritmo minimax, marque em sua árvore os valores para as posições nas profundidades 1 e 0, e use estes valores para escolher a melhor jogada inicial.
   - e. Circule os nós na profundidade 2 que não seriam avaliadas se o algoritmo alfa-beta fosse utilizado, assumindo que os nós são gerados na ordem ótima para a poda alfa-beta.

4. Descreva como os algoritmos minimax e alfa-beta mudariam para jogos de soma-NÃO-zero (com dois jogadores) na qual cada jogador tem uma função utilidade distinta e ambas são conhecidas pelos jogadores. Se não existem restrições nas duas utilidades terminais, é possível que os nós sejam podados pelo alfa-beta? E se as funções de utilidade dos jogadores diferirem por, no máximo, uma constante k, tornando o jogo praticamente cooperativo?

5. A Figura 5.19 do livro mostra uma árvore completa para um jogo trivial. Assuma que os nós folha serão avaliados da esquerda para a direita e que, antes de um nó folha ser avaliado, não sabemos nada sobre seu valor (o domínio é $(-\infty, \infty)$).

- a. Copie a figura, marque o valor dos nós internos e indique a melhor jogada na raiz com uma seta.
- b. Dados os valores dos seis primeiros nós, é necessário avaliar o sétimo e o oitavo nó? Dado os valores dos primeiros sete nós, é necessário avaliar o oitavo? Explique suas respostas.
- c. Suponha que os nós folhas assumem valores entre -2 e 2 (inclusive). Depois que os dois primeiros nós são avaliados, qual é o intervalo de valores para o nó probabilístico à esquerda?
- d. Circule todos os nós folha que não precisam ser avaliados sob o pressuposto do item anterior.

6. A seguir, uma árvore “max” consiste apenas de nós max, enquanto que uma árvore “expectimax” consiste apenas de nós max na raiz, alternando entre camadas de nós probabilísticos e nós max. Em todos os nós probabilísticos, todas as probabilidades de eventos são não nulas. O objeto é *encontrar o valor da raiz* através de uma busca por profundidade limitada. Para cada um dos itens abaixo, ou dê um exemplo ou explique por que isto é impossível.
    - a. Assumindo que os nós folha são finitos mas ilimitados, seria o método de poda (como no algoritmo alfa-beta) possível em uma árvore max?
    - b. Seria possível usar a poda em uma árvore expectimax sob as mesmas condições?
    - c. Se os valores dos nós folhas são todos não negativos, é possível podar em uma árvore max? Dê um exemplo ou explique por que não.
    - d. Se os valores dos nós folhas são todos não negativos, é possível podar em uma árvore expectimax? Dê um exemplo ou explique por que não.
    - e. Se todos os valores estão no intervalo [0,1], é possível podar em uma árvore max? Dê um exemplo ou explique por que não.
    - f. Se todos os valores estão no intervalo [0,1], é possível podar em uma árvore expectimax?
    - g. Considere os possíveis eventos de um nó probabilístico em uma árvore expectimax. Quais das seguintes ordens de avaliação permitiriam mais oportunidades de poda? (*i*) Menor probabilidade primeiro, (*ii*) Maior probabilidade primeiro, (*iii*) Não faz diferença.