



Faculdade de Engenharia

Departamento de Electrotecnia

Licenciatura em Engenharia Informática

Inteligência Artificial I

Tema: Capítulo 3 e Problemas NP

Estudante	Lino, Miro Pedro Tipaneque
Docentes	Ruben Manhiça e Roxan Cadir

1. Leitura do capítulo 3 do livro de Russell e Norvig

O capítulo 3 do livro *Inteligência Artificial: Uma Abordagem Moderna* de Stuart Russell e Peter Norvig é intitulado "Soluções por Busca". Esse capítulo explora os conceitos fundamentais sobre como agentes podem resolver problemas por meio de busca em um espaço de estados.

O capítulo começa descrevendo problemas como uma combinação de estados, ações e objetivos. Discute representações de problemas, estados iniciais, estados objetivos, e as várias maneiras de modelar ações. O foco principal é em algoritmos de busca não informada (ou seja, sem informação extra sobre o problema além da própria definição), como busca em largura (*breadth-first search*), busca em profundidade (*depth-first search*), busca de custo uniforme e busca de profundidade limitada.

Além disso, o capítulo aborda questões sobre a complexidade computacional e eficiência desses algoritmos, comparando suas vantagens e desvantagens em diferentes cenários.

2. Definição dos Problemas NP: Equivalente, Fácil, Completo e Difícil

No contexto da Teoria da Complexidade Computacional, os problemas NP (*Nondeterministic Polynomial time*) são aqueles cujas soluções podem ser verificadas em tempo polinomial por uma máquina de Turing não-determinística. Dentro dessa classe, existem diferentes categorias que ajudam a classificar a dificuldade dos problemas:

a) Problemas NP-completo:

Um problema é considerado NP-completo se ele está na classe NP e é, ao mesmo tempo, pelo menos tão difícil quanto qualquer outro problema em NP. Em termos técnicos, se um problema NP-completo puder ser resolvido em tempo polinomial, então todos os problemas em NP podem ser resolvidos em tempo polinomial.

Exemplos: Problema de Satisfatibilidade Booleana (SAT), Problema do Caixeiro-Viajante (TSP).

No problema de Satisfatibilidade Booleana, imagine que tem uma expressão lógica composta por várias variáveis booleanas (que podem ser verdadeiras ou falsas) e operadores lógicos (AND, OR, NOT). O problema é determinar se existe uma atribuição de valores (verdadeiro ou falso) para essas variáveis que torna toda a expressão verdadeira. A aplicação prática pode ser em circuitos digitais, verificar se existe uma configuração de entradas que faz com que um circuito produza uma saída desejada.

Importância: Resolver um problema NP-completo eficientemente significa que todos os problemas na classe NP podem ser resolvidos eficientemente.

b) Problemas NP-difícil:

Um problema é NP-difícil se ele é pelo menos tão difícil quanto qualquer problema em NP. No entanto, diferente dos problemas NP-completos, ele não precisa estar na classe NP, o que significa que a verificação da solução pode não ser possível em tempo polinomial.

Exemplos: Otimização do TSP (encontrar o caminho mais curto), problemas de decisão mais complexos que não têm verificação eficiente.

Na otimização do TSP, dado um conjunto de cidades e as distâncias entre elas, o problema é encontrar a rota mais curta que passe por todas as cidades exatamente uma vez e retorne ao ponto de partida. A versão de otimização do TSP é NP-difícil porque, apesar de ser fácil verificar uma solução (uma rota), encontrar a rota mais curta entre todas as possíveis é extremamente complexo. A aplicação prática pode ser no planeamento de rotas para entrega de mercadorias, onde se deseja minimizar o tempo ou a distância percorrida.

Importância: Esses problemas são reconhecidos como altamente complexos e servem como *benchmarks* para medir a dificuldade de outros problemas.

c) Problemas NP-fácil:

Um problema NP-fácil é aquele que pode ser transformado em um problema NP através de uma redução em tempo polinomial. Esses problemas podem ser resolvidos eficientemente se forem "facilmente" redutíveis a um problema em NP.

Exemplos: Embora o termo seja menos comum, um exemplo pode ser visto em problemas de otimização que podem ser reduzidos a problemas de decisão em NP.

Considere o problema de determinar o menor número de cores necessárias para colorir um gráfico (onde nenhuma cor se repete entre vértices adjacentes). Este é um problema de otimização. Se você perguntar se é possível colorir o gráfico com exatamente k cores (problema de decisão), este pode ser reduzido ao problema NP de encontrar uma coloração válida. A aplicação prática seria alocação de frequências em redes sem fio, onde cada área precisa ser "colorida" com uma frequência diferente para evitar interferências.

Importância: Eles são utilizados para entender melhor a estrutura e as relações entre problemas de diferentes classes de complexidade.

d) Problemas NP-equivalente:

Este termo refere-se a problemas que têm complexidade equivalente a problemas NP-completos, sem necessariamente pertencer à classe NP.

Exemplos: Esse termo não é amplamente utilizado na literatura clássica, mas pode ser aplicado a problemas cuja solução é comparável, em termos de dificuldade, aos problemas NP-completos.

Considere um quebra-cabeça deslizante em uma grade de 4x4 com 15 peças numeradas e um espaço vazio. O objetivo é organizar as peças em ordem numérica deslizando-as. Determinar se uma configuração inicial pode ser transformada na configuração final é um problema NP-equivalente, pois a complexidade da solução é comparável a um problema NP-completo, mas não pertence necessariamente à classe NP. A aplicação prática pode ser resolver problemas de reconfiguração em sistemas embarcados ou robóticos onde se precisa reorganizar componentes de acordo com certas regras.

Importância: Embora menos utilizado, ajuda a categorizar problemas que possuem características de complexidade semelhantes aos problemas NP-completos.