

Branch and Bound e Backtracking

Lucas Saliba Lucas Santiago de Oliveira

Novembro de 2021

Sumário

1	<i>Branch and Bound</i>	2
1.1	Conceito	2
1.2	Quando usar	2
1.3	Exemplo	2
2	<i>Backtracking</i>	2
2.1	Conceito	2
2.2	Quando usar	2
2.3	Exemplo	2
3	Relação entre <i>Branch and Bound</i> e <i>Backtracking</i>	3
4	Relação com a Abordagem Gulosa	3
5	Relação com Programação Dinâmica	3
6	Relação com Divisão e Conquista	3

Resumo

Para começar falando sobre os algoritmos de otimização de problemas, começarei falando sobre o que é um problema linear na programação. Se considerarmos que exista um problema X que leva Y de tempo para ser resolvido e que Y cresce diretamente proporcional ao crescimento de X , então temos um problema linear. A ideia de todos os algoritmos que serão citados nesse documento é fazer Y crescer mais lentamente. Dessa forma, se o problema X dobrar de tamanho Y não dobrará de tamanho junto.

1 *Branch and Bound*

1.1 Conceito

Branch and Bound ou ramificar e limitar é um estilo de algoritmo que aplica otimizações para tornar problemas lentos em problemas rápidos.

1.2 Quando usar

1.3 Exemplo

2 *Backtracking*

2.1 Conceito

2.2 Quando usar

2.3 Exemplo

Explicacao do problema Ciclo Hamiltoniano

O caminho hamiltoniano em um gráfico não direcionado é um caminho que visita cada vértice exatamente uma vez. Um ciclo hamiltoniano é um caminho hamiltoniano tal que existe uma aresta (no gráfico) do último vértice ao primeiro vértice do caminho hamiltoniano. Determine se um determinado gráfico contém

Ciclo Hamiltoniano ou não. Se contiver, imprime o caminho. A seguir estão as entradas e saídas da função necessária.

Entrada do problema

Um gráfico de matriz 2D $[V][V]$ onde V é o número de vértices no gráfico e o gráfico $[V][V]$ é a representação da matriz de adjacência do gráfico. Um gráfico de valor $[i][j]$ é 1 se houver uma borda direta de i para j , caso contrário, gráfico $[i][j]$ é 0.

Saida do problema

Um caminho de matriz $[V]$ que deve conter o Caminho Hamiltoniano. o caminho $[i]$ deve representar o i^o vértice no caminho hamiltoniano. O código também deve retornar falso se não houver um ciclo hamiltoniano no gráfico.

Algoritmo Backtracking

Cria uma matriz de caminho vazia e adiciona o vértice 0 a ela. Adiciona outros vértices, começando do vértice 1. Antes de adicionar um vértice, verifica se ele é adjacente ao vértice adicionado anteriormente e se já não foi adicionado. Se encontrarmos tal vértice, adicionamos o vértice como parte da solução. Se não encontrarmos um vértice, retornamos falso.

3 Relação entre *Branch and Bound* e *Backtracking*

4 Relação com a Abordagem Gulosa

5 Relação com Programação Dinâmica

6 Relação com Divisão e Conquista