

made for free at coggle.it

A ideia principal é construir soluções um componente de cada vez e avaliar tais candidatos parciais da seguinte forma. Se uma solução parcial pode se desenvolver ainda mais sem violar as restrições do problema ao tomar a primeira opção válida restante para a próxima componente. Se não houver opção válida para o próxima componente, não é necessário considerar alternativas para os restantes. Assim, o algoritmo retrocede para substituir a última componente da solução parcial pela sua próxima opção.

Backtracking

Lidando com as Limitações do Poder dos
Algoritmos

Duas técnicas de design de algoritmos, backtracking
e branch-and-bound, que frequentemente permitem
resolver algumas instâncias grandes de problemas
combinatórios difíceis.

Comparado ao backtracking, o branch-and-bound requer dois itens adicionais: uma maneira de fornecer, para cada nó de uma árvore de espaço de estados, um limite para o melhor valor da função objetivo em qualquer solução que possa ser obtida adicionando mais componentes à solução parcialmente construída representada pelo nó, e o valor da melhor solução vista até agora.

Em geral, terminamos um caminho de busca do nó atual em uma árvore de espaço de estados de um algoritmo de branch-and-bound por uma das três razões a seguir:

- O valor do limite do nó não é melhor do que o valor da melhor solução vista até agora.
- O nó representa soluções inviáveis porque as restrições do problema já estão violadas.
- O subconjunto de soluções viáveis representado pelo nó consiste em um único ponto, neste caso, comparamos o valor da função objetivo para esta solução viável com o da melhor solução vista até agora e atualizamos o último com o primeiro se a nova solução for melhor.