

Problemas NP-Completo

Neste capítulo, pretende-se abordar problemas cujo melhor algoritmo que conhecemos para os resolver tem complexidade exponencial, o que implica um longo tempo de espera para inputs razoavelmente grandes.

Problemas de otimização vs problemas de decisão

A resolução de um problema de otimização consiste na seleção da melhor solução para outro problema.

A cada problema de otimização está geralmente associado um problema de **decisão**, i.e., um problema cuja solução será uma resposta do tipo **sim ou não**.

Classe de problemas P

Um algoritmo é limitado polinomialmente caso tenha um comportamento no pior caso em $O(P(n))$, com $P(n)$ um polinómio em n (a dimensão do input). Dizemos que um problema é limitado polinomialmente caso exista um algoritmo limitado polinomialmente que o resolva.

A classe P é precisamente constituída por problemas do tipo descrito acima. Esta classe inclui problemas razoáveis mas também outros de difícil resolução, dado que existem polinómios cujo crescimento é muito rápido.

Classe de problemas NP – problemas de decisão cuja verificação de soluções pode ser feita em tempo polinomial.

Problema de decisão = obtenção de resposta para um problema de existência de um objeto. Solução para o problema = objeto que justifica uma resposta verdadeira.

Para cada problema faz sentido que exista um processo ou algoritmo que dada uma solução proposta verifica se ela é ou não solução do problema. Uma solução proposta será descrita por uma *string*. Verificar a solução implica verificar que a string obedece ao formato utilizado para descrever as soluções, i.e., que é válida. Por outro lado, a verificação de que a solução proposta pela string respeita o critério do problema é também muito importante.

Algoritmos Não – determinísticos



(*) escrita na memória uma string arbitrária s . Em cada execução do algoritmo esta string pode variar.

(**) algoritmo lê s e processa-a, podendo suceder uma das seguintes situações: o algoritmo pára ao obter uma resposta sim; pára ao obter resposta não; algoritmo não pára.

A primeira fase é uma “tentativa de adivinhar” uma solução. A segunda fase verifica o “palpite” inicial.

P e NP

Teorema: $P \subseteq NP$.

[Estratégia de força bruta resolve os NP problemas em tempo exponencial]