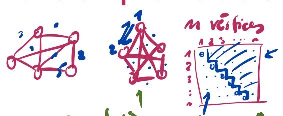
Busca heurística: solução não-exata de problemas de otimização; sem garantia de qualidade.

PCV (I)  n vértices $I = \{\ldots\}$

$S = \{\ldots, \text{etc} \ldots\}$

1. Introdução

instâncias soluções

Um *problema de busca* é uma relação binária $\mathcal{P} \subseteq I \times S$ com instâncias $x \in I$ e soluções $y \in S$. O par $(x, y) \in \mathcal{P}$ caso y é uma solução para x.

1, *

poly. não det. poly.

$P \subseteq NP$ Sim, não Decisão

Definição 1.1
A classe de complexidade FNP contém os problemas de busca com relações $\mathcal{P}$ polinomialmente limitadas (ver definição 1.3) tal que $(x, y) \in \mathcal{P}$ pode ser decidido em tempo polinomial.
A classe de complexidade FP contém os problemas em FNP para quais existe um algoritmo polinomial A com

PCV ∈ FNP

x∈I ① y_1 ② y_2 Test O(n)

$(x, y_1) \in P$? Sim $(x, y_2) \notin P$? NÃO

$$A(x) = \begin{cases} y & \text{para um } y \text{ tal que } (x, y) \in \mathcal{P} \\ \text{“insolúvel”} & \text{caso não existe } y \text{ tal que } (x, y) \in \mathcal{P} \end{cases}.$$

Teorema 1.1
FP=FNP se e somente se P=NP.

$FP \subseteq FNP$

Prova. Ver por exemplo C. Papadimitriou (1993, cáp. 10.3). ∎

Definição 1.2
função objetivo
Um *problema de otimização* $\Pi = (\mathcal{P}, \varphi, \text{opt})$ é uma relação binária $\mathcal{P} \subseteq I \times S$ com instâncias $x \in I$ e soluções $y \in S$, junto com

PCV
φ: Distância da rota
$\varphi(x, y) = 210$

- uma função de otimização (função de objetivo) $\varphi : \mathcal{P} \to \mathbb{N}$ (ou $\mathbb{Q}$).

$\varepsilon = 10^{-6}$

opt ∈ {min, max}

- um objetivo: Encontrar mínimo ou máximo

$$\text{OPT}(x) = \text{opt}\{\varphi(x, y) \mid (x, y) \in \mathcal{P}\}$$

junto com uma solução y^* tal que $f(x, y^*) = \text{OPT}(x)$.

O par $(x, y) \in \mathcal{P}$ caso y é uma solução para x.

PO } problemas de otim. podem ser em tempo polinomial
NPO } recalculados

Uma instância x de um problema de otimização possui soluções $S(x) = \{y \mid (x, y) \in \mathcal{P}\}$.

Convenção 1.1
Escrevemos um problema de otimização na forma

NOME

Instância x

Solução y

Objetivo Minimiza ou maximiza $\varphi(x, y)$.

Com um dado problema de otimização correspondem três problemas:

PCV

Resolve

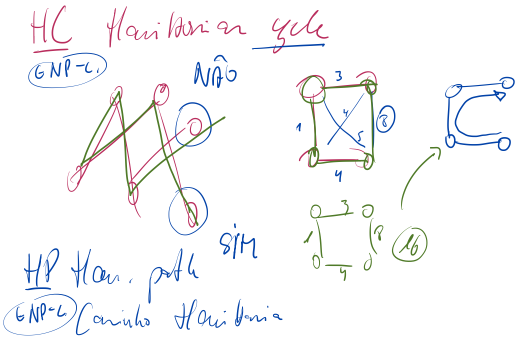
Valor ótimo

Força bruta ≤120?

- Construção: Dado x, encontra a solução ótima y^* e seu valor $\mathrm{OPT}(x)$. * Otimalidade
- Avaliação: Dado x encontra valor ótimo $\mathrm{OPT}(x)$. log. Poly
- Decisão: Dado x e k, decide se $\mathrm{OPT}(x) \geq k$ (maximização) ou $\mathrm{OPT}(x) \leq k$ (minimização). Poly

Definição 1.3
Uma relação binária R é *polinomialmente limitada* se

$$\exists p \in \mathrm{poly} : \forall (x, y) \in R : |y| \leq p(|x|).$$



Definição 1.4 (Classes de complexidade)
A classe PO consiste dos problemas de otimização tal que existe um algoritmo polinomial A com $\varphi(x, A(x)) = \mathrm{OPT}(x)$ para $x \in I$. OPT(x)
A classe NPO consiste dos problemas de otimização tal que

(i) As instâncias $x \in I$ são reconhecíveis em tempo polinomial.

(ii) A relação $\mathcal{P}$ é polinomialmente limitada.

(iii) Para y arbitrário, polinomialmente limitado: $(x, y) \in \mathcal{P}$ é decidível em tempo polinomial.

(iv) φ é computável em tempo polinomial.

1.1. Não tem almoço de graça

> "Sire in eight words I will reveal to you all the wisdom that I have distilled through all these years from all the writings of all the economists who once practiced their science in your kingdom. Here is my text: 'There ain't no such thing as free lunch' " (NN 1938)

A frase "there ain't no such thing as free lunch" (TANSTAFEL) expressa que uma vantagem (p.ex. o almoço de graça em bares dos EUA no século 19) tipicamente é pago de outra forma (p.ex. comida salgada e bebidas caras). Para problemas de busca e de otimização, Wolpert e Macready (1997) provaram teoremas que mostram que uma busca universal não pode ter uma vantagem em todos problemas de otimização.
Para um problema de otimização supõe que $\varphi : \mathcal{P} \to \Phi$ é restrito para um conjunto finito Φ, e seja $\mathcal{F} = \Phi^{S(x)}$ o espaço de todas funções objetivos para uma instância do problema. Um algoritmo de otimização avalia pares de soluções e valores $(s, v) \in S(x) \times \Phi$. Seja $D =$

$$\mathcal{F} = \{ f \mid f : S(x) \to \Phi \}$$

$(s_1, v_1) \to (s_2, v_2) \to (s_3, v_3) \to (s_4, v_4) \to \cdots$

$p = (s, v)$ $\quad p^*$

$m \geq 0$

$\bigcup_{m \geq 0} (S(x) \times \Phi)^m$ o conjunto de todas sequencias de pares. Um algoritmo de otimização que não repete avaliações pode ser modelado por uma função $a : d \in D \to \{s \mid s \neq s_i, \text{para } d_i = (s_i, v_i), i \in [|d|]\}$ que mapeia a sequencia atual para a próxima solução a ser avaliada (observe que o algoritmo toma essa decisão em função das soluções anteriormente visitadas e os seus valores). A avaliação de um algoritmo de otimização é através uma função $\Psi(d)$. Ela pode, por exemplo, atribuir a d o valor mínimo encontrado durante a busca.

Teorema 1.2 (Wolpert e Macready (1997)) NFL

Para algoritmos a, a', um número de passos m e uma sequencia de valores $v \in \Phi^m$

SA $\quad$ Loteria

função objetivo

$$\sum_{f \in \mathcal{F}} P(v \mid f, m, a) = \sum_{f \in \mathcal{F}} P(v \mid f, m, a'). \quad / |\mathcal{F}|$$

Prob. ver a seq. v | dado $\quad$ # passos alg.

O teorema mostra que uma busca genérica não vai ser melhor que uma busca aleatória em média sobre todas funções objetivos. Porém, uma grande fração das funções possíveis não ocorrem na prática (uma função aleatória é incompressível, i.e. podemos especificá-la somente por tabulação, funções práticas muitas vezes exibem localidade). Além disso, algoritmos de busca frequentemente aproveitam a estrutura do problema em questão.

1.2. Representação de soluções

A representação de soluções influencia as operações aplicáveis e a sua complexidade. Por isso a escolha de uma representação é importante para o desempenho de uma heurística. A representação também define o tamanho do espaço de busca, e uma representação compacta (e.g. 8 coordenadas versus permutações no problema das 8-rainhas) é preferível. Para problemas com muitas restrições uma representação implícita que é transformada para uma representação direta por um algoritmo pode ser vantajoso.

Para uma discussão abstrata usaremos frequentemente duas representações elementares. Na *representação por conjuntos* uma solução é um conjunto $S \subseteq U$ de um universo U. Os conjuntos válidos são dados por uma coleção $\mathcal{V}$ de subconjuntos de U. Na *representação por variáveis* uma instância é um subconjunto $I \subseteq U$, e uma solução é uma atribuição de valores de um universo V aos elementos em I.

Exemplo 1.1 (Representação do PCV por conjuntos)

Uma representação por conjuntos do PCV sobre um grafo $G = (V, A)$ é o universo de arestas $U = A$, com $\mathcal{V}$ todos subconjuntos que formam ciclos. ◊

Exemplo 1.2 (Representação do PCV por variáveis)

Uma representação por variáveis do PCV sobre um grafo $G = (V, A)$ usa um universo de vértices U. Uma instância $I = V$ atribui a cada cidade a próxima cidade no ciclo. Uma representação alternativa usa $I = [n]$ a atribui a cada variável $i \in I$ a i-ésima cidade no ciclo. ◊