



Algoritmia (e Estruturas de Dados) 3 **utad**
TIPOLOGIAS DE ALGORITMOS Pedro Melo Pinto 2022

Existem diversos métodos de classificação de algoritmos

só vamos abordar 2 dessas classificações

01. tipo de implementação
02. paradigma da abordagem (*design*)

Algoritmia (e Estruturas de Dados) 4 **utad**
TIPOLOGIAS DE ALGORITMOS Pedro Melo Pinto 2022

classificação de algoritmos 01.implementação

- A** Recursivos / Iterativos
- B** Lógicos
- C** Sequenciais / Paralelos
- D** Determinísticos / Não determinísticos
- E** Exactos / Aproximados

Algoritmia (e Estruturas de Dados)
TIPOLOGIAS DE ALGORITMOS

5

utad
Pedro Melo Pinto 2022

classificação de algoritmos 01.implementação

A Recursivos / Iterativos

Alguns problemas são naturalmente mais indicados para uma destas duas implementações. As implementações recursivas tem uma versão iterativa (mais ou menos complexa) equivalente, e vice-versa (Church-Turing thesis, 1930s).

exemplo:

<pre>function factorial - iterative pass in: INT n var: INT n_f, i n_f ← 1 repeat for i = 2 to n n_f ← n_f × i end repeat pass out: n_f end function</pre>	<pre>function factorial - recursive pass in: INT n if (n ≤ 1) pass out: 1 else pass out: n × (call: factorial(n-1)) end if end function</pre>
---	---

Algoritmia (e Estruturas de Dados)
TIPOLOGIAS DE ALGORITMOS

6

utad
Pedro Melo Pinto 2022

classificação de algoritmos 01.implementação

B Lógicos (programação em lógica)

É um paradigma declarativo
o programador declara os objectivos da computação,
mas não detalha o algoritmo pelo qual esses objectivos deverão ser alcançados

Os programas são expressos através de lógica simbólica

exemplo:

```

dado um elemento x e uma lista L, provar que x está em L
Provar que L é [x].
Senão, dividir L em duas sub-listas L1 e L2 e provar uma das
seguintes condições:
  x está em L1 ou x está em L2
  
```

Algoritmia (e Estruturas de Dados)
7
utad
Pedro Melo Pinto 2022

TIPOLOGIAS DE ALGORITMOS

classificação de algoritmos 01.implementação

C
Sequenciais / Paralelos

Usam um só processador de forma sequencial / usam mais de um computador ou um computador com múltiplos processadores.
Os algoritmos iterativos são, em geral, implementáveis de modo paralelo.
Os problemas que não possuem uma solução paralela são designados problemas sequenciais inatos.

Algoritmia (e Estruturas de Dados)
8
utad
Pedro Melo Pinto 2022

TIPOLOGIAS DE ALGORITMOS

classificação de algoritmos 01.implementação

C
Sequenciais / Paralelos

usam, para a resolução de um problema,
mais de um computador ou um
computador com múltiplos processadores.

Para alguns problemas (computacionais) uma solução eficaz passa pela utilização de computação paralela em que os algoritmos podem especificar múltiplas operações num só passo.

Desempenho do algoritmo:
$$\text{speedup} = \frac{\text{worst case execution time of the fastest known sequential for a particular problem}}{\text{worst case execution time of the parallel algorithm}}$$

Número de processadores utilizados:

Este é um importante factor na análise a eficiência de um algoritmo paralelo.
Quanto maior este número, mais cara (custo, manutenção, etc.) é a solução.

Custo Total:

total cost = Time complexity × Number of processors used

Algoritmia (e Estruturas de Dados)

TIPOLOGIAS DE ALGORITMOS

9

utad

Pedro Melo Pinto 2022

classificação de algoritmos 01.implementação

usam, para a resolução de um problema, mais de um computador ou um computador com múltiplos processadores.

C Sequenciais / Paralelos

Para alguns problemas (computacionais) uma solução eficaz passa pela utilização de computação paralela em que os algoritmos podem especificar múltiplas operações num só passo.

Tipos de computação Paralela:
multiprocessador com memória partilhada - modelo PRAM (Parallel Random Access Machines);
multicomputador com memória distribuída

Algoritmia (e Estruturas de Dados)

TIPOLOGIAS DE ALGORITMOS

10

utad

Pedro Melo Pinto 2022

classificação de algoritmos 01.implementação

C Sequenciais / Paralelos

EXEMPLO 01.03 Algoritmo Paralelo para Determinar o Mínimo

Problema: Determinar o mínimo de uma colecção de n elementos

Solução paralela:

- as comparações entre números podem ser executadas independentemente
- o resultado é encontrado utilizando escrita simultânea/concorrente

C

9725

P1P2P3P4P5P6P7P8P9P10P11P12P13P14P15P16

1101

M

Algoritmo A1

for each $1 \leq i \leq n$ do in parallel
 $M[i] \leftarrow \emptyset$
for each $1 \leq i, j \leq n$ do in parallel
 if $i \neq j$ and $C[i] \leq C[j]$ then $M[j] \leftarrow 1$
for each $1 \leq i \leq n$ do in parallel
 if $M[i] = 0$ then output $:= i$

$O(1)$

UTAD | Algoritmia (e Estruturas de Dados)

5

Algoritmia (e Estruturas de Dados)

TIPOLOGIAS DE ALGORITMOS

11

utad

Pedro Melo Pinto 2022

classificação de algoritmos 01.implementação

Sequenciais / Paralelos

EXEMPLO 01.03 Algoritmo Paralelo para Determinar o Mínimo

Problema: Determinar o mínimo de uma colecção de n elementos

Solução paralela (redução para $n^{1+\epsilon}$ processadores)

Algoritmo A_{k+1} :

- c1 $\epsilon = 1/2$
- c2 divisão do conjunto C em blocos disjuntos de tamanho n^ϵ
- c3 aplicar o algoritmo A_k a cada um dos blocos resultantes
- c4 aplicar o algoritmo A_k ao conjunto C' constituído pelos $\frac{n}{n^\epsilon}$ mínimos dos blocos considerados

Teorema

Podemos encontrar o mínimo de n elementos com uma complexidade $O(1/\epsilon)$, num modelo CRCW PRAM com $O(n^{1+\epsilon})$ processadores, para qualquer $0 < \epsilon \leq 1$.

Algoritmia (e Estruturas de Dados)

TIPOLOGIAS DE ALGORITMOS

12

utad

Pedro Melo Pinto 2022

classificação de algoritmos 01.implementação

Determinísticos / Não determinísticos

Os algoritmos em que o resultado de uma qualquer operação é definida de maneira única são designados Determinísticos (está de acordo com o funcionamento dos sistemas de computação actuais).

Se retirarmos essa restrição na saída das operações, temos operações que não estão definidas de modo único mas sim limitadas a um conjunto especificado de possibilidades (as máquinas que os executam poderão escolher uma de várias possíveis saídas para determinada operação).

Uma possível interpretação de Não Determinismo em computação é “paralelismo sem restrições”

Algoritmia (e Estruturas de Dados)
13
utad
Pedro Melo Pinto 2022

TIPOLOGIAS DE ALGORITMOS

classificação de algoritmos 01.implementação

E

Exactos / Aproximados

Um algoritmo exacto oferece apenas soluções óptimas. Se o problema tiver uma solução, o algoritmo encontra a solução óptima, de acordo com os critérios de pesquisa.

Muitos problemas são tão difíceis, que tentar uma solução exacta é ambicioso demais para o poder computacional (ou a matemática) actualmente disponível.

Do you remember the Travelling Salesman problem?

Portanto, usamos algoritmos aproximados, que encontram maneiras inteligentes (mais “baratas”) de chegar o mais próximo possível da solução ideal, por exemplo, evitando a enumeração, no espaço de pesquisa, de muitas das possíveis soluções.

Geralmente, com esses métodos, tenta-se obter (provar) alguma garantia (limite) de que a solução não estará mais longe do ideal que um determinado valor x .

Algoritmia (e Estruturas de Dados)
14
utad
Pedro Melo Pinto 2022

TIPOLOGIAS DE ALGORITMOS

classificação de algoritmos 01.implementação

E

Exactos / Aproximados

exemplo:

Problema da cobertura de vértices de um grafo (não orientado) G - Dado um grafo (não orientado) G , encontrar a cobertura de vértices de G de tamanho mínimo.

A vertex cover of an undirected graph is a subset of its vertices such that for every edge (u, v) of the graph, either 'u' or 'v' is in the vertex cover.

Algoritmia (e Estruturas de Dados) 15 utad Pedro Melo Pinto 2022

TIPOLOGIAS DE ALGORITMOS

classificação de algoritmos 02.paradigma da abordagem

- A Força Bruta (*exhaustive search*)
- B1 Divisão-e-Conquista
- B2 Redução-e-Conquista
- B3 Transformação-e-Conquista
- C Programação Dinâmica
- D Algoritmos Gulosos (*greedy*)
- E Programação Linear
- F Pesquisa e Enumeração
- G Probabilísticos e Heurísticos
- H Baseados nos Dados

Algoritmia (e Estruturas de Dados) 16 utad Pedro Melo Pinto 2022

TIPOLOGIAS DE ALGORITMOS

classificação de algoritmos 02.paradigma da abordagem

A

Força Bruta (*exhaustive search*)

Aplica uma solução elementar (geralmente) baseada no conceito subjacente ao problema. Executa repetidamente uma operação simples até à solução (problemas numéricos → força bruta) ou verifica exaustivamente todas as soluções do problema até encontrar a correcta (problemas combinatórios + pesquisa e ordenação → *exhaustive search*).

exemplos

- 1 ordenação por selecção e por flutuação (*bubble sort*);
- 2 pesquisa sequencial;
- 3 selecção de máximo ou mínimo em sequências;
- 4 travessia em profundidade e largura (grafos).

Algoritmia (e Estruturas de Dados)
17
utad
Pedro Melo Pinto 2022

TIPOLOGIAS DE ALGORITMOS

classificação de algoritmos 02.paradigma da abordagem

A Força Bruta (*exhaustive search*)

Aplica uma solução elementar (geralmente) baseada no conceito subjacente ao problema. Executa repetidamente uma operação simples até à solução (problemas numéricos → força bruta) ou verifica exaustivamente todas as soluções do problema até encontrar a correcta (problemas combinatórios + pesquisa e ordenação → *exhaustive search*).

exemplos

- 1 ordenação por selecção e por flutuação (*bubble sort*);
- 2 pesquisa sequencial;
- 3 selecção de máximo ou mínimo em sequências;
- 4 travessia em profundidade e largura (grafos).

```

function linear search
pass in: ARRAY INT a[], INT n, INT x
var: INT i

repeat for i = 1 to n
  if v[i] = x
    pass out: i
  end if
end repeat

pass out: -1
end function

```

Algoritmia (e Estruturas de Dados)
18
utad
Pedro Melo Pinto 2022

TIPOLOGIAS DE ALGORITMOS

classificação de algoritmos 02.paradigma da abordagem

B1 Divisão-e-Conquista

Divide sucessivamente um problema em um ou mais problemas menores (habitualmente de modo recursivo) até que esse problema esteja de tal modo dividido que a solução é simples/conhecida.

exemplos:

- 1 método de ordenação *quicksort*;
- 2 método de ordenação *mergesort*;
- 3 algoritmos recursivos de árvores binárias.

Algoritmia (e Estruturas de Dados)

TIPOLOGIAS DE ALGORITMOS

19

utad

Pedro Melo Pinto 2022

classificação de algoritmos 02.paradigma da abordagem

B1 Divisão-e-Conquista

Divide sucessivamente um problema em um ou mais problemas menores (habitualmente de modo recursivo) até que esse problema esteja de tal modo dividido que a solução é simples/conhecida.

exemplos:

- ① método de ordenação *quicksort*;
- ② método de ordenação *mergesort*;
- ③ algoritmos recursivos de árvores binárias

```
function mergesort
pass in: ARRAY INT a[ ], INT left, right
var: INT m
if left < right
    m ← (left + right) / 2
    call: mergesort(a, left, m)
    call: mergesort(a, m+1, right)
    call: merge(a, left, m, right)
endif
end function
```

Algoritmia (e Estruturas de Dados)

TIPOLOGIAS DE ALGORITMOS

20

utad

Pedro Melo Pinto 2022

classificação de algoritmos 02.paradigma da abordagem

B2 Redução-e-Conquista

Variante da metodologia Divisão-e-Conquista

Resolve um sub-problema (por redução/diminuição sucessiva da dimensão da entrada).

A diminuição da dimensão da entrada pode ser feita de modo recursivo ou iterativo.

O decremento da diminuição da dimensão da entrada pode ser constante (geralmente unitário), pode ser um factor constante (por exemplo 2, ou seja, diminuir a dimensão para metade) ou variável.

exemplos:

- ① método de ordenação por inserção;
- ② pesquisa binária (decremento com factor constante: 2);
- ③ algoritmos de inserção, remoção e pesquisa em árvores (decremento com factor variável: depende da árvore).

Algoritmia (e Estruturas de Dados) 21 utad
TIPOLOGIAS DE ALGORITMOS Pedro Melo Pinto 2022

classificação de algoritmos 02.paradigma da abordagem

B2 Redução-e-Conquista

Variante da metodologia Divisão-e-Conquista

Resolve um sub-problema (por redução/diminuição sucessiva da dimensão da entrada).
A diminuição da dimensão da entrada pode ser feita de modo recursivo ou iterativo.
O decréscimo da diminuição da dimensão da entrada pode ser constante (geralmente unitário), pode ser um factor constante (por exemplo 2, ou seja, diminuir a dimensão para metade) ou variável.

exemplos:

- 1 método de ordenação por inserção;
- 2 pesquisa binária (decréscimo com factor constante: 2);
- 3 algoritmos de inserção, remoção e pesquisa em árvores (decréscimo com factor variável: depende da árvore).

Algoritmia (e Estruturas de Dados) 22 utad
TIPOLOGIAS DE ALGORITMOS Pedro Melo Pinto 2022

classificação de algoritmos 02.paradigma da abordagem

B Divisão-e-Conquista **vs.** Redução-e-Conquista

O algoritmo *mergesort* (divisão-e-conquista) divide a lista em duas listas separadas para ordenar recursivamente cada uma das sub-listas, enquanto o algoritmo de pesquisa binária (redução-e-conquista) utiliza sequencialmente metade da lista anterior para efectuar a pesquisa.

A divisão-e-conquista, ao dividir um problema em múltiplos sub-problemas, resulta numa fase de conquista mais complexa do que num algoritmo de redução-e-conquista.

Algoritmia (e Estruturas de Dados)
TIPOLOGIAS DE ALGORITMOS

23

utad
Pedro Melo Pinto 2022

classificação de algoritmos 02.paradigma da abordagem

B3 Transformação-e-Conquista

Modifica o problema (e pode diminuir a dimensão da entrada), para obter um problema de mais fácil resolução (e de menor dimensão).

Esta estratégia pode ser feita de modo recursivo ou iterativo.

Por exemplo: encontrar a mediana de um conjunto de valores pode ser transformado no problema de encontrar o elemento do meio, se esse conjunto de valores for previamente ordenado.

exemplos:

- 1 método de ordenação *heapsort*;
- 2 algoritmos de inserção e remoção e pesquisa em árvores equilibradas de tipo *red-black*.

Algoritmia (e Estruturas de Dados)
TIPOLOGIAS DE ALGORITMOS

24

utad
Pedro Melo Pinto 2022

classificação de algoritmos 02.paradigma da abordagem

C Programação Dinâmica

Resolve problemas compostos por sub-problemas duplicados (normalmente em resultado da recursividade) que são repetidamente recalculados.

O cálculo destes sub-problemas é efectuado uma só vez, sendo "memorizados" para posterior utilização. Evita-se assim a recomputação de soluções já efectuadas. Para tal, um problema tem de apresentar uma sub-estrutura ótima, ou seja, a solução ótima para esse problema pode ser obtida a partir das soluções ótimas dos sub-problemas.

exemplos:

- 1 algoritmos numéricos repetitivos
ex: números de *Fibonacci*;
- 2 algoritmo de Floyd em grafos
(todos os pares de caminhos mais curtos).

Algoritmia (e Estruturas de Dados)

TIPOLOGIAS DE ALGORITMOS

25

utad

Pedro Melo Pinto 2022

classificação de algoritmos 02.paradigma da abordagem

C Programação Dinâmica

Resolve problemas compostos por sub-problemas duplicados (normalmente em resultado da recursividade) que são repetidamente recalculados.

O cálculo destes sub-problemas é efectuado uma só vez, sendo "memorizados" para posterior utilização.

exemplo: *números de Fibonacci*

```
function Fibonacci numbers - solução recursiva
pass in: INT n
var: ARRAYINT f[], i
if n < 2
  pass out: n
end if
pass out: (call: Fibonacci(n-1)) + (call: Fibonacci(n-2))
end function
```

```
function Fibonacci numbers - dynamic programming
pass in: INT n, ARRAYINT f[]
if n < 2
  pass out: n
end if
if f[n] ≠ 0
  pass out: f[n]
end if
f[n] ← (call: Fibonacci(n-1, f)) + (call: Fibonacci(n-2, f))
pass out: f[n]
end function
```

top-down version

Algoritmia (e Estruturas de Dados)

TIPOLOGIAS DE ALGORITMOS

26

utad

Pedro Melo Pinto 2022

classificação de algoritmos 02.paradigma da abordagem

C Programação Dinâmica **vs.** Divisão-e-Conquista

A diferença principal entre a programação dinâmica e a divisão-e-conquista é que nesta os sub-problemas são (mais ou menos) independentes, enquanto que na programação dinâmica eles são sobrepostos.

Algoritmia (e Estruturas de Dados)
TIPOLOGIAS DE ALGORITMOS

27

utad
Pedro Melo Pinto 2022

classificação de algoritmos 02.paradigma da abordagem

C Programação Dinâmica **vs.** Recursividade

A diferença entre a programação dinâmica e a recursividade está no *caching* dos valores já calculados (ou *memoization*) das chamadas recursivas. Uma função recursiva, que não utiliza *memoization*, vai calcular diversas vezes o mesmo problema, sendo assim ineficaz.

Algoritmia (e Estruturas de Dados)
TIPOLOGIAS DE ALGORITMOS

28

utad
Pedro Melo Pinto 2022

classificação de algoritmos 02.paradigma da abordagem

D Algoritmos Gulosos (*greedy*)

Decompõe o problema num conjunto de etapas e tenta otimizar a solução partindo do pressuposto que a escolha sucessiva das melhores soluções em cada etapa conduz à melhor solução do problema.

As escolhas das melhores soluções (imediatistas ou locais) em cada etapa podem conduzir a soluções óptimas ou subóptimas (*less-than-optimal*).

exemplos

- 1 algoritmos MST (árvore abrangente de menor custo) de Prim e Kruskal;
- 2 algoritmo de caminho mais curto (em grafos) de Dijkstra.

Algoritmia (e Estruturas de Dados)

TIPOLOGIAS DE ALGORITMOS

utad

Pedro Melo Pinto 2022

classificação de algoritmos 02.paradigma da abordagem

Algoritmos Gulosos vs. Programação Dinâmica

Um algoritmo guloso é semelhante a um algoritmo de programação dinâmica mas, em vez de ser necessário o prévio conhecimento das soluções para os sub-problemas, é feita uma escolha de maior lucro (gulosa - ótimo local) em cada altura, baseada na melhor decisão (e não todas) da etapa anterior.

Algoritmia (e Estruturas de Dados)

TIPOLOGIAS DE ALGORITMOS

utad

Pedro Melo Pinto 2022

classificação de algoritmos 02.paradigma da abordagem

Algoritmos Gulosos

EXEMPLO 02.06A

Algoritmo Guloso para Cálculo do Troco de uma Compra

Problema: Fazer o troco de determinado montante, utilizando o mínimo número possível de moedas

algoritmo informal

Começar com nada.

Em cada altura, e não ultrapassando o montante necessário:
Acrescentar a moeda maior às moedas já escolhidas.

A moeda de maior valor, menor do que o troco restante, é um ótimo local.
(em geral, para encontrar a solução ótima, este problema necessita de programação dinâmica)

Troco: 2,61€

2,61 - 2,00 = 0,61€

0,61 - 0,50 = 0,11€

0,11 - 0,10 = 0,01€

0,01 - 0,01 = 0,00€

Algoritmia (e Estruturas de Dados) 31 utad
TIPOLOGIAS DE ALGORITMOS Pedro Melo Pinto 2022

classificação de algoritmos 02.paradigma da abordagem
D Algoritmos Gulosos

EXEMPLO 02.06B *Algoritmo Guloso para Cálculo do Troco de uma Compra*
Problema: Fazer o troco de determinado montante, utilizando o mínimo número possível de notas

algoritmo informal
Começar com nada.
Em cada altura, e não ultrapassando o montante necessário:
Acrescentar a nota maior às notas já escolhidas.

A nota de maior valor, menor do que o troco restante, é um ótimo local.
(em geral, para encontrar a solução ótima, este problema necessita de programação dinâmica)

Troco: \$15,00

A melhor solução melhor seria utilizar duas notas de 7 e uma de 1 (total = 3 notas).

A solução gulosa não é a solução ótima neste caso.

notas existentes




Algoritmia (e Estruturas de Dados) 32 utad
TIPOLOGIAS DE ALGORITMOS Pedro Melo Pinto 2022

classificação de algoritmos 02.paradigma da abordagem

E Programação Linear

A resolução de um problema através de programação linear envolve a maximização (ou minimização) das entradas de um conjunto de desigualdades lineares.

função linear a maximizar/minimizar $f(x_1, x_2, \dots, x_n) = c_1x_1 + c_2x_2 + \dots + c_nx_n$

sabendo que (restrições do problema) $g_1(x_1, x_2, \dots, x_n) =, \geq, \leq b_1$
 $g_2(x_1, x_2, \dots, x_n) =, \geq, \leq b_2$
...
 $g_m(x_1, x_2, \dots, x_n) =, \geq, \leq b_m$

e que (restrições das variáveis) $x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0$

A solução pode ser obtida através de um algoritmo genérico tipo *simplex* (dual) ou do uso de métodos de pontos interiores (p.e. algoritmo de Karmarkar).

exemplo
1 fluxo máximo em grafos orientados.

Algoritmia (e Estruturas de Dados)
TIPOLOGIAS DE ALGORITMOS

33

utad
Pedro Melo Pinto 2022

classificação de algoritmos 02.paradigma da abordagem

F Pesquisa e Enumeração

Muitos problemas (como o xadrez) podem modelar-se através de grafos.

Os algoritmos de pesquisa (percurso) em grafos especificam as regras para a exploração desses grafos.

Esta categoria inclui os algoritmos de pesquisa, *branch and bound* e *backtracking*.

Algoritmia (e Estruturas de Dados)
TIPOLOGIAS DE ALGORITMOS

34

utad
Pedro Melo Pinto 2022

classificação de algoritmos 02.paradigma da abordagem

F Pesquisa e Enumeração

1 Branch and Bound

Enumeração sistemática de todos os candidatos a solução através de pesquisa no espaço de estados.

O problema é organizado através de uma árvore de sub-problemas (candidatos) que se vão constituindo à medida que o algoritmo avança. O problema original é o "problema raiz" dessa árvore.

O algoritmo vai explorando os ramos (*branches*) da árvore, que contêm os candidatos e cuja solução resultante é antes verificada em relação aos limites superior e inferior estimados da solução ótima.

Os candidatos são descartados se não produzirem uma solução melhor do que a encontrada até aí.

exemplo

1 problema do caixeiro viajante.

Algoritmia (e Estruturas de Dados)
TIPOLOGIAS DE ALGORITMOS

35

utad
Pedro Melo Pinto 2022

classificação de algoritmos 02.paradigma da abordagem

F Pesquisa e Enumeração

2 *Backtracking* (recursividade com retrocesso)

A pesquisa exaustiva verifica todas as soluções possíveis do problema.

A recursividade com retrocesso elimina recursivamente soluções possíveis (em análise) que não respeitem as condições do problema, otimizando a pesquisa.

Os candidatos são descartados se não produzirem uma solução melhor do que a encontrada até aí.

exemplo

- 1 determinação do caminho Hamiltoniano.
(passar por todos os vértices de um grafo não orientado G, não repetindo nenhum)

Algoritmia (e Estruturas de Dados)
TIPOLOGIAS DE ALGORITMOS

36

utad
Pedro Melo Pinto 2022

classificação de algoritmos 02.paradigma da abordagem

G Probabilísticos e Heurísticos

Para alguns problemas (computacionais) pode ser difícil a formulação de uma solução com boa eficácia (tempo de execução) ou essa solução pode apresentar um grande crescimento do tempo de execução com o número de entradas.

Neste caso recorre-se habitualmente a:

- algoritmos probabilísticos
- heurísticas eficientes
- algoritmos aproximados

Algoritmia (e Estruturas de Dados)
37
utad
Pedro Melo Pinto 2022

TIPOLOGIAS DE ALGORITMOS

classificação de algoritmos 02.paradigma da abordagem

G Probabilísticos e Heurísticos

1 Algoritmos Probabilísticos (*randomized algorithms*)

Um algoritmo probabilístico faz, pelo menos, uma escolha aleatória (ou pseudo-aleatória).

Esta abordagem revela-se útil na obtenção de soluções aproximadas, em casos em que a determinação de soluções exacta seja impraticável (tal como acontece no uso de heurísticas)

exemplos

1 método de ordenação *quicksort* aleatório.

a escolha do pivot é feita aleatoriamente obedecendo a uma condição:

o nº de elementos mais pequenos (ou maiores) do que o pivot têm de ser maior do que $n/4$ - *central splitter*.

Algoritmia (e Estruturas de Dados)
38
utad
Pedro Melo Pinto 2022

TIPOLOGIAS DE ALGORITMOS

classificação de algoritmos 02.paradigma da abordagem

G Probabilísticos e Heurísticos

2 Algoritmos Heurísticos

O objectivo geral não é encontrar a solução óptima, mas uma solução aproximada de forma fácil e rápida (para problemas em que o tempo ou os recursos são limitados), reduzindo o espaço de pesquisa das soluções.

Estes algoritmos procuram a solução de entre todas as possíveis, mas não garantem que a solução encontrada seja a melhor, sendo no entanto uma solução próxima da melhor (ou uma solução que não se aplica a todas entradas).

Os métodos utilizado são métodos conhecidos, tais como o menor custo (dos algoritmos gulosos), mas para ser fácil e rápido, o algoritmo ignora ou elimina algumas das exigências/necessidades do problema.

exemplos

1 caminhos mais curtos (em grafos).

Algoritma (e Estruturas de Dados)

TIPOLOGIAS DE ALGORITMOS

39

uta
Pedro Melo-Pinto, 2020

classificação de algoritmos

02.paradigma da abordagem

G

Probabilísticos e Heurísticos

2

Algoritmos Heurísticos

O objectivo geral não é encontrar a solução óptima, mas uma solução aproximada de forma fácil e rápida (para problemas em que o tempo ou os recursos são limitados), reduzindo o espaço de pesquisa das soluções.

A. Algoritmos Genéticos.

Tentam encontrar as soluções mimetizando os processos evolutivos (biológicos), ao emularem a reprodução e sobrevivência do mais apto (através de sucessivas gerações de soluções) e incorporando mutações aleatórias.

Uma solução do problema é representada como um "genoma" (ou cromossoma). O algoritmo gera uma população de soluções, utilizando operadores genéticos tais como mutação, herança, selecção e crossover fazendo evoluir as soluções até encontrar a(s) melhor(es).

utad
Pedro Melo-Pinto 202

Algoritmos (e Estruturas de Dados)

TIPOLOGIAS DE ALGORITMOS

40

classificação de algoritmos 02.paradigma da abordagem

- G Probabilísticos e Heurísticos
- 2 Algoritmos Heurísticos

EXEMPLO 02.09B Algoritmos Genéticos

O tipo mais comum de algoritmo genético funciona assim:

algoritmo genético

- a população inicial é um grupo de indivíduos criados aleatoriamente.
- os indivíduos da população são avaliados:
 - a função de avaliação é fornecida
 - e dá aos indivíduos uma pontuação baseada no seu desempenho
- seleccionam-se dois elementos com base na sua aptidão.
 - quanto maior for a aptidão, maior a probabilidade de ser seleccionado
- estes elementos "reproduzem-se" (um ou mais descendentes).
- os descendentes sofrem mutações de forma aleatória.
- o processo repete-se até que seja encontrada uma solução adequada.
 - (ou até que um certo número de gerações tenham passado)

O diagrama ilustra o ciclo iterativo de um algoritmo genético. No topo, dentro de um círculo laranja rotulado "Genetic Algorithms", há três etapas principais: "Mutate some randomly" (representado por ícones de cápsulas azuis e brancas), "Create random Specimen" (representado por ícones de cápsulas) e "Breed new Generation" (representado por ícones de cápsulas). Uma seta curva indica o fluxo de "Find best Specimen" (representado por ícones de cápsulas) para a etapa de reprodução. O ciclo se repete continuamente.

Algoritmia (e Estruturas de Dados) 41 utad Pedro Melo Pinto 2022

TIPOLOGIAS DE ALGORITMOS

classificação de algoritmos 02.paradigma da abordagem

- A Força Bruta (*exhaustive search*)
- B1 Divisão-e-Conquista
- B2 Redução-e-Conquista
- B3 Transformação-e-Conquista
- C Programação Dinâmica
- D Algoritmos Gulosos (*greedy*)
- E Programação Linear
- F Pesquisa e Enumeração
- G Probabilísticos e Heurísticos
- H Baseados nos Dados

Algoritmia (e Estruturas de Dados) 42 utad Pedro Melo Pinto 2022

TIPOLOGIAS DE ALGORITMOS

classificação de algoritmos 02.paradigma da abordagem

- H Baseados nos Dados

Em vez de utilizar algoritmos que só têm garantias de desempenho de pior caso, podemos otimizar grandes famílias de algoritmos parametrizados e ajustar os parâmetros desses algoritmos usando um processo/conjunto de treino de amostras (exemplos).

Contudo, este processo não tem desempenho garantido.

Para muitos problemas combinatórios significativos, como sejam os problemas particionamento ou a selecção de sub-conjuntos, um pequeno ajuste nos parâmetros pode causar uma cadeia de mudanças no comportamento do algoritmo, de modo que o desempenho do algoritmo é uma função descontínua dos seus parâmetros.

O seu desenho pode ser feito através de aprendizagem estatística (*statistical learning*) ou aprendizagem online (*online learning*)

exemplos

- 1 classificação (*clustering*) de imagens utilizando *deep learning*.

Algoritmia (e Estruturas de Dados)

TIPOLOGIAS DE ALGORITMOS

43

utad

Pedro Melo Pinto 2022

classificação de algoritmos

02.paradigma da abordagem

Baseados nos Dados

EXEMPLO 02.10A

Clustering

Problema: Classificar uma imagem utilizando um modelo Deep Learning

O problema de agrupar (*clustering*) é um dos mais básicos em Ciência dos Dados – dado um grande conjunto de dados complexos (p.e., imagens ou artigos de jornal), o objectivo é organizá-los em conjuntos de elementos semelhantes/do mesmo tipo.

AlexNet (2012)

Input

CONV1

CONV2

CONV3

CONV4

CONV5

FC6

FC7

FC8

227 × 227 × 3

55 × 55 × 96

27 × 27 × 256

13 × 13 × 384

13 × 13 × 384

13 × 13 × 256

4096

4096

1000

extração de características

classificador

Algoritmia (e Estruturas de Dados)

TIPOLOGIAS DE ALGORITMOS

44

utad

Pedro Melo Pinto 2022

classificação de algoritmos

02.paradigma da abordagem

Baseados nos Dados

EXEMPLO 02.10A

Clustering

Problema: Classificar uma imagem utilizando um modelo Deep Learning

Amostras do conjunto de treino - ImageNet (> 14 milhões)

AlexNet (2012)

Input

CONV1

CONV2

CONV3

CONV4

CONV5

FC6

FC7

FC8

227 × 227 × 3

55 × 55 × 96

27 × 27 × 256

13 × 13 × 384

13 × 13 × 384

13 × 13 × 256


4096

4096

1000

extração de características

classificador



Algoritmia (e Estruturas de Dados)

TIPOLOGIAS DE ALGORITMOS

45

utad

Pedro Melo Pinto 2022

classificação de algoritmos

02.paradigma da abordagem

Baseados nos Dados

EXEMPLO 02.10A

Clustering

Problema: Classificar uma imagem utilizando um modelo Deep Learning

Exemplos de resultados

AlexNet (2012)

Input

CONV1

CONV2

CONV3

CONV4

CONV5

FC6

FC7

FC8

224 x 224 x 3

55 x 55 x 96

27 x 27 x 128

13 x 13 x 192

13 x 13 x 128

13 x 13 x 128


1000

1000


1000




mite



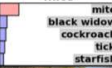
container ship



motor scooter



leopard




black widow




lifeboat




go-kart




jaguar




cockroach




amphibian



moped



cheetah



tick



fireboat



bumper car



snow leopard



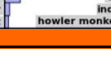
starfish




drilling platform




golfcart



Egyptian cat




grille




convertible




agaric




dalmatian



grille



mushroom




mushroom




grape



pickup




jelly fungus



elderberry



spider monkey




beach wagon



gill fungus




ffordshire bullterrier



titi




fire engine



dead-man's-fingers



currant



howler monkey

Leitura Adicional:

- 1

Levitin A., 2012.
Introduction to The Design and Analysis of Algorithms 3rd Edition. **CAPÍTULOS 3-10**
- 2

Muitos outros livros abordam algoritmos das diferentes tipologias de forma mais sistemática ou fragmentada.