ALGORITMOS GENÉTICOS

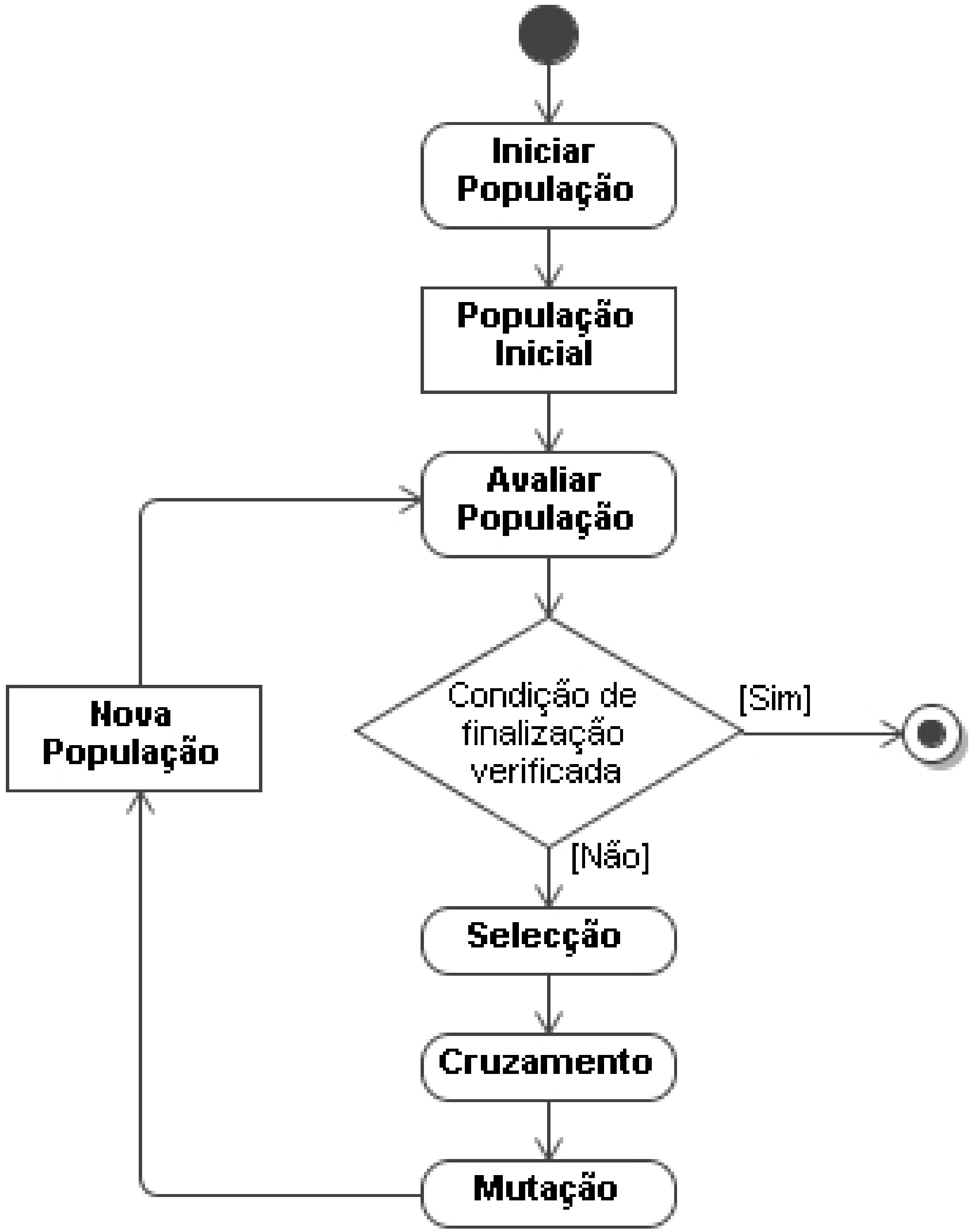
Luís Morgado

ISEL-ADEETC

POPULAÇÃO

- **Conjunto de soluções candidatas** (indivíduos - *fenótipo*)
  - Representadas por cromossomas - *genótipo*
- **Geração**
  - População no instante $t + 1$, constituída por indivíduos resultantes da aplicação das operações *selecção* (*reprodução*), *cruzamento* e *mutação* à população no instante t
- **Evolui com o tempo**

ALGORITMO GENÉTICO



ALGORITMO GENÉTICO

```
p(t) - População no instante de tempo t (geração)
f(x) - Função de adequação (de um indivíduo x)

1) Iniciar t = 0
2) Inicializar população p(t)
   (por exemplo gerar n indivíduos de forma aleatória)
3) Enquanto critério de finalização não se verificar
   (adequação, tempo ou outro)
    4)  Para cada x existente em p(t) calcular f(x)
    5)  t = t + 1
    6)  Seleccionar p(t) a partir de p(t-1)
    7)  Recombinar p(t)
    8)  Aplicar Mutações a p(t)
```

ALGORITMO GENÉTICO: EXEMPLO

```
function GENETIC-ALGORITHM(population, FITNESS-FN) returns an individual
  inputs: population, a set of individuals
          FITNESS-FN, a function that measures the fitness of an individual

  repeat
     new_population ← empty set
     for i = 1 to SIZE(population) do
        x ← RANDOM-SELECTION(population, FITNESS-FN)
        y ← RANDOM-SELECTION(population, FITNESS-FN)
        child ← REPRODUCE(x, y)
        if (small random probability) then child ← MUTATE(child)
        add child to new_population
     population ← new_population
  until some individual is fit enough, or enough time has elapsed
  return the best individual in population, according to FITNESS-FN
```

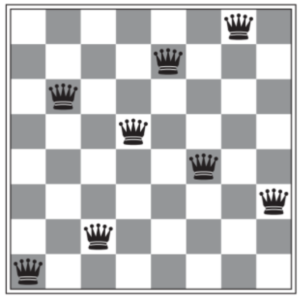
[Russel & Norvig, 2010]

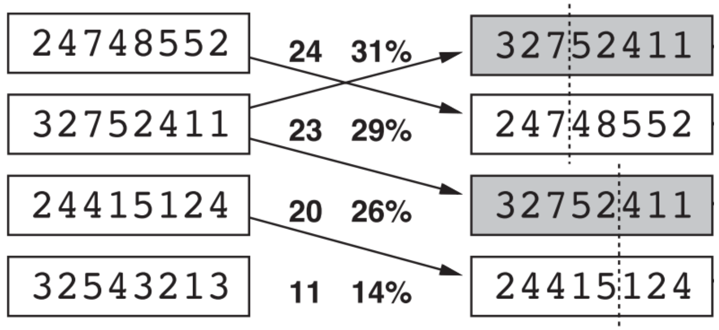
REPRESENTAÇÃO DO PROBLEMA

ESCOLHA DE UMA CODIFICAÇÃO

Importância da representação

- Garantir restrições do problema
- Reduzir multiplicidade
- **Exemplo**
  - Problema das N-Rainhas





Princípios base

- Escolher o **menor alfabeto que permite uma representação natural** do problema
- Escolher a codificação de tal forma que os **elementos simbólicos sejam significativos para o problema em análise e independentes de outros elementos**

Problemas a ter em conta

- Pendor de procura (*search bias*) eventualmente não adequado
- Semântica posicional

TÉCNICAS BASE DE CODIFICAÇÃO

Lista de características: sequência de bits

b_1	b_2	b_3		b_i		b_n

Cada bit representa uma única propriedade, característica, ou predicado.

bit i = 1 : Característica i presente

bit i = 0 : Característica i ausente

Linguagem de descrição de conceitos correspondente:

1 - Característica presente nas instâncias

0 - Característica não presente nas instâncias

- Característica com valores inconsistentes para instâncias distintas

Aplicação

Classes cuja descrição possa ser realizada com base num conjunto de características essenciais, cuja presença ou ausência seja obrigatória

TÉCNICAS BASE DE CODIFICAÇÃO

Ordenação linear

- Cadeia de elementos utilizada para representar parâmetros pertencentes a um conjunto ordenado de valores mutuamente exclusivos
  - Valores inteiros
  - Parâmetros contínuos com uma determinada precisão
  - Valores simbólicos

Exemplo

Problema: Encontrar a configuração de pesos de uma rede neuronal

Codificação: Blocos com valores reais representam pesos das ligações

Cromossoma: [0.3254, -2.5310, 0.5287, 1.4629, 0.7461]

REPRESENTAÇÃO DO PROBLEMA

- **Técnicas de codificação**
  - **Codificação binária**
    - Sequência de bits
  - **Codificação inteira**
    - Atributos inteiros ordenados ou não ordenados
  - **Codificação real**
    - Aproximação de valores reais
  - **Permutações**
    - Para problemas que implicam restrições de ordem
    - Valores não podem ser repetidos no mesmo genótipo
  - **Codificação em árvore**
    - Representação de árvores sintácticas (e.g. programa)
    - Programação genética

EXEMPLO: Problema do Caixeiro Viajante

Descrição do problema

- **Informação a representar**
  - Cidades e respectivas distâncias entre cidades
- **Resultado a obter**
  - Percurso tal que:
    - Cada cidade seja visitada uma única vez
    - Termine no ponto de partida
    - Minimize a distância total percorrida

EXEMPLO: Problema do Caixeiro Viajante

Codificação dos percursos

Codificação 1: *Codificação binária e operadores genéticos base*

- Possível repetição de cidades num percurso - *percursos inválidos*
- Possível obtenção de códigos que não representam qualquer cidade

Codificação 2: *Codificação através de um vector de inteiros*

- Necessidade de operadores genéticos específicos
- O vector (c_1, c_2, …, c_n) representa o percurso

$$\begin{tikzcd} c_1 \arrow[r] & c_2 \arrow[r] & \ldots \arrow[r] & c_n \arrow[lll, bend left] \end{tikzcd}$$

EXEMPLO: Problema do Caixeiro Viajante

Operadores de cruzamento

Cruzamento com preservação de ordem

Escolha aleatória de dois pontos de corte que determinam um sub-percurso a copiar:

Percurso 1 :	(1 2 3 | 4 5 6 | 7 8 9)	⟶	Percurso 1_d:	(_ _ _ 6 9 4 _ _ _)
Percurso 2 :	(5 8 7 | 6 9 4 | 3 2 1)		Percurso 2_d:	(_ _ _ 4 5 6 _ _ _)

Preservação da ordem das cidades restantes:

- **Obtenção da ordem das cidades a partir do segundo ponto de corte**

  Percurso 1: 7 → 8 → 9 → 1 → 2 → 3 → 4 → 5 → 6

- **Eliminação das cidades já visitadas**

  Percurso 1: 7 → 8 → 9 → 1 → 2 → 3 → 4 → 5 → 6

- **Colocação das cidades restantes a partir do segundo ponto de corte**

Percurso 1 :	(_ _ _ 6 9 4 _ _ _)	⟶	Percurso 1_d:	(2 3 5 6 9 4 7 8 1)
Percurso 2 :	(_ _ _ 4 5 6 _ _ _)		Percurso 2_d:	(8 7 9 4 5 6 3 2 1)

EVOLUÇÃO DA POPULAÇÃO

Como caracterizar a evolução de uma população no âmbito de um algoritmo genético ?

Indivíduo ⟷ **Cadeia**

Indivíduos com *semelhanças* ⟷ **Esquema**

Um ***esquema*** descreve um conjunto de cadeias

ESQUEMAS

Representam **padrões de semelhança** entre cadeias

Permitem relacionar as semelhanças entre cadeias com altos valores de adequação

Cada esquema representa várias cadeias

Cada cadeia corresponde a múltiplos esquemas

Alfabeto para construção de esquemas:

Alfabeto de cadeias: $\boldsymbol{V}$

Alfabeto de esquemas: $\boldsymbol{V} \cup \{*\}$

Exemplo:

$\boldsymbol{V} = \{0, 1\}$

$\boldsymbol{V} \cup \{*\} = \{0, 1, *\}$

Esquema **H** = 0*00*1**1 (Representa várias cadeias)

CARACTERIZAÇÃO DE UM ESQUEMA

Ordem de um esquema H: $\boldsymbol{o}$**(H)**

Número de posições fixas do esquema

Exemplo: H = 1**** $\quad o(H) = 1$

H = 10*1* $\quad o(H) = 3$

Comprimento de definição de um esquema: $\boldsymbol{\delta}$**(H)**

Distância máxima entre posições fixas

Exemplo: H = 10*1* $\quad \delta(H) = 3$

H = **11* $\quad \delta(H) = 1$

H = 1**** $\quad \delta(H) = 0$

ESQUEMAS

Como são os esquemas afectados pela aplicação de operadores?

- **Selecção**
  - As cópias de uma cadeia representam os mesmos esquemas
- **Cruzamento**
  - Quanto menor for o comprimento de definição δ menor será a probabilidade de o cruzamento fazer desaparecer o esquema
- **Mutação**
  - Quanto menor for a ordem ***o*** menor será a probabilidade de a mutação fazer desaparecer o esquema

EQUAÇÃO DE CRESCIMENTO DE UMA POPULAÇÃO

Considerando apenas a reprodução

$$\underbrace{m(H,t+1)}_{\substack{\text{Número de representantes} \\ \text{de H no instante } t+1}} = m(H,t) \cdot \frac{f(H)}{\overline{f}}$$

Probabilidade de sobrevivência de um esquema a um cruzamento

$$p_{sc}(H) \geq 1 - p_c \cdot \underbrace{\frac{\delta(H)}{l-1}}$$

Probabilidade de cruzamento

Probabilidade máxima de destruição em caso de cruzamento

EQUAÇÃO DE CRESCIMENTO DE UMA POPULAÇÃO

Probabilidade de sobrevivência de um esquema a uma mutação:

$$(1-p_m)^{o(H)}$$

Probabilidade de mutação de uma posição

Sendo $p_m << 1$:

$$(1-p_m)^{o(H)} \approx 1 - o(H) \cdot p_m$$

Equação de crescimento geral (***Schema theorem***):

$$\boxed{m(H,t+1) \geq m(H,t) \cdot \frac{f(H)}{\overline{f}}\left[1 - p_c \cdot \frac{\delta(H)}{l-1} - o(H) \cdot p_m\right]}$$

[Holland, 1975]

EQUAÇÃO DE CRESCIMENTO DE UMA POPULAÇÃO

A equação de crescimento geral (*Schema theorem*) permite concluir que **esquemas com maior adequação vão aumentando o seu número de representantes em gerações sucessivas** (de forma exponencial).

Cada indivíduo representa um conjunto variado de esquemas, assim sendo o **número de esquemas efectivamente processado é superior ao número de indivíduos (*n*) da população**.

$\downarrow$

Paralelismo implícito

BLOCOS CONSTRUTORES (*BUILDING BLOCKS*)

Esquemas com **alta adequação média**, pequeno comprimento de definição e pequena ordem

Em termos gerais os algoritmos genéticos

Descobrem

Reforçam

Recombinam

blocos construtores de soluções de uma forma massivamente paralela

REFERÊNCIAS

[Goldberg , 1989]
David E. Goldberg, “Genetic Algorithms”, *Addison-Wesley*, 1989.

[Michalewicz, 1994]
Zbigniew Michalewicz, “Genetic Algorithms + Data Structures = Evolution Programs”, *Springer Verlag*, 1994.

[Mitchell, 1996]
Melanie Mitchell, “An Introduction to Genetic Algorithms”, *MIT Press*, 1996.

[Booker, 1991]
Lashon B. Booker, “Representing Attribute-Based Concepts in a Classifier System”, *in “Foundations of Genetic Algorithms”, Morgan Kaufmann*, 1991.

[Buckles & Petry , 1992]
B. Buckles, F. Petry, “Genetic Algorithms”, *IEEE Press*, 1992.

[Mitchell, 1997]
Tom M. Mitchell, “Machine Learning”, *McGraw-Hill*, 1997.

[Hornby *et al.*, 2006]
G. Hornby, A. Globus, D. Linden, J. Lohn, "Automated antenna design with evolutionary algorithms". American Institute of Aeronautics and Astronautics, 2006

[Holland, 1975]
J. Holland, "Adaptation in Natural and Artificial Systems". MIT Press, 1975