INTRODUÇÃO À INTELIGÊNCIA ARTIFICIAL
22-23

07. ALGORITMOS GENÉTICOS

Carlos Pereira
ISEC

Índice

Introdução

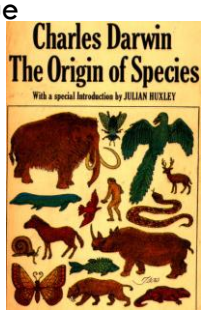
3

- Computação Evolucionária (CA)
  - A área de investigação designada por "Computação evolucionária (EC)" envolve:
    - Algoritmos Genéticos (GA)
    - Estratégias Evolucionárias (ES)
    - Programação Evolucionária (EP)

Introdução

4

- Algoritmos Genéticos
  - Técnica para resolução de problemas que necessitem de optimização
  - Baseados na Teoria de Evolução de Darwin

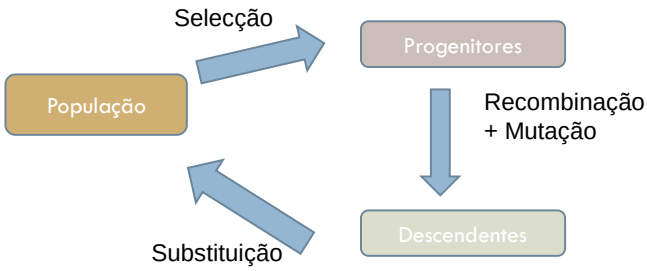


    - Sub-classe da computação evolucionária.
      - A Computação evolucionária desenvolveu-se nos anos 60
      - Os GAs foram criados a meio da década de 70 por John Holland

Introdução

5

- Princípio Básico de Funcionamento



Funcionamento

6

- Selecção
  - As "melhores hipóteses" são as de maior "aptidão". Esta aptidão é avaliada por uma função (*fitness function*).
- Recombinação (crossover) e Mutação:
  - Em vez de procurarem sistematicamente uma solução (hipótese h), os AGs geram hipóteses sucessoras das actuais (descendência/offspring) recombinando **propabilisticamente** as melhores hipóteses entre si, e "mutando" algumas outras.

Funcionamento

7

- Protótipo

- *Fitness: Função de avaliação*
- *Fitness_threshold: Critério de Fim de Ciclo*
- *p = Número de hipóteses da população*
- *r = Fracção da população a ser recombinada*
- *m = Mutation Rate*

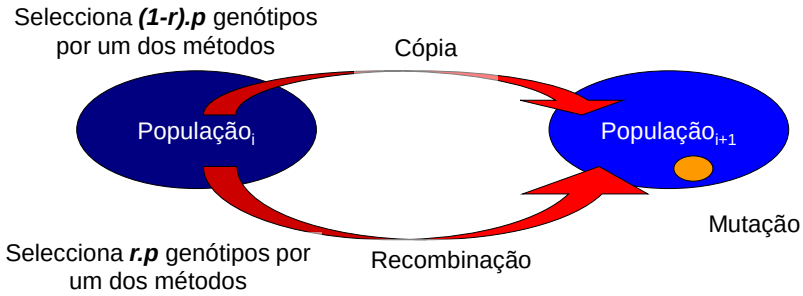
$\mathrm{GA}(Fitness, Fitness\_threshold, p, r, m)$

- *Initialize:* $P \leftarrow p$ random hypotheses
- *Evaluate:* for each h in P, compute $Fitness(h)$
- While $[\max_h Fitness(h)] < Fitness\_threshold$
  1. *Select:* Probabilistically select $(1-r)p$ members of P to add to P_S.
     $$\Pr(h_i) = \frac{Fitness(h_i)}{\sum_{j=1}^{p} Fitness(h_j)}$$
  2. *Crossover:* Probabilistically select $\frac{r \cdot p}{2}$ pairs of hypotheses from P. For each pair, $\langle h_1, h_2 \rangle$, produce two offspring by applying the Crossover operator. Add all offspring to P_s.
  3. *Mutate:* Invert a randomly selected bit in $m \cdot p$ random members of P_s
  4. *Update:* $P \leftarrow P_s$
  5. *Evaluate:* for each h in P, compute $Fitness(h)$
- Return the hypothesis from P that has the highest fitness.

Funcionamento

8

- …



Funcionamento

9

- …
  - Função de Fitness (aptidão)
    - Define o critério que avalia cada hipótese de acordo com o objectivo a atingir.
      - É a base de qualquer critério de selecção na geração seguinte.
    - Exemplos:
      - Mochila: Lucro da solução
      - N-Rainhas: A função de fitness calcula a soma dos ataques que as rainhas produzem, em cada configuração.

Funcionamento

10

- …
  - Selecção
    - Com base na função fitness, há vários métodos de selecção das hipóteses a incluir na próxima geração:
      - (1-r).p hipóteses são copiadas
      - r.p hipóteses são sujeitas a recombinação produzindo outros tantos descendentes

Selecção

11

- Selecção Proporcional
  - A probabilidade de selecção de uma hipótese é proporcional ao quociente q entre a sua aptidão e a soma das aptidões das restantes
  - As hipóteses com maior valor de q são seleccionadas mais vezes)

Selecção

12

- …
  - Exemplo: a solução de maior fitness foi seleccionada duas vezes!

Working Sheet of a GA

The problem is to optimize $f(x)=x$

String Number	Before Crossover (generation 0)			After Fitness Prop. Selection		After Crossover (generation 1)		
i	*String* X_i	*Fitness* $f(X_i)$	$\frac{f(X_i)}{\Sigma f(X_i)}$	*String* X_i	*Fitness* $f(X_i)$	*Cross-Point*	X_i	$f(X_i)$
1	011	3	0.25	011	3	2	111	7
2	001	1	0.08	110	6	2	010	2
3	110	6	0.50	110	6	-	110	6
4	010	2	0.17	010	2	-	010	2
Total		12			17			17
Worst		1			2			2
Average		3.00			4.25			4.25
Best		6			6			7

Creating generation 1 from generation 0 by application of selection and crossover

Selecção

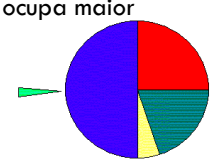
13

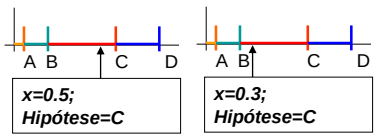
- Implementação da Selecção Proporcional
  - Método da Roleta
    - Cada hipótese de uma dada população possui uma fitness f_i
      - f1=1/6, f2=1/3, f3=1, f4=1/2
    - Calculam-se os valores acumulados:
      - A=f1=1/6; B=f1+ f2=1/2; C=f1+f2+f3=3/2; D=f1+f2+f3+f4=2
    - Normalizam-se este valores
      - A=0.0833, B=0.25, C=0.75, D=1

Selecção

14

- …
  - Gera-se um número aleatório ***x*** entre 0 e 1 e verifica-se sobre qual das hipóteses ele "cai"
    - Como qualquer ***x*** é igualmente provável, ele cairá mais vezes sobre a zona correspondente à hipótese que ocupa maior espaço na recta
    - Exemplos





Selecção

15

- …
  - Seleção por Torneio:
    - Selecionar k hipóteses (Tamanho do Torneio) de entre a população.
    - De entre elas, selecionar a de maior fitness.
      - Duas hipóteses são selecionadas aleatoriamente de entre a população.
      - Com uma probabilidade pré-definida p, a de maior aptidão é selecionada (a outra é selecionada com probabilidade (1-p)).

Selecção

16

- …
  - Seleção por Posicionamento (Ranking Selection)
    - As hipóteses são ordenadas de acordo com a sua aptidão, da melhor para a pior.
    - O **valor do ranking** (posição depois da ordenação) é usado (em vez da aptidão) por uma função que determina a probabilidade de seleção da hipótese (o espaço que ocupará na roleta)

Selecção

17

- …
  - Exemplo:
    - Escolher um número k entre 0 e 1: Seja k=0.6
    - O indivíduo de maior aptidão, ID1, ocupará 60% da área da roleta
    - D2 ocupará 60% da área restante: (1-0.6)*0.6=24%
    - ID3 ocupará 60% da área restante: (1-0.6-0.24)*0.6=0.16*0.6=9.6%
    - …
    - IDn ocupará a área restante

Recombinação

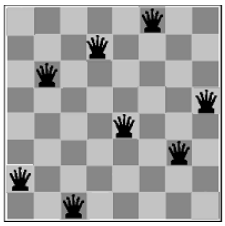
18

- Operadores de recombinação
  - As hipóteses são, muitas vezes, representadas por strings, o que permite uma implementação simples das operações de recombinação e mutação
    - Exemplos
      - Problema da mochila
        - "1" = objecto na mochila, "0" = objecto fora da mochila)
      - 8 Rainhas
        - Cada hipótese é um "estado do tabuleiro", representado por uma string do tipo "q1q2q3q4q5q6q7q8"
        - a hipótese "62714053" é uma solução "Rainha 1 = C1,L6… Rainha 8 = C8, L3"

Recombinação

19

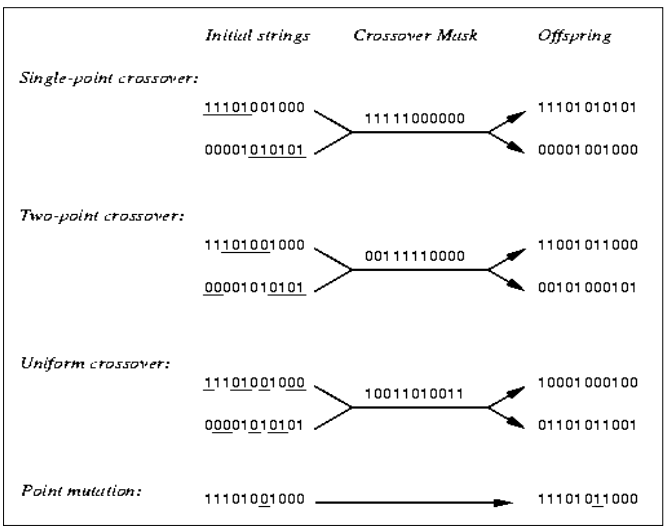
- ...
  - A string “62714053” representa o genótipo (alusão ao material genético que caracteriza cada indivíduo)
  - A configuração do tabuleiro representa o fenótipo (a tradução no mundo real daquilo que o genótipo determina)



Recombinação e Mutação

20

- ...



Recombinação

21

- …
  - Single-Point Crossover

Posição escolhida aleatoriamente

Progenitores: 1010001110 0011010010

Descendentes: 1010010010 0011001110

Recombinação

22

- …
  - Double-Point Crossover

Posições escolhidas aleatoriamente:

Progenitores: 1010001110 0011010010

Descendentes : 1010010010 0011001110

Recombinação

23

- …
  - Uniform Crossover

Máscara: 0110011000 **(Gerada aleatoriamente)**

Progenitores: 1010001110 0011010010

Descendentes: 0011001010 1010010110

Mutação

24

- Operadores de Mutação

Gera posição aleatória

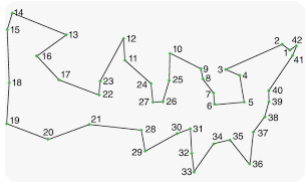
Progenitor: 1010001110

Descendente: 1010101110

Problema do Caixeiro Viajante

25

- TSP
  - Neste caso, são necessários novos operadores.
  - Para a recombiação é importante não perder informação, tais como:
    - Ordem pela qual as cidades são visitadas
    - Adjacência entre cidades
    - Posição absoluta das cidades na sequência



Problema do Caixeiro Viajante

26

- ...
  - Recombinação por ordem
    - Considerem-se os progenitores P1e P2
    - O descendente D1 é criado da seguinte forma:
      - Seleccionar dois pontos de corte C1 e C2 (C2>C1)
      - Copiar secção entre C1 e C2 de P1 para D1
      - Com início em C2, copiar as cidades de P2 para D1, omitindo as que já se encontram na sequência
    - O Descendente D2 é criado de forma análoga

Problema do Caixeiro Viajante

27

- ...

P1: 123456789

P2: 248139576

- **D1** 4567
- ... 456728
- ... 139 456728

Problema do Caixeiro Viajante

28

- ...
  - Operadores de Mutação
    - Inserção
      - I: 1 2 3 4 5 6 → I: 1 2 4 5 3 6
    - Troca
      - I: 1 2 3 4 5 6 ↔ I: 1 2 6 4 5 3
    - Inversão
      - I: 1 2 3 4 5 6 → I: 1 2 4 3 5 6