

ALGORITMOS GENÉTICOS (AG)

Optimização	optimização numérica e problemas de optimização combinatorial tal como o desenho de circuitos, planeamento, e ao problema do caixeiro viajante.
Programação automática	evolução de programas de computador para tarefas específicas, e no desenho de outras estruturas computacionais como autómatos celulares e redes de ordenação.
Aprendizagem automática	aplicações de aprendizagem automática, onde se inclui a classificação e previsão (e.g., a previsão do tempo ou a estrutura de proteínas). Melhoramento de sistemas de aprendizagem, tal como na caracterização dos pesos associados às ligações em redes neuronais, regras para os SC (a utilização estudada no contexto deste trabalho) ou sistemas de produção simbólicos, e sensores para robots.
Economia	modelação de processos de inovação, desenvolvimento de estratégias de licitação, e nascimento de mercados económicos.
Sistemas imunitários	modelação de características de sistemas imunitários naturais, incluindo a mutação somática que ocorre durante a vida de um indivíduo, e a descoberta de famílias de multi-gene durante o tempo de evolução.
Ecologia	modelação de fenómenos ecológicos tais como a corrida às armas biológicas, coevolução de parasita-portador, simbiose, e fluxo de recursos.
Populações genéticas	estudo de questões ligadas às populações genéticas: “Sob que condições um gene para recombinação será evolutivamente viável?”
Evolução e aprendizagem	estudo da correlação entre a aprendizagem individual e a evolução da sua espécie.
Processamento de imagem	alinhamento e análise de imagens.
Sistemas sociais	estudo dos aspectos evolutivos dos sistemas sociais, tais como a evolução do comportamento de colónias de insectos e, mais genericamente, na evolução da cooperação e comunicação em sistemas multi-agente.

Proc AG≡

seja $P(t)$ a população de cromossomas (soluções) no ciclo de tempo t

seja t o ciclo de tempo

seja pc a probabilidade de cruzamento

seja pm a probabilidade de mutação

$t \leftarrow 0$

inicializar $P(0)$

avaliar $P(0)$

enquanto $\neg(\text{condição de paragem})$ fazer

$t \leftarrow t+1$

$P'(t) \leftarrow \text{selecção}(P(t-1))$

$P''(t) \leftarrow \text{cruzamento}(P'(t), pc)$

$P(t) \leftarrow \text{mutação}(P''(t), pm)$

avaliar $P(t)$

CARACTERIZAÇÃO DE UM AG

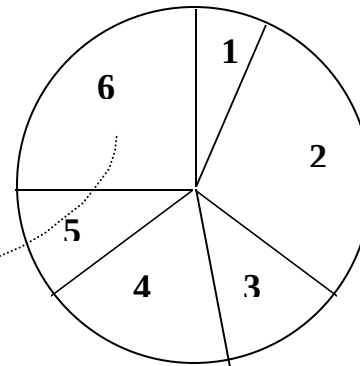
- uma representação genética das soluções;
- procedimentos para codificação/descodificação das soluções em sequências binárias;
- um método de gerar a população inicial $P(0)$;
- uma função de avaliação que classifique as soluções em termos da sua utilidade;
- operadores genéticos que alterem a composição das soluções (cromossomas);
- valores para os vários parâmetros que o algoritmo usa (e.g., cardinalidade da população, probabilidades associadas aos operadores genéticos, condição(ões) de paragem).

OPERADORES GENÉTICOS

SELECÇÃO

Método da Roleta

População POP(t)	
Indivíduo	Valor
1	5
2	30
3	10
4	20
5	10
6	25.0



func SEL^o

seja n o tamanho da população de cromossomas

seja Sum a totalidade dos valores entálpicos dos n elementos da população

seja $R \in [0, \text{Sum}]$ um número gerado aleatoriamente

seja m a ordem do elemento seleccionado (primeiro elemento da população de cromossomas cujo valor entálpico adicionado ao dos restantes elementos da população precedente, seja maior ou igual a R)

seja *random* uma função geradora de números aleatórios

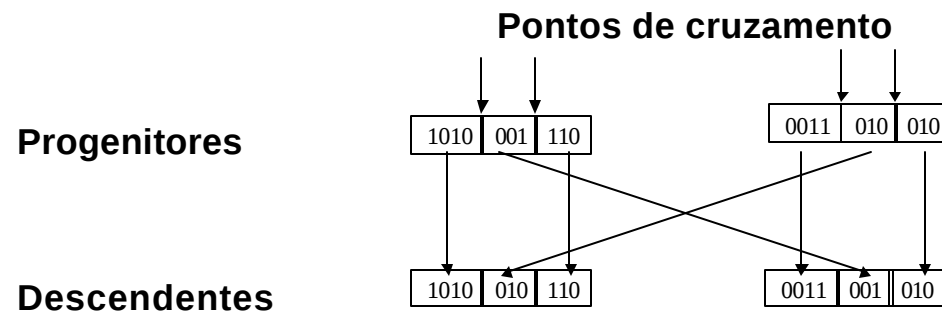
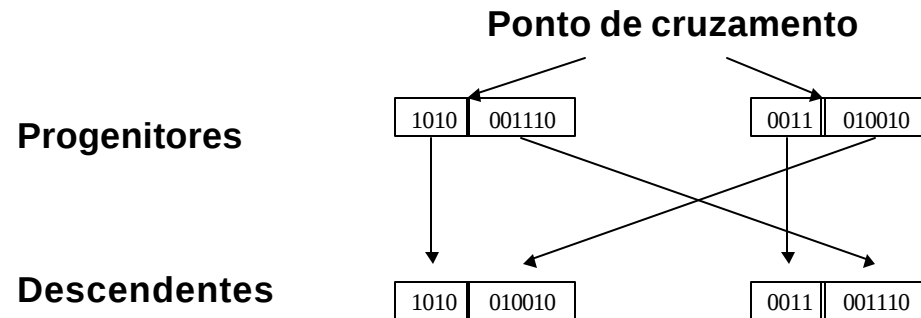
seja ST_i a entalpia de um cromossoma i

seja x um número real

seja m um número inteiro

{
Sum $\leftarrow$ 0
para $i \leftarrow 1$ **até** n **fazer**
 { Sum $\leftarrow$ Sum + ST_i
R $\leftarrow$ *random*(0, Sum)
 $m \leftarrow 1$
 $x \leftarrow 0$
enquanto $x < R$ e $m < n$ **fazer**
 {
 $x \leftarrow x + ST_m$
 $m \leftarrow m + 1$
retornar m
}

CRUZAMENTO



proc CRUZ°

seja pro1 e pro2 os cromossomas progenitores a cruzar de tamanho l

seja pro1_i o bit de ordem i do cromossoma pro1

seja pro2_i o bit de ordem i do cromossoma pro2

seja des1 e des2 os cromossomas descendentes

seja l o tamanho de um cromossoma (número de bits)

seja cruz uma medida da ocorrência ou não da operação de cruzamento

seja p_c a probabilidade de cruzamento

seja pcruz o ponto de cruzamento

seja random uma função geradora de número aleatórios $\in \mathbf{N}^+$

```

{
  cruz ← falso
  se pc = 1.0 então
    { cruz ← verdadeiro
    senão
      { cruz ← (random ≤ pc)
    }
  se cruz então
    {
      pcruz ← random(1,l)
      des1 ← pro11,..., pcruz, pro2pcruz+1,..., l
      des2 ← pro21,..., pcruz, pro1pcruz+1,..., l
    }
}
```

MUTAÇÃO

func *MUT* °

seja bit o bit a mutar

seja mut uma medida da ocorrência ou não da operação de mutação

seja p_m a probabilidade de mutação

seja random uma função geradora de números aleatórios $\in [0.0, 1.0[$

$\left\{ \begin{array}{l} \text{mut} \leftarrow \text{falso} \\ \textbf{se } p_m = 1.0 \textbf{ então} \\ \quad \{ \text{mut} \leftarrow \text{verdadeiro} \\ \quad \textbf{senão} \\ \quad \quad \{ \text{mut} \leftarrow (\text{random} \leq p_m) \\ \textbf{se mut} \textbf{ então} \\ \quad \quad \{ \text{bit} \leftarrow \text{não bit} \\ \textbf{retornar bit} \end{array} \right.$

EXEMPLO

Optimizar a função $f(x)=x$, com $x \in [0, 31]$

Considerar:

tamanho da população $n=4$

$P(0) \equiv \{00110, 10011, 11000, 00000\}$

$pc=1$

$pm=0.2$

Números Aleatórios

N		N		N		N		N		N		N		N		N	
1	0,8	11	0,6	21	0,8	31	0,8	41	0,5	51	0,6	61	0,8	71	0,1	81	0,5
2	0,4	12	0,9	22	0,7	32	0,6	42	0,1	52	0,4	62	0,7	72	0,9	82	0,8
3	0,3	13	0,8	23	0,1	33	0,3	43	0,6	53	0,5	63	0,4	73	0,3	83	0,3
4	0,8	14	0	24	0,6	34	0,6	44	0,7	54	0,4	64	0,9	74	0,1	84	0,5
5	0,5	15	0,6	25	0,9	35	0,7	45	0,5	55	0,7	65	0,9	75	0,3	85	0,4
6	0,3	16	0,3	26	0,5	36	0,7	46	0,9	56	0,4	66	0,2	76	0,4	86	0,3
7	0,7	17	0,5	27	0,1	37	0,6	47	0,7	57	0,9	67	0	77	0,2	87	0,6
8	0,5	18	0,1	28	0,6	38	0,2	48	0,1	58	0,4	68	0	78	0,6	88	0,7
9	0,1	19	0,4	29	0,2	39	0,6	49	0,5	59	0,6	69	0,1	79	0,1	89	0,6
10	0,4	20	0,1	30	0,1	40	0,6	50	0,2	60	0,8	70	0,4	80	0,8	90	0
																100	0