

Algoritmos Genéticos

1 Introdução

- Os algoritmos genéticos (AGs) foram desenvolvidos por J. H. HOLLAND (1975; 1992), da Universidade de Michigan.
- **Metas:**
 - Abstrair e explicar os processos adaptativos em sistemas naturais.
 - Desenvolver simulações em computador que retenham os mecanismos originais encontrados em sistemas naturais.
- **Características principais:**
 - Existência de **dois** espaços de trabalho: espaço **genotípico** e espaço **fenotípico**.
 - AGs realizam a busca com base em uma **população** de soluções candidatas (pontos no espaço de busca) e não sobre um único ponto.

- AGs fazem uso de descrições genéricas do que se quer ver presente na solução através de **funções de *fitness*** (funções-objetivo).
- AGs utilizam regras de transição **probabilísticas**, e não regras determinísticas.

2 Algoritmo Genético Clássico

- i. População de tamanho fixo (N indivíduos).
- ii. Estrutura de dados baseada em cadeias binárias.
- iii. Estratégia de seleção proporcional ao *fitness* via algoritmo da roleta.
- iv. *Crossover* simples (*crossover* de um ponto).
- v. Mutação pontual.

2.1 Representação (Estrutura de Dados)

- Cadeias binárias de comprimento fixo.

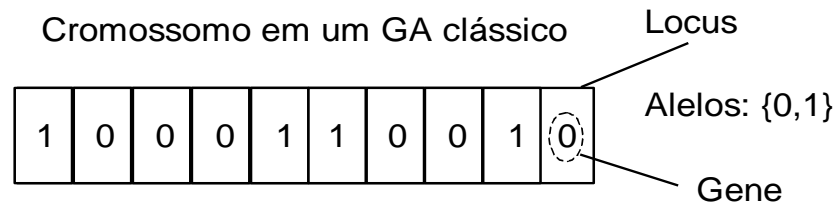


Figura 1: Cadeia binária de comprimento $l = 10$ correspondente à estrutura de dados de um GA clássico.

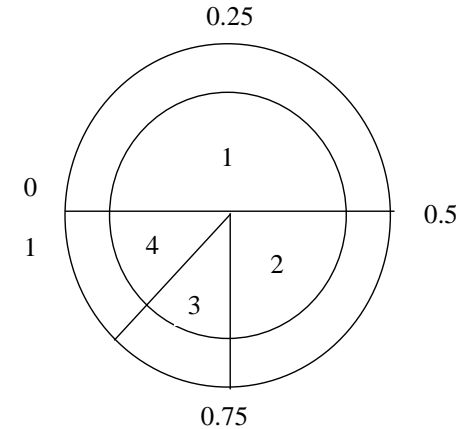
- O problema a ser resolvido é capturado em uma função objetivo que indica o *fitness* (adaptabilidade) de cada candidato a solução.

2.2 Mecanismo de Seleção

- A seleção no GA clássico é proporcional ao *fitness* e é geralmente implementada utilizando um algoritmo da roleta (*Roulette Wheel*).

- **Exemplo:**

N	Cromossomo	Fitness	Graus
1	0001100101010	6.0	180
2	0101001010101	3.0	90
3	1011110100101	1.5	45
4	1010010101001	1.5	45



- **Implementação típica:** a partir de um gerador de números pseudoaleatórios com distribuição uniforme.
- Note que esse procedimento permite a perda (“morte”) do melhor indivíduo e também permite que um indivíduo seja selecionado mais que uma vez.
- As probabilidades de reprodução de cada indivíduo vão resultar na geração de uma nova população composta por indivíduos probabilisticamente selecionados a partir da população atual.
- Os indivíduos selecionados vão gerar probabilisticamente filhos (descendentes) através de operadores genéticos específicos, particularmente *crossover* e *mutação*.

2.3 Operador de Recombinação (Crossover Simples)

- Nos sistemas biológicos, o *crossover* pode ocorrer durante a meiose (no processo de geração dos gametas). A ideia de recombinação em computação evolutiva também pode ser entendida em termos de um processo reprodutivo (com a geração de filhos a partir da combinação do material genético de pais).
- Esse processo pode ser abstraído, levando a um operador geral para as estruturas de dados do tipo cadeia binária utilizadas no GA clássico (HOLLAND, 1975):
 - Duas cadeias $\mathbf{x} = x_1x_2\dots x_l$ e $\mathbf{y} = y_1y_2\dots y_l$ de comprimento l são selecionadas com probabilidade de *crossover* pc .
 - Um número $r \in \{1, 2, \dots, l-1\}$ indicando o ponto de cruzamento (*crossover*) é selecionado.
 - Duas novas cadeias são formadas a partir de \mathbf{x} e \mathbf{y} através da troca de um conjunto de atributos à direita da posição r , resultando em $\mathbf{x}' = x_1\dots x_r y_{r+1}\dots y_l$ e $\mathbf{y}' = y_1\dots y_r x_{r+1}\dots x_l$.

- Os dois novos cromossomos gerados x' e y' são os filhos (*offspring*) de x e y .

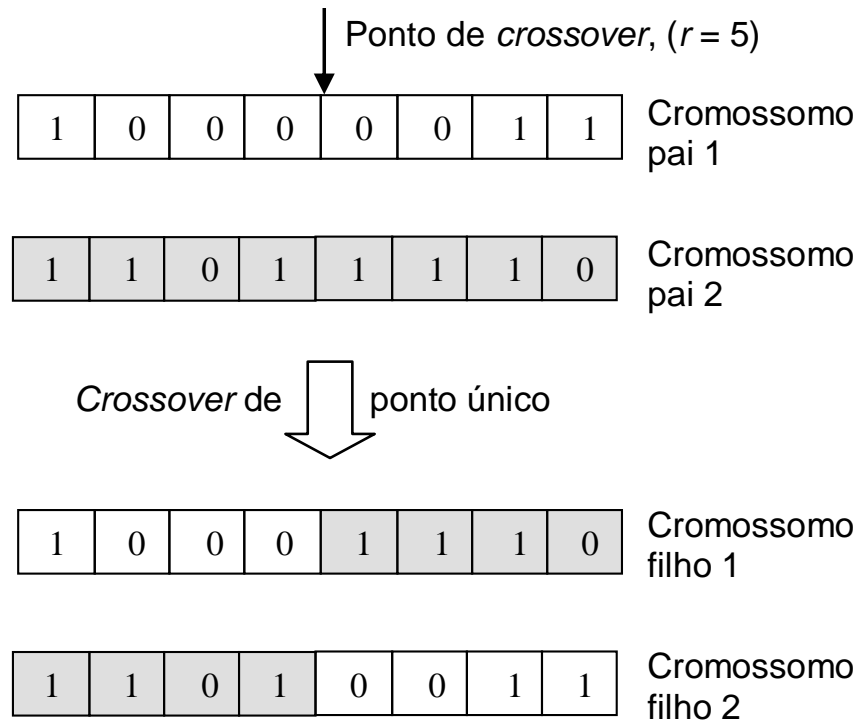
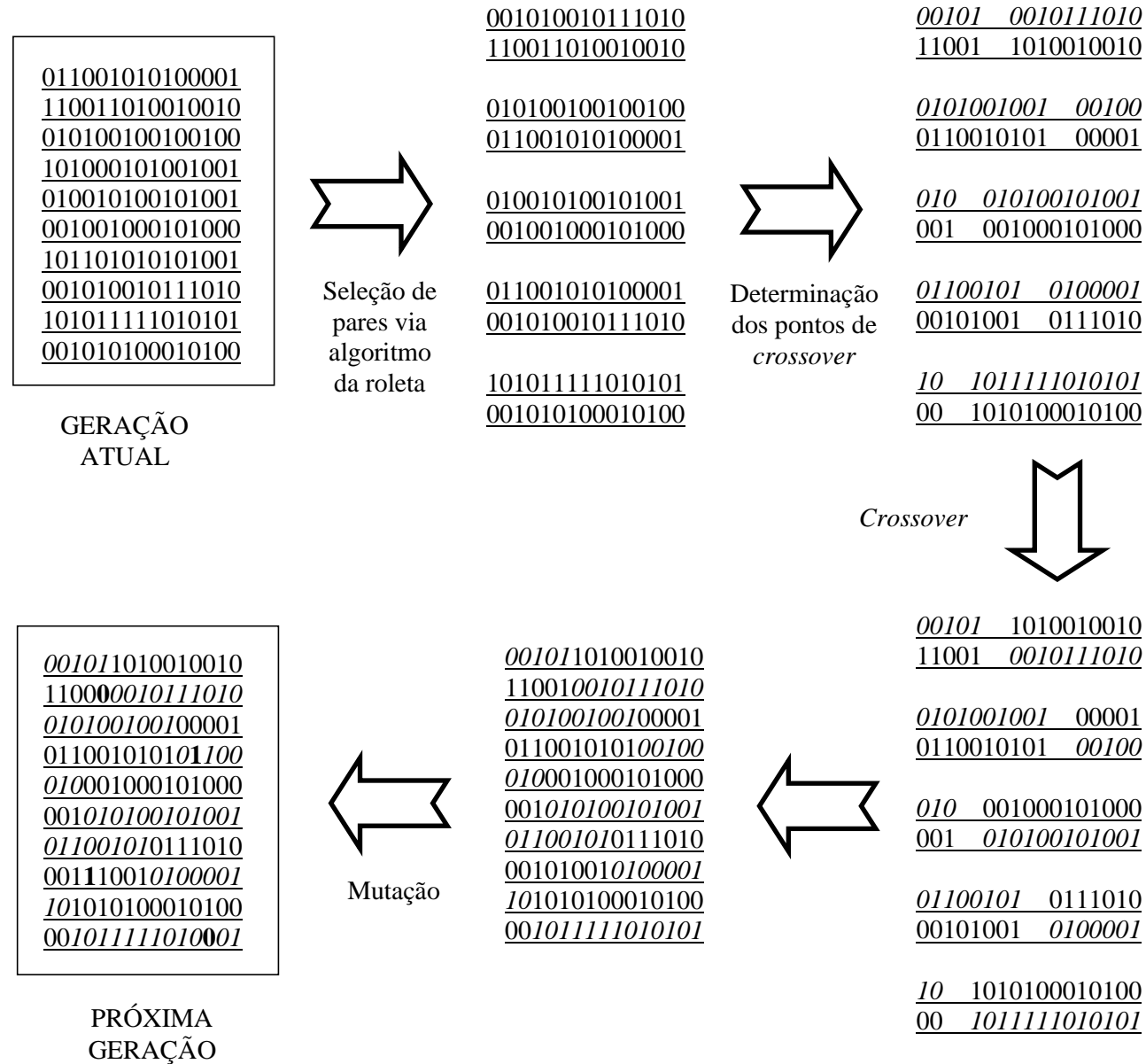


Figura 2: *Crossover* de um único ponto para cadeias de comprimento $l = 8$.

2.4 Operador de Mutação

- Em genética, a mutação pontual é um processo no qual um alelo de um gene é estocasticamente substituído (ou modificado) por outro, o que leva a um novo cromossomo.
- Geralmente existe uma baixa probabilidade de “mutar” cada gene de um cromossomo.
- Isso significa que cada bit na população P é “tratado” da seguinte forma:
 - Os números r, \dots, u que indicam as posições que vão sofrer mutação são determinados aleatoriamente, de forma que cada posição possui uma pequena probabilidade pm de sofrer mutação independentemente das outras posições.
 - Uma nova cadeia $\mathbf{x}' = x_1 \dots x_r \dots x_u \dots x_l$ é gerada, sendo $x_r \dots x_u$ determinados aleatoriamente partindo do conjunto de alelos para cada gene. No caso de cadeias binárias, se uma posição possui alelo ‘0’, então ela se torna ‘1’ e vice-versa.



- **Problemas com o algoritmo clássico:**

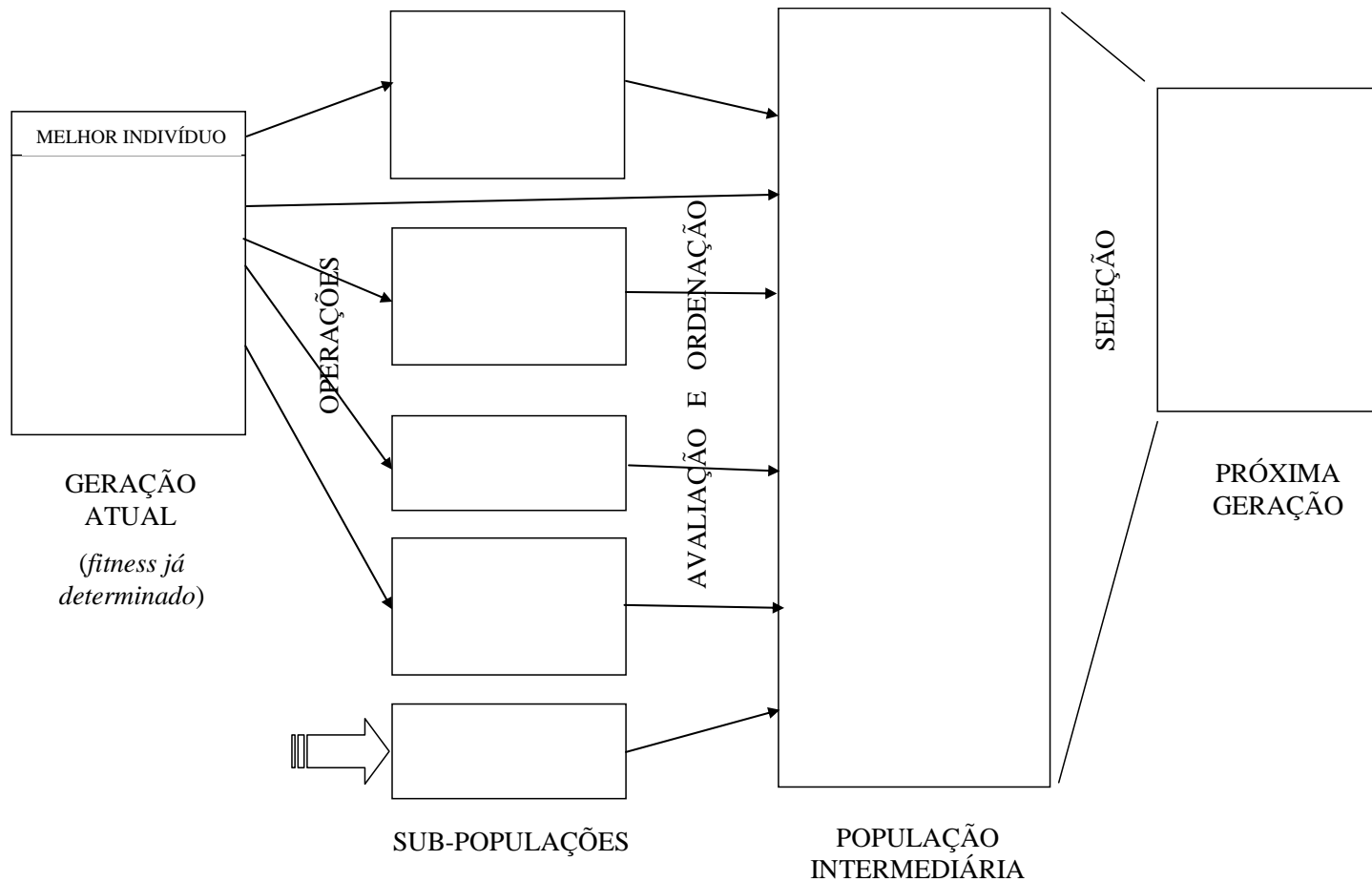
- Estratégia de reprodução/seleção permite a perda do melhor indivíduo.
- A posição dos genes no cromossomo influencia a probabilidade de que permaneçam no cromossomo gerado via *crossover*.
- Dificuldades relacionadas à codificação binária no contexto de otimização de parâmetro reais.

- **Algumas estratégias de solução:**

- Uso de mecanismos alternativos de seleção (e.g., torneio).
- *Crossover* uniforme.
- Codificação em arranjos de números reais.

3 Algoritmo Genético Modificado

- Geração de subpopulações por meio da aplicação de operadores genéticos e outros operadores sobre membros da geração atual.



- Avaliação (cálculo do *fitness*) e ordenação da população intermediária.
- Seleção para nova geração.

4 Exemplos

Dada a função de uma única variável $g(x) = 2^{-2((x-0.1)/0.9)^2} \text{sen}(5\pi x)^6$, apresentada na Figura 3, tente determinar seu ótimo global utilizando o algoritmo genético padrão. A variável x está definida no intervalo $[0,1]$, $x \in [0,1]$. Escolha como representação para a variável x uma cadeia binária m ($m = \langle m_l, \dots, m_2, m_1 \rangle$) de comprimento $l = 22$, correspondente a uma precisão de seis casas decimais.

- O mapeamento de uma cadeia binária para um número real pode ser feito em dois passos:
 - Converta a cadeia binária $m = \langle m_l, \dots, m_2, m_1 \rangle$ da base 2 para a base 10:

$$(\langle m_l, \dots, m_2, m_1 \rangle)_2 = \left(\sum_{i=0}^{l-1} m_i \cdot 2^i \right)_{10} = x'$$

- Encontre o correspondente valor real (\Re) para x : $x = x_{\min} + x' \cdot \frac{x_{\max} - x_{\min}}{2^l - 1}$, onde $x_{\min} = 0$ e $x_{\max} = 1$, $x \in [x_{\min}, x_{\max}]$ (domínio da variável).

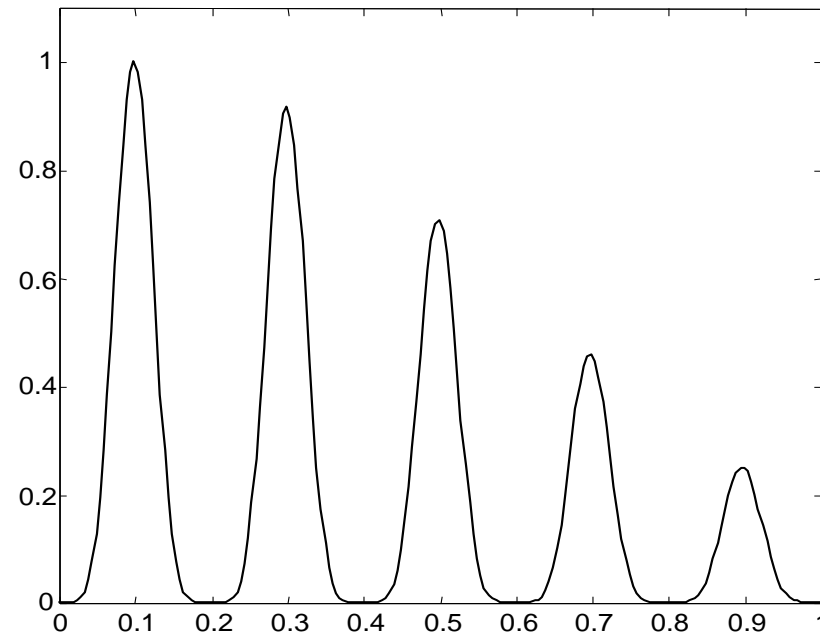


Figura 3: Função $g(x) = 2^{-2((x-0.1)/0.9)^2} \sin(5\pi x)^6$ a ser maximizada.

Função de Avaliação: $\phi(x) = g(x)$

Operadores Genéticos: *Crossover* simples, mutação pontual

O problema do **caixeiro viajante** (*travelling salesman problem* – TSP) é de natureza combinatória e aparece em diversas aplicações, do projeto de circuitos integrados à entrega de *fast food*. Em sua forma mais simples, o caixeiro deve visitar cada cidade em um dado território somente uma vez, e depois retornar a cidade de origem. Dado o custo da viagem entre cada uma das cidades, qual o itinerário que resulta no custo mínimo total?

- Exemplo:

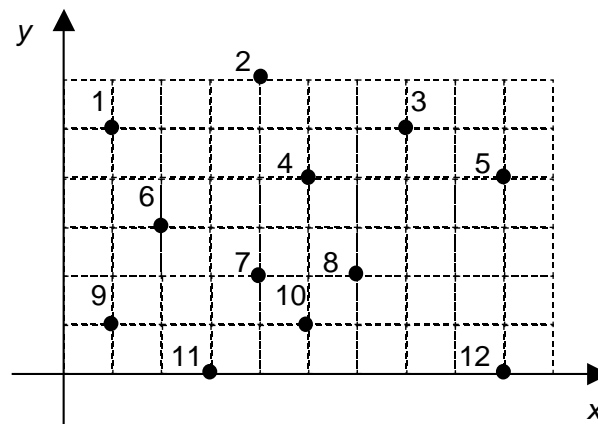


Figura 4: TSP simples com 12 cidades.

Codificação: arranjo de números inteiros: (1,3,4,5,6,8,9,12,11,10,7,2)

Função de *fitness*: inverso da soma dos custos de movimentação entre cada par de cidades.

Operadores

- Crossover OX:

1	2	3	4	5	6	7	8	9	10	11	12
7	3	1	11	4	12	5	2	10	9	6	8
			4	5	6	7					
1	11	12	4	5	6	7	2	10	9	8	3

- Mutação Inversiva

1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	----	----	----

1	2	3	8	5	6	7	4	9	10	11	12
---	---	---	---	---	---	---	---	---	----	----	----

- Variação de Mutação Inversiva

1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	----	----	----

1	2	3	5	4	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	----	----	----

1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	----	----	----

12	2	3	4	5	6	7	8	9	10	11	1
----	---	---	---	---	---	---	---	---	----	----	---

5 Teoria dos Esquemas (*schemata theory*)

- A teoria dos esquemas foi proposta por HOLLAND (1975; 1992) para tentar explicar por que os algoritmos genéticos funcionam.
- Nesta seção, apresentaremos os principais resultados da teoria dos esquemas: o teorema de crescimento dos esquemas e a hipótese dos blocos construtivos (*building blocks*).
- Um *esquema* é uma representação capaz de descrever diversos cromossomos simultaneamente. Ele é construído inserindo um caractere *don't care* (*) no alfabeto dos genes, indicando que aquele gene pode ser preenchido por qualquer alelo.
 - Por exemplo, o esquema $[1 * 0 1 0 0 1]$ representa os cromossomos $[1 \text{ 0 } 0 1 0 0 1]$ e $[1 \text{ 1 } 0 1 0 0 1]$. O esquema $[1 * 0 * 1 1 0]$ representa quatro cromossomos: $[1 \text{ 0 } 0 \text{ 0 } 1 1 0]$, $[1 \text{ 0 } 0 \text{ 1 } 1 1 0]$, $[1 \text{ 1 } 0 \text{ 0 } 1 1 0]$ e $[1 \text{ 1 } 0 \text{ 1 } 1 1 0]$.

- O esquema $[1\ 1\ 1\ 0\ 0\ 1\ 0]$ representa apenas um cromossomo, enquanto o esquema $[*\ * \ * \ * \ * \ * \ *]$ representa todos os cromossomos de comprimento 7.
 - Observe que cada esquema representa 2^r cromossomos, onde r é o número de caracteres *don't care* “*” presentes. Por outro lado, cada cromossomo de comprimento m é representado por 2^m esquemas.
- Por exemplo, considere o cromossomo $[0\ 1\ 0\ 0\ 1\ 0\ 0]$. Este cromossomo é representado pelos seguintes 2^7 esquemas:

$[0\ 1\ 0\ 0\ 1\ 0\ 0]$
 $[*\ 1\ 0\ 0\ 1\ 0\ 0]$
 $[0\ *\ 0\ 0\ 1\ 0\ 0]$
 \vdots
 $[0\ 1\ 0\ 0\ 1\ 0\ *]$
 $[*\ * \ 0\ 0\ 1\ 0\ 0]$
 $[*\ 1\ *\ 0\ 1\ 0\ 0]$
 \vdots
 $[0\ 1\ 0\ 0\ 1\ *\ *]$
 $[*\ *\ * \ 0\ 1\ 0\ 0]$

$$\begin{array}{c} \vdots \\ [* * * * *] \end{array}$$

- Considerando cromossomos de comprimento m , há um total de 3^m possíveis esquemas. Numa população de tamanho n , entre 2^m e $n \cdot 2^m$ diferentes esquemas podem ser representados.
- A **ordem** de um esquema S , $o(S)$, é definida como o número de 0's e 1's presentes no esquema, isto é, o número de *posições fixas* (caracteres diferentes de *don't care*) presentes no esquema. A ordem de um esquema define sua especificidade, de modo que, quanto maior a ordem, mais específico é o esquema.
- O **comprimento definidor** de um esquema S , denotado por $\delta(S)$, é a maior distância entre posições fixas de um cromossomo. O comprimento definidor estabelece o nível de compactação da informação contida no esquema.
- O *fitness* de um esquema S na geração t , $\text{eval}(S, t)$, é definido como a média dos *fitness* de todos os cromossomos na população representados pelo esquema S .

Assuma que há p cromossomos $\{\mathbf{x}_{i_1}^t, \dots, \mathbf{x}_{i_p}^t\}$ representados pelo esquema S_i na geração t . Então:

$$\text{eval}(S_i, t) = \frac{1}{p} \sum_{j=1}^p \text{eval}(\mathbf{x}_{i_j}^t),$$

onde $\text{eval}(\mathbf{x}_{i_j}^t)$ é o *fitness* do indivíduo $\mathbf{x}_{i_j}^t$.

- Seja tam_pop o tamanho da população. O *fitness* médio da população na geração t , $\bar{F}(t)$, é dado por

$$\bar{F}(t) = \frac{1}{\text{tam_pop}} \sum_{i=1}^{\text{tam_pop}} \text{eval}(\mathbf{x}_i^t).$$

- Sejam p_c e p_m as probabilidades de *crossover* e mutação, respectivamente, e m o comprimento dos cromossomos. Seja $\xi(S_i, t)$ o número de cromossomos representados pelo esquema S_i na geração t . Pode-se mostrar que (MICHALEWICZ, 1996):

$$\xi(S_i, t+1) \geq \frac{\xi(S_i, t) \text{eval}(S_i, t)}{\bar{F}(t)} \left[1 - p_c \frac{\delta(S_i)}{m-1} - o(S_i) p_m \right]$$

- A equação acima é conhecida como *equação de crescimento reprodutivo do esquema*. Esta equação é deduzida supondo que a função de *fitness* $f(\cdot)$ produz apenas valores positivos. Se a função a ser otimizada produz valores negativos, um mapeamento entre as funções de otimização (objetivo) e de *fitness* é necessário.
- Esta equação de crescimento mostra que a seleção aumenta a amostragem de esquemas cujo *fitness* está acima da média da população, e este aumento é exponencial (MICHALEWICZ, 1996).
- A seleção, por si só, não introduz nenhum esquema novo (não representado na geração inicial em $t = 0$). Esta é a razão da introdução do operador de *crossover*: possibilitar a troca de informação estruturada, ainda que aleatória. Além disso, o operador de mutação introduz uma variabilidade maior na população.

- O efeito (destrutivo) combinado destes operadores não é significativo se o esquema for curto e de ordem baixa. O resultado final da equação de crescimento pode ser formulado como segue:

Teorema dos Esquemas: Esquemas com comprimento definidor curto, de ordem baixa, e com *fitness* acima da média, têm uma tendência estatística de um aumento exponencial de sua participação em gerações consecutivas de um algoritmo genético.

Prova: Veja HOLLAND (1975; 1992).

- Uma consequência imediata deste teorema é que os algoritmos genéticos tendem a explorar o espaço por meio de esquemas curtos e de baixa ordem que, subsequentemente, são usados para troca de informação durante o *crossover*.

Hipótese dos Blocos Construtivos: Um algoritmo genético busca desempenho “quase-ótimo” através da justaposição de esquemas curtos, de baixa ordem e alto desempenho, chamados de *blocos construtivos*.

- HOLLAND (1975; 1992) mostrou que, em uma população de tamanho tam_pop , pelo menos tam_pop^3 esquemas são processados de forma útil. Esta propriedade foi denominada *paralelismo implícito*, pois é obtida sem nenhuma exigência extra de memória e processamento. Entretanto, BERTONI & DORIGO (1993) mostraram que a estimativa tam_pop^3 é válida apenas para o caso particular em que tam_pop é proporcional a 2^l , onde $l = \frac{1}{2}m\varepsilon$ e ε é a probabilidade de um esquema ser rompido por *crossover*.
- Note, entretanto que, em alguns problemas, alguns blocos construtivos (esquemas curtos, de ordem baixa) podem direcionar erroneamente o algoritmo, levando-o a convergir a pontos sub-ótimos. Este fenômeno é conhecido como **decepção**. Assim, a hipótese dos blocos construtivos não fornece uma explicação definitiva do porquê os algoritmos genéticos funcionam. Ela é apenas uma indicação do porquê os algoritmos genéticos funcionam para uma certa classe de problemas.

5.1 Deception Problem

- Alguns blocos construtivos podem direcionar erroneamente o GA, levando a uma convergência a pontos sub-ótimos:
 - $\langle 1\ 1\ 1\ *\ *\ *\ *\ *\ *\ *\ *\rangle$ – *fitness* acima da média
 - $\langle *\ *\ *\ *\ *\ *\ *\ *\ *\ 1\ 1\rangle$ – *fitness* acima da média
 - $\langle 1\ 1\ 1\ *\ *\ *\ *\ *\ *\ 1\ 1\rangle$ – *fitness* muito menor que $\langle 0\ 0\ 0\ *\ *\ *\ *\ *\ *\ 0\ 0\rangle$
 - Solução ótima – $\langle 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\rangle$
 - Tendência a convergir para pontos como $\langle 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0\rangle$
- Algumas alternativas foram propostas para combater o problema da decepção (GOLDBERG, 1989; MICHALEWICZ, 1996). A primeira assume que há algum conhecimento a priori da função-objetivo para que seja possível codificá-la de forma apropriada (que forme blocos construtivos “coesos”).

- A segunda é a utilização de um operador de *inversão*: selecionam-se dois pontos em um cromossomo e inverte-se a ordem dos bits entre os pontos selecionados (alterando-se a codificação em conjunto com a operação).
- A terceira opção é utilizar algoritmos genéticos *messy*, que diferem do algoritmo genético clássico de várias maneiras: codificação, operadores, presença de cromossomos de tamanho distinto e fases evolutivas.
- A quarta opção é o emprego de algoritmos de estimação de distribuição, que serão vistos em um dos tópicos seguintes deste curso.

6 Referências

- GOLDBERG, D. E. “Messy Genetic Algorithms: Motivation, Analysis, and First Results”, *Complex Systems*, 3: 493-530, 1989.
- HOLLAND, J.H. “Adaptation in Natural and Artificial Systems”, University of Michigan Press, 1975.
- HOLLAND, J.H. “Adaptation in Natural and Artificial Systems”, 2nd edition, The MIT Press, 1992.
- MICHALEWICZ, Z. “Genetic Algorithms + Data Structures = Evolution Programs”, 3rd edition, Springer, 1996.