

Otimização contínua $\rightarrow \mathbb{R}$
VS
Otimização discreta $\rightarrow \mathbb{N}$

▪ Procura local vs procura global

- Algumas meta-heurísticas aplicam métodos de procura local, onde as novas soluções exploradas são “vizinhas” de soluções anteriores (e.g. *Simulated Annealing*, *Tabu Search*);
- Outras meta-heurísticas distribuem o processo de procura por todo o espaço de procura (normalmente através de abordagens baseadas em populações).

▪ Solução única vs *Population-based*

- As abordagens de solução única, são iterativas, e orientam o processo de procura através da melhoria da solução anterior;
- As abordagens baseadas em populações utilizam uma procura em paralelo por parte de vários membros da população, podendo, ou não, existir a troca de informação entre os indivíduos (e.g. *Particle Swarm optimization*, *Genetic Algorithms*, *Ant Colony optimization*).

▪ Algoritmos Genéticos

- Definição do estado como um cromossomo
- Gerar soluções (cromossomos) a partir de uma população de estado inicial
- Reprodução, Mutação e Seleção

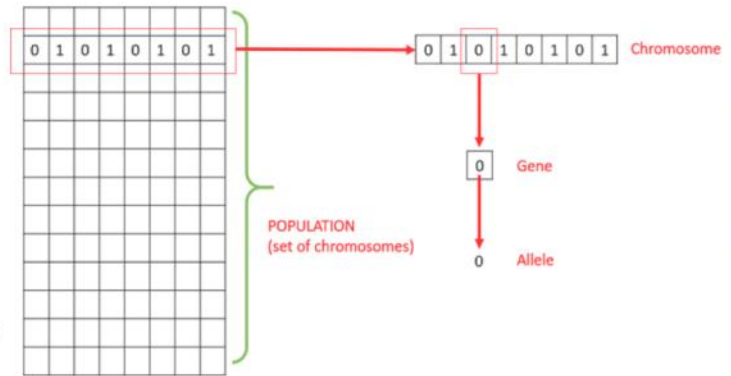
▪ Colónia de Formigas (*Ant Colony*)

- Iniciando vários estados (colónia de formigas)
- A probabilidade de um caminho ser melhor é determinada pelo número de “formigas” que passam por ele

▪ Enxame de Partículas (*Particle Swarm*)

- Vários estados de partida (enxame)
- A vizinhança é explorada e mantida, a melhor solução e o melhor estado
- “Partículas” caminham na direção da melhor solução encontrada até agora
- A velocidade do movimento depende das distâncias para a melhor solução e o melhor estado e a posição do estado

- **Indivíduo ou Cromossoma**
 - Uma solução específica, constituída por um ou mais elementos
- **Gene**
 - Uma posição no cromossoma (um elemento)
- **Alelo**
 - O valor de um gene
- **População**
 - Conjunto de indivíduos
- **Geração**
 - Conjunto de todos os novos indivíduos "nascidos" a partir de uma mesma população pai
 - Mede a passagem do tempo
- **Espaço de procura**
 - Os limites dos valores dos Genes



18

- O objetivo do cruzamento é o de conseguir, nos descendentes, uma combinação do material genético dos progenitores;
- O operador Cruzamento é aplicado a dois indivíduos da população, produzindo outros dois indivíduos para a população da geração seguinte;
- Existem diferentes formas de implementar o operador:



Estrutura de um algoritmo genético básico

1. **[Início]** Gerar uma população aleatória de n cromossomas (apropriados para a solução do problema)
2. **[Fitness]** Avaliar cada cromossoma x na população através da função de fitness $f(x)$
3. **[Nova População]** Criar uma nova população através da repetição dos seguintes passos até a população estar completa
 - **[Seleção]** Escolher 2 cromossomas da população de acordo com as suas avaliações de fitness (quanto melhor o fitness melhor a sua probabilidade de ser selecionado)
 - **[Recombinação ou Crossover]** Tendo em conta a probabilidade de crossover, combinando 2 cromossomas formando assim um novo cromossoma.
 - **[Mutação]** Tendo em conta a probabilidade de mutação, mutar cromossomas na população).
 - **[Aceitação]** Por o novo cromossoma na nova população
4. **[Substituir]** Use os novos cromossomas criados na população de cromossomas para correr de novo o programa substituindo cromossomas da população anterior
5. **[Testar]** Se a condição de paragem é satisfeita, parar e devolver a melhor solução encontrada na população corrente.
6. **[Loop]** Ir para o passo 2

Crossover:

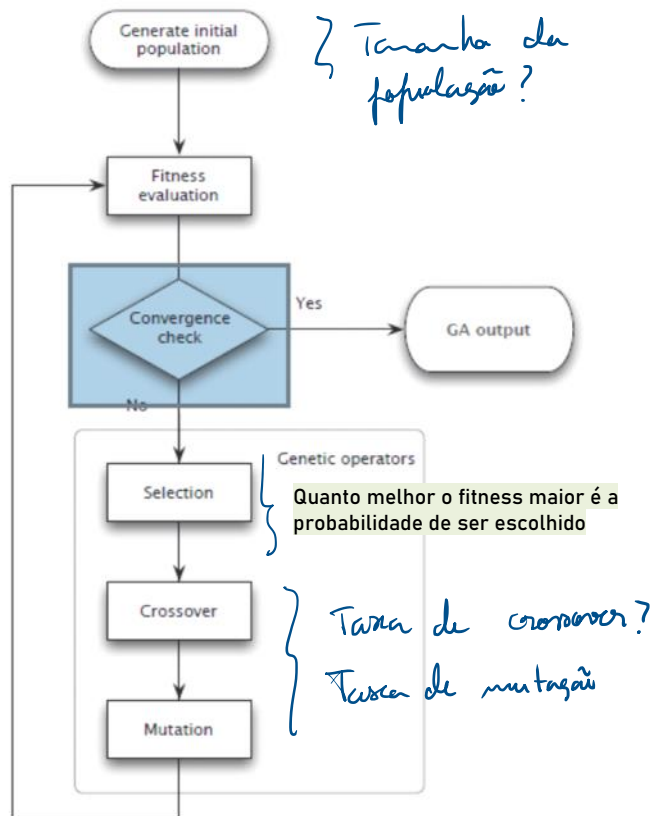
- * Também conhecido como recombinação, o crossover envolve a troca de informações genéticas entre dois indivíduos (pais) para criar novos indivíduos (descendentes).

Crossover:

- Também conhecido como recombinação, o crossover envolve a troca de informações genéticas entre dois indivíduos (pais) para criar novos indivíduos (descendentes).
- Geralmente, um ponto de crossover é escolhido aleatoriamente nos cromossomos dos pais, e os segmentos antes ou depois desse ponto são trocados entre os pais para gerar os descendentes.
- O objetivo do crossover é explorar combinações favoráveis de características presentes nos pais.

Mutações:

- A mutação é um operador genético que introduz mudanças aleatórias em um indivíduo.
- Durante a mutação, um ou mais genes no cromossomo de um indivíduo são alterados de forma aleatória.
- A mutação ajuda a manter a diversidade genética na população e evita que o algoritmo genético fique preso em ótimos locais locais.
- Embora a taxa de mutação seja geralmente baixa, ela desempenha um papel crucial na introdução de variação genética.



MODELOS DE POPULAÇÃO

○ Steady State:

- Em cada iteração são gerados N novos indivíduos
- Estes substituem N indivíduos da geração anterior
- Também conhecidos como Algoritmos Genéticos incrementais
- N deve ser inferior ao tamanho da população

=> mantém-se alguns de
gerações anteriores

○ Generational:

- São gerados N novos indivíduos
- N é igual ao tamanho da população
- Toda a população é substituída pela nova geração

■ Tamanho da População (N):

- uma **população pequena** pode provocar um mau desempenho, por não cobrir adequadamente o espaço do problema, gerando soluções locais;
- uma **população grande** pode evitar os problemas anteriores, mas pode afetar a eficiência computacional do sistema.

■ Taxa de Cruzamento (Cr):

- quantidade de cromossomos utilizados para cruzamento: $N \times Cr$.

■ Taxa de Mutação (Mr):

- a mutação é utilizada para aumentar a variabilidade da população;
- cada gene tem uma probabilidade finita de mudar;
- uma **baixa taxa** de mutação permite que um gene “gele” num valor;
- uma **alta taxa** de mutação resulta numa procura aleatória de soluções;
- sendo L o comprimento do cromossoma, ocorrerão $Mr \times N \times L$ mutações.

Particle Swarm Optimization

$$\left. \begin{aligned} \vec{X}_1^d &= [x_1^d, y_1^d] \\ \vec{X}_2^d &= [x_2^d, y_2^d] \\ \vec{X}_3^d &= [x_3^d, y_3^d] \end{aligned} \right\} \vec{X}_i^d = [x_i^d, y_i^d, z_i^d, \dots]$$

PSO search strategy

$$\overrightarrow{V_i^{d+1}} = 2r_1 \overrightarrow{V_i^d} + 2r_2 \left(\overrightarrow{P_i^d} - \overrightarrow{X_i^d} \right) + 2r_3 \left(\overrightarrow{G^d} - \overrightarrow{X_i^d} \right)$$

Next velocity (tomorrow)

Current velocity (today)

Personal best solution

Global best solution

Distance to the personal best

Distance to the global best

$$\overrightarrow{X_i^{d+1}} = \overrightarrow{X_i^d} + \overrightarrow{V_i^{d+1}}$$

Position in day $d+1$

Position in day d

Velocity in day $d+1$



Mathematical model of PSO

$$\overrightarrow{X_i^{t+1}} = \overrightarrow{X_i^t} + \overrightarrow{V_i^{t+1}}$$

$$\overrightarrow{V_i^{t+1}} = w \overrightarrow{V_i^t} + c_1 r_1 \left(\overrightarrow{P_i^t} - \overrightarrow{X_i^t} \right) + c_2 r_2 \left(\overrightarrow{G^t} - \overrightarrow{X_i^t} \right)$$

Inertia

maintains the current direction

Cognitive component

individual

↳ Exploration

Social component

↳ Exploitation

Initialize the controlling parameters (N , $c1$, $c2$, $Wmin$, $Wmax$, $Vmax$, and $MaxIter$)

Initialize the population of N particles

```
do
  for each particle
    calculate the objective of the particle
    Update PBEST if required
    Update GBEST if required
  end for

  Update the inertia weight
  for each particle
    Update the velocity (V)
    Update the position (X)
  end for
```

while the end condition is not satisfied

Return GBEST as the best estimation of the global optimum