

The background features a light blue and white color palette. It contains several semi-transparent DNA double helix structures and a network of interconnected nodes and lines, suggesting a genetic or computational theme.

ALGORITMOS GENÉTICOS

Profa Mariana Kleina

INTRODUÇÃO

Algoritmo Genético (AG) é um método de Computação Evolucionária inspirado na Teoria da Evolução das Espécies de Charles Darwin, usando as seguintes hipóteses:

- Na natureza sobrevivem os indivíduos com maior capacidade de adaptação;
- Suas características genéticas são repassadas para as gerações seguintes e melhoradas;
- A nova geração será composta por indivíduos com material genético melhor do que os da população anterior.

Aplicações:

- Otimização (máquinas × processos, alocação de recursos, roteamento de veículos);
- Busca (mineração de dados, indução de classificadores).

INTRODUÇÃO

Requisitos para implementação de AGs:

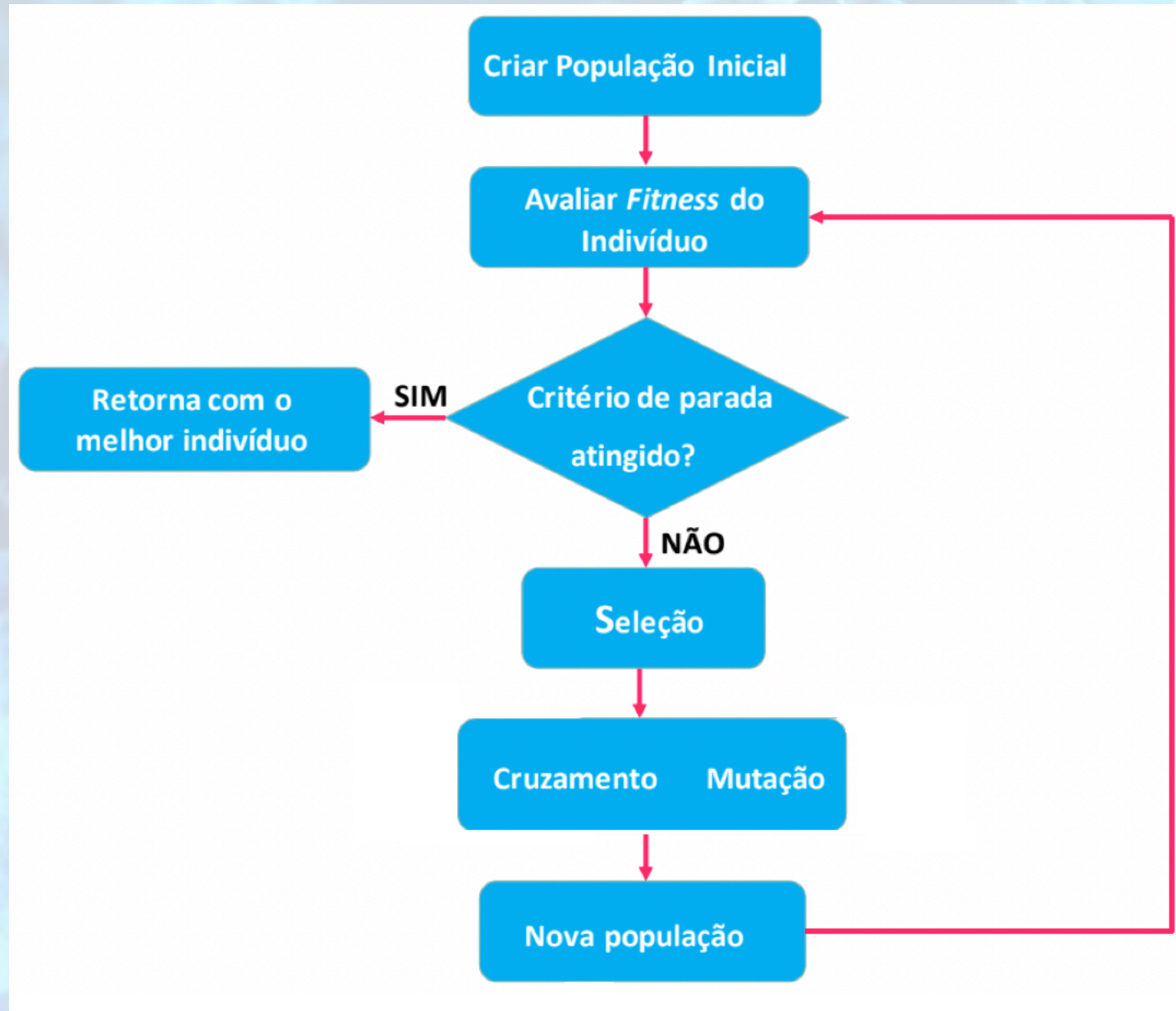
- Codificação genética das possíveis soluções do problema;
- População inicial com diversidade suficiente para permitir ao algoritmo combinar características e produzir novas soluções;
- Definição de um método para medir a qualidade de uma solução potencial;
- Definição de procedimentos de combinação de soluções para gerar novos indivíduos na população.

INTRODUÇÃO

É preciso definir:

- Critério de escolha das soluções que permanecerão na população ou que serão retiradas;
- Procedimento para introduzir, periodicamente, alterações em algumas soluções da população. Desse modo mantém-se a diversidade da população e a possibilidade de se produzir soluções inovadoras para serem avaliadas pelo critério de seleção dos mais aptos.

FLUXOGRAMA AG



CRIAÇÃO DA POPULAÇÃO INICIAL

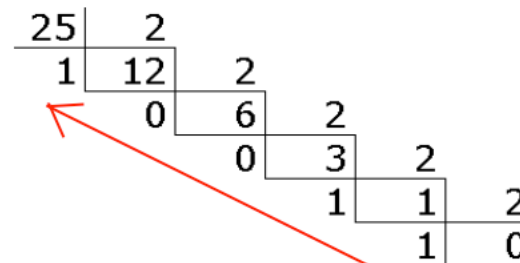
Para iniciar o algoritmo é necessário criar uma população de indivíduos (cromossomos) que representam soluções para o problema.

Assim, os indivíduos são codificados por meio de uma associação dos elementos do espaço de **genótipos** ao espaço de busca, os **fenótipos**.

Por exemplo, em uma codificação binária, a cadeia 10011011 é uma possível solução de um dado problema com 8 elementos. O conjunto dos genótipos é formado por todos os números binários de 00000000 a 11111111 ($2^8 = 256$ neste exemplo).

Codificação binária

$$(1\ 1\ 0\ 0\ 1) = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 16 + 8 + 0 + 0 + 1 = 25$$



CRIAÇÃO DA POPULAÇÃO INICIAL

Quantos indivíduos gerar?

- População pequena – grande chance de não explorar todo o espaço de soluções;
- População grande – perda de eficiência pelo alto custo computacional.

Como gerar a população inicial?

Realizada aleatoriamente utilizando um gerador de números aleatórios com **distribuição uniforme** dos indivíduos numa faixa definida pelo usuário. Pode-se introduzir na população inicial um ou mais indivíduos “interessantes” como por exemplo, soluções aproximadas conhecidas ou contendo algum tipo de informação prévia.

AVALIAR *FITNESS* DO INDIVÍDUO

A função *fitness* (aptidão) avalia a capacidade de que o indivíduo ofereça uma solução satisfatória para o problema (adaptação ao ambiente). Por intermédio dela, alguns indivíduos são selecionados (com regra probabilística) para passar por um processo de recombinação (operadores genéticos).

A função *fitness* pode representar um ganho (lucro) para um problema de maximização ou um custo, despesa, distância para um problema de minimização.

CRITÉRIOS DE PARADA

- Alcançar um número máximo de gerações (iterações);
- Solução ótima encontrada (quando é conhecida);
- Perda de diversidade;
- Convergência: nas últimas k gerações não houve melhoria na aptidão (valor *fitness*).

SELEÇÃO

Serve para escolher os indivíduos sobre os quais serão aplicados os operadores genéticos (dois indivíduos são selecionados).

Existem diversas formas de seleção, sendo as mais utilizadas:

- Roleta;
- *Ranking*;
- Torneio;

SELEÇÃO

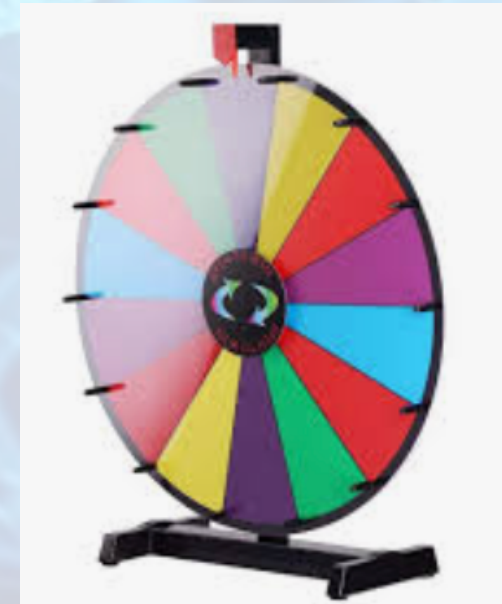
- **Roleta**

Especifica a probabilidade de que cada indivíduo seja selecionado para a próxima geração:

$$p_i = \frac{f_i}{\sum_{j=1}^n f_j}$$

em que:

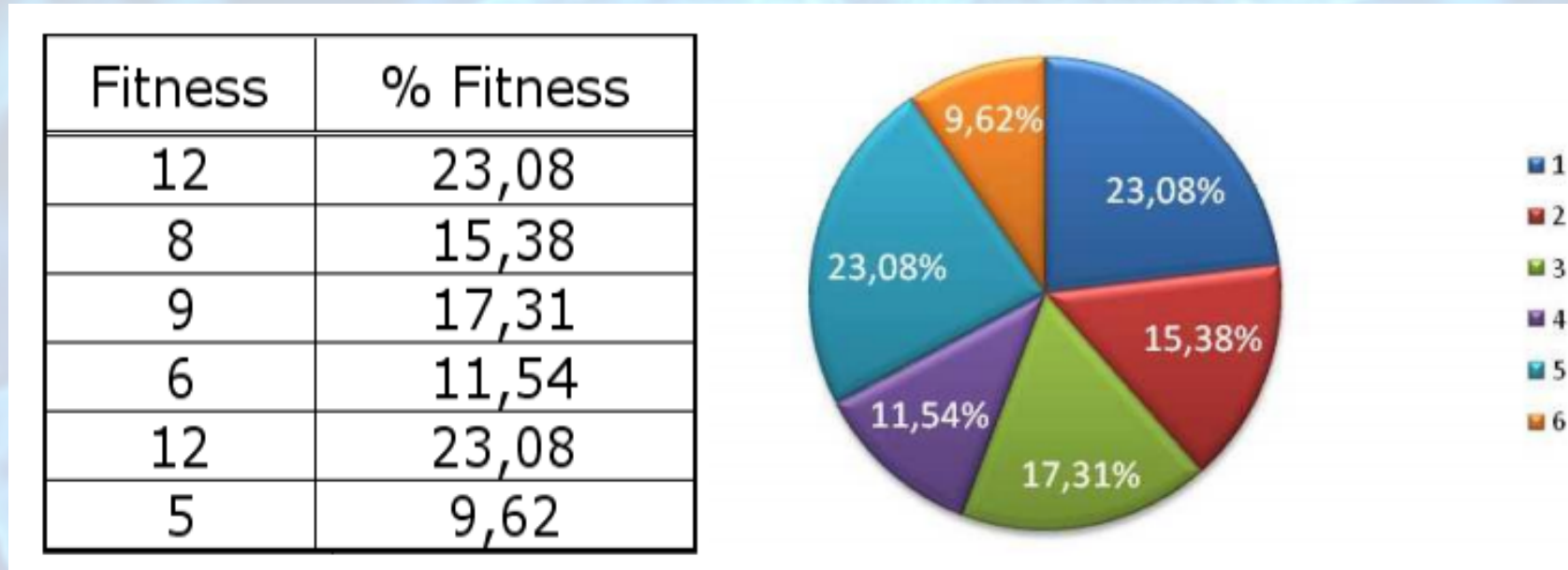
- n – tamanho da população;
- f_i – valor *fitness* do indivíduo;
- $\sum_{j=1}^n f_j$ – valor acumulado de fitness de todos os indivíduos da população.



SELEÇÃO

- **Roleta**

Cada indivíduo i da população recebe uma porção da roleta proporcional ao seu valor p_i .



O sorteio dos elementos é feito por meio de um “jogo de roleta”, onde a probabilidade de cada indivíduo ser selecionado é proporcional ao seu *fitness*.

SELEÇÃO

- **Roleta**

Pode ocorrer que os indivíduos com melhor *fitness* não sejam selecionados, pois sua chance de escolha não é de 100%.

Em contrapartida, pode-se adotar o critério de **elitismo**, em que uma porcentagem da população com os melhores *fitness* é preservada para a próxima geração automaticamente.

OPERADORES GENÉTICOS

São operações para transformar a população por meio de sucessivas gerações até chegar a um resultado satisfatório. Desta forma, a população se diversifica e mantém características de adaptação adquiridas nas gerações anteriores.

Os principais operadores genéticos são:

- Cruzamento;
- Mutação.



CRUZAMENTO (*CROSSOVER*)

Dois indivíduos (pais) que foram selecionados na etapa de seleção trocam materiais genéticos, permutando uma parte de um dos pais por uma parte de outro, gerando dois novos indivíduos (filhos), esperando-se que sejam melhores do que os pais.

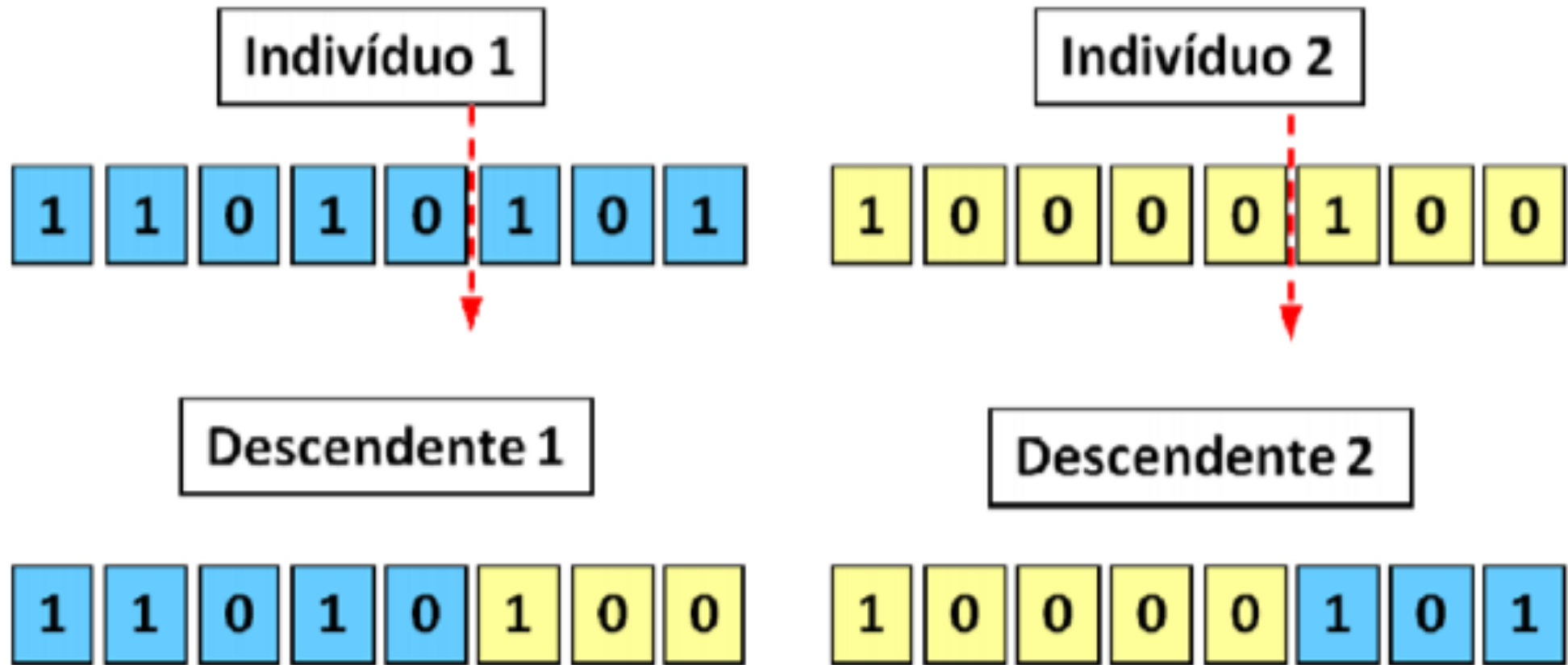
CRUZAMENTO (*CROSSOVER*)

O *crossover* é feito da forma:

1. Os dois indivíduos escolhidos na etapa de Seleção serão os pais, que trocarão material genético;
2. Seleciona-se, aleatoriamente, um ou dois pontos de cruzamento;
3. Permutam-se os genes dos dois pais no(s) ponto(s) de cruzamento, gerando dois filhos, que poderão fazer parte da nova população.

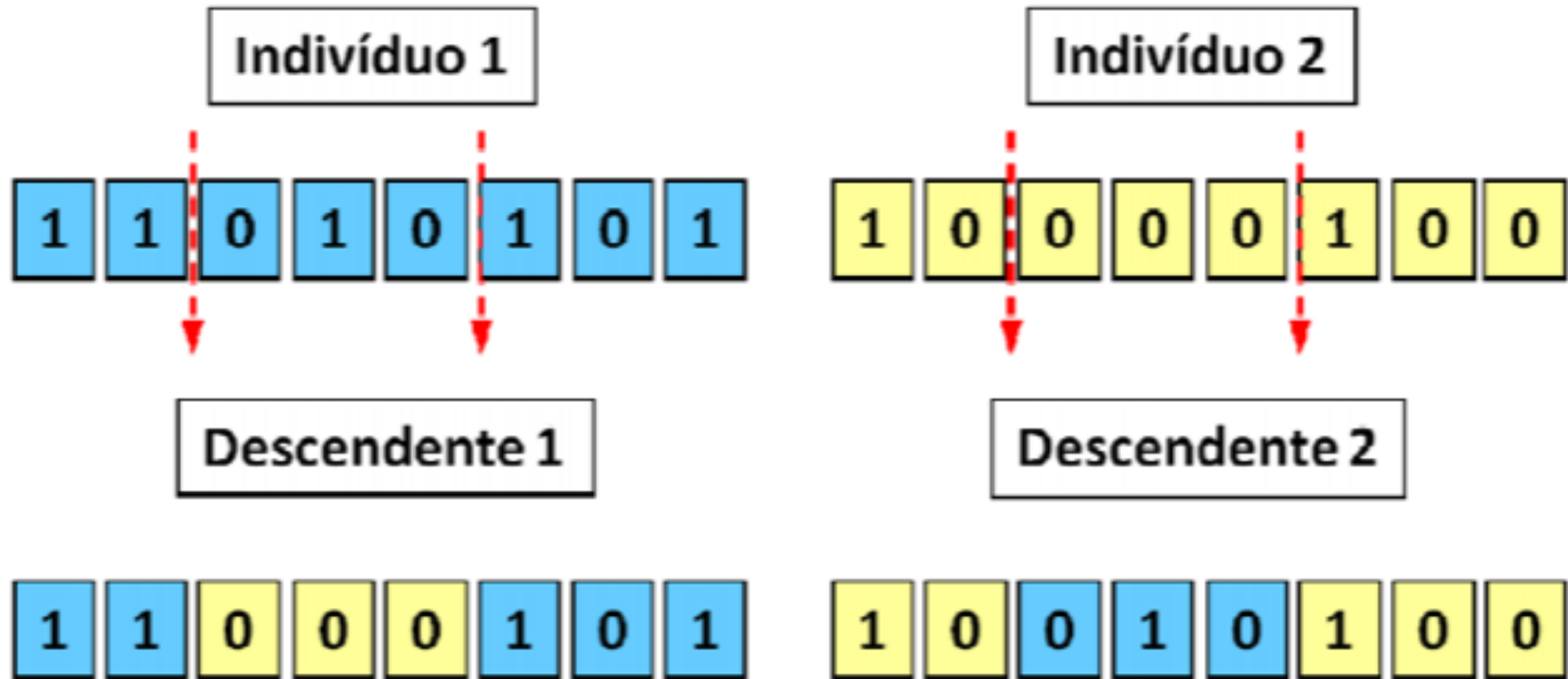
CRUZAMENTO (*CROSSOVER*)

Exemplo com 1 ponto de cruzamento



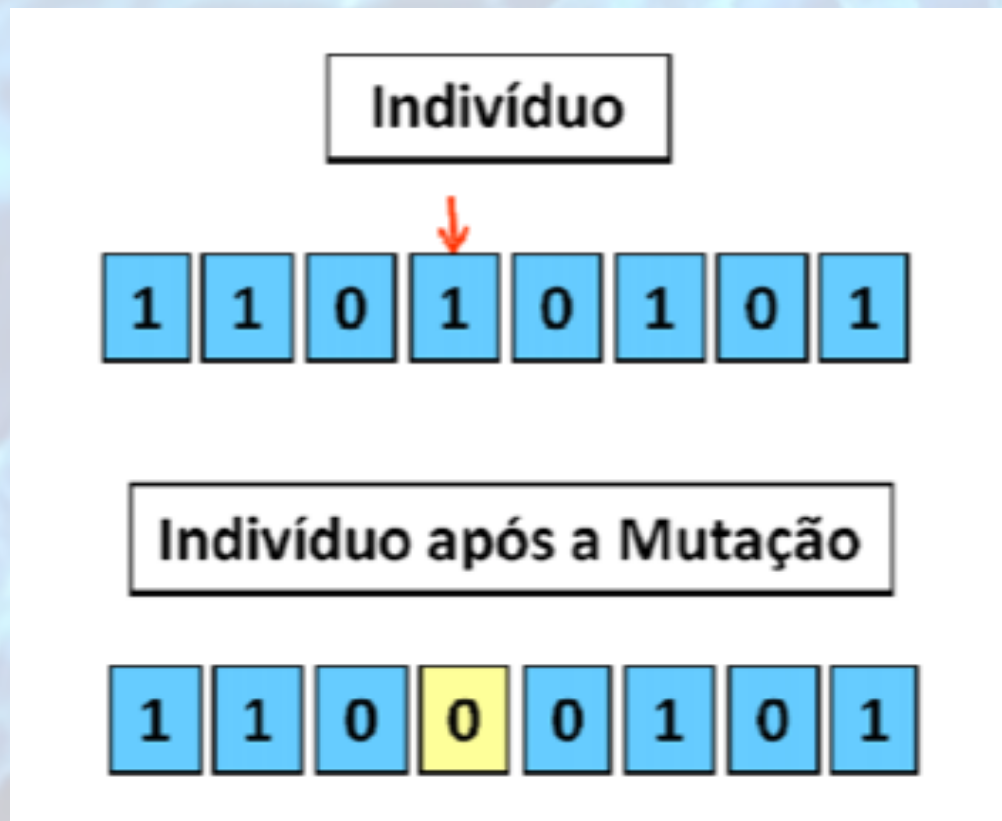
CRUZAMENTO (*CROSSOVER*)

Exemplo com 2 pontos de cruzamento



MUTAÇÃO

Uma posição do cromossomo é sorteada e o *bit* correspondente é invertido, isto é, se for “1” ele passa a ser “0” e vice-versa.



Oferece oportunidade para que mais pontos do espaço de busca sejam avaliados. Evita estagnação em ótimos locais.

TAXAS DE *CROSSOVER* E MUTAÇÃO

O *crossover* deve ser realizado com uma taxa relativamente alta (75 a 95%).

A mutação deve ser aplicada a uma taxa relativamente baixa (de 5 a 10%), caso contrário o algoritmo fará uma busca aleatória, dificultando a convergência.

NOVA POPULAÇÃO

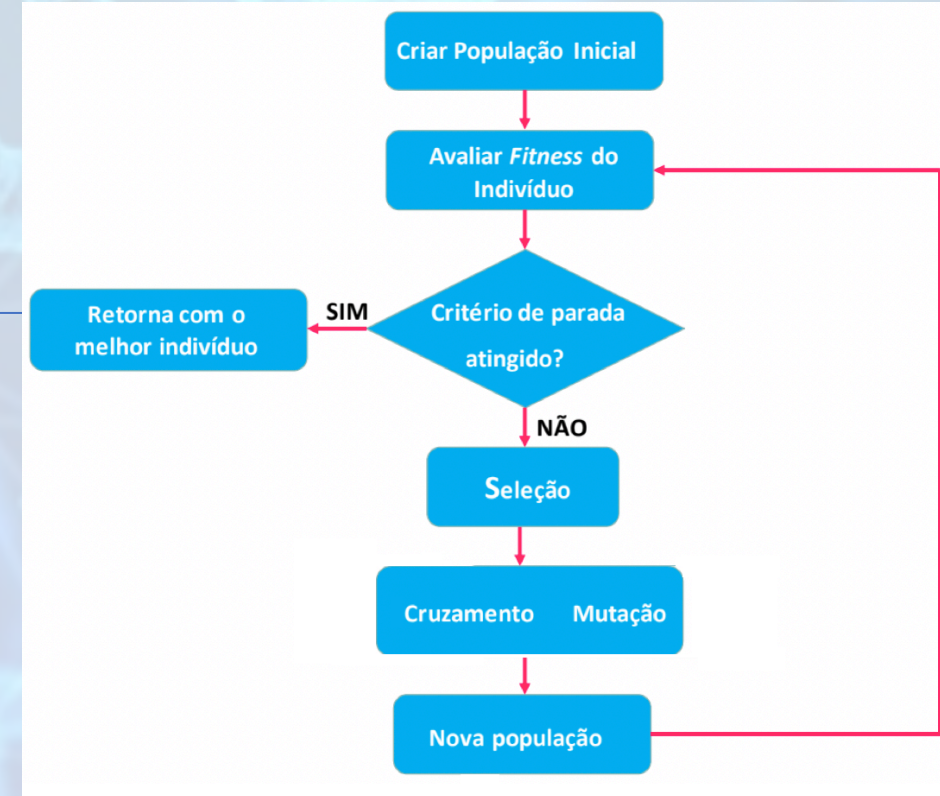
- *Steady-State*: Insere-se nenhum, um ou dois filhos na população, retirando os piores indivíduos de acordo com o *fitness* e recomeça-se o processo;

OU

- Generacional: Toda a população é substituída a cada geração.

ALGORITMO

- PASSO 1: Gerar população inicial;
PASSO 2: Avaliar o *fitness*;
PASSO 3: Critério de parada;
PASSO 4: Seleção;
PASSO 5: Operadores genéticos (*crossover* e *mutação*);
PASSO 6: Substituir a população e voltar ao passo 2.



PARÂMETROS AG

- Tamanho da população inicial;
- Tipo da seleção (roleta, torneio, *ranking*,...);
- Taxa de crossover;
- Quantidade de pontos de cruzamento;
- Taxa de mutação;
- Critério de parada.

EXEMPLO 1 – MÍNIMO/MÁXIMO DE UMA FUNÇÃO

Utilize AG para encontrar o ponto de mínimo (inteiro) da função $f(x) = (x - 4)^2 - (x - 8)^3 + 5$ no intervalo $[0, 10]$ com os quatro indivíduos abaixo, usando 5 gerações.

| |
|----------------------|
| $i_1 = (1\ 0\ 0\ 1)$ |
| $i_2 = (1\ 0\ 0\ 0)$ |
| $i_3 = (0\ 0\ 1\ 0)$ |
| $i_4 = (0\ 0\ 0\ 1)$ |

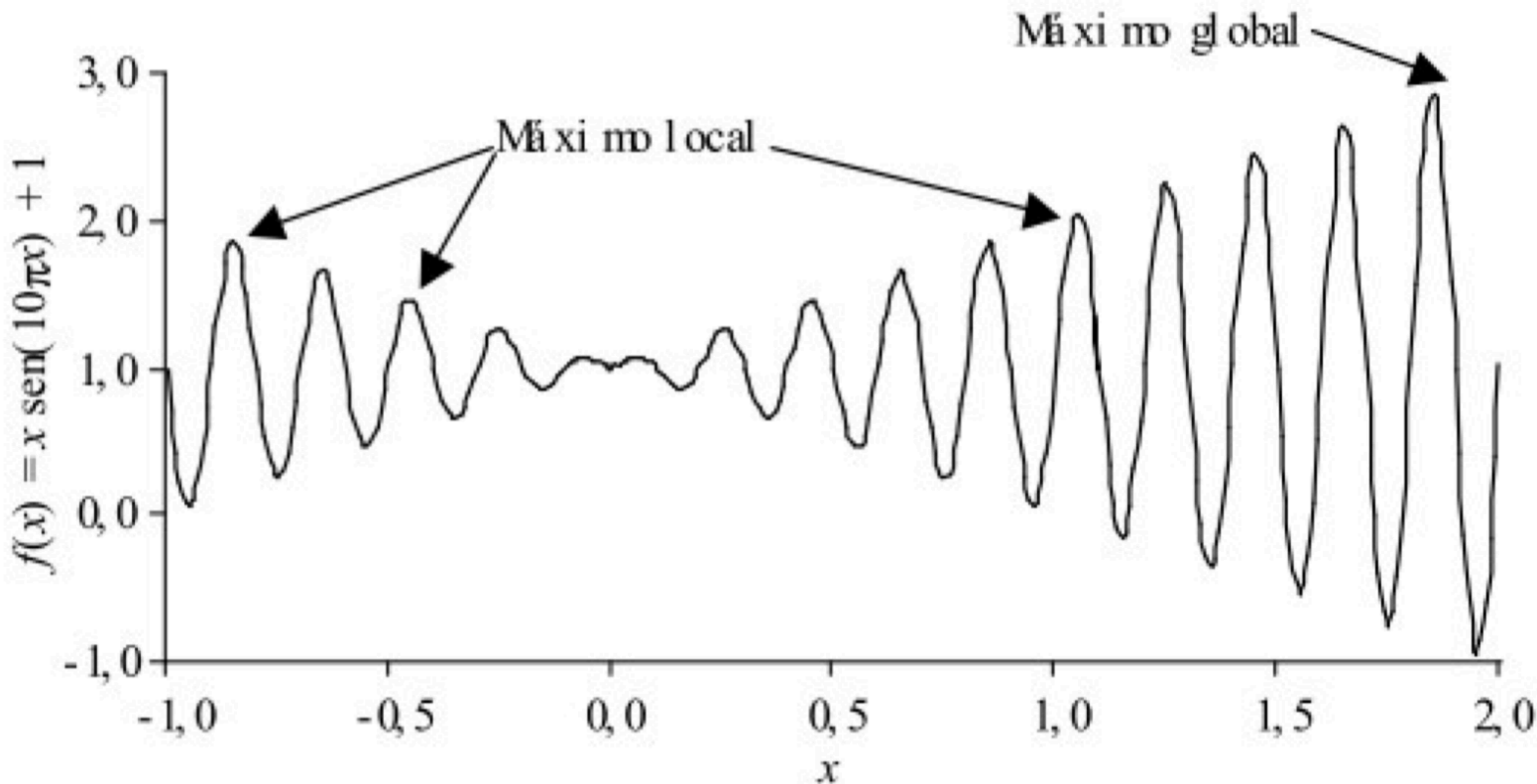
Codificação binária de 4 *bits* pois o maior inteiro (10) em binário é 1010.

| DECIMAL | BINARIO |
|---------|---------|
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 4 | 0100 |
| 5 | 0101 |
| 6 | 0110 |
| 7 | 0111 |
| 8 | 1000 |
| 9 | 1001 |
| 10 | 1010 |

EXEMPLO 1.1 – MÍNIMO/MÁXIMO DE UMA FUNÇÃO

Achar ponto máximo da função

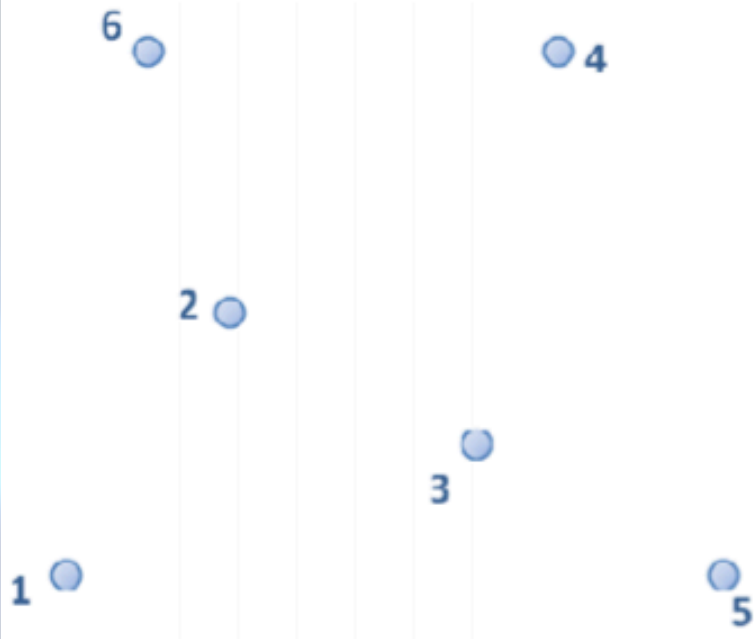
■ $f(x) = x \operatorname{sen}(10\pi x) + 1, \quad -1 \leq x \leq 2$



Máximo global:
1,85055

EXEMPLO 2 – PROBLEMA DE p -MEDIANAS

Utilize o AG para encontrar 4 medianas para o conjunto de 6 vértices dado abaixo com os 5 indivíduos da população inicial indicados.



| pontos | pesos |
|--------|-------|
| 1 | 2 |
| 2 | 3 |
| 3 | 2 |
| 4 | 4 |
| 5 | 4 |
| 6 | 5 |

| d | 1 | 2 | 3 | 4 | 5 | 6 |
|-----|---|---|---|---|---|---|
| 1 | 0 | 2 | 2 | 4 | 4 | 4 |
| 2 | 2 | 0 | 1 | 2 | 3 | 2 |
| 3 | 2 | 1 | 0 | 2 | 2 | 3 |
| 4 | 4 | 2 | 2 | 0 | 3 | 3 |
| 5 | 4 | 3 | 2 | 3 | 0 | 5 |
| 6 | 4 | 2 | 3 | 3 | 5 | 0 |

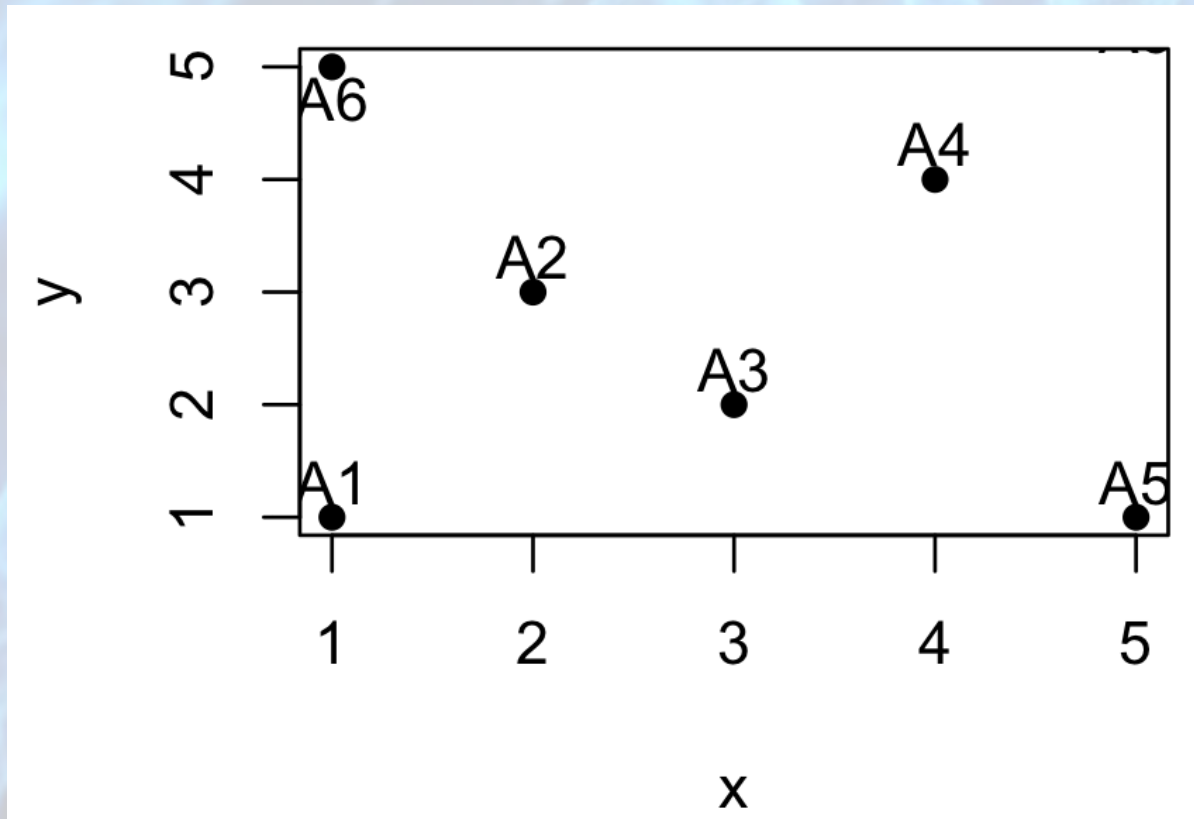
População inicial:

| | | | | |
|-------|---|---|---|---|
| r_1 | 1 | 3 | 5 | 6 |
| r_2 | 2 | 4 | 6 | 1 |
| r_3 | 3 | 5 | 6 | 2 |
| r_4 | 1 | 4 | 5 | 6 |
| r_5 | 6 | 3 | 2 | 4 |

EXEMPLO 3 – CAIXEIRO VIAJANTE

Dados os pontos com as seguintes coordenadas:

$A_1 = (1, 1)$, $A_2 = (2, 3)$, $A_3 = (3, 2)$, $A_4 = (4, 4)$, $A_5 = (5, 1)$ e $A_6 = (1, 5)$, a serem visitados por um caixeiro viajante, determine a solução ótima (ou quase ótima) utilizando AG. (Fazer apenas duas iterações).



Utilize 4 indivíduos e o método da roleta para selecionar os indivíduos.

EXEMPLO 4 – PROBLEMA DE TRANSPORTE

Dado o seguinte Problema de Programação Linear Inteira, determine uma solução ótima (ou quase ótima) utilizando o Algoritmo Genético. Faça duas iterações completas explicando bem cada passo.

Um transportador autônomo possui 10 m^3 de espaço disponível sobre um caminhão com partida programada para Ponta Grossa. Um distribuidor com grandes quantidades de 3 diferentes tipos de utensílios, todos destinados à cidade de Ponta Grossa, ofereceu ao transportador as seguintes taxas de pagamento para transportar tantos itens quantos o caminhão possa acomodar:

| Utensílios | Taxa (R\$/item) | Volume (m^3/item) | Quantidade |
|------------|-----------------|-------------------------------------|------------|
| I | 11 | 2 | 5 |
| II | 32 | 3 | 4 |
| III | 58 | 5 | 3 |

Determine quantas unidades de cada item o transportador deve levar de forma a maximizar o seu lucro.

EXEMPLO 5 – PROBLEMA DE DESIGNAÇÃO

Dado o seguinte problema de designação, determine uma solução ótima (ou quase ótima) utilizando o Algoritmo Genético. Faça duas iterações completas explicando bem cada passo.

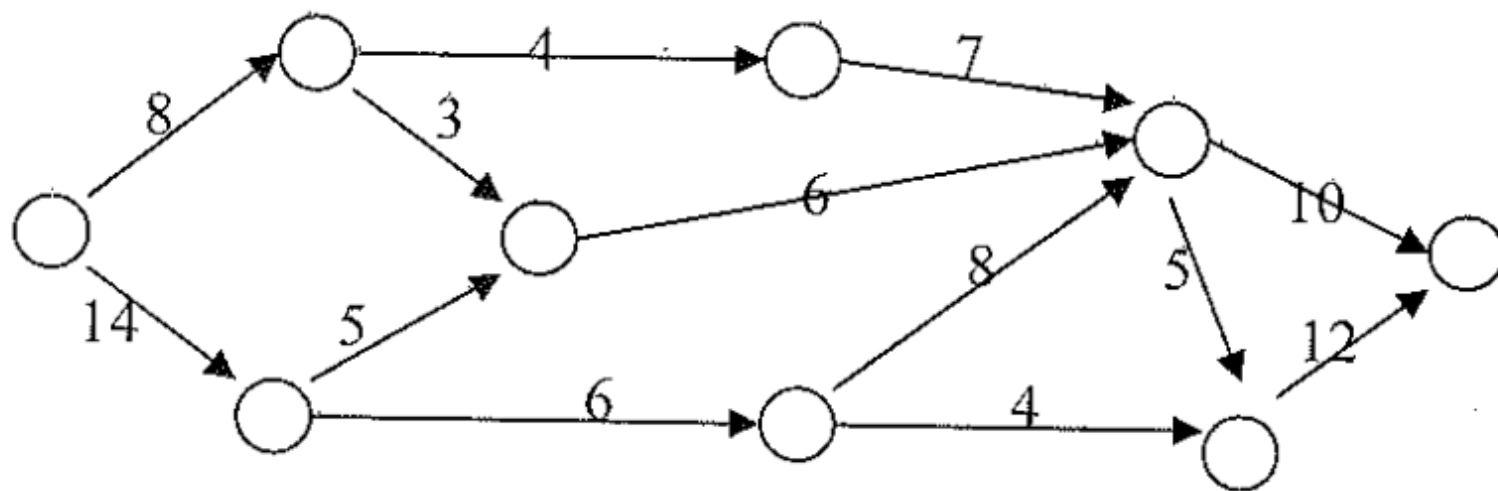
A Metalúrgica Araucária S/A, dentro de 60 dias, deverá começar a funcionar em sua nova sede localizada na Cidade Industrial de Curitiba (CIC). O Presidente da Metalúrgica deseja que a distribuição das salas, dessa nova instalação, seja feita de modo a atender, na medida do possível, as preferências já manifestadas. Em uma pesquisa realizada, os Diretores manifestaram as suas preferências:

| | Sala 1 | Sala 2 | Sala 3 | Sala 4 | Sala 5 | Sala 6 |
|-----------|--------|--------|--------|--------|--------|--------|
| Diretor 1 | 2 | 4 | 3 | 1 | 5 | 6 |
| Diretor 2 | 1 | 5 | 4 | 6 | 3 | 2 |
| Diretor 3 | 5 | 3 | 4 | 2 | 1 | 6 |
| Diretor 4 | 1 | 3 | 2 | 4 | 6 | 5 |
| Diretor 5 | 3 | 2 | 5 | 6 | 1 | 3 |

Se você fosse convidado a opinar sobre a distribuição das salas qual seria a sua recomendação, de forma a satisfazer ao máximo as preferências dos diretores?

EXEMPLO 6 – PROBLEMA DE MÍNIMA ARBORESCÊNCIA

Dado o seguinte grafo com 9 nós e 13 arcos, determine a MÍNIMA ARBORESCÊNCIA utilizando a metaheurística Algoritmo Genético. Faça duas iterações completas explicando bem cada passo. Obs: para a obtenção de uma mínima arborescência em um grafo $G(N, A)$, deve-se ter para n nós dados, $(n - 1)$ arcos e, além disso, o grafo deve ser conexo e não pode formar ciclos; os "sentidos" nos arcos do grafo em questão, podem ser desconsiderados.



REFERÊNCIAS

SIQUEIRA, P. H. Metaheurísticas e Aplicações. Parte III – Outras Metaheurísticas. PPGMNE/UFPR. Notas de Aula.

GENDREAU, M., POTVIN, J., Y. Handbook of Metaheuristics. Springer, vol. 146, 2 ed., 2010.