

### 3º Atividade

## Algoritmo Genético (AG) X Simulated Annealing (S.A)

### Benefícios

Variação  $\rightarrow$   $X$ : conjunto de cidades  
 $w(x_i, x_j)$ : custo entre cidades

3 principais métodos (principais):  
Ordem, por caminhos (ou intuitivo) e adjacências

Em Benefícios:

$\hookrightarrow$  por caminhos, cromossomo é formado pela seq. dos nós na solução

$\hookrightarrow$  Solução mais simples e natural

Cromossomo  $C_i = \{1, 2, 3, 4, 5, 6, 7\}$  representa solução  $r_i = \{1, 2, 3, 4, 5, 6, 7\}$

$\rightarrow$  Função Aptidão - Fitness  
 $\hookrightarrow$  custo total do circuito

$$Fitness(r_i) = w(x_n, x_1) + \sum_{j=1}^{n-1} w(x_j, x_{j+1})$$

$\hookrightarrow$  Menor custo  $\Rightarrow$  + chance de reprodução

$\rightarrow$  Seleção

$\hookrightarrow$  Seleção Elitista segundo Maynard

$\rightarrow$  Operador de cruzamento

Preservar a posição absoluta de cidades  
e preservar ordem relativa

$\rightarrow$  Simulação de Recozimento  $\rightarrow$  versão adaptada do HX  
(Heuristic Crossover)  
melhor resultado, principalmente com mais nós

$\rightarrow$  Operador de Mutação

$\hookrightarrow$  entre mínimos locais

### Algoritmo

$\hookrightarrow$  Def. anteriores e princípios da evolução das espécies

P1: Construção da população inicial: gere os  $m$  cromossomos aleatórios representando os roteiros Hamiltonianos. Calcule os custos  $C_i = [Fitness(r_i)]$ , ( $1 \leq i \leq m$ ) de todos os cromossomos gerados, construindo uma lista  $R = (r_1, r_2, \dots, r_m)$ , de forma que  $C_1 \leq C_2 \leq \dots \leq C_m$ ; faça  $k = 0$ , defina o erro  $\epsilon$  e o número máximo de iterações  $k_{max}$ ;

Obs: Os cromossomos são construídos de modo que a probabilidade de uma cidade suceder outra, seja inversamente proporcional ao quadrado da distância que as separa;

P2: Teste: se  $C_n - C_1 \leq \epsilon$  ou  $k \geq k_{max}$ , então PARE e apresente o cromossomo  $r_1$ ;

P3: Seleção Natural: selecione dois cromossomos  $r_p = Select(R)$  e  $r_q = Select(R)$  com  $r_p \neq r_q$ ;

P4: Reprodução: faça  $r_f = Crossover(r_p, r_q)$ ;

P5: Mutação: aplique o operador de mutação com uma probabilidade  $p$ ;

P6: Calcule  $F_f = Fitness(r_f)$  e caso o fitness do cromossomo filho for melhor do que o fitness do pior cromossomo, elimine esse cromossomo, inserindo o cromossomo  $r_f$  na lista  $R$ , mantendo a ordem crescente dos custos; faça  $k = k + 1$ , e volte ao P2.

Procura Gradativa

$\Delta E \leq 0$ , passo seguinte

$\Delta E > 0$ , aceite se não parar a ser uma decisão probabilística

Analogia:

$\hookrightarrow$  Identifica-se a função energia com a função objetivo e os estados do sistema são associados às variáveis.

$\hookrightarrow$  Para cada Temperatura de uma seq. de Temperaturas decrescente realiza-se a simulação, assim espera-se que o sistema estacione em um mínimo global.

$\rightarrow$  Verificação da verdadeira condição de equilíbrio é muito difícil

$\therefore$  H: n.º máx de iterações

Os passos do algoritmo contidos em Barbosa (1989) são os seguintes:

Seja  $x_0 \in D$  a solução inicial e  $T_0$  a temperatura inicial;

$k = 0$ ;  $x_k = x_0$ ;  $T_k = T_0$

While  $T_k \geq T_{min}$  do

begin

While ainda não em equilíbrio a temperatura  $T_k$  do

begin

selecione aleatoriamente um ponto  $x' \in A(x_k) \subset D$

$\Delta = f(x') - f(x)$

if  $\Delta \leq 0$  then

$x_k = x'$

else

$x_k = x'$  com probabilidade  $e^{-\frac{\Delta}{T_k}}$

end

$T_{k+1} = r(T_k)$ ;  $x_{k+1} = x_k$ ;  $k = k + 1$

End

A solução é  $x_k$ .

### Conclusão Benefícios

- Na dos casos há uma melhoria na rota.
- Maioria dos casos o A.G. é melhor.
- S.A. é mais estatístico

### A.C.

- Gera e Armazena mais soluções
- Função de Avaliação
- Espaço de Solução Maior
- Busca Ampla
- Baseado em população
- Cruzamento e Mutação
- Alta sensibilidade a parâmetros
- Complexo de implementar
- Alto requisito de memória
- Melhor desempenho em muitos nos
- Sensível a condições iniciais
- Paralelização
- Vários linguagens

### Outros X

### S.A.

- Escolhe uma solução e melhora
- Critério Probabilístico
- Menor Tempo
- Busca Local
- Baseado em melhores soluções
- Insensível a perturbação
- Menor sensibilidade a parâmetros
- Fácil de implementar
- Baixo requisito de memória
- Dificuldades com muitos nos
- Sensível a função objetivo
- Avaliação de soluções
- Vários linguagens

Para escolha do algoritmo, temos:

- Convergência
- Tamanho da população
- nº de iterações
- Memória disponível
- Sensibilidade a ruídos
- Facilidade de Implementação
- Robustez
- Parâmetros