

Licenciatura em Engenharia Informática

Regime Pós-Laboral

Introdução à Inteligência Artificial

2020/2021

Liliana Faustino 2017012944

Algoritmos de Pesquisa Local

- Trepa Colinas First Choice
- Trepa Colinas aceita soluções com qualidade igual
- Trepa Colinas com 3 vizinhanças e aceita soluções de qualidade igual
- Trepa Colinas Probabilístico

Algoritmos Genéticos

- Algoritmo Base
- Algoritmo Recombinação de 2 Pontos de Corte + Mutação por Troca
- Algoritmo Recombinação Uniforme + Mutação por Troca

Algoritmos Genéticos - Algoritmo Base

```
function crossover (problema) retorna os descendentes
  inputs: parents
          tamanho_populacao
          numGenes
          parametro_PR
  variáveis locais: ponto_corte
                   probabilidade_gerada
  loop for i = 0 until tamanho_populacao incrementando 2
    if probabilidade_gerada < parametro_PR then
      ponto_corte = random(0, numGenes-1)
      loop for j = 0 until ponto_corte incrementando 1
        descendente[i].posicao[j] <- parents[i].posicao[j]
        descendente[i+1].posicao[j] <- parents[i+1].posicao[j]

      loop for j = ponto_corte until numGenes incrementando 1
        descendente[i].posicao[j] <- parents[i+1].posicao[j]
        descendente[i+1].posicao[j] <- parents[i].posicao[j]
    else
      descendente[i] <- parents[i]
      offspdescendering[i+1] <- parents[i+1]
```

Algoritmos Genéticos - Algoritmo Base

```
function mutacao_por_troca(problema) retorna os descendentes trocados
  inputs: descendentes
          tamanho_populacao
          numGenes
          parametro_PM
  variáveis locais: posicao_1
                    posicao_2
                    probabilidade_gerada
  loop for i = 0 until tamanho_populacao incrementando 1
    if probabilidade_gerada < parametro_PM then
      posicao_1 = random(0, numGenes-1)
      loop do
        posicao_2 = random(0, numGenes-1)
      until posicao_2 != posicao_1
      descendente[i].posicao[posicao_1] <-
descendente[i].posicao[posicao_2]
      descendente[i].posicao[posicao_1] <-
descendente[i].posicao[posicao_2]
```

Algoritmos Genéticos - Algoritmo Recombinação com 2 pontos de Corte

```
function recombinaao_2_pontos_corte retorna os descendentes
  inputs: parents
          tamanho_populacao
          numGenes
          parametro_PR
  variáveis locais: ponto_corte_1
                    ponto_corte_2
                    probabilidade_gerada
  loop for i <- 0 until tamanho_populacao incrementando 2
    if probabilidade_gerada < parametro_PR then
      loop do
        ponto_corte_1 <- random(0, numGenes-1)
        ponto_corte_2 <- random(0, numGenes-1)
      until ponto_corte_1 != ponto_corte_2
      loop for j <- 0 until ponto_corte_1 incrementando 1
        descendente[i].posicao[j] <- parents[i].posicao[j]
        descendente[i+1].posicao[j] <- parents[i+1].posicao[j]

      loop for j <- ponto_corte_1 until ponto_corte_2
        incrementando 1
        descendente[i].posicao[j] <- parents[i+1].posicao[j]
        descendente[i+1].posicao[j] <- parents[i].posicao[j]
      loop for j <- ponto_corte_2 until numGenes incrementando 1
        descendente[i].posicao[j] <- parents[i].posicao[j]
        descendente[i+1].posicao[j] <- parents[i+1].posicao[j]

    else
      descendente[i] <- parents[i]
      offspdescendering[i+1] <- parents[i+1]
```

Algoritmos Genéticos - Algoritmo Recombinação uniforme

```
function recombinacao_uniforme retorna os descendentes
  inputs: parents
          tamanho_populacao
          numGenes
          parametro_PR
  variáveis locais: probabilidade_gerada

  loop for i <- 0 until tamanho_populacao incrementando 2
    if probabilidade_gerada < parametro_PR then

      loop for j <- 0 until numGenes incrementando 1
        if random (0, 1) == 1 then
          descendente[i].posicao[j] <- parents[i].posicao[j]
          descendente[i+1].posicao[j] <- parents[i+1].posicao[j]

        else
          descendente[i].posicao[j] <- parents[i+1].posicao[j]
          descendente[i+1].posicao[j] <- parents[i].posicao[j]

      else
        descendente[i] <- parents[i]
        offspdescendering[i+1] <- parents[i+1]
```

Algoritmos Híbridos

- Trepa colinas na População Inicial
- Trepa colinas na População Final
- Trepa colinas no Ciclo de Gerações

Conclusões Finais