



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
CIÊNCIA DA COMPUTAÇÃO

Bruno Ferreira Tomé - 12011BCC050

ATIVIDADE 01 - COMPUTAÇÃO EVOLUTIVA

Uberlândia - MG

2024

1. Estratégia evolutiva:

1.1. Resumo:

O código implementa uma Estratégia Evolutiva para otimização, utilizando o modelo $(\mu + \lambda)$ -EE. A função a ser otimizada é $f(x) = x * \sin 10\pi x + 1$. A estratégia começa com uma população inicial de pais escolhidos aleatoriamente, e em cada iteração, cria descendentes ao introduzir perturbações nos pais. A aptidão é avaliada tanto para os pais quanto para os descendentes, sendo selecionados os melhores indivíduos para formar a próxima geração. Esse processo é repetido por um número predefinido de gerações, exibindo o resultado mais eficaz em cada iteração. Os parâmetros, como o tamanho da população (μ), o número de descendentes (λ) e o número máximo de gerações (tmax), são ajustáveis para explorar diversas configurações da estratégia evolutiva.

O código-fonte foi encaminhado junto com o arquivo compactado (.zip), denominado "código-fonte". Adicionalmente, está disponível no GitHub por meio do seguinte link: <https://github.com/brunoftom/Computacao-Evolutiva/tree/main/Atividade01>.

1.2. Função de Aptidão:

```
# Função de Aptidão
funcao_aptidao <- function(x) {
  return(x * sin(10 * pi * x) + 1)
}
```

A função '*funcao_aptidao*' define a função de aptidão que será otimizada. Neste caso, é a função $f(x) = x * \sin 10\pi x + 1$, sujeito a $:- 1 \leq x \leq 2$

1.3. Estratégia Evolutiva $(\mu + \lambda)$ -EE:

```
# Estratégia Evolutiva  $(\mu + \lambda)$ -EE
estrategia_evolutiva <- function(mu, lambda, tmax) {
  # Inicialização da população de pais
  pais <- runif(mu, -1, 2)

  for (t in 1:tmax) {
    # Geração de filhos
    descendencia <- matrix(rnorm(mu * lambda, mean = pais, sd = 0.1),
                          nrow = lambda, ncol = mu)
  }
```

- A função '*estrategia_evolutiva*' implementa a estratégia evolutiva $(\mu + \lambda)$ -EE.
- Inicialmente, gera-se uma população de pais aleatória '*pais*' com tamanho '*mu*'.
- Em cada geração, são gerados filhos '*descendencia*' por meio de uma permutação aleatória de pais.

```
# Avaliação da aptidão
aptidao_pais <- funcao_aptidao(pais)
aptidao_descendencia <- apply(descendencia, 1, funcao_aptidao)

# Combinação de pais e filhos para seleção
populacao_total <- c(pais, descendencia)
todas_aptidoes <- c(aptidao_pais, aptidao_descendencia)
```

- A aptidão é avaliada para os pais e filhos em '*aptidao_pais*' e '*aptidao_descendencia*'.

```

# Seleção dos melhores  $\mu$  indivíduos
indices <- order(todas_aptdoes, decreasing = TRUE)[1:mu]
pais <- populacao_total[indices]

# Exibição do melhor resultado a cada geração
melhor_individuo <- populacao_total[which.max(todas_aptdoes)]
cat(sprintf("Geração %d: Melhor indivíduo - x=%.4f, Aptidão=%.4f\n",
            t, melhor_individuo, max(todas_aptdoes)))
}

return(pais)

```

- A população completa (pais + filhos) é combinada, e os melhores ‘mu’ indivíduos são selecionados com base na aptidão.
- O processo é repetido por um número de gerações ‘tmax’, exibindo o melhor resultado a cada geração.

1.4. Variação de Parâmetros:

```

# Parâmetros reduzidos
valores_mu <- c(1, 3)
valores_lambda <- c(2, 5)
valores_tmax <- c(10, 20)

```

- Os valores de ‘mu’, ‘lambda’ e ‘tmax’ são variados conforme solicitado para explorar diferentes configurações da estratégia evolutiva.

1.5. Laços de Repetição:

```

for(i in 1:30) {
  for (mu in valores_mu) {
    for (lmbda in valores_lambda) {
      for (tmax in valores_tmax) {
        cat("\nμ=", mu, ", λ=", lmbda, ", tmax=", tmax, "\n")
        estrategia_evolutiva(mu, lmbda, tmax)
      }
    }
  }
}

```

- Os loops aninhados percorrem as combinações de valores de ‘mu’, ‘lambda’ e ‘tmax’ especificados, executando a estratégia evolutiva para cada combinação. A repetição é feita 30 vezes, para gerar vários resultados.

2. Resultados:

2.1. Tabela com resultados:

- Nas iterações, foram utilizados valores idênticos para o parâmetro 'tmax', tanto para tmax com valores 10 e 20, com o intuito de possibilitar uma comparação direta. Foram registrados três indicadores principais para avaliação do desempenho: o melhor resultado (representando o mínimo obtido nas 30 execuções), a média, e o desvio padrão (indicado pelo símbolo \pm). A tabela a seguir apresenta esses valores obtidos:

tmax	μ	λ	melhor	média
10	1	2	1.9344	2.0452 \pm 0.0699
20	1	2	2.0407	2.0724 \pm 0.0376
10	1	5	1.7436	1.9277 \pm 0.1174
20	1	5	1.5800	1.8237 \pm 0.1782
10	3	2	2.6493	2.8020 \pm 0.1256
20	3	2	1.9593	2.0396 \pm 0.0452
10	3	5	1.8520	2.0188 \pm 0.1134
20	3	5	2.0472	2.5536 \pm 0.2377

2.2. Melhor resultado:

- A seleção da configuração ideal de parâmetros na estratégia evolutiva implica encontrar um equilíbrio entre uma exploração inicial abrangente e uma busca a longo prazo. Neste cenário específico, a eficácia máxima é alcançada ao adotar um tamanho de população (μ) de 3 para promover uma diversidade inicial, associado a $\lambda=5$ para uma exploração mais profunda, e tmax=20 para permitir uma evolução contínua. Essa harmonização cuidadosa resulta em soluções mais eficientes ao término das 20 iterações, enfatizando a necessidade de ajustar os parâmetros de acordo com as peculiaridades inerentes ao problema de otimização em questão.

Portanto a combinação que leva ao melhor resultado, é:

- $\mu = 3$
- $\lambda = 5$
- tmax = 20

Essa combinação, na geração 20, apresentou o melhor indivíduo, com $x = 1.6537$ e fitness de 2.6426.

- A tabela com todos os valores também pode ser acessada no arquivo "Tabela - Resultados".