

Primeiro trabalho Prático
Disciplina: Computação Natural
Professor Laércio Ives Santos
Data da Entrega via Sala de Aula do Google: Até 13/03/2022
Data da Apresentação: 14/03/2022

Orientações Gerais

O trabalho deve ser implementado em linguagem Python 3.*. ou C++;

Após implementação e realização dos testes solicitados, o aluno deve escrever um relatório contendo: Introdução (caracterização do problema atacado), Materiais e Métodos (falar do algoritmo implementado), Resultados e Discussão.

O relatório, bem como o código implementado, devem ser postados na sala de aula da disciplina até a data prevista.

Cada aluno terá no máximo 10 minutos de apresentação;

Problema 1 – Implementação de um AG para o problema “Caixa Preta”

1. 30 Indivíduos e 40 gerações.
 2. Atualização da população: Filhos substituem Pais com Elitismo
 3. Cruzamento com um ponto de corte aleatório
 4. Mutação com escolha aleatória do bit
- Comparar Métodos de Seleção :os 3 estudados em sala: roleta, torneio binário e uniforme (setar taxa de mutação em 0.03)
 - Comparar o algoritmo Com Mutação e Sem Mutação (setar taxa em 0.03 e escolher o melhor método de seleção do primeiro passo)
 - Comparar taxas de mutação em 0.03, 0.05, 0.1, 0.4
 - Para cada comparação o algoritmo deve ser executado ao menos 10 vezes
 - Fazer Estatísticas dos 10 testes:
 - Utilizar o gráfico BoxPlot da ultima geração (nas 10 vezes que foi executado)
 - Número de sucessos(chegou ao valor 27 como sinal de saída).

O problema:

Sabendo-se que cada botão da Caixa Preta apresentada na Figura 1 pode ser colocado em 16 posições distintas, e que se deseja maximizar o sinal de saída, encontre a melhor combinação de posições.

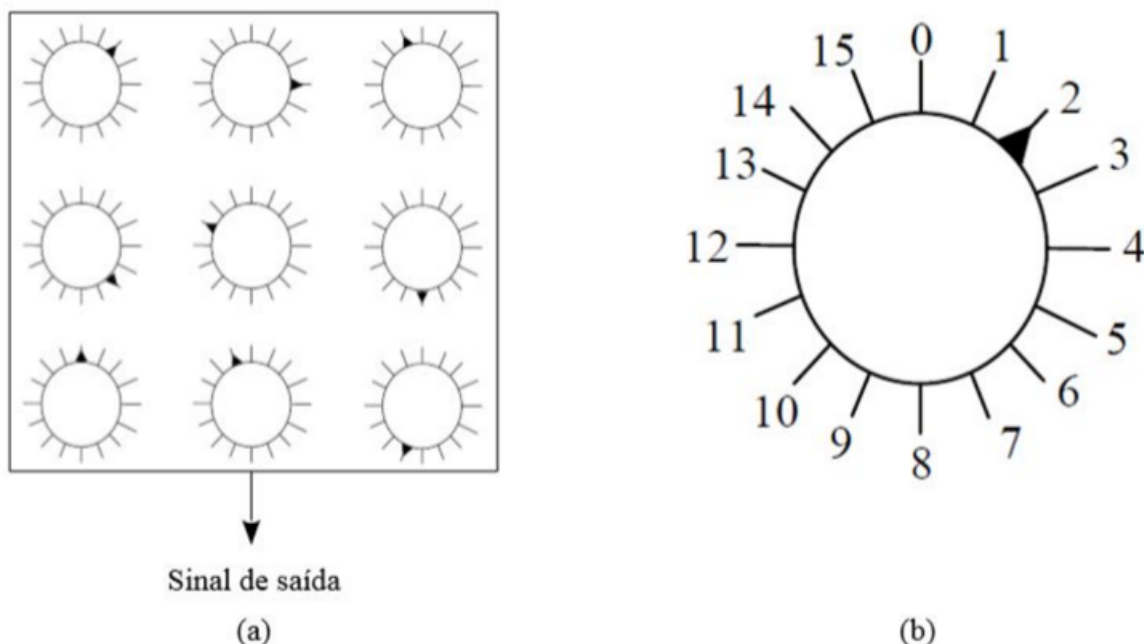


Figura 01: Representação da Caixa Preta

Sabendo-se que cada botão da Caixa Preta pode ser colocado em uma de dezesseis posições (posições de zero a quinze), cada posição pode ser representada por um número binário de quatro bits. A posição 2, por exemplo, apresentada na parte (b) da Figura 1, será representada pelo número 0010. A Tabela apresenta as representações para cada uma das dezesseis posições possíveis para cada um dos botões da Caixa Preta. Cada indivíduo (cromossomo) representará uma combinação de posições para os botões e, portanto, possuirá trinta e seis bits (quatro bits para cada um dos nove botões), totalizando 2^{36} (aproximadamente $68,72 \times 10^9$) soluções possíveis. Considerando-se essa codificação, um indivíduo representando a combinação de posições apresentada na parte (a) da Figura 1 terá a seguinte estrutura: 001001001111011011011000000011111001 (2, 4, 15, 6, 13, 8, 0, 15, 9).

Tabela 01: Representação de todas as posições

Posição	Representação	Posição	Representação
0	0000	8	1000
1	0001	9	1001
2	0010	10	1010
3	0011	11	1011
4	0100	12	1100
5	0101	13	1101
6	0110	14	1110
7	0111	15	1111

No problema “real”, o mapeamento entre as 2^{36} posições é desconhecido: só se tem acesso ao valor de saída após a aplicação de uma configuração de posições para os botões (motivo para a denominação Caixa Preta). Para fins de simulação, a Equação abaixo, apresenta o

mapeamento (arbitrário) entre uma posição de o respectivo valor de saída (b representa o bit i da posição i, da esquerda para a direita).

$$\begin{aligned} \text{sinál de saída} = & 9 + b_2 b_5 - b_{23} b_{14} + b_{24} b_4 - b_{21} b_{10} + b_{36} b_{15} - b_{11} b_{26} + b_{16} b_{17} + b_3 b_{33} \\ & + b_{28} b_{19} + b_{12} b_{34} - b_{31} b_{32} - b_{22} b_{25} + b_{35} b_{27} - b_{29} b_7 + b_8 b_{13} - b_6 b_9 + b_{18} b_{20} \\ & - b_1 b_{30} + b_{23} b_4 + b_{21} b_{15} + b_{26} b_{16} + b_{31} b_{12} + b_{25} b_{19} + b_7 b_8 + b_9 b_{18} + b_1 b_{33} \end{aligned} \quad (1)$$

A solução ótima, determinada por meio da análise da função utilizada para simulação (no problema “real” a solução ótima é desconhecida), cujo valor correspondente de saída é vinte e sete, é 111110111001101111111011111100101111.

Problema 2 - Implementação de um método baseado em Evolução Diferencial para **Minimizar** a função de Zakharov

A Função: problema de Minimização

$$Z_n(x) = \left(\sum_{j=1}^n x_j^2 \right) + \left(\sum_{j=1}^n 0,5j \times x_j \right)^2 + \left(\sum_{j=1}^n 0,5j \times x_j \right)^4$$

Onde: e n é a quantidade de variáveis do problema.

$x_j \in [-5, 10]$ com $j=1, \dots, n$

Implementar um algoritmo do tipo Diferencial Evolution para o problema Função Zakharov sem restrições para valores de $n = 3, 5$ e 10 (configurações possíveis do problema);

Para cada configuração deve-se executar o algoritmo ao menos 10 vezes e computar (para as populações finais) : plotar um gráfico do tipo Boxplot, analisar e discutir.

1. Comparar os valores de Cr em 0.3, 0.7 (F = 0,5 e implementação sem variação).
2. Comparar os valores de F em 0.5, 1.3 e 1.7 (melhor valor de Cr encontrado em 1 e implementação sem variação)
3. Implementar uma variação do algoritmo e comparar com a forma tradicional. Essa variação será na seleção, sendo que não se compara o Vetor X_i com o Vetor U_i corresponde, e sim A população X com a população U. Os melhores em cada geração (g) irão compor a população da Próxima Geração (g+1). (melhor valor encontrado em 1 e 2)

Quantidade vetores da População: 40

Quantidade de Gerações: 50

Divisão dos Temas

<i>Problema 1: Implementação de um AG para a caixa Preta</i>	<i>Problema 2: Evolução Diferencial para a função Zakharov</i>
Filipe	Ana Júlia
Reinaldo	Anne
João Vitor Amorim	Graciely
Pedro Henrique	João Vitor Ribeiro