

Primeiro Trabalho Prático da Disciplina Computação Evolutiva

Professor: Renato Dourado Maia

Curso: Especialização em Engenharia de Sistemas

1 – Objetivos

Este trabalho tem como objetivos a implementação de um Algoritmo Genético para a resolução do Problema da Caixa Preta e a realização de um estudo simplificado dos efeitos de cada um dos parâmetros no desempenho do algoritmo implementado (análise de sensibilidade paramétrica).

2 – Detalhamento

2.1 – O Problema da Caixa Preta

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.

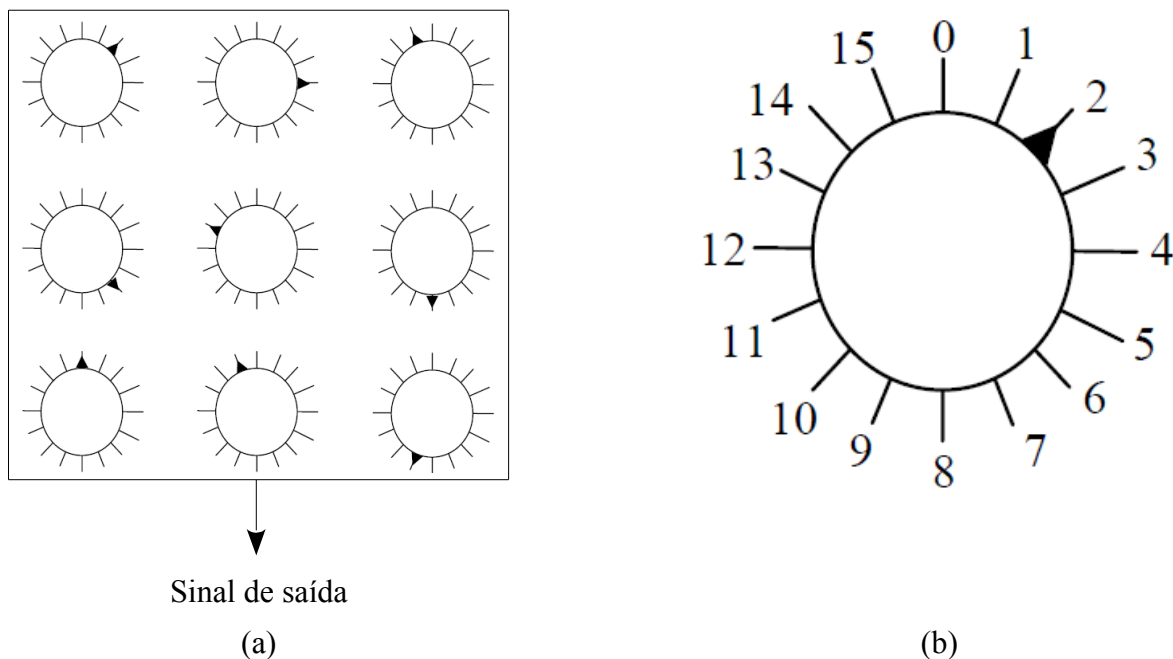


Figura 1: (a) Caixa Preta. (b) Cada botão da Caixa Preta pode ser colocado em uma de 16 posições. Cada combinação de posições gera um sinal de saída diferente.

2.2 – Codificação

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 1 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 1: Representações para cada uma das posições possíveis para os botões da Caixa Preta.

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 1, apresenta o mapeamento (arbitrário) entre uma posição de o respectivo valor de saída (b_i representa o bit 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 **1111101110011011111101111100101111**.

2.3 – Características do Algoritmo Genético

O Algoritmo Genético a ser implementado deve apresentar (**no mínimo**) as seguintes características:

- Possibilidade de utilização de elitismo (opção do usuário).
- Cruzamento com um ponto de corte ou uniforme (opção do usuário).
- Seleção por roleta ou por torneio de 2 indivíduos (opção do usuário).
- Mutação por escolha aleatória do *bit* ou *bit a bit* (escolha do usuário).
- Apresentar, ao fim da execução, um gráfico que apresente a evolução do *fitness* médio da população e a evolução do *fitness* do melhor indivíduo com o passar das gerações (opção do usuário: mostrar ou não).
- Possibilidade de configurar as probabilidades de cruzamento e mutação, o número de indivíduos da população e o número de gerações.

2.4 – Testes

Para os testes a serem realizados, uma execução será considerada como um sucesso quando conseguir encontrar a solução ótima apresentada na seção 2.2. Para cada configuração de parâmetros, deverão ser realizadas cem execuções do algoritmo e apresentados as seguintes informações, que serão utilizadas para a definição da melhor configuração de parâmetros:

- Número de sucessos.
- Maior valor de *fitness* (melhor caso).
- Menor valor de *fitness* (pior caso).
- Valor médio de *fitness*.
- Desvio padrão dos valores de *fitness*.
- Primeiro, segundo (mediana) e terceiro quartis dos valores de *fitness* (utilize *Box-plots* dos quando julgar pertinente).

Para cada um dos testes, você deverá verificar se os resultados são estatisticamente diferentes utilizando o Teste da Soma de Postos de Wilcoxon, um teste não paramétrico, tomando como referência a série de valores correspondente ao maior valor médio de *fitness*.

Primeiro Teste: Avaliação do Efeito do Tipo de Cruzamento

Considere a seguinte configuração de parâmetros:

- Sem elitismo.
- Seleção por roleta.
- Mutação bit a bit.
- Probabilidade de cruzamento: 0,8.
- Probabilidade de mutação: 0,025.
- Número de indivíduos da população: 30.
- Número de gerações: 50.

Utilize a configuração apresentada com o cruzamento com um ponto de corte e uniforme e comente os resultados.

Segundo Teste: Avaliação do Efeito do Tipo de Seleção

Considere a seguinte configuração de parâmetros:

- Sem elitismo.
- Cruzamento (escolher o método de acordo com os resultados do primeiro teste).
- Mutação bit a bit.
- Probabilidade de cruzamento: 0,8.
- Probabilidade de mutação: 0,025.

- Número de indivíduos da população: 30.
- Número de gerações: 50.

Utilize a configuração apresentada com seleção por roleta e por torneio e comente os resultados.

Terceiro Teste: Avaliação do Efeito do Tipo de Mutação

Considere a seguinte configuração de parâmetros:

- Sem elitismo.
- Cruzamento (escolher o método de acordo com os resultados do primeiro teste).
- Seleção (escolher o método de acordo com os resultados do segundo teste).
- Probabilidade de cruzamento: 0,8.
- Probabilidade de mutação: 0,025.
- Número de indivíduos da população: 30.
- Número de gerações: 50.

Utilize a configuração apresentada com mutação *bit a bit* e por escolha aleatória do *bit* e comente os resultados.

Quarto Teste: Avaliação do Efeito da Probabilidade de Cruzamento

Considere a seguinte configuração de parâmetros:

- Sem elitismo.
- Cruzamento (escolher o método de acordo com os resultados do primeiro teste).
- Seleção (escolher o método de acordo com os resultados do segundo teste).
- Mutação (escolher o método de acordo com os resultados do terceiro teste).
- Probabilidade de mutação: 0,025.
- Número de indivíduos da população: 30.
- Número de gerações: 50.

Utilize a configuração apresentada com probabilidades de cruzamento iguais a 0,2, 0,5 e 0,8 (perceba que essa configuração já foi utilizada no quarto teste) e comente os resultados.

Quinto Teste: Avaliação do Efeito da Probabilidade de Mutação

Considere a seguinte configuração de parâmetros:

- Sem elitismo.
- Cruzamento (escolher o método de acordo com os resultados do primeiro teste).
- Seleção (escolher o método de acordo com os resultados do segundo teste).
- Mutação (escolher o método de acordo com os resultados do terceiro teste).

- Probabilidade de cruzamento (escolher o valor de acordo com os resultados do quarto teste).
- Número de indivíduos da população: 30.
- Número de gerações: 50.

Utilize a configuração apresentada com probabilidades de mutação iguais a 0,025 (perceba que essa configuração já foi utilizada no quarto teste), 0,05, 0,1, 0,25 e 0,75 e comente os resultados.

Sexto Teste: Avaliação do Efeito do Elitismo

Considere a seguinte configuração de parâmetros:

- Cruzamento (escolher o método de acordo com os resultados do primeiro teste).
- Seleção (escolher o método de acordo com os resultados do segundo teste).
- Mutação (escolher o método de acordo com os resultados do terceiro teste).
- Probabilidade de cruzamento (escolher o valor de acordo com os resultados do quarto teste).
- Probabilidade de mutação (escolher o valor de acordo com os resultados do quinto teste).
- Número de indivíduos da população: 30.
- Número de gerações: 50.

Utilize a configuração apresentada sem e com o elitismo e comente os resultados.

Avaliação Final do Desempenho do Algoritmo

Considerando os resultados dos sete testes realizados, pode ser definida uma configuração “ótima” para o algoritmo. Utilizando essa configuração, faça um teste em que os números de gerações e de indivíduos são variados e compare e comente os resultados.

3 – Algumas Observações

3.1 – O trabalho deve ser feito **em equipes de no máximo três acadêmicos**: a discussão dos problemas e das estratégias de solução com os colegas é **permitida e aconselhável**. Todavia, a implementação e a documentação das soluções deve ser feita **apenas pelos membros da equipe**.

3.2 – Caso você(s) utilize(m) quaisquer fontes **externas** para elaborar as suas respostas, elas devem ser **citadas**: artigos ou livros, amigos ou colegas, informações encontradas na Internet, **qualquer coisa encontrada em qualquer lugar!** É melhor tentar solucionar os problemas, pois **solucionar problemas é um componente fundamental** para a nossa área de estudo. Não haverá penalidades no caso de utilização de ajuda externa, desde que devidamente citada, e desde que essa ajuda **não seja a cópia** do trabalho de um colega. **Utilizar o trabalho dos outros, como se fosse de você, é plágio, é desonestidade acadêmica.**

3.3 – O algoritmo pode ser implementado em **Matlab**, em **Python 2** (utilize(m) **Numpy para array computing** e **Matplotlib para gerar gráficos**) ou em **C/C++** (utilizar o **Code::Blocks** e linguagem padrão). A sua implementação deve possuir um **script geral de execução** do algoritmo

em que os **parâmetros** configuráveis possam ser **alterados**, no caso de utilização e Matlab ou Python ou um programa principal de teste no caso de utilização de C/C++.

Links:

1. *Code:Blocks*: <http://www.codeblocks.org>.
2. *Python(x, y)*: <https://code.google.com/p/pythonxy>.

3.4 – Uma **documentação escrita** deve ser elaborada, explicando os **objetivos** do trabalho, e apresentando **resultados** e **discussões**. Naturalmente, a documentação deverá apresentar uma **conclusão**, argumentando sobre as **dificuldades**, sobre o **aprendizado**, e sobre a **relevância** do trabalho no contexto do **curso** e da **disciplina**, e apontando eventuais **sugestões** para melhorias em futuras “edições”. **Todas as decisões de implementação deverão ser explicadas e comentadas na documentação.**

3.5 – Os códigos devem ser bem **comentados**, e os nomes de atributos, métodos, classes, funções, variáveis, etc, devem ser **consistentes** e **informativos**.

4 – Data de Entrega

O Trabalho Prático **deverá ser entregue até o dia 18/10/2013, por e-mail**: deverá ser enviado **um único arquivo compactado (7z, zip ou rar)**, com o assunto AG – TRABALHO PRÁTICO, para o endereço ag@renatomaia.net contendo o arquivo da **documentação** (envie no formato do editor de textos que você utilizar e também em formato pdf), além **dos arquivos com os códigos**.

5 – Valor

O Trabalho Prático vale 40 pontos.