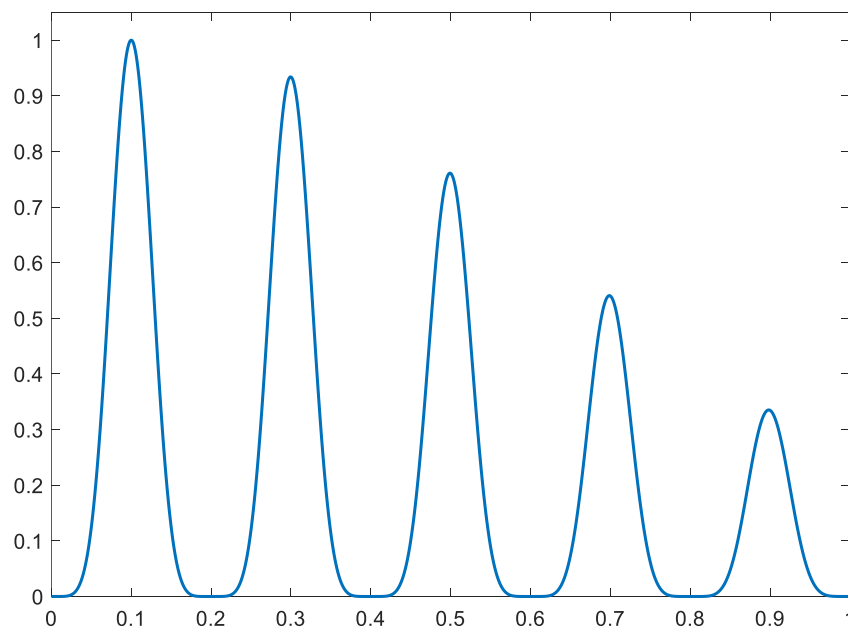


Universidade Estadual Paulista "Júlio de Mesquita Filho" (UNESP)
Programa de Pós-Graduação em Ciência da Computação
Disciplina: Computação Inspirada pela Natureza
Prof.: Fabricio Breve – Trabalho nº 1 – Data de Entrega: 18/04/2024

- 1) Implemente um Algoritmo Genético para o exemplo de reconhecimento de padrões apresentado em aula. Em vez de reconhecer o número 1 seu algoritmo deve reconhecer o número 0, representado pela *bitstring* [1 1 1 1 0 1 1 0 1 1 1]. Verifique quantas gerações são necessárias em média para atingir o alvo. Teste diferentes taxas de *crossover* e mutação e compare os resultados. Faça experimentos apenas com *crossover* e apenas com *mutação* e compare também os resultados.
- 2) Implemente um Algoritmo Genético para maximizar a função $g(x) = 2^{-2((x-0,1)/0,9)^2} (\sin(5\pi x))^6$, já utilizada nos exercícios feitos em aula, dentro do intervalo [0 1]. Utilize uma representação de *bitstring*, com precisão suficiente para pelo menos 3 casas decimais. Considere o alvo como desconhecido e verifique quais os maiores valores obtidos para $g(x)$. Compare os resultados obtidos com os resultados que você obteve com os algoritmos Subida da Colina e Recozimento Simulado aplicados a esta mesma função em relação aos valores obtidos para $g(x)$ e tempos de execução. Aproveite para explorar diferentes formas de seleção, como roleta, torneio e amostragem estocástica universal.



Dica: você também pode aplicar Subida da Colina e Recozimento Simulado em uma *bitstring*, utilizando uma perturbação semelhante ao operador de mutação dos algoritmos genéticos, com a vantagem de não ter de se preocupar com o domínio de x , visto que a própria representação binária dá conta disso.

- 3) Utilize um Algoritmo Genético para minimizar a seguinte função no intervalo contínuo $[-10 \quad +10]$:

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2$$

Considere o alvo como desconhecido e verifique quais os maiores valores obtidos para $g(x)$. Aproveite para testar diferentes tamanhos de população, sem elitismo e com elitismo com diferentes quantidades de indivíduos.

Para todos os exercícios, faça um relatório contendo os dados de todos os seus experimentos, configurações utilizadas, resultados obtidos e suas conclusões. Lembre-se de otimizar os hiperparâmetros dos algoritmos para obter os melhores resultados e de repetir os experimentos nas mesmas condições diversas vezes para obter média e desvio padrão, visto que os algoritmos são estocásticos. Você também pode usar as diferentes versões do algoritmo apresentadas (representação, seleção, etc.) para verificar qual se adequa melhor a cada problema.

Além de encontrar um indivíduo com boa aptidão, também é desejável que o algoritmo tenha uma convergência rápida. Portanto, registre também o número de iterações e tempo que os algoritmos demoram para convergir, dados os valores atribuídos aos hiperparâmetros.

Para cada experimento, inclua em seu relatório um ou mais gráficos que mostrem o valor mínimo e médio da função de aptidão ao longo das iterações.

Observações:

- 1) Qualquer linguagem de programação e ambiente de desenvolvimento podem ser utilizados. Porém o código-fonte deve ser devidamente comentado.
- 2) O relatório deve ser entregue em formato PDF, através de upload no Google Classroom. Deixe o PDF separado dos códigos (**não** comprima o PDF em um arquivo .zip ou similar).
- 3) Não é necessário anexar códigos-fontes no PDF. Os códigos-fontes podem ser postados em um ou mais arquivos separados no Google Classroom. Estes podem ser compactados (.zip ou similar).