

Project 2: Evolutionary Computing

Eduardo S. Ito RA 159086 / Thales E. Nazatto / Professora: Esther Luna Colombini / UNICAMP-IC

Problema / Questão

Seria possível preencher uma mochila de uma forma otimizada com vários itens, que tenham atributos peso e valor, de forma a não exceder sua capacidade máxima em peso e ao mesmo tempo maximizar em valor total os itens carregados? Isto é conhecido como o Problema da Mochila (ou Knapsack Problem)

Hipótese

Seria possível resolver esse dilema com computação evolucionária, mais especificamente utilizando algoritmo genético (GA), em vez da programação dinâmica, comumente utilizada para resolver essa questão? Seria possível fazer uma combinação otimizada dos itens a serem carregadas na mochila? Seria aplicável em outros casos, como otimizar carga em caminhões?

Materiais

Foram utilizados os seguintes materiais:

- Anaconda ou miniconda
- Pycharm. IDE para desenvolvimento dos scripts.
- Jupyter Notebook
- Python 3
- Aplicativo reverse para obtenção das classes em UML dos scripts python.

Procedimento

Os seguintes procedimentos foram realizados

- Revisão da aula 7 de evolutionary computing
- Discussão sobre qual tema abordar para GA. Por decisão conjunta, o tema escolhido foi o Knapsack problem.
- Revisão da literatura, exemplos da internet (e.g. Wikipedia, youtube, etc)
- Desenvolver scripts simples para entender o mecanismo do GA
- Mecanismo de fitness e critério de parada.
- Modelar o problema em classe UML.
- Evoluir os scripts para programação orientada à objeto (OOP)
- Eventualmente, aplicar o reverse para obtenção das classes dos scripts em formato UML.
- Generalizar a entrada dos parâmetros (taxa de reprodução, mutação, população inicial, estratégia de seleção, mutação, etc).
- Testes de casos de uso (mochila e cargo)

Dados / Observações

Os dados de entrada (parâmetros, população inicial, taxas de reprodução e mutação, etc) foram realizados a partir do Jupyter notebook e executados na nova biblioteca knapsack. Os dados de saída são dois. O file1.txt que contém dados para o gráfico e file2.txt que contém o log detalhado da execução dos scripts, para auditoria.

Resultados

A combinação de estratégias de seleção, mutação, taxa de seleção, população inicial, etc. podem levar à um término prematuro da execução, com um valor de fitness bastante pobre, mais detalhes no jupyter notebook

Conclusão

Foi possível resolver o problema da mochila e também do cargo de forma eficiente utilizando o algoritmo GA. O valor do fitness se estabiliza num patamar ótimo dependendo dos dados de entrada. Se há uma população inicial baixa ou extremamente alta, há uma grande tendência de uma parada prematura, isso ocorre também com cromossomos com baixo números de alelos.

Trabalhos Citados

Referências

https://pt.wikipedia.org/wiki/Problema_da_mochila
<https://www.youtube.com/watch?v=9kbzMeEBvUY>