PUC-Rio
Prof. Marcus Vinicius S. Poggi de Aragão
18 de novembro de 2015

Departamento de Informática
Período: 2015.2
Horário: 4as. 13-16 - Sala 512 RDC

OTIMIZAÇÃO COMBINATÓRIA (INF 2912)

Trabalho de Implementação

Descrição

A entrega do trabalho consiste de:

- OBRIGATÓRIO: Um e-mail para poggi@inf.puc-rio.br com ASSUNTO (ou SUBJECT) OC152T contendo os arquivos correspondentes ao trabalho. O NÃO ENVIO DESTE E-MAIL, COMO SOLICITADO, IMPLICA QUE O TRABALHO NÃO SERÁ CONSIDERADO.
- Um documento contendo o roteiro de desenvolvimento dos algoritmos (e dos códigos), os itens pedidos acima, comentários e análises sobre a implementação e os testes realizados (papel).
- A impressão dos trechos relevantes dos códigos fonte (papel).
- O trabalho pode ser feito em grupo de até 5 alunos.

Este trabalho prático consiste em desenvolver códigos para diferentes algoritmos e estruturas de dados para resolver os problemas descritos abaixo e, principalmente, analisar o desempenho das implementações destes algoritmos com respeito ao tempo de CPU. O desenvolvimento destes códigos e a análise devem seguir o seguinte roteiro:

- Descrever os algoritmos informalmente.
- Demonstrar o entendimento do algoritmo explicando, em detalhe, o resultado que o algoritmo deve obter e justificá-lo.
- Explicar a fundamentação do algoritmo e justificar a sua corretude. Apresentar e explicar a complexidade teórica esperada para cada algoritmo.
- Apresentar as tabelas dos tempos de execução obtidos pelos algoritmos sobre as instâncias testadas.
- Documente o arquivo contendo o código fonte de modo que cada passo do algoritmo esteja devidamente identificado e deixe claro como este passo é executado.
- Para a medida de tempo de CPU das execuções utilize as funções disponíveis no link correspondente na página do curso.

- Obrigatoriamente apresente tabelas contendo colunas para cada algoritmo aplicado às instâncias, uma com o tempo de CPU utilizado, outras com o valores do limite inferior e o limite superior obtidos, assim como uma com o número de nós na enumeração. Cada linha da tabela é associada a uma instância e contém a identificação da mesma. Nesta tabela coloque as instâncias em ordem crescente de tamanho.
- A corretude e a qualidade do código será testada sobre um conjunto de instâncias que será distribuido.

- **Descrição do Problema Original**

  Considere objetos descritos por c características binárias. Estes objetos podem vir de g grupos diferentes. Um grupo é descrito pelas probabilidades ter cada uma das c caraterísticas. Uma possível definição de um grupo é que este tenha c_y características com probabilidade p ($p > 0.5$) de ocorrer, c_n com probabilidade $1-p$ e as demais com ocorrência indiferente ($p = 0.5$). Sugere-se que os conjuntos das c_y características de cada grupo formem conjuntos com interseção par a par vazia.

  Dado que n objetos são gerados conforme descrito acima, propor um classificador que indique o grupo de cada objeto.

  A abordagem deve considerar entre n_{min} e n_{max} objetos gerados a partir de cada um dos g grupos, determinando o conjunto para aprendizado. Em seguida, gera-se e clasifica-se n objetos gerados sorteando com probabilidades iguais o grupo de origem e as características do objeto de acordo com o seu grupo.

  Propõe-se executar os testes assumindo o conhecimento de g e sem conhecer g, quando um algoritmo para propor g deve ser testado.

  Um conjunto de instâncias será disponibilizado.

- **Métodos**

  O trabalho consiste na implementação e teste de 4 métodos.

  1. **K-Means**

     Consiste em executar o algoritmo *K-means* determinar os pontos *centrais* de cada grupo e classificar cada objeto como sendo do grupo com ponto central *mais próximo*

  2. **ULP - Problema de Localização sem Capacidade**

     Consiste em resolver o *ULP* determinar os objetos representantes de cada grupo e classificar cada objeto como sendo do grupo com representante *mais próximo*

  3. **P-Center - Problema de Localização de Centróides**

     Consiste em resolver o *P-Center* determinar os objetos representantes de cada grupo e classificar cada objeto como sendo do grupo com representante *mais próximo*

  4. **Propor um método alternativo**