PROGRAMA INSTITUCIONAL DE BOLSAS DE INICIAÇÃO CIENTÍFICA

PIBIC/CNPq - INPE

RELATÓRIO DE BOLSA DE INICIAÇÃO CIENTÍFICA

PROJETO:

ESTUDO E IMPLEMENTAÇÃO DE ALGORITMOS PARA RESOLUÇÃO DE PROBLEMAS COMBINATÓRIOS EM GRAFOS

BOLSISTA:

GRAZIELE DE OLIVEIRA VICTOR

ORIENTADOR:

DR. HORÁCIO HIDEKI YANASSE

PERÍODO:

08/2001- 01/2002

Sumário

Resumo

Quando analisamos um conjunto de elementos de dados (não necessariamente dados computacionais), podemos estar preocupados com o seu conteúdo ou com as relações existentes entre eles. A Teoria dos Grafos preocupa-se com o estudo das relações existentes entre diversos objetos de análise, podendo ser utilizada em qualquer área que necessite de organização de dados: sociologia, pesquisa operacional, química, etc.

A área de grafos trata de abstrações de certos problemas práticos encontrados na indústria (projetos de chips VLSI, administração de frotas de veículos, sistemas operacionais de computadores, etc.) e em outras áreas da matemática e da pesquisa operacional. Muitos destes problemas podem ser descritos sobre um grafo. Os problemas consistem em encontrar uma configuração ótima (máxima ou mínima, conforme o caso) de um certo tipo no grafo. Para resolução de tais problemas foi escolhida a linguagem de programação C++, por utilizar o paradigma da orientação por objetos.

Apresentaremos nesse relatório uma descrição do que foi realizado no período de 08/2001 a 01/2002

Introdução

Neste projeto tem-se como objetivo o estudo e implementação de algoritmos sugeridos na literatura para resolução de problemas de fluxo máximo, cobertura de arcos e eventualmente , localização de facilidades. Pretende-se realizar testes computacionais para análise de desempenho e comparação dos algoritmos propostos.

Para a consecução deste trabalho está sendo utilizada a linguagem de programação C++, escolhida por utilizar o paradigma da orientação por objetos , ser uma linguagem de médio nível, visivelmente mais elaborada e compreensível, ser uma linguagem estruturada permitindo a declaração de funções e procedimentos facilitando a manutenção do programa por completo, ter a característica de portabilidade, a sua compartimentalização do programa, sua velocidade e portabilidade entre outras características.

Objetivos

Como a implementação desses algoritmos está sendo realizada, tais algoritmos implementados deverão fazer parte de uma biblioteca de algoritmos que se pretende utilizar no Laboratório de Computação e Matemática Aplicada.

Procedimento

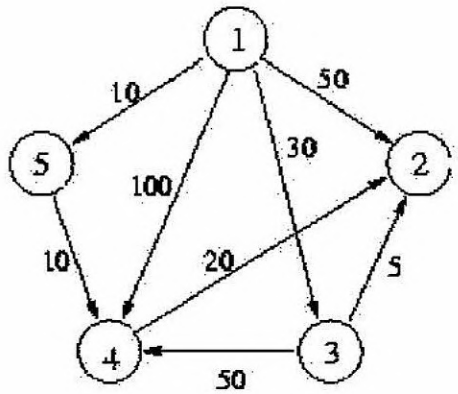
No período de 08/2001 a 01/2002 realizou-se um estudo mais detalhado e um tanto demorado da área de programação e teoria de grafos visto que era totalmente leiga na área de programação por estar cursando Licenciatura de Matemática, foram feitas análises e testes de mesa, para a familiarização do problema proposto, definindo estruturas e processos que serão utilizados na implementação dos algoritmos.

Os algoritmos implementados estão sugeridos na literatura, e foram escolhidos por serem supostamente os melhores em termos de desempenho computacional, e suas estruturas matemáticas serem mostradas a seguir para se ter um melhor entendimento do que foi implementado. São elas:

Utilizei o algoritmo de Dijkstra que identifica, a partir de um vértice do grafo, qual é o custo mínimo entre esse vértice e todos os outros do grafo. No início, o conjunto S contém somente esse vértice, chamado origem. A cada passo, selecionamos no conjunto de vértices sobrando, o que é o mais perto da origem. Depois atualizamos, para cada vértice sobrando, a sua distância em relação à origem. Se passando pelo novo vértice acrescentado, a distância fica menor, é essa nova distância que será memorizada.

Suponhamos que o grafo é representado por uma matriz de adjacência onde temos o valor se não existe aresta entre dois vértices. Suponhamos também que os vértices do grafo são enumerados de 1 até n, isto é, o conjunto de vértices é N = {1, 2, ..., n}. Utilizaremos também um vetor D[2..n] que conterá a distância que separa todo vértice do vértice 1 (o vertíce do grafo que é o vertice 1 é escolhido arbitrariamente). Eis o algoritmo:

```
função Dijkstra(L = [1..n, 1..n]: grafo): vetor[2..n]
    C := {2,3,...,n}  {Implicitamente S = N - C}
    Para i := 2 até n:
        D[i] :=  L[1,i]
    Repetir n-2 vezes:
        v :=  Elemento de C que minimiza D[v]
        C :=  C - {v}
        Para cada elemento w de C:
            D[w] :=  min(D[w],D[v]+ L[v,w])
    Retornar D
```



Para entender melhor o algoritmo, considere o grafo direcionado ilustrado na figura 1.

As etapas da execução do algoritmo são as seguintes:

Passo

v

C

D Início

{2,3,4,5}

[50,30,100,10]

1

5

{2,3,4}

[50,30,20,10]

2

4

{2,3}

[40,30,20,10]

3

3

{2}

[35,30,20,10]

Da maneira que o algoritmo foi apresentado, obtemos o custo do caminho mínimo para qualquer vértices, mas não obtemos o
caminho mínimo. Para identificar esse caminho mínimo, é só acrescentar mais um vetor P[2..n], onde P[v] indica o vértice
que precede v no caminho mais curto. A modificação do algoritmo para obter esse vetor é simples. Primeiro devemos inicializar esse vetor. Segundo, no mesmo momento que o vetor D é atualizado, atualizamos também o vetor P. O algoritmo modificado é o seguinte:

```
função Dijkstra(L = [1..n, 1..n]: grafo): vetor[2..n]
    C := {2,3,...,n}
    Para i := 2 até n:
        D[i] :=  L[1,i]
        P[i] :=  1
    Repetir n-2 vezes:
        v :=  Elemento de C que minimiza D[v]
        C :=  C - {v}
Para cada elemento w de C:
```

Se D[w] > D[v]+ L[v,w] então
D[w] := D[v]+ L[v,w]
P[w] := v
Retornar P
[50,30,100,10]
[1,1,1,1]
1
5
{2,3,4}
[50,30,20,10]
[1,1,5,1]
2
Com esse algoritmo modificado, as etapes da execução são as seguintes:

Passo
v
C
D
P
Início
-
{2,3,4,5}
4
{2,3}
[40,30,20,10]
[4,1,5,1]
3
3
{2}
[35,30,20,10]
[3,1,5,1]

Vemos que o estado final do vetor P é [3,1,5,1]. Para saber qual é o caminho mais curto entre os vértices 1 e 2, procuramos o valor na posiçao 2 desse vetor (não esqueça que P e D são indexados a partir de 2). O vetor indica que o último vértice antes do vértice 2 é o vértice 3. Repetimos de novo o mesmo processo para ver o caminho mais curto entre 1 e 3. No vetor, na posição 3, temos o valor 1, que é a origem. Então, o caminho mais curto é 1,3,2.

Discussões e conclusões

Com este estudo realizado pude obter um conhecimento básico nessa área , utilizando-o para implementação de algoritmos da bibliografia recomendada, além de poder aperfeiçoar-me e aplica-lo futuramente com maior complexidade.
Infelizmente não realizei grandes progressos pois tive muita dificuldade para aprender a programar na linguagem C++ e conhecer a teoria de grafos, pois como já foi dito estou fazendo graduação em Matemática, por isso acabei realizando quase que o perído todo um estudo mais aprofundado.

Perspectivas e trabalhos futuros

Espera- se estudar e aumentar ainda mais o meu conhecimento a ser utilizado no projeto, visto que ainda sinto muita dificuldade na área de programação, principalmente da que utiliza o paradigma da orientação ao objeto, para assim poder melhor implementar os algoritmos, obtendo os resultados esperados.
Os algoritmos implementados deverão fazer parte de um banco de algoritmos implementados que poderão ser acessados futuramente por usuários interessados (como já foi citado em relatório anterior). Este danço de dados deverá ser integrado a um Banco de Dados Geográfico, servindo como uma ferramenta adicional.

Agradecimentos

Agradeço ao CNPq pela bolsa PIBIC concedida e ao Laboratório Associado de Computação e Matemática Aplicada (LAC)- INPE pelas ótimas condições de infra-estrutura oferecidas para a realização deste trabalho, e ao meu orientador pela paciência que durante todo o projeto possuiu comigo e pela grande oportunidade de trabalhar neste projeto.

Referência

- Cormen, Leiserson, Rivest – Introduction to Algorithms. McGraw-Hill Book Company, 1989.
- Jungnickel,Dieter Jungnickel – Graphs, Networks and Algorithms. Springer, 1999.
- Ellis, Stroustrup – C++ Manual de Referência Comentado. Editora Campus,1991 .

Forbellone, Eberspacher – Lógica de Programação. McGraw-Hill Ltda,1993.

- Larson, Odoni – Urban Operations Research, Prentice Hall, 1981.
- Lucchesi – Introdução a Teoria dos Grafos. Instituto de Matemática Pura e Aplicada, 1979.
- Tenenbaum, Langsam, Augenstein – Estruturas de Dados usando C, Prentice Hall,1989.
- htttp:/www.ime.usp.br/~pf/mac338/1999.htm
- http:/www.cos.ufrj.br/~grafos/
- http://dmoz.org/computer/programing/language
- http:/www.inf.ufpr.br/~michel/Disciplinas/Bac/Grafos/Intro/intro.html#def-basicas
- http://www.ime.usp.br/~pf/mac328/aulas/grafossgb.html
- http://www.cos.ufrj.br/~grafos/DGP97.htm
- http://www.dcc.ufjf.br/emen/grafos.htm
- http://www.coordinf.ufjf.br/grafos.htm

- http://www.mat.puc-rio.br/~nicolau/olimp/obm-l.200002/msg00027.html
- http://www.inf.ufsc.br/grafos/temas/euleriano/euleriano.htm
- http://www.dcc.pucminas.br/computacao/disciplinas/atp4/semestre1_2000/Seminarios/grafos.html