

MC658 - PROJETO E ANÁLISE DE ALGORITMOS III

Prof. Flávio Keidi Miyazawa / PED: Gustavo

Laboratório 4 - 2o. Semestre de 2015

Laboratório 4: Transmissão múltipla de dados

Uma motivação: Suponha que a empresa Skype¹ te contrata para resolver o seguinte problema: Dado um grafo $G = (V, E)$ com as seguintes funções nas arestas:

- função de latência $\ell : E \rightarrow \mathbb{R}^+$, que representa o tempo para um dado ser transmitido pela aresta, em milissegundos.
- Função de capacidade da largura de transmissão, $d : E \rightarrow \mathbb{R}^+$.
- Função de custo para a transmissão por byte transmitido na aresta, dado por $c : E \rightarrow \mathbb{R}^+$.

Além disso, você dispõe de k pares de vértices $P = \{(s_1, t_1), (s_2, t_2), \dots, (s_k, t_k)\}$ e para cada par (s_i, t_i) , há uma quantidade de dados q_i (largura dos dados do par i , mesma largura durante toda a transmissão) que deve ser transmitida de s_i para t_i e um tempo máximo T_i (soma dos tempos de latência de todas as arestas do caminho) para fazer a transmissão. Cada par (s_i, t_i) pode representar uma conversa online entre uma pessoa s_i e uma pessoa t_i (ou de uma maneira macro, todas as conversas juntas de pessoas de uma cidade s_i com pessoas da cidade s_j).

Para garantir uma qualidade de serviço mínima, os dados que trafegam pela rede, de s_i até t_i não podem demorar mais que T_i milissegundos e com isso, pode haver necessidade de se pagar por conexões mais rápidas, porém mais caras.

Para que a conversa de s_i com t_i não fique com problemas, toda a transmissão dos q_i bytes de s_i para t_i deve ser feita por um mesmo caminho (i.e., não se pode separar os frames do vídeo nem separar dados de vídeo e de voz). A capacidade de cada cabo (largura do cabo/aresta) limita a quantidade total das conversas que trafegam por segundo naquele cabo (soma das larguras das transmissões dos pares que usam o cabo é no máximo sua capacidade).

O custo c_e de transmissão e de latência ℓ_e do cabo $e = u, v$ pode ser ditado pelas diferentes tecnologias que o provedor do serviço oferece para

¹No geral, empresas que utilizam tecnologia P2P (e a utilização de seus aplicativos) que fazem grande quantidade de transmissões par a par, como Whatsapp, Google/Youtube,...

isso (alguns podem ter instalado cabos físicos ligando u até v , outros por antenas wireless, outros fazem via satélites, etc).

O tempo de transmissão por segundo, T_i pode ser ditado pelo tipo dos seus usuários, algumas são companhias que pagam mais caro para ter mais qualidade na transmissão (seja dados, telefonia, VoIP, dados, emails,...). Outros usuários usam o serviço gratuito e menor prioridade na qualidade de serviço.

Seu objetivo é obter uma transmissão dos dados² para cada par com custo total mínimo e respeitando as restrições impostas pela qualidade de serviço e capacidade das arestas (conexões).

Sobre a Resolução:

1. Sua implementação deve ser baseada em formulação de programação linear inteira e utilização de solver escolhido na disciplina. Para isso, você pode utilizar as bibliotecas fornecidas pelo professor para resolver problemas envolvendo conectividade, ou utilizar formulações baseadas em fluxo com modelos mais compactos (pode ser interessante fazer as duas e comparar os modelos).
2. Você receberá um programa (a ser divulgado em breve) que já lê uma entrada para o problema e imprime o valor da solução que sua rotina produzir. A assinatura da rotina que você irá desenvolver, tem a seguinte forma:

```
bool transmissoes(Problem_Data &G, long maxtime)
```

3. O parâmetro `maxtime` é o tempo máximo de execução da sua rotina em milissegundos. O parâmetro `G` é uma classe (como o `TSP_Data`, do arquivo `ex_tsp_gurobi.cpp`) e é formado pela seguinte declaração:

```
class Problem_Data {
public:
    Graph_Data(ListGraph &g,
                NodeName &nodenname,
                NodePos &posx,
                NodePos &posy,
                vector<Node> &s,
                vector<Node> &t,
```

²Note que apesar de parecer um problema de fluxo/transporte, neste problema não se pode dividir os dados de um mesmo par nem enviar um dado de uma fonte s_i para um outro destino t_j que não seja o correspondente.

```

        NodeWeight &tmax,
        NodeWeight &q;
        EdgeWeight &custo,
        EdgeWeight &capacidade,
        EdgeWeight &latencia);
ListGraph &g;
int NNodes,NEdges,NPairs; // número de nós, arestas e pares
NodeName &nodename;      // nome dos vértices
NodePos &posx;            // (posx,posy) são as coordenadas
NodePos &posy;            // de cada vértice
vector<Node> &s,           // s[i] é o nó de origem do par i
vector<Node> &t,           // t[i] é o nó de destino do par i
NodeWeight &Tmax;         // latência máxima do caminho s[i]--t[i]
NodeWeight &q;            // quantidade de dados por ms
EdgeWeight &custo;        // custo para transmitir um byte na aresta
EdgeWeight &capacidade;   // capacidade de transmissão na aresta por ms
EdgeWeight &latencia;     // latência para transmitir pela aresta
vector<vector<Node> > BestSol; // matriz que representa melhor solução
                                //BestSol[i][0],BestSol[i][1],BestSol[i][2],...
                                //é a sequência de nós do caminho s[i]---t[i]

double BestVal;            // valor da melhor solução encontrada
double BestLB;            // Se obteve solucao ótima, BestVal==BestLB,
                                // senão, BestLB é um limitante inferior
};

```

A entrega do laboratório está dividida em duas partes:

1. Você deve entregar um texto contendo: *(i)* Formulação em programação linear inteira que você utilizou, e uma explicação resumida das variáveis e restrições. *(ii)* Descrição do algoritmo que você implementou, em alto nível, para que uma pessoa possa entender facilmente as idéias e estruturas de dados utilizadas no algoritmo. *(iii)* Testes computacionais e gráficos para as entradas disponibilizadas, de maneira que possamos entender o impacto das suas idéias a medida que o tamanho das entradas aumentam.
2. Você deve entregar seu código bem documentado e modularizado de maneira que fique fácil para ser entendido.

Observação: Qualquer tentativa de fraude implicará em *aproveitamento zero na disciplina para todos os envolvidos*.