PROFESSORES: CID C. DE SOUZA E ZANONI DIAS (SALA 8 - IC1) PRIMEIRO SEMESTRE DE 2005

TRABALHO PRÁTICO MC538/MC548 – PROJETO E ANÁLISE DE ALGORITMOS II

Introdução

Neste trabalho você deverá modelar alguns problemas usando Programação Linear Inteira (*PLI*) e resolvê-los utilizando o `Xpress`, um dos melhores resolvedores comerciais de programação matemática disponíveis no mercado atualmente. Para poder ter acesso à uma versão estudantil do *software*, a qual **só roda no Windows (!)**, siga os seguintes passos:

1. Vá ao *site* `www.dashoptimization.com`, entre no menu *Education* e escolha a opção *Free Student Edition.*

2. Siga as instruções para baixar e instalar esta versão gratuita do resolvedor de programação linear na sua máquina.

   Note que esta versão estudantil do `XPRESS` já está instalada em alguns dos laboratórios do IC-3. Assim, você pode optar por fazer o trabalho no IC.

3. Leia o manual `Xpress-MP Essentials` dando atenção especial para os capítulos `Xpress-IVE`, `Modeling with Xpress-MP` e `Further Mosel Topics`. Este manual pode ser obtido em: `http://www.ic.unicamp.br/~cid/XPRESS/docs/essentials.pdf`

4. O manual citado no item acima mais o exemplo do item seguinte devem ser suficientes para você conseguir fazer o trabalho. Contudo, considere a possibilidade de ler também algumas partes do manual `Xpress-Mosel User's Guide` disponibilizado em `http://www.ic.unicamp.br/~cid/XPRESS/docs/moselug.pdf`

5. Execute o exemplo contendo o modelo PLI para o problema do conjunto independente que está disponibilizado em `www.ic.unicamp.br/~cid/MC548/200501/Trabalho/independente.mos` usando como entrada os arquivos de teste (`independente1.dat` e `independente2.dat`) neste mesmo diretório. Veja o código deste exemplo ! Note que para executar uma instância cujos dados estão num arquivo *exemplo.dat*, você deve passar este mesmo nome como resposta da pergunta feita pelo programa.

   Este mecanismo de entrada do nome do arquivo contendo os dados da instância a ser resolvida deverá ser implementado nos programas feitos neste trabalho.

ATENÇÃO: o trabalho deve ser feito individualmente ou em grupos de no máximo 2 (dois) integrantes !

Abaixo estão enunciados os problemas que fazem parte do trabalho. Para modelá-los, utilize a linguagem `Mosel` do `Xpress`. Os seus códigos devem ser testados com a versão gratuita do `Xpress-MP` disponível no site da `Dash Optimization` conforme discutido acima. A versão final do trabalho deverá ser entregue por *e-mail* para o docente (`(cid,zanoni)@ic.unicamp.br`, conforme a turma) até o dia 01/07/2005 às 24:00 PM. Como `subject` da mensagem coloque *Trabalho de MC548: raXXXXXX/raYYYYYY*, onde `raXXXXXX` e `raYYYYYY` são os RA's dos integrantes do grupo (se só houver um, omita o segundo RA do subject da mensagem). **Os alunos cujos trabalhos não forem recebidos na máquina do docente até o horário fixado acima receberão nota *zero !***

Entregue os códigos dos programas Mosel referentes ao modelo de cada um dos problemas como anexos (*attachments*) da sua mensagem. O nome do arquivo correspondente ao i-ésima questão deverá ter a forma raXXXXXX-i.mos onde XXXXXX corresponde ao número do RA de um dos elementos da dupla. Se a questão estiver dividida em itens, o arquivo referente ao item j da i-ésima questão terá o nome RAXXXXXX-i-j.mos. Por exemplo, a formulação do problema da questão 3 item a para um aluno com RA 123456 estará anexada na mensagem no arquivo denominado ra123456-3-a.mos.

Na mesma mensagem ao docente deverá ser anexado ainda um arquivo texto de no máximo 8 páginas no formato *pdf*, html ou *ps* contendo, para cada um dos problemas propostos, as seguintes informações:

1. descrição das variáveis do modelo;
2. descrição das restrições (significado das mesmas);
3. descrição da função objetivo

Coloque o nome e o RA de todos integrantes do grupo em um cabeçalho na primeira página do texto.

Note que todos os problemas deverão ser resolvidos de forma inteira (restrições is_binary, is_integer sobre variáveis mpvar).

Os códigos Mosel corresponderão a 50% da nota e o texto corresponderá aos outros 50%. No texto será avaliado a corretude das formulações propostas, a apresentação e a organização. O código Mosel deverá ser bem comentado, isto é, os seus comentários devem ser curtos e precisos. A ausência de comentários no código será penalizada.

Observação: Um programa Mosel que tenha sido entregue e que dê erro ao ser executado pelo docente terá nota *zero*. Portanto, preste **muita atenção** para os exemplos de entrada de cada problema. Mantenha as variáveis que vão receber os dados do arquivo com o mesmo nome das variáveis nos arquivos de entrada dados como exemplo (atenção para letras maiúsculas e minúsculas). Caso você não faça isto, seu programa não conseguirá executar com as entradas de teste preparadas pelos docentes e, por conseqüência, a nota do trabalho prático será *zero*.

Assim como no exemplo do conjunto independente, mantenha uma variável chamada **nomearq** que será lida no início do programa e que especifica o nome do arquivo de entrada.

Você pode baixar arquivos exemplos de teste para cada um dos problemas no endereço

www.ic.unicamp.br/~cid/MC548/200501/Trabalho/.

Neste endereço existem 2 arquivos de teste para cada uma das questões. Os nomes dos arquivos são teste-i<-j>-1.dat e teste-i<-j>-2.dat, onde i é o número da questão e j um item da mesma, quando aplicável.

Problemas

1. Dado um grafo não-orientado $G = (V, E)$ sem auto-laços, com $n = |V|$, seja U um subconjunto de V. O *corte* de U, denotado por $\delta(U)$, é o conjunto de todas as arestas (i, j) onde $i \in U$

e $j \notin U$. Se $|U| = \lfloor \frac{|V|}{2} \rfloor$ (U tem cerca da metade do número de vértices do grafo), $\delta(U)$ é chamado de um *equicorte* de G. O ***problema do equicorte mínimo*** recebe como entrada um grafo G como descrito acima e deve retornar o tamanho do menor equicorte de G. Ou seja, deve ser encontrado um conjunto $U^\star$ tal que $|\delta(U^\star)| \leq |\delta(U)|$ para todo U satisfazendo $|U| = \lfloor \frac{|V|}{2} \rfloor$. Este problema é $\mathcal{NP}$-difícil.

Formule-o como um PLI usando o `Mosel`. O seu programa deve imprimir o número de arestas do equicorte mínimo bem como os vértices do conjunto $U^\star$ e as arestas presentes em $\delta(U^\star)$.

2. Uma cidade está tentando resolver o problema anual da coleta de lixo de modo a minimizar os seus custos operacionais. A cidade está dividida em n setores e existem m locais potenciais para instalação dos depósitos de lixo. O custo de instalação de um depósito no local j, $j \in \{1, \ldots, m\}$, é dado por `custo_fixo`(j) reais. Os demais custos decorrem do transporte do lixo entre os setores e os depósitos aos quais estão alocados. As leis ambientais determinam que cada setor deve estar alocado a um **único** depósito. Para cálculo do custo do transporte entre um setor i e um depósito j, usa-se como estimativa de distância entre eles a distância entre o centro geográfico do setor e o local do depósito. Este valor é dado por `distancia`(i, j). O custo por kilômetro e por metro cúbico de lixo transportado independe do percurso e tem valor igual a `custo_km` reais. Cada depósito j tem uma capacidade anual de armazenamento de lixo dada em milhares de metros cúbicos pelo valor `capacidade`(j). Por sua vez, cada setor i tem sua capacidade anual de produção de lixo dada por `req`(i) milhares de metros cúbicos.

   Formule o problema com o um PLI usando o `Mosel`. O seu programa deve imprimir o custo total de operação da solução ótima, bem como a relação de depósitos a serem instalados e a alocação dos setores aos depósitos.

3. Um grande grupo do setor de alimentação possui 2 cadeias de restaurantes. Na primeira cadeia, os restaurantes são do tipo *fast food*, com um cardápio não muito sofisticado. A segunda cadeia é composta por restaurantes com *menu à la carte*, portanto, mais requintados. O grupo está chegando em uma cidade onde pretende instalar seus primeiros restaurantes de ambas as cadeias. O departamento de *marketing* do grupo já fez um estudo de mercado na cidade e identificou n possíveis locais para instalação de restaurantes. Cada um destes locais é adequado para receber restaurante de uma ou de outra cadeia mas, obviamente, só um restaurante pode ser instalado em cada local. Além disso, para um local i e uma cadeia j, chegou-se a conclusão que o lucro mensal esperado com a instalação de um restaurante da cadeia j no local i é de `lucro`(i, j) reais.

   Por outro lado, o pessoal de *marketing* também constatou que não é lucrativo instalar dois restaurantes **da mesma cadeia** em locais distantes menos de 5 km um do outro. Por isso, foi criada uma tabela $n \times n$ de nome `Menos5` de modo que `Menos5`(i, j) $= 1$ se os locais i e j distam menos de 5 km, caso contrário, `Menos5`(i, j) $= 0$.

   É claro que o objetivo do grupo é decidir aonde e que tipo restaurante instalar de modo a maximizar seu lucro. O grupo considerou então duas estratégias:

   (a) não instalar restaurantes da mesma cadeia em locais distantes menos de 5 km (restaurantes de cadeias diferentes são permitidos).

(b) se instalar um restaurante de uma cadeia num local então **obrigatoriamente** irá instalar um restaurante da outra cadeia a uma distância não superior a 5 km, mantida a restrição da estratégia anterior.

Resolva o problema para as duas estratégias formulando-os com PLI e usando o `Mosel`. Lembre-se que os programas `Mosel` referentes a cada estratégia devem ser colocados em arquivos diferentes: `raXXXXXX-3-`a`.mos` para a estratégia 1 e `raXXXXXX-3-`b`.mos` para a estratégia 2. Em cada um dos casos, o programa deve imprimir o valor do lucro mensal auferido pelo grupo e a relação dos locais onde serão instalados restaurantes da cadeias 1 e 2, respectivamente.

4. Uma pequena distribuidora de combustíveis possui um único caminhão-tanque para fazer entregas aos seus clientes. Este caminhão é compartimentado em n tanques menores de capacidades dadas por `capacidade`(j) litros, para $j \in \{1, \ldots, n\}$.

   Suponha que a distribuidora recebeu encomendas de clientes para m tipos de combustíveis diferentes nas quantidades `demanda`(i) litros para cada $i \in \{1, \ldots, m\}$.

   Infelizmente, a quantidade de combustível toda demandada supera a capacidade de carga do caminhão. O problema então é decidir quais tipos de combustível levar, em que quantidades e em qual compartimento do caminhão de modo a minimizar a quantidade total de combustível que deixa de ser transportado.

   Obviamente, em um mesmo compartimento do caminhão não podem ser misturados dois ou mais combustíveis diferentes, o que caracterizaria adulteração dos produtos.

   Formule e resolva este problema como um PLI usando o `Mosel`. O seu programa deve imprimir qual a quantidade total de combustível que deixou de ser transportada bem como os tipos e quantidades de cada produto carregados em cada um dos compartimentos do caminhão.