Trabalho Prático 2

Este trabalho será a continuação do primeiro trabalho.

Como será a entrega?

Existem 5 tarefas no Moodle, uma para cada um dos problemas descritos na prova. Os alunos deverão enviar **UM APENAS UM ARQUIVO .jl** em cada tarefa. A submissão de qualquer outro formato de arquivo ou de mais de um arquivo implicará em **ZERO**. O arquivo também deve ter uma nomenclatura especifica. Para cada tarefa, o arquivo submetido pelo aluno de matricula xxxxxx deve se chamar `tp1_xxxxxx.jl`.

O que faremos?

No primeiro trabalho os alunos foram convidados a utilizar resolvedores para modelar e resolver cinco problemas propostos. Neste segundo trabalho, os alunos serão convidados a solucionar os mesmos problemas com instâncias de um tamanho mais expressivo para avaliar as limitações dos resolvedores e o uso das técnicas de programação inteira.

Os arquivos enviados pelos alunos para cada problema serão testados para 3 instâncias de testes (que são semelhantes as instâncias de testes que foram disponibilidades aos alunos, mas não são iguais). Os arquivos serão testados com a seguinte linha de comando:

`julia tp1_xxxxxx.jl <instancia>`

Onde `instancia` é o caminho para o arquivo que será testado. O programa deve então executar e imprimir na **TELA**:

`TP1 xxxxxx = <valor>`
`<certificado>`

Onde xxxxxx é o número de matrícula do aluno, `<valor>` é o valor da solução ótima obtida pelo método e `<certificado>` é um certificado de que a solução está correta (será especificado para cada problema como deve ser o seu certificado).

Cada execução será limitada a 2 minutos, tome os devidos cuidados para que seu código execute dentro do tempo limite. Observe que não é razoável exigir que os programas **resolvam exatamente** as instâncias para esses problemas em tempo tão restrito. Assim, os alunos não serão obrigados a dar soluções exatas, mas sim soluções heurísticas (aproximadas).

Os alunos terão acesso a 3 instâncias de testes para realizar validações em seus códigos. Os arquivos estão disponíveis no Moodle e as soluções heurísticas, assim como a formatação e interpretação das instâncias de testes estão indicadas abaixo para cada problema:

Quais são os problemas?

Empacotamento

Considere um conjunto de objetos $O = \{o_1, o_2, \ldots, o_n\}$ cada objeto com um peso w_i. Dispomos de um várias caixas de papel, cada uma delas com o limite de peso 20Kg. Desejamos empacotar nossos objetos, utilizando o menor número de caixas possíveis, dado que em nenhuma caixa o valor da soma dos pesos dos objetos ultrapasse seu limite de peso.

0.0.1 Formatação da Entrada

A primeira linha do arquivo é no formato

`n <num>`

onde `<num>` é um inteiro representando o número de objetos. As demais linhas são no formato

`o <id> <peso>`

onde `<id>` é um inteiro para representar o objeto e `<peso>` é um ponto flutuante que representa o peso do objeto em Kg.

Observe que o separador é uma tabulação e não um espaço.

Certificado: Os objetos de cada caixa, separados por tabulação com uma caixa em cada linha.

Conjunto Independente

Dado um grafo $G = (V, E)$, um *conjunto independente* é um conjunto de vértices dois a dois não adjacentes, ou seja, sem arestas entre eles. Desejamos determinar um conjunto independente de cardinalidade máxima (maior tamanho em número de vértices).

0.0.2 Formatação da Entrada

A primeira linha do arquivo é no formato

`n <num>`

onde `<num>` é um inteiro representando o número de vértices no grafo. Assumimos que os vértices do grafo estão nomeados de 1 até `<num>`. As demais linhas são no formato

`e <v> <u>`

onde `<v>` e `<u>` são números inteiros representando vértices e essa linha no arquivo significa que temos uma aresta entre os vértices v e u.

Observe que o separador é uma tabulação e não um espaço.

Certificado: Os vértices do conjunto independente em ordem crescente separados por tabulação

Lotsizing com Backlog

Estamos auxiliando um produtor a planejar sua produção. Essa produtor quer que planejemos sua produção para um horizonte de tempo com n períodos. O produtor produz um único produto, conhece as demandas dos clientes para cada período de tempo i (d_i), o custo

de produzir uma unidade do produto no tempo $i(c_i)$ e o custo de armazenar uma unidade do tempo i para o tempo $i+1(h_i)$. Entretanto, devido a sazonalidade de seu produto, pode ser que os pedidos dos clientes não sejam satisfeitos em um período, esse caso, podemos entregar o produto atrasado para o cliente, mas pagamos uma multa de p_i por unidade de produto pedida pelo cliente e ainda não entregue no período i.

0.0.3 Formatação da Entrada

A primeira linha do arquivo é no formato

`n <num>`

onde `<num>` é um inteiro que representa o tamanho do horizonte de planejamento. As demais linhas são no formato

`<id> <num> <valor>`

onde `<valor>` é um ponto flutuante, `num` é um inteiro e `<id>` pode assumir 4 valores:

Se `<id>` $=$ **c** : `valor` é o custo de produção no período `<num>`

Se `<id>` $=$ **d** : `valor` é a demanda pelo produto no período `<num>`

Se `<id>` $=$ **s** : `valor` é o custo de estocagem no período `<num>`

Se `<id>` $=$ **p** : `valor` é o valor da multa no período `<num>`

Observe que o separador é uma tabulação e não um espaço.

Certificado: Os valores produzidos em cada período (incluindo os zeros para períodos sem produção) separados por tabulação

Coloração de Grafos

Dado um grafo $G = (V, E)$, uma coloração própria é uma atribuição de cores aos vértices do grafo de tal forma que vértices adjacentes recebem cores diferentes. Desejamos determinar o menor número de cores necessárias para colorir de maneira própria um grafo dado de entrada..

A primeira linha do arquivo é no formato

`n <num>`

onde `<num>` é um inteiro representando o número de vértices no grafo. Assumimos que os vértices do grafo estão nomeados de 1 até `<num>`. As demais linhas são no formato

`e <v> <u>`

onde `<v>` e `<u>` são números inteiros representando vértices e essa linha no arquivo significa que temos uma aresta entre os vértices v e u.

Observe que o separador é uma tabulação e não um espaço.

Certificado: Os vértices em cada classe de cor, em ordem crescente, separados por tabulação e com uma cor por linha.

Maior Subgrafo Induzido

Dado um grafo $G = (V, E)$ com pesos em suas arestas w_{ij}, um subgrafo induzido por um conjunto de vértices S em G é definido como sendo o grafo que possui como conjunto de vértices S e como conjunto de arestas, as arestas de G que possuem as duas extremidades em S, ou seja $uv \in E$ tal que $u \in S$ e $v \in S$. O peso de um subgrafo induzido é definido como sendo a soma de todos os pesos de suas arestas, desejamos determinar o conjunto de vértices que produz o subgrafo induzido de maior peso em G. Observe que as arestas podem ter pesos negativos. .

A primeira linha do arquivo é no formato

`n <num>`

onde `<num>` é um inteiro representando o número de vértices no grafo. Assumimos que os vértices do grafo estão nomeados de 1 até `<num>`. As demais linhas são no formato

`e <v> <u> <w>`

onde `<v>` e `<u>` são números inteiros representando vértices e essa linha no arquivo significa que temos uma aresta entre os vértices v e u. E `<w>` é um ponto flutuante indicando o valor do peso da aresta vu

Observe que o separador é uma tabulação e não um espaço.

Certificado: Os vértices do subgrafo induzido, em ordem crescente, separados por tabulação.

Como será a avaliação?

Os códigos serão executados em uma máquina Ubuntu 20.04 com Julia 1.8.1, JuMP 1.9. Seu código deve executar nessa configuração, caso ele não execute ou produza uma erro será atribuída a nota 0. Caso o aluno queira usar um solver, será disponibilizado o Gurobi 9.0.

Cada problema valerá ao todo 3 pontos e cada execução de uma instância valerá 1 ponto. Essa pontuação será atribuída da seguinte forma, considerando uma execução para uma instância:

Programa não finalizou a execução: 0.

Erro de execução: 0.

Impressão errada: 0.

Execução correta:

Certificado errado: 0.0

Solução não atingiu o mínimo requirido: 0.0

Solução atingiu exatamente o mínimo requirido e este não é o ótimo: 0.3.

AVISO: As soluções para os problemas de exemplo vieram de programas utilizados para resolver de forma SIMPLES os problemas selecionados dentro do tempo estimado. As soluções dos alunos devem ser pelo menos atingir esse valor mínimo, caso não atinja, será atribuída nota zero