

## 1 Introdução

1. *Data/hora de entrega:* 11/04/2019 (até as 23:59.59, de acordo com horário do servidor de *email* do IC).
2. *Número de integrantes por grupo:* 2 (dois). Excepcionalmente poderá ser aceito um **único** grupo com 3 (três) alunos.

3. *Descrição do trabalho:*

O trabalho consiste em implementar o algoritmo de *branch-and-bound* ( $B\&B$ ) para o problema do *flowshop* visto em aula e descrito nas páginas 444–448 do livro “Combinatorial Optimization” de Papadimitriou e Steiglitz, Prentice-Hall INC., 1982. Uma cópia deste texto está disponível em

[www.ic.unicamp.br/~cid/cursos/MC658/201901/tp1-bnb-texto.zip](http://www.ic.unicamp.br/~cid/cursos/MC658/201901/tp1-bnb-texto.zip).

É **importante notar** que, para abrir esse arquivo, você vai precisar usar a senha dada em aula. Trata-se de material protegido por **direitos autorais**. Portanto, você não pode redistribuí-lo ou colocá-lo sob domínio público na rede.

O algoritmo de *branch-and-bound* deverá ser implementado usando a estratégia de *best-bound* para a exploração do espaço de estados. O grupo é incentivado a propor e testar outras estratégias que acredite serem mais eficientes para este fim. Evidentemente, isso poderá dar origem a outras variantes do algoritmo de *branch-and-bound* em relação àquela proposta na referência acima.

Em uma tentativa de melhorar o desempenho deste algoritmo, sua implementação deve fazer também o seguinte. Em cada nó da árvore de espaço de estados, além dos dois limitantes duais, deverão ser computados adicionalmente os limitantes primais correspondentes às ordens das tarefas usadas no cálculo dos dois limitantes duais. Isso eventualmente permitirá a poda de um nó por *otimalidade* e/ou a atualização do menor limitante primal e, conseqüentemente, da melhor solução viável conhecida.<sup>2</sup>

4. *Instâncias para teste:*

As instâncias de teste estão disponíveis para *download* na página da disciplina.

5. *Relatório:*

O relatório a ser entregue deverá atender aos seguintes requisitos:

- (a) O arquivo do relatório deverá estar no formato **pdf** e conter **no máximo** 8 páginas em fonte 11 ou 12pt. **Arquivos que não atendam a esses requisitos não serão aceitos!**

---

<sup>1</sup>Preparado pelo docente em colaboração com o PED da disciplina, Natanael Ramos.

<sup>2</sup>Lembre-se que isto foi feito em aula e levou à redução da porção da árvore de espaço de estados que foi explorada em relação àquela que é mostrada nas transparências que foram disponibilizadas!

- (b) Deverá ser apresentada no texto uma tabela descrevendo os resultados obtidos pelo algoritmo de *branch-and-bound* implementado. Esta tabela deverá conter, para cada instância testada: quantos nós foram explorados na árvore de espaço de estados, o valor dos melhores limitantes primal e dual, o tempo de computação necessário para obter cada um destes limitantes, a melhor solução viável encontrada (dada pelo índice das tarefas na ordem em que serão processadas) e o tempo total de execução do seu programa. Todos os tempos de execução deverão ser reportados em segundos.

A implementação das regras de dominância explicadas na referência citada acima será considerada para uma eventual atribuição de pontuação extra na nota do grupo. Caso isso seja feito, o texto deverá trazer detalhes sobre como a dominância foi implementada.

- (c) O texto deverá conter ainda uma descrição do equipamento utilizado (*hardware*), incluindo memória RAM disponível, tipo de CPU, frequência do *clock*.
- (d) Quando aplicável, o texto deverá conter uma seção especialmente dedicada à análise **comparativa** dos resultados obtidos pelas diferentes implementações do algoritmo testadas pelo grupo (qual fornece os melhores limitantes duais? e os primais? qual resolve mais instâncias no tempo especificado? o quanto demorou para encontrar o melhor limitante primal/dual?, etc) Procure tirar **conclusões** a partir dos dados obtidos. Ou seja, sua análise não deve ser uma mera descrição dos dados tabelados.

Análises bem feitas serão consideradas para uma eventual atribuição de pontuação extra na nota do grupo.

#### 6. Informações adicionais:

- (a) Você não deve assumir o conhecimento de nenhuma solução viável a priori para as instâncias de teste dos algoritmos. Ou seja, elas devem ser obtidas ao longo da execução dos respectivos códigos, sem que haja qualquer tipo de preprocessamento.
- (b) **Todos os testes devem ser executados na mesma máquina**, a fim de que os tempos de computação das diferentes alternativas possam ser comparados adequadamente.
- (c) Ainda no algoritmo de *branch-and-bound*, o seu código deve ler um arquivo **param** contendo parâmetros de entrada (um por linha) representados por números inteiros que especificam os limites superiores para o número de nós que poderão ser explorados na enumeração e para o tempo total de execução do algoritmo (em segundos), respectivamente. Isto porque as instâncias que estarão sendo passadas para você podem obrigar o algoritmo a abrir **muitos** nós. Se, em alguma destas instâncias o limite for ultrapassado, indique o fato nas tabelas de resultados. O objetivo aqui é só evitar que o algoritmo possa ficar longas horas rodando por nada. O seu programa deverá executar dentro do tempo limite especificado com uma tolerância de poucos segundos acima do que foi estabelecido. **Trabalhos que excederem o tempo de execução serão penalizados**.
- (d) Os códigos deverão ser implementado em linguagem C ou C++, compiláveis no gcc (g++) e passíveis de serem executados em uma distribuição Linux recente (Ubuntu 16.04 ou mais moderna). A versão do gcc/g++ utilizada na correção do trabalho será a **7.4.0**. Códigos que não compilem corretamente receberão nota **zero**.
- (e) As instâncias são disponibilizadas no formato texto, constando na primeira linha um único inteiro representando a quantidade de tarefas a serem processadas. As demais linhas possuem cada uma dois inteiros, separados por espaços, representando os tempos de processamento da primeira e segunda subtarefas, respectivamente. Por exemplo, um arquivo contendo o seguinte texto seria a instância correspondente ao exemplo 18.6 da referência fornecida:

3  
2 1  
3 1  
2 3

## 2 Forma de entrega do trabalho

A entrega deve ser feita por *email* enviado ao docente, **com cópia para o PED**, sendo que:

- o campo `subject` deverá vir preenchido **obrigatoriamente** com os seguintes dizeres:

[MC658-2019s1] TP1 - grupoXX

onde `XX` é o identificador do grupo (a ser divulgado oportunamente).

- A mensagem deverá conter um anexo composto de um **único** arquivo compactado (com o comando `tar`) e chamado `grupoXX.tgz`. Ao descompactar este arquivo, além dos arquivos do código fonte (extensões `.c`, `.cpp`, `.h`, etc), deverão também estar presentes o arquivo `grupoXX-relatorio.pdf` contendo o texto do relatório e o arquivo `Makefile` que permite a compilação do código implementado ao se executar o comando `make`<sup>3</sup>. O executável gerado pela compilação deverá ter o nome `bnb-fs` e o mesmo deve ser capaz de receber dois parâmetros na forma:

`bnb-fs <instância> <param>`

onde `<instância>` é o nome do arquivo de instância e `<param>` é o arquivo de parâmetros descrito no item (6c) das “*Informações Adicionais*”. A saída do seu programa deve ser uma única linha impressa na saída padrão, na forma:

`<instância>,<primal>,<dual>,<nós-total>,<t-primal>,<t-dual>,<t-total>`

onde:

`<instância>`: nome do arquivo de instância.

`<primal>`: valor do melhor limitante primal (inteiro).

`<dual>`: valor do melhor limitante dual (inteiro).

`<nós-total>`: número de nós explorados na árvore de  $B\mathcal{E}B$  (inteiro).

`<t-primal>`: tempo no qual o melhor limitante primal foi obtido (*float* com 2 casas decimais).

`<t-dual>`: tempo no qual o melhor limitante dual foi obtido (*float* com 2 casas decimais).

`<t-total>`: tempo total de execução (*float* com 2 casas decimais).

Os caracteres `<>` **não** devem ser explicitamente escritos na saída, são apenas indicadores de campos usados nesse enunciado, atente-se somente aos separadores de campos dados por vírgulas (,). *Não coloque espaços em branco antes ou depois das vírgulas*. Caso algum valor esteja ausente, quando seu programa não foi capaz de encontrar um limitante primal, por exemplo, substitua o valor do campo pela *string* vazia. Note que os resultados apresentados devem ser aqueles que foram obtidos até a parada do programa por (a) otimalidade comprovada, (b) tempo limite alcançado ou (c) número máximo de nós explorados atingido. Os trabalhos que fugirem a essa especificação serão devidamente penalizados.

---

<sup>3</sup>[www.gnu.org/software/make/](http://www.gnu.org/software/make/)

Apesar de que se exija que seja enviado o código de todas as estratégias que foram implementadas, o ideal é que você envie seu código com as melhores estratégia de exploração de nós e de obtenção de limitantes primais e duais ativadas, uma vez que essa será a versão utilizada na execução para fazer a comparação entre os trabalhos dos grupos.

### 3 Rotinas de obtenção do tempo de execução

Um programa com exemplo de uso de rotinas de medição de tempo de execução em C e C++ está disponibilizado para *consulta* na página da disciplina. As implementações devem usar estas rotinas.

### 4 Considerações finais

As notas terão um fator comparativo que levará em consideração a qualidade dos programas, das soluções obtidas por eles e dos textos entregues pelos grupos (*sim, o mundo é competitivo!*).

Os códigos devem ser implementados pelos grupos **separadamente** e não serão toleradas de forma alguma cópias parciais ou totais de códigos entre os grupos ou de material disponível na rede. Ou seja, a implementação de **todas** as linhas de código deve ser feita **exclusivamente** pelos integrantes do grupo. Qualquer desvio em relação a essa norma resultará em média semestral ZERO para todos os envolvidos, sem prejuízo de outras sanções previstas pelas regras da universidade.

### 5 Critérios de Correção

A distribuição de pontos do trabalho será feita do seguinte modo:

- Implementação (código): até 5 pontos, dependendo da qualidade do código e dos resultados;
- Relatório: até 4 pontos, dependendo da qualidade do documento;
- Comparativo dos grupos: até 1 ponto (ver detalhes abaixo);
- Bônus previstos (acréscimo na nota dependendo da qualidade da implementação e das análises feitas no relatório):
  - regra de dominância implementada corretamente e com análise comparativa de desempenho feita adequadamente no relatório: até 1 ponto;
  - implementação e análise de desempenho de estratégias alternativas para exploração do espaço de estados: até 0.5 ponto.

*Sobre o comparativo dos grupos.* Para cada grupo e instância de teste será calculado o *gap de otimalidade*  $g$  segundo a fórmula:  $g = \frac{v-d}{p}$ , onde  $p$  ( $d$ ) é o valor do melhor limitante primal (dual) obtido pelo grupo para aquela instância. Em seguida, cada grupo recebe um *rank*  $r$  para aquela instância que é dado por  $r = (\text{número de grupos com } \text{gap} < g) + 1$ . Finalmente, calcula-se o *rank médio* do grupo  $r_m$  que é a soma dos seus *ranks* em todas as instâncias de teste dividido pela quantidade de instâncias. Os *ranks médios* serão computados com truncamento na sexta casa decimal. Finalmente, seja  $n_g$  o número total de grupos. Para um grupo com *rank médio*  $r_m$ , o seu *fator multiplicativo*  $f$  dado por

$$f = \frac{n_g - (\text{número de grupos com } \text{rank médio} < r_m)}{n_g}.$$

A nota do grupo neste item de avaliação será dada, então, pelo valor de  $f$  truncado na primeira casa decimal.