

MC658: Análise de Algoritmos III

Turma A – Prof. Cid C. de Souza

PED: Natanael Ramos

Instituto de Computação – UNICAMP – 1º semestre de 2019

1º Trabalho de Prático - Notas

1 Critérios de Correção

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

1. Implementação (código): até 5 pontos, dependendo da qualidade do código e dos resultados;
 - 1.1. Implementação das estratégias de poda por limitante, inviabilidade e otimalidade. (1.0 pt)
 - 1.2. Implementação das duas estratégias de obtenção de limitantes duais. (1.0 pts)
 - 1.3. Implementação da estratégia de exploração *best bound*. (1.0 pt)
 - 1.4. Implementação da estratégia de obtenção de limitantes primais apresentada na descrição do trabalho. (1.0 pt)
 - 1.5. Uso adequado de estrutura de dados e algoritmos eficientes. (1.0 pt)
 - 1.6. **Bônus** previstos (acréscimo na nota dependendo da qualidade da implementação e das análises feitas no relatório):
 - 1.6.1. regra de dominância implementada corretamente e com análise comparativa de desempenho feita adequadamente no relatório: até 1 ponto;
 - 1.6.2. implementação e análise de desempenho de estratégias alternativas para exploração do espaço de estados: até 0.5 ponto.
2. Relatório: até 4 pontos, dependendo da qualidade do documento;
 - 2.1. Apresentação de todos os resultados. (1.0 pt)
 - 2.2. Comparação entre as diferentes combinações de estratégias. (1.5 pts)
 - 2.3. Discussão das conclusões obtidas a partir das análises. (1.5 pts)
3. Comparativo dos grupos: até 1 ponto (ver detalhes abaixo);

Sobre o comparativo dos grupos. Para cada grupo e instância de teste será calculado o *gap de otimalidade* g segundo a fórmula: $g = \frac{p-d}{p}$, 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.

2 Avaliação

Em cada subseção, as notas de um grupo são apresentadas. A coluna **Item** se refere aos itens dos critérios de correção descritos na Seção 1 e a coluna **Nota** se refere à nota obtida no respectivo item. Detalhes sobre o comparativo dos grupos podem ser vistos na Seção 3.

2.1 Grupo 01

Item	Nota	Item	Nota
1.1.	0.5	2.1.	0.2
1.2.	1.0	2.2.	1.0
1.3.	1.0	2.3.	1.0
1.4.	1.0	3	0.6
1.5.	0.5		
1.6.1.	-		
1.6.2.	-		
Total	6.8		

Observações:

- Poda por limitante está errada. O **único** nó explorado é o de melhor limitante, enquanto os outros são todos podados (mesmo que tenham limitante dual < melhor limitante primal). Isso pode impedir que a solução ótima seja encontrada.
- No relatório, os resultados foram discutidos em cima de apenas três instâncias e não ficou claro porque tal escolha foi feita. Também não foram descritos os parâmetros utilizados (tempo/número de nós máximo) para obtenção dos resultados, isso impede que o experimento possa ser reproduzido.

2.2 Grupo 02

Item	Nota	Item	Nota
1.1.	0.8	2.1.	0.8
1.2.	1.0	2.2.	0.8
1.3.	1.0	2.3.	0.8
1.4.	1.0	3	0.8
1.5.	0.5		
1.6.1.	-		
1.6.2.	-		
Total	$7.5 - 0.1 - 0.05 = 7.35$		

Observações:

- Implementação:
 - Tenta deletar memória apontada pelo ponteiro `best_solution`, mas o mesmo pode ter valor NULL. -0.1 pt
 - Imprime informações adicionais na saída padrão que não foram solicitadas. -0.05 pt

- Faz a poda de um nó quando seu limitante primal é melhor que o limitante primal global, mas isso não quer dizer necessariamente que o limitante dual é igual ao primal, portanto a poda por otimalidade não foi feita.
- No relatório, faltou apresentar os resultados comparativos de diferentes estratégias. Apesar da restrição de espaço, poderia ter dado foco nas instâncias onde a diferença de desempenho foi mais notória. Por exemplo, foi mencionado que a estratégia de exploração de nós pelo nó mais profundo foi implementada também, mas nenhum resultado foi apresentado (o critério de mais profundo foi utilizado somente como desempate). As discussões dos resultados são um pouco vagas, por exemplo, como o uso de uma estratégia de obtenção de limitantes primais influenciou no tempo de computação/número de nós explorados.

2.3 Grupo 03

Item	Nota	Item	Nota
1.1.	1.0	2.1.	1.0
1.2.	1.0	2.2.	1.5
1.3.	1.0	2.3.	1.5
1.4.	1.0	3	1.0
1.5.	1.0		
1.6.1.	1.0		
1.6.2.	-		
Total	11.0		

Observações:

2.4 Grupo 04

Item	Nota	Item	Nota
1.1.	0.5	2.1.	1.0
1.2.	1.0	2.2.	0.2
1.3.	0.5	2.3.	0.5
1.4.	1.0	3	0.1
1.5.	0.5		
1.6.1.	-		
1.6.2.	-		
Total	5.1 - 1.0 = 4.1		

Observações:

- Existe um erro na atualização do melhor limitante dual. Para muitas instâncias, o valor do melhor limitante dual é maior que o melhor primal, sob a suposição que o limitante primal é dado por uma solução viável isso implicaria que o valor do limitante dual é maior que o ótimo, algo que não pode ocorrer. Tal erro resulta em podas indevidas por limitante/otimalidade, bem como em falhas na estratégia de exploração *best bound*.
- Foi utilizada uma fila de prioridades para implementar a estratégia *best bound*, mas a inserção em tal fila ainda é feita em $O(n)$.

- As conclusões do trabalho ficaram influenciadas pelos erros na implementação. Faltou uma maior discussão dos resultados. Por exemplo, como o uso de uma estratégia de obtenção de limitantes primais ajudou na resolução do problema?
- Atraso de 1 dia na entrega. -1.0 pt

2.5 Grupo 05

Item	Nota	Item	Nota
1.1.	0.8	2.1.	1.0
1.2.	1.0	2.2.	0.8
1.3.	1.0	2.3.	1.2
1.4.	0.0	3	0.7
1.5.	0.0		
1.6.1.	-		
1.6.2.	-		
Total	6.5		

Observações:

- Atualiza o melhor limitante dual como o menor valor de limitante dual encontrado durante a exploração, enquanto deveria ser o maior.
- Faltou a implementação da poda por otimalidade.
- No relatório, apesar de ter sido mencionado que foi também implementada a estratégia de exploração *best bound* (e foi verificado no código que foi de fato feita), faltou a apresentação de dados que mostrassem a diferença entre as diferenças estratégias, tomando, por exemplo, instâncias onde as diferenças de tempo/nós explorados foram mais discrepantes.

2.6 Grupo 07

Item	Nota	Item	Nota
1.1.	1.0	2.1.	0.8
1.2.	1.0	2.2.	0.5
1.3.	1.0	2.3.	0.5
1.4.	1.0	3	0.4
1.5.	0.5		
1.6.1.	-		
1.6.2.	-		
Total	6.7		

Observações:

- Atualiza o melhor limitante dual como o menor valor de limitante dual encontrado durante a exploração, enquanto deveria ser o maior.
- No relatório, poderiam ter reportado pelo menos os limitantes duais nas instâncias onde não foram obtidas soluções.
- Faltou uma maior discussão dos resultados. Por exemplo, como o uso de uma estratégia de obtenção de limitantes primais ajudou na resolução do problema?

2.7 Grupo 09

item	nota	item	nota
1.1.	0.8	2.1.	1.0
1.2.	0.8	2.2.	0.0
1.3.	0.5	2.3.	0.0
1.4.	0.8	3	0.2
1.5.	0.5		
1.6.1.	-		
1.6.2.	0.25		
total	4.85 - 0.05 = 4.0		

Observações:

- compilação:
 - erro no makefile: esperava que o arquivo `.cpp` estivesse no diretório `src` que não existe.
-0.05 pt
- atualiza o melhor limitante dual (chamado de `optimal_dual`) como o menor valor de limitante dual encontrado durante a exploração, enquanto deveria ser o maior. O limitante dual reportado está incorreto, pois é o do nó completo (todas tarefas escalonados) o qual teve melhor limitante primal.
- implementação da estratégia de exploração considera o menor limitante primal para fazer o *best bound*, mas deveria ser o dual. Foi considerada como uma estratégia alternativa de exploração da árvore, mas nenhuma análise/comparação com a mesma foi feita.
- no cálculo de s_1 e s_2 o valor de n é o total de tarefas e não o total de tarefas não escalonadas.
- faltou a implementação da poda por otimalidade. (implementou somente por limitante).
- a estratégia de obtenção de limitantes primais não é necessariamente a descrita no enunciado (não aproveita nenhuma ordenação das tarefas).
- no relatório, não foi feita nenhuma análise dos resultados.

2.8 Grupo 10

item	nota	item	nota
1.1.	0.8	2.1.	0.0
1.2.	1.0	2.2.	0.0
1.3.	1.0	2.3.	0.0
1.4.	1.0	3	0.9
1.5.	1.0		
1.6.1.	-		
1.6.2.	-		
total	5.7 - 0.05 = 5.65		

Observações:

- Arquivo do relatório estava corrompido aparentemente. O aluno foi contactado para enviar uma nova cópia, mas não houve resposta.
- faltou a implementação da poda por otimalidade. (implementou somente por limitante).
- Código pouco inteligível e mal documentado. -0.05 pt

2.9 Grupo 12

item	nota	item	nota
1.1.	0.8	2.1.	0.8
1.2.	1.0	2.2.	1.5
1.3.	1.0	2.3.	1.5
1.4.	1.0	3	0.3
1.5.	1.0		
1.6.1.	1.0		
1.6.2.	-		
total	9.9		

Observações:

- atualiza o melhor limitante dual como o menor valor de limitante dual encontrado durante a exploração, enquanto deveria ser o maior.
- faltou a implementação da poda por otimalidade. (implementou somente por limitante).

3 Comparativo dos grupos

Nessa seção, descrevemos como foi conduzido o experimento para realização da análise comparativa dos grupos. As instâncias utilizadas foram as mesmas distribuídas juntas do enunciado do trabalho. Todos os códigos foram compilados com a flag “-O3” do compilador gcc sob o seguinte ambiente computacional:

- Processador: Intel(R) Xeon(R) CPU E5-2603 v3 @ 1.60GHz
- 32GB de memória RAM
- Sistema operacional Ubuntu 16.04.6 LTS
- gcc versão 5.4.0 20160609

Os parâmetros estabelecidos foram de 300 segundos (5 minutos) como limite de tempo e no máximo $1e8$ nós poderiam ser explorados pelas estratégias. Foi estabelecida uma tolerância de 2 segundos que poderiam ser excedidos do tempo limite (nenhum grupo excedeu esse limite).

Como algumas implementações estavam erradas, na maioria dos casos no cômputo/atualização do limitante dual, mesmo quando o *gap* de alguma instância é 0, o mesmo não indica que a solução é ótima, nesses casos. Então, em uma instância, quando o *gap* é 0 e o limitante primal é maior que o melhor limitante primal dentre todos os grupos, o *gap* recebe valor ∞ . Note que foi verificado que

o menor limitante primal dentre todos os grupos para todas as instâncias configura de fato uma solução viável.

A Tabela 1 apresenta os valores de *rank* médio e *f* (veja a Seção 1) para todos os grupos. Note que se um subconjunto de grupos tem o mesmo valor de *gap*, então todos esses receberão o mesmo *rank*.

Tabela 1: Valores de *rank* médio e *f* (veja a Seção 1) para todos os grupos.

grupo	rank médio	f
01	4.148148	0.6
02	2.296296	0.8
03	1.333333	1.0
04	5.481481	0.1
05	3.000000	0.7
07	4.296296	0.4
09	5.333333	0.2
10	1.555556	0.9
12	4.814815	0.3

As Figuras 1 e 2 apresentam o número de nós explorados por cada grupo por instância.

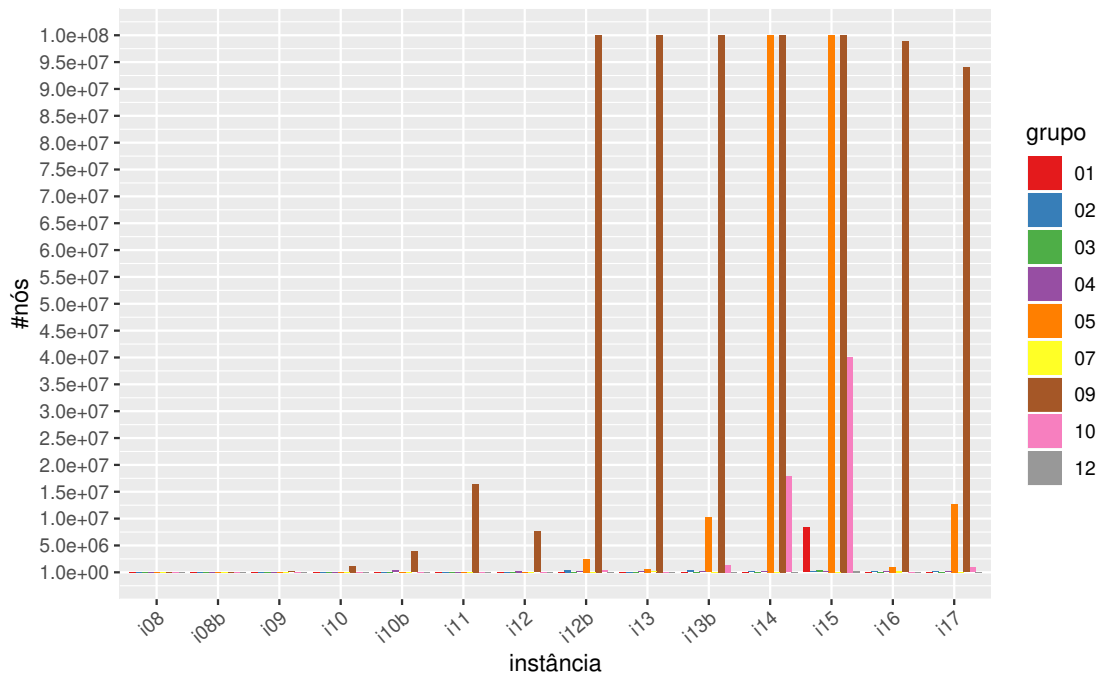


Figura 1: Número de nós explorados para $8 \leq n \leq 17$

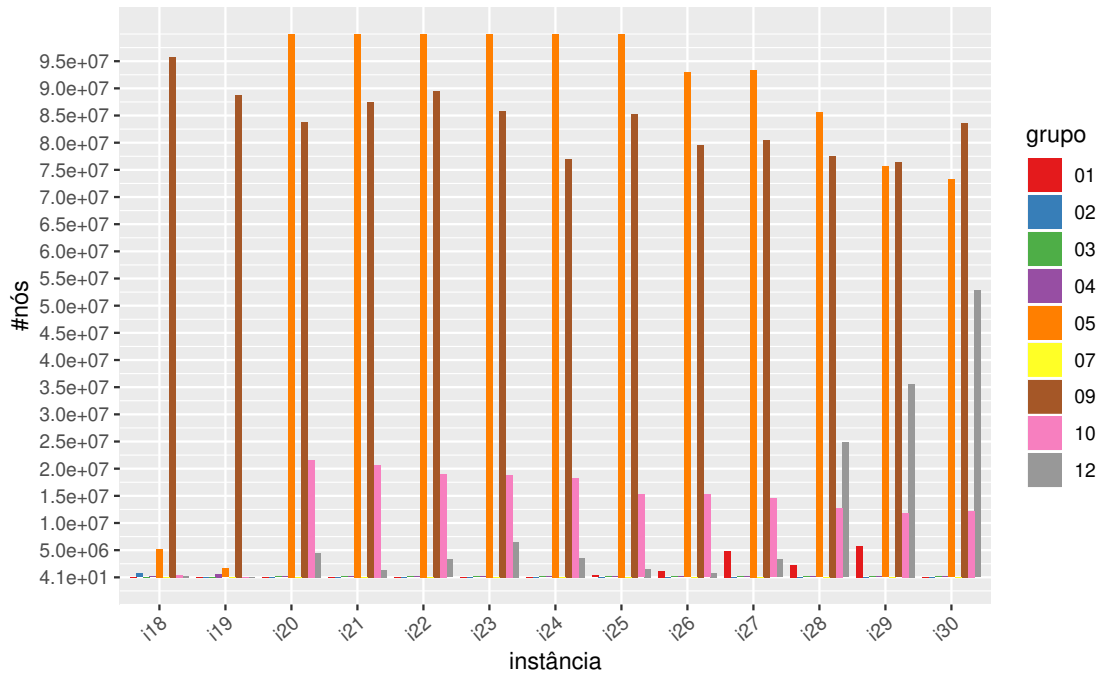


Figura 2: Número de nós explorados para $18 \leq n \leq 30$

As Figuras 3 e 4 apresentam o tempo de computação gasto por cada grupo por instância.

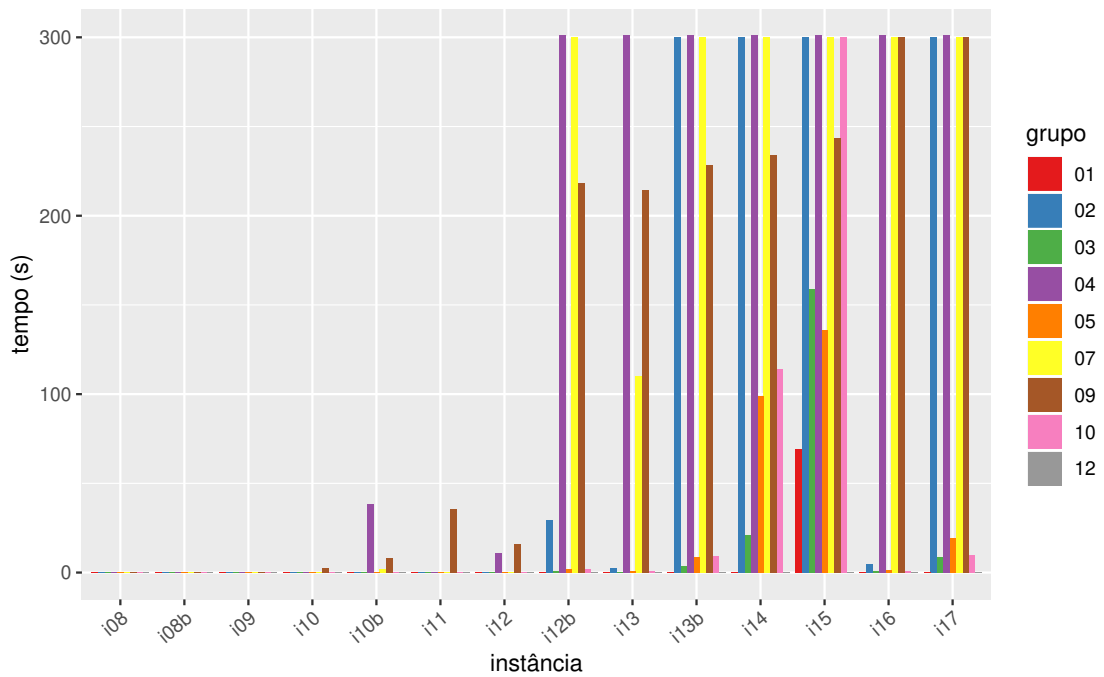


Figura 3: Tempo de computação (em segundos) para $8 \leq n \leq 17$

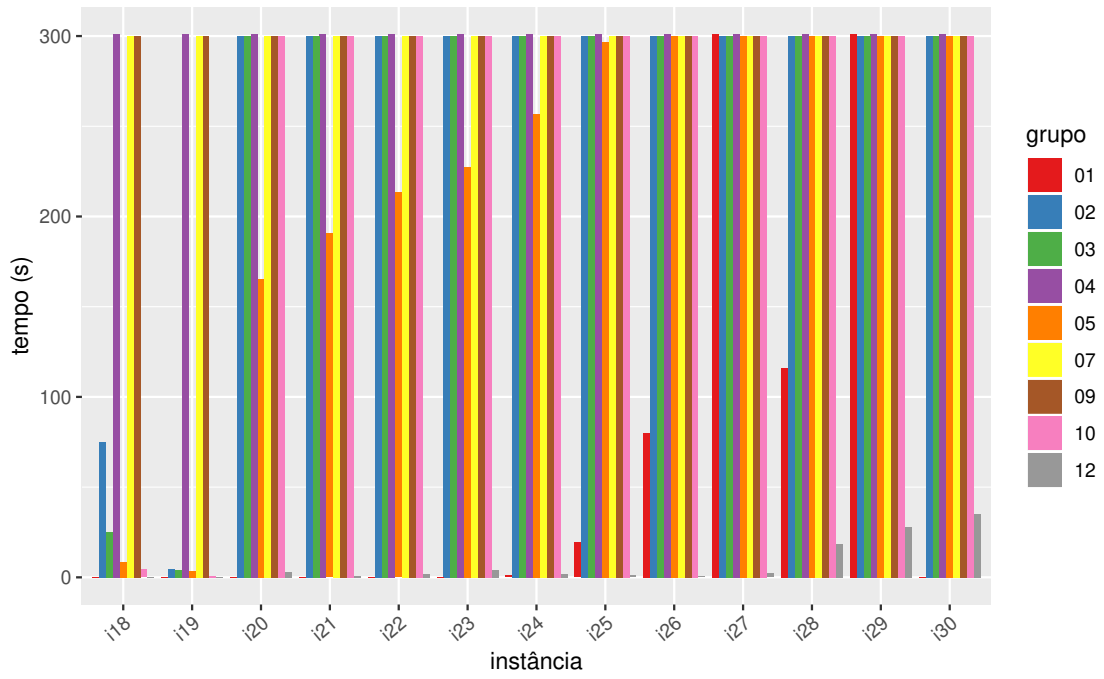


Figura 4: Tempo de computação (em segundos) para $18 \leq n \leq 30$