MC658: Análise de Algoritmos III

Turma A – Prof. Cid C. de Souza

Instituto de Computação – UNICAMP – 1º semestre de 2019

4º Trabalho de Prático[1]

1 Introdução

1. *Data/hora de entrega*: 26/06/2019 (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 heurísticas lagrangianas e meta-heurísticas para o Problema da Árvore Gerador Mínima com Restrições de Grau (`DCMSTP` do inglês *Degree Constrained Minimum Spanning Tree Problem*), que é descrito como segue. Seja $G = (V, E)$ um grafo não direcionado e conexo, com conjunto de vértices V e arestas E. Suponha que custos reais $\{c_e : e \in E\}$ são associados às arestas de G e que, para cada vértice i de G, seja associado um valor inteiro d_i tal que $1 \leq d_i \leq d_i^*$, onde d_i^* é o grau de i em V. Uma árvore geradora T de G é dita ser *restrita nos graus* se não possui mais do que d_i arestas incidentes em $i \in V$. Portanto, o `DCMSTP` consiste em encontrar uma árvore geradora de G que seja restrita nos graus e que tenha custo mínimo. Aplicações práticas do `DCMSTP` aparecem em áreas como, por exemplo, projeto de computadores, telecomunicações e redes de transporte.

   A versão de decisão do `DCMSTP` foi provada pertencer à classe NP-completo [4], então, não existe algoritmo determinístico polinomial para resolvê-lo, a não ser que $P = NP$.

2 Relaxação Lagrangiana para o `DCMSTP`

Inicialmente, apresentamos uma formulação de Programação Linear para o Problema da Árvore Geradora Mínima (`MSTP`), a qual pode ser naturalmente estendida para o `DCMSTP` introduzindo as restrições de grau.

Para cada aresta $e \in E$, associe uma variável binária x_e, a qual tem valor um se e somente se a aresta e está na solução. Para $S \subseteq V$, denote por $E(S) \subseteq E$ o conjunto de arestas $\{(i, j) \in E : i \textbf{ e } j \in S\}$. Ademais, para $i \in V$, seja $\delta(i) \subseteq E$ o conjunto de arestas incidentes em i. Uma descrição do espaço de soluções $R_{\texttt{MSTP}}$ de todas as árvores geradoras de um grafo G é dado pelas equações (1)-(3) (veja [3]).

$$\sum_{e \in E} x_e = |V| - 1 \tag{1}$$

$$\sum_{e \in E(S)} x_e \leq |S| - 1, \qquad \forall S \subset V \tag{2}$$

$$x_e \geq 0, \qquad \forall e \in E \tag{3}$$

A restrição (1) impõe que exatamente $|V| - 1$ arestas de G devem ser utilizadas, enquanto a restrição (2) impede a formação de ciclos. Então, o problema `MSTP` pode ser resumidamente descrito pela equação (4).

[1]Preparado pelo docente em colaboração com o PED da disciplina, Natanael Ramos.

$$\min \left\{ \sum_{e \in E} c_e x_e : x \in R_{\texttt{MSTP}} \right\} \quad (4)$$

Mesmo com o número exponencial de restrições correspondentes à equação (2), o MSTP pode ser resolvido em tempo polinomial por algoritmos como o de Kruskal [5] ou Prim [6].
Agora, para obter uma formulação para o DCMSTP a partir da formulação do MSTP, basta que sejam incluídas no modelo anterior as restrições (5) abaixo:

$$\sum_{e \in \delta(i)} x_e \leq d_i, \qquad \forall i \in V \quad (5)$$

Denote por $R_{\texttt{DCMSTP}}$ o espaço de soluções descrito pelas restrições (1)-(3) e (5). Desta forma, o DCMSTP pode ser formulado pela equação (6).

$$z = \min \left\{ \sum_{e \in E} c_e x_e : x \in R_{\texttt{DCMSTP}} \cap \mathbb{Z}^{|E|} \right\} \quad (6)$$

Se um vetor de multiplicadores não negativos $\lambda \in \mathbb{R}_+^{|V|}$ for associado às restrições (5), podemos dualizá-las, obtendo o Problema Primal Lagrangiano (PPL) descrito pela equação (7) a seguir.

$$z(\lambda) = \min \left\{ \sum_{e=(i,j) \in E} (c_e + \lambda_i + \lambda_j) x_e - \sum_{i \in V} \lambda_i d_i : x \in R_{\texttt{MSTP}} \right\} \quad (7)$$

Como o valor de $\sum_{i \in V} \lambda_i d_i$ é constante para um dado λ, resolver o PPL é o mesmo que resolver o MSTP com custo de arestas $\{(c_e + \lambda_i + \lambda_j) : e = (i,j) \in E\}$. Então, $z(\lambda)$ é um limitante inferior para o DCMSTP.
De maneira a obter o melhor valor de λ, λ^*, associado com o melhor limitante inferior obtido por (7), resolve-se o Problema Dual Lagrangiano (PDL), descrito pela Equação (8).

$$z(\lambda^*) = \max_{\lambda \geq 0} \left\{ \sum_{e=(i,j) \in E} (c_e + \lambda_i + \lambda_j) x_e - \sum_{i \in V} \lambda_i d_i : x \in R_{\texttt{MSTP}} \right\} \quad (8)$$

O método de otimização do subgradiente pode ser utilizado para resolver o PDL, como visto em classe e também descrito (incluindo detalhes de implementação) por Beasley [2].

3 Instâncias para teste

As instâncias de teste para este trabalho foram geradas no artigo Andrade et al. Andrade et al. [1] [2] e podem ser baixadas da página da disciplina. O formato do arquivo de entrada é o seguinte. A primeira linha contém dois inteiros, o número de vértices (n) e o número de arestas (m) do grafo. As m linhas seguintes contêm a descrição das arestas, sendo cada uma representada por um par de rótulos inteiros "u v", correspondentes às extremidades da aresta, e seu respectivo custo c_{uv}. Por fim, existem n linhas na forma "i d_i" indicando o limite de grau d_i do i-ésimo vértice. Um exemplo de um grafo com 4 vértices e 6 arestas é apresentado abaixo. Note que todas as instâncias teste caracterizam grafos completos e os custos das arestas são inteiros.

[2] O conjunto completo de instâncias disponibilizadas pelos autores encontra-se em `https://github.com/malbarbo/dcmstp-instances`

```
4 6
1 2 10
1 3 20
1 4 40
2 3 60
2 4 80
3 4 100
1 1
2 1
3 2
4 2
```

São no total 40 instâncias, com $n \in \{100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 2000\}$.

4 Relatório

O relatório a ser entregue deverá atender aos seguintes requisitos:

1. O arquivo do relatório deverá estar no formato `pdf` e conter **<u>no máximo</u>** 10 páginas em fonte `11` ou `12pt`. **Arquivos em outros formatos <u>não</u> serão aceitos**!
2. Deverá ser dada uma breve descrição da(s) heurística(s) Lagrangiana(s) implementada(s) e também da(s) meta-heurística(s), junto da análise de complexidade assintótica das mesmas. Além disso, inclua os valores dos parâmetros utilizados nos métodos (caso existam). Deixe claro qual o tipo de meta-heurística você usou e o funcionamento dos seus procedimentos básicos como, por exemplo, o de busca local (caso haja um).
3. Deverá ser feita uma análise dos resultados obtidos pelas estratégias implementadas, apresentados no texto sob a forma de uma tabela. Especificamente, esta tabela deverá conter, para cada instância testada, os valores dos limitantes duais e primais encontrados e também o tempo de computação. Note que, para a(s) meta-heurística(s), somente os limitantes primais devem ser portados. Ademais, na sua discussão de resultados, inclua (não limitado a) respostas a questões como: (i) Em quantas instâncias o valor ótimo foi encontrado? (ii) Comparando a qualidade dos limitantes primais da heurística lagrangiana e da meta-heurística, qual foi melhor? Qual o *trade-off* entre tempo de computação e qualidade de solução? (iii) Por que os particulares valores de parâmetros foram utilizados? Nessa parte, busque evitar conclusões precipitadas como, por exemplo, concluir que um método é melhor que o outro baseando-se em média aritmética, mesmo se os dados considerados não são uniformemente distribuídos. Por fim, busque ser criativo na apresentação de resultados, usando, por exemplo, gráficos comparativos.
4. O texto deverá conter ainda uma descrição do equipamento utilizado (*hardware*), incluindo memória RAM disponível, tipo de CPU, frequência do *clock*, etc.

5 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] TP4 - 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, deverá ser gerada uma pasta de nome `codigo` com todo o código fonte do trabalho e também o arquivo `grupoXX-relatorio.pdf` contendo o texto do relatório. Dentro da pasta `codigo` deve existir um arquivo `Makefile` que permite a compilação do código ao se executar o comando `make`. O executável gerado pela compilação deverá ter o nome `dcmstp-solver` e o mesmo dever ser capaz de receber três parâmetros na forma:

  ```
  dcmstp-solver <instância> <tempo> <método>
  ```

  Onde <instância> corresponde ao caminho completo para o arquivo de instância, <tempo> é o tempo limite (em segundos) que será imposto para a aplicação e `método` é um único caractere indicando qual método deve ser executado, sendo "`l`" para a heurística Lagrangiana e "`m`" para a meta-heurística. Haverá uma tolerância de ≈ 2 segundos de tempo que pode ser excedido sobre o limite. Como saída do seu programa, imprima na saída padrão as seguintes informações:

  ```
  <instância>,<lim-dual>,<lim-primal>
  ```

  caso o <método>=l e

  ```
  <instância>,<lim-primal>
  ```

  caso o <método>=m, onde <lim-dual> (<lim-primal>) é o valor do melhor limitante dual (primal) encontrado. Os valores impressos devem ser truncados em 4 casas decimais. Além disso, sua aplicação deve gerar um arquivo de nome <instância>.out que corresponde à lista das arestas (uma por linha) utilizadas na sua melhor solução. A descrição de cada aresta é dada pelos seus vértices extremos, separados por espaço, como é mostrado abaixo. Você deve assumir que os vértices são rotulados de 1 a n. Por fim, para cada aresta, os rótulos de seus extremos devem estar em ordem crescente no arquivo de saída. Para uma aresta $(3, 1)$, por exemplo, sua descrição deve ser 1 3.

  $u_1 \;\; v_1$
  ...
  $u_{n-1} \;\; v_{n-1}$

- *Informações adicionais:* **Atente-se** ao formato de saída, trabalhos que fugirem a essa especificação serão penalizados. Todo o código deve ser enviado, porém, caso você tenha feito diferentes estratégias de heurísticas Lagrangianas ou meta-heurísticas, deixe ativas somente aquelas que você observou serem as melhores. O mesmo vale para os valores de parâmetros.

6 Critérios de Correção

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

- Implementação (código): até 6 pontos, dependendo da qualidade do código e dos resultados;
  - Implementação da heurística Lagrangiana (3 pontos);
  - Implementação da meta-heurística (3 pontos);
- Relatório: até 4 pontos, dependendo da qualidade do documento;

- Bônus:
  - Heurística(s) Lagrangiana(s) alternativa(s), junto de sua descrição e análise comparativa com as demais variações. (+1 ponto)
  - Meta-heurística(s) alternativa(s), junto de sua descrição e análise comparativa com as demais variações. (+1 ponto)
- Comparativo dos grupos: *bônus* de 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. Note que p será o melhor valor entre o retornado pela heurística Lagrangiana e a meta-heurística. Em seguida, cada grupo recebe um *rank* r para aquela instância que é dado por $r =$ (número de grupos com 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

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

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

7 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 *(você já sabe, 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.

Referências

[1] R. Andrade, A. Lucena, and N. Maculan. Using lagrangian dual information to generate degree constrained spanning trees. *Discrete Applied Mathematics*, 154(5):703–717, 2006.

[2] J. E. Beasley. Lagrangian relaxation. In *Modern heuristic techniques for combinatorial problems*, pages 243–303. John Wiley & Sons, Inc., 1993.

[3] J. Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, 1(1):127–136, 1971.

[4] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979. ISBN 0-7167-1044-7.

[5] J. B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, 1956.

[6] R. C. Prim. Shortest connection networks and some generalizations. *The Bell System Technical Journal*, 36(6):1389–1401, 1957.