Universidade Estadual de Campinas - UNICAMP
Instituto de Computação - IC

MO420/MC908: Programação Linear Inteira
Prof. Cid Carvalho de Souza - IC/UNICAMP
1^o Trabalho Prático – 1^o semestre de 2018

1 Introdução

O Problema da Árvore Geradora de Custo Mínimo com Conflitos (AGMCC) tem sido alvo de pesquisas recentes [2, 4, 3]. Aplicações do mesmo são reportadas na área de projetos de redes e também na teoria da escolha social. O problema consiste no seguinte.

Seja um grafo $G = (V, E)$, não direcionado, com $|V| = n$ e $|E| = m$. A cada aresta e de E está associado um custo c_e. Além disso, é dado um conjunto S de pares de arestas, chamado de *conjunto de conflito*. Cada elemento de $\{e, f\}$ em S é chamado de um *par (de arestas) conflitante*. Uma árvore geradora T de G é dita ser *livre de conflitos* se nenhum par de arestas em T é conflitante. No AGMCC o que se deseja é encontrar uma árvore geradora de custo mínimo que seja livre de conflitos. Sabe-se que este problema $\mathcal{NP}$-difícil.

O objetivo deste trabalho é estudar e implementar algoritmos lagrangianos para o AGMCC.

2 Relaxação Lagrangiana para o AGMCC

Seja $\mathcal{F}$ o conjunto de todos os vetores de incidência (ou característico) que representam árvores geradoras de G. Denotando por x um destes vetores, temos que $x \in \mathbb{B}^{|m|}$. Assim, o AGMCC pode ser modelado por:

$$
\begin{aligned}
(IP) \qquad \min \qquad & z = \sum_{e \in E} c_e x_e & (1)\\
\text{s.a} \qquad & x_e + x_f \leq 1, \qquad \forall\ \{e, f\} \in S & (2)\\
& x \in \mathcal{F}. & (3)
\end{aligned}
$$

Denote por S_e o conjunto de pares conflitantes de S envolvendo a aresta e. Seja ainda $e^\star$ a aresta de E com menor custo para a qual S_e não é vazio. Finalmente, no caso da inequação (2) referente ao par $\{e, f\}$ de S seja dualizada (penalizada) em uma relaxação lagrangiana, seja u_{ef} o multiplicador de Lagrange correspondente. A partir daí, vamos considerar duas possíveis relaxações lagrangianas para o AGMCC. São elas:

- **Relaxação 1:**

$$(RL^1) \qquad \min \qquad z^1 = \sum_{e \in E} \big(c_e + \sum_{\{e,f\} \in S_e} u_{ef}\big) x_e - \sum_{\{e,f\} \in S} u_{ef} \tag{4}$$

$$\text{s.a} \qquad x \in \mathcal{F}. \tag{3}$$

- **Relaxação 2:**

$$(RL^2) \qquad \min \qquad z^2 = \sum_{e \in E} \Big(c_e + \sum_{\{e,f\} \in S_e \setminus S_{e^\star}} u_{ef}\Big) x_e - \sum_{\{e,f\} \in S \setminus S_{e^\star}} u_{ef} \tag{5}$$

$$\text{s.a} \qquad x_{e^\star} + x_f \leq 1, \qquad \forall \{e^\star, f\} \in S_{e^\star} \tag{6}$$

$$x \in \mathcal{F}. \tag{3}$$

Note que nos dois casos, para um vetor $u \in \mathbb{R}_+^{|S|}$, o problema primal lagrangiano pode ser resolvido em tempo polinomial (veja a seção seguinte). Além disso, foi visto em aula que o sistema linear que descreve a envoltória convexa dos vetores de incidência de todas as geradoras mínimas árvores de G é conhecido, porém, tem tamanho exponencial em $n = |V|$. Isto faz com que, mesmo para instâncias com baixo valor de n, já não seja possível carregar todo o modelo linear em memória. Seja z_{LP} o valor da relaxação linear dada por este modelo e w_{LD}^k o valor ótimo do dual lagrangiano para a relaxação RL^k, $k = 1, 2$.

3 Questões teóricas a serem respondidas no relatório

1. Formule o problema dual lagrangiano w_{LD}^k associado a cada relaxação RL^k.

2. Algoritmos para o resolver o *problema da árvore geradora mínima* (PAGM) e, por consequência, o problema primal lagrangiano (z^1) da primeira relaxação, podem ser encontrados em bons livros-texto de projeto e análise de algoritmos. Diga qual o algoritmo que você usou para resolver o PAGM na sua implementação, destacando a complexidade de pior caso do mesmo e citando a fonte das suas informações (referências bibliográficas). Justifique a escolha do algoritmo utilizado, levando em consideração as características dos grafos usados nos testes.

3. Usando resultados teóricos vistos em aula, diga se a afirmação seguinte é falsa ou verdadeira: *"A relaxação RL^1 tem a propriedade de integralidade, ou seja, $w_{LD}^1 = z_{LP}$."* Em caso afirmativo, qual seria então a vantagem de usar relaxação RL^1?

4. Em teoria, os limitantes duais gerados por uma das relaxações domina aqueles obtidos pela outra? Justifique.

5. Descreva um algoritmo polinomial para resolver o problema primal lagrangiano (z^2) da relaxação RL^2. Dê a complexidade de pior caso do seu algoritmo e argumente porque ele está correto.

4 Tarefas a serem cumpridas e resultados a serem reportados/analisados

Você deve cumprir as seguintes tarefas:

1. Implementar o **método do subgradiente** (MS) para resolver o problema dual lagrangiano associado a cada uma das duas relaxações apresentadas;
2. Projete e implemente uma heurística para cada relaxação para gerar limitantes primais durante a execução do MS, citando referências caso use um algoritmo descrito na literatura;
3. Para todas as instâncias de teste, executar o MS implementado (para cada relaxação) armazenando os valores dos melhores limitantes primal e dual encontrados e verificando se a solução que gerou o melhor limitante dual é ótima para o AGMCC;

Note que não está sendo solicitada a implementação do algoritmo de *branch-and-bound*, mas apenas a parte do MS!

No seu relatório, você deve:

4. Reportar, para todas as instâncias de teste, os valores dos melhores limitantes primal e dual encontrados com cada relaxação;
5. Descrever sucintamente a(s) heurística(s) primal(is) desenvolvidas no item 2, reportando a(s) complexidade(s) da(s) mesma(s);
6. Descrever como a verificação citada no item 3 foi implementada (ou seja, qual o resultado teórico que permite esta verificação);
7. Reportar, se houver, todas as instâncias para as quais você conseguiu comprovar a otimalidade, incluindo o valor ótimo e a(s) relaxação(ões) que forneceu(ram) tal valor;
8. Apresentar tabelas com os tempos computacionais totais e os de obtenção de cada um dos melhores limitantes duais e primais para todas as instâncias de teste e relaxações;
9. Fazer uma análise comparativa dos tempos citados no item 8;
10. Fazer uma análise comparativa sucinta do desempenho das duas relaxações;
11. Reportar as características da máquina onde os experimentos foram realizados (*clock*, RAM, processador, etc);
12. Reportar quais os valores dos parâmetros usados no MS e outros que alterem o comportamento do algoritmo (tamanho do passo, taxa de redução do tamanho do passo, número de iterações, etc). [*Dica:* considere realizar testes preliminares que mostrem como o comportamento do MS é afetado por estes parâmetros];
13. Apresentar, para pelo menos uma instância, um gráfico mostrando a variação dos limitantes inferior e superior (se tiver) em função do número de iterações do MS;
14. Interpretar o gráfico citado no item 13 analisando a variação dos limitantes, especialmente do limitante dual, em função das mudanças que ocorreram em parâmetros durante a execução do MS (alteração no tamanho do passo, melhora do limite primal, etc).

5 Instâncias de teste

As instâncias de teste e a descrição do seu formato de entrada estão disponibilizadas na página da disciplina.

6 Formato de saída do programa

A especificação da saída do seu programa está disponiblizada na página da disciplina.

Siga **rigorosamente** o formato de saída especificado. Não acrescente **nada** à saída impressa pelo seu programa além do que foi pedido, inclusive textos de qualquer tipo. Caso o seu programa não siga essa formatação de saída, a nota do seu trabalho será penalizada.

7 Forma de entrega do trabalho

O trabalho deve ser entregue por e-mail ao docente sob a forma de um arquivo "raXXXXXX.tar.gz" (ou "raXXXXXX.tgz") que ao ser descompactado deverá criar (no diretório corrente) um subdiretório "raXXXXXX/", onde XXXXXX é o número do seu RA. Este diretório deverá conter dois subdiretórios: (1) "texto/", contendo um arquivo texto.pdf (nenhum outro formato será aceito) com o relatório reportando os resultados que você obteve e as suas análises sobre os mesmos, e (2) "codigo/", contendo os programas fonte que você implementou e um `Makefile` que permita sua compilação. **A ausência deste arquivo ou de equivalente será penalizada com rigor**.

O arquivo executável gerado pelo `Makefile` deverá obrigatoriamente se chamar `relaxlag`. Este último diretório deverá conter ainda dois aquivos de parâmetros que deverão ser lidos pelo programa e conter todos os parâmetros de execução do MS, sendo um para cada relaxação e nomeados `param1` e `param2` de acordo com as relaxações descritas neste documento.

Os nomes dos seus arquivos deverão respeitar estritamente o uso de letras maiúsculas e minúsculas na descrição acima!

Para executar o programa, a seguinte linha de comando deverá ser dada:

```
./relaxlag k ENTRADA SAIDA
```

onde `k` é o número da relaxação que será executada (RL^k, para $k = 1, 2$) e `ENTRADA` e `SAIDA` são, respectivamente, caminhos para um arquivo de uma instância de teste e para um arquivo de saída.

8 Considerações gerais

- Os trabalhos deverão ser feitos individualmente.
- O texto do relatório deverá ser preparado em formato carta, fonte tamanho 12, espaçamento simples e, em hipótese alguma, poderá ultrapassar 12 (doze) páginas.
- Os programas devem ser obrigatoriamente escritos em linguagem `C` ou `C++` e devem ser compiláveis em uma plataforma Linux usando o `gcc` (ou `g++`).
- Recomenda-se fortemente a leitura do excelente artigo [1] de Beasley. O livro que contém este artigo está disponível em algumas bibliotecas da UNICAMP e cópias de pequenas partes do capítulo de interesse serão disponibilizadas em local a ser informado oportunamente. Vários

detalhes de implementação do MS estão muito bem discutidos neste artigo e serão de grande valia para a realização deste trabalho. *Cópias deste trabalho e do artigo [4] podem ser baixadas da página da disciplina. Para abrir os arquivos você precisará da senha dada em aula.*

- A entrega do trabalho está marcada para o dia 16/05. O arquivo contendo o trabalho deve ser recebido pelo professor impreterivelmente até as 24:00 horas desta data (de acordo com o horário do servidor do IC).

- IMPORTANTE I: A avaliação terá uma componente **comparativa**, ou seja, a sua nota poderá aumentar ou diminuir em função também do que os outros alunos fizeram no trabalho!

- IMPORTANTE II: O código `C` ou `C++` deve ser **inteiramente desenvolvido por você**. Isto quer dizer que o uso de (partes de) códigos baixados da rede será considerado uma tentativa de fraude, contra a qual serão aplicadas todas as sanções previstas nas regras da Universidade.

Referências

[1] J. E. Beasley. Lagrangian relaxation. In C. R. Reeves, editor, *Modern Heuristic Techniques for Combinatorial Problems*, chapter 6, pages 243–303. John Wiley & Sons, Inc., New York, NY, USA, 1993.

[2] A. Darmann, U. Pferschy, e J. Schauer. Determining a minimum spanning tree with disjunctive constraints. In F. Rossi and A. Tsoukias, editors, *Algorithmic Decision Theory*, volume 5783 of *Lecture Notes in Computer Science*, pages 414–423. Springer Berlin Heidelberg, 2009.

[3] P. Samer e S. Urrutia. A branch and cut algorithm for minimum spanning trees under conflict constraints. *Optimization Letters*, 9(1):41–55, 2015.

[4] R. Zhang, S. N. Kabadi, e A. P. Punnen. The minimum spanning tree problem with conflict constraints and its variations. *Discrete Optimization*, 8(2):191 – 205, 2011.