Exercícios de Projeto e Análise de Algoritmos*

Flávio K. Miyazawa

Exercícios de Programação Linear e de Fluxos

Nas questões em que se pede para resolver um problema através de formulação via programação linear, descreva o significado de cada variável, a função objetivo da formulação e explique cada uma das restrições. Justifique porque sua formulação modela o problema em questão. Em algumas situações, a resolução da questão exige que você faça algo com a solução do programa linear. Nestes casos, descreva o algoritmo e prove que sua resolução está correta dentro da complexidade de tempo solicitada.

1. Faça os exercícios do Capítulo 1 do livro "Linear Programming and Network Flows", M. S. Bazaraa, J. J. Jarvis, H. D. Sherali.

2. Uma industria de empacotamento de carne produz 480 kilos de hamburguers, 400 de pork bellies e 230 kilos de picnic hams, todo dia. Cada um destes produtos pode ser vendido ou fresco ou defumado. O número total de hams, bellies e picnics que podem ser defumados durante um dia normal é 420 kilos. Além disso, até 250 kilos podem ser defumados durante a noite, mas com um custo maior. Os valores de lucro para cada produto (números dados em dólares) são dados pela seguinte tabela:

	Fresco	Defumada	Defumada a noite
Hams	8	14	11
Bellies	4	12	7
Picnics	4	13	9

Por exemplo, o seguinte escalonamento (números dados em kilos) permite um lucro total de $9965,00:

	Fresco defumada	Defumada	Defumada a noite
Hams	165	280	35
Bellies	295	70	35
Picnics	55	70	105

O objetivo é encontrar um escalonamento que maximiza o lucro total.
Sugestão: Use variáveis H_F, H_D e H_N para (Hamburger fresco, defumada de dia e defumada a noite, resp.) O mesmo para B_F.... - Bellies e P_F... - Picnics.

3. Considere o problema de cobertura por vértices com pesos nos vértices. Resolva este problema por duas abordagens diferentes quando o grafo G é bipartido:

*Peço que envie sugestões, adições e correções para `fkm@ic.unicamp.br`.

- Reduza o problema para o problema de fluxo máximo / corte mínimo. Sugestão: Suponha que a partição dos vértices seja dada pelos conjuntos X e Y. Acrescente dois vértices s e t e coloque arcos de s para X com pesos dados pelo vértice em X. Oriente as arestas de X para Y colocando peso infinito nos arcos. Coloque arcos de Y para t, colocando o peso nos arcos igual ao dos vértices de Y. Mostre a relação do corte mínimo com cobertura mínima.
- Formule o problema como programação linear inteira e relaxe as variáveis. Mostre que o poliedro associado é inteiro.

4. Considere o problema do item anterior. Mostre que também é possível aplicar as mesmas ideias para resolver o problema do conjunto independente em grafos bipartidos.

5. Uma rede de fluxos é dado pelo seguinte grafo orientado $G = (V, A, s, t, c)$ onde V é o conjunto de vértices, A é o conjunto de arcos (arestas), s é o vértice fonte e t é o vértice destino e c é uma função de capacidades nas arestas $c : A \to R^+$ (i.e., c_e é a capacidade do arco e, as capacidades são valores positivos). Dizemos que um arco $e \in A$ tem um fluxo de x_e se $0 \leq x_e \leq c_e$ (imagine que você está transportando uma quantidade de material pelo arco, mas você não pode transportar mais que a capacidade do arco c_e). Denotamos por $\delta^+(v)$ o conjunto de arestas saindo de v e por $\delta^-(v)$ o conjunto de arestas entrando em v. Denotamos por $\Delta(v) = \sum_{e\in\delta^+(v)} x_e - \sum_{e\in\delta^-(v)} x_e$ (diferença do fluxo entrando em v e saindo de v).

   Um fluxo nesta rede é dado por uma atribuição x, $x : A \to R$, não negativa, tal que $\Delta(v) = 0$ para todo vértice v diferente de s e t (conservação do fluxo). O valor do fluxo dado por x é $f(x) = \Delta(s)$.

   (a) Faça uma formulação em programação linear para encontrar um fluxo de valor máximo.
   (b) O item anterior foi para grafos orientados. Descreva a versão deste problema para grafos não-orientados e apresente uma formulação para este problema e construa a solução do problema.
   (c) Um resultado relacionado aos exercícios anteriores é o do fluxo máximo - corte mínimo. Um corte $K \subset V$ é um conjunto de vértices tal que $s \in K$ e $t \notin K$. O valor da capacidade deste corte é igual a $\sum_{e\in\delta^+(K)} c_e$ (onde $\delta^+(K)$ é o conjunto de arcos que tem origem em K e destino fora de K). O teorema diz que se o fluxo máximo entre dois vértices é limitado, então seu valor é igual ao de um corte mínimo. Tendo em mãos um fluxo x (definido para cada aresta), descreva um algoritmo que encontra um corte de capacidade mínima ?

6. Seja S um conjunto de pontos contido em $\mathbb{R}^d$ e $conv(S)$ o conjunto de pontos que são combinação convexa dos pontos em S. Se temos pontos x, y e z, tal que $x \in conv(S)$, $y \in conv(S)$ e $z \in conv(x, y)$, então temos também que $z \in conv(S)$.

7. Se A é matriz formada por elementos em $\{-1, 0, 1\}$, e cada coluna tem no máximo uma ocorrência de -1 e no máximo uma ocorrência de 1, então A é totalmente unimodular (TU). Prove esta afirmação.

8. Mostre que se A é a matriz de incidência de um grafo orientado e B uma matriz obtida a partir de A transformando alguns elementos não nulos em 0, então B também é totalmente unimodular. Prove esta afirmação.

9. Seja $G = (N, A)$ um grafo orientado com conjunto de nós N e arcos A e dois nós distintos s e t. Apresente um poliedro definido por variável $x \in \{0, 1\}^{|A|}$ indexada em A, com número polinomial de restrições, e conjunto de vértices $\mathcal{V}$, tal que, se $\hat{x} \in \mathcal{V}$, então $\hat{x}$ representa os caminhos de s a t, com $\hat{x}_a = 0$ para todo $a \in \delta^-(s) \cup \delta^+(t)$. Isto é, são caminhos sem arcos entrando em s ou saindo de t. Dado um vértice, solução deste poliedro, descreva um algoritmo de tempo polinomial que imprime um possível conjunto de caminhos de s a t representados neste vértice. Justifique sua solução/algoritmo.

10. Considere o seguinte problema de seleção de itens. São dados um conjunto de itens $N = \{1, \dots, n\}$, um conjunto de m subconjuntos de N, dado pelo conjunto $\{S_1, \dots, S_m\}$. Seja $M = \{1, \dots, m\}$. Se o item $i \in N$ for selecionado, então deve-se pagar c_i pelo item. Porém, caso todos os itens de um conjunto S_j forem selecionados, temos um benefício (e.g., itens poderiam ser máquinas e um conjunto S_j significa que podemos fabricar certo produto que depende destas máquinas), e com isso, temos um benefício de b_j. Os valores de c_i e b_j são positivos. O objetivo é selecionar os itens, de maneira que o valor total dos benefícios conseguidos, menos o valor total dos gastos de todos os itens, é máximo. Projete um algoritmo de tempo polinomial para resolvê-lo. Justifique a corretude e complexidade de tempo do seu algoritmo.

11. No problema do conjunto fechado de peso máximo, são dados: um grafo orientado $G = (N, A)$, com conjunto de nós N e conjunto de arcos A, e uma função de peso nos nós, $w : V \to \mathbb{Q}$, que podem dar valor positivo ou negativo. Um conjunto $F \subseteq N$ é dito ser fechado, se não há arcos saindo de F para $V \setminus F$. Faça um algoritmo exato para encontrar um conjunto fechado de peso total máximo. Justifique a corretude e complexidade de tempo do seu algoritmo.

12. Projete algoritmos de tempo polinomial, utilizando formulações em programação linear, para os seguintes problemas (conhecidos como versões do teorema de Menger). Para cada caso, justifique a corretude e complexidade de tempo do seu algoritmo.

    (a) Problema dos Caminhos Disjuntos em Grafos orientados: Dado grafo orientado e vértices s e t, encontrar o maior número de caminhos orientados disjuntos nos arcos, de s a t.

    (b) Problema do Corte Separador de s e t em grafos orientados: Dado grafo orientado e vértices s e t, encontrar o menor número de arcos cuja remoção desconecta todos os caminhos de s a t.

    (c) Problema dos Caminhos Disjuntos em Grafos não-orientados: Dado grafo não-orientado e vértices s e t, encontrar o maior número de caminhos disjuntos nas arestas, de s a t.

    (d) Problema do Corte Separador de s e t em grafos não-orientados Dado grafo não-orientado e vértices s e t, encontrar o menor número de arestas cuja remoção desconecta todos os caminhos de s a t.

13. Exercícios de busca de caminhos de custo mínimo. Para estes exercícios, considere que são dados: *(i)* um grafo orientado $G = (V, E, c)$, onde V é o conjunto de nós do grafo, E é o conjunto de arcos e c é uma função de custo nas arestas, $c : E \to \mathbb{Q}^+$ e *(ii)* dois nós $s \in V$ e $t \in V$. Utilize formulações por programação linear como parte da resolução. Para cada caso, justifique a corretude e complexidade de tempo do seu algoritmo.

    (a) Resolva o problema de se encontrar um caminho de custo mínimo em G, de s para t através de algoritmos baseados em programação linear.

    (b) Considere o problema dos k-caminhos de custo total mínimo: O objetivo é encontrar um conjunto de k caminhos $P_1, \dots, P_k$, cada um começando em s e terminando em t tal que:

        - Se E_r é o conjunto de arestas em P_r, onde $r \in \{1, \dots, k\}$, então $E_i \cap E_j = \emptyset$ para quaisquer $1 \leq i < j \leq k$.
        - Se $F = E_1 \cup \ldots \cup E_k$ (conjunto de todas os arcos na solução), então $\sum_{e \in F} c_e$ é mínimo.

        Projete um algoritmo de tempo polinomial, utilizando resolução por programação linear, que resolva o problema dos k-caminhos de custo total mínimo. Seu algoritmo deve imprimir cada um dos k caminhos separadamente.

    (c) Descreva o problema dos exercícios anteriores, mas para grafos não-direcionados. Projete um algoritmo de tempo polinomial para resolvê-lo.

(d) Descreva os problemas dos dois exercícios anteriores, mas para caminhos disjuntos nos vértices internos (os caminhos só se interseptam nos vértices s e t).

(e) Descreva os problemas dos exercícios anteriores, mas para caminhos disjuntos nos vértices internos (os caminhos só se interseptam nos vértices s e t).

14. Considere a seguinte variação do problema do Fluxo de valor Máximo, que é definido como no problema do Fluxo de valor Máximo, mas com a restrição adicional que cada nó interno i possuiu um limitante c_i (um valor numérico positivo) para o valor do fluxo total entrando em i. Vamos denotar o problema do Fluxo de valor Máximo por FM e a variação do problema do Fluxo de valor Máximo com Limitação nos nós internos por FML. Primeiro enuncie o problema do Fluxo de valor Máximo. Depois, faça uma redução de tempo linear do problema FML para FM. Justifique a corretude da sua redução.

15. Reduza o problema de encontrar um Emparelhamento de Cardinalidade Máxima em grafos bipartidos ao Problema do Fluxo Máximo. Mostre que a aplicação do Algoritmo Ford-Fulkerson nos dá um algoritmo de tempo polinomial para este problema. Justifique a corretude e complexidade de tempo do seu algoritmo.

16. Suponha que a empresa Google queira melhorar as chances dos usuários clicarem em propagandas de empresas (e consequentemente aumentar o faturamento) considerando o seguinte:

    - Em um certo momento, a Google detecta que há n usuários de sua ferramenta de busca, dado pelo conjunto $U = \{1, \ldots, n\}$, e não mostrou o resultado das respectivas buscas para nenhum deles. E temos m empresas, dadas pelo conjunto $E = \{1, \ldots, m\}$.
    - Cada empresa $j \in E$ quer aparecer para no máximo d_j usuários (como limitador da valor disponível que tem para gastar nestas propagandas).
    - A Google sabe o valor esperado $p_{i,j}$ que irá receber, caso a empresa j seja atribuída ao usuário i (isso poderia ser calculado considerando o número de palavras da busca que estão associadas à empresa, considerando o quanto cada empresa é conhecida pelos usuários,...).
    - O usuário i pode receber até m_i propagandas (a quantidade pode variar, dependendo do tamanho da tela utilizada pelo equipamento, que pode ser computador, notebook ou smartphone).

    Projete um algoritmo de tempo polinomial para atribuir as propagandas atuais aos usuários, utilizando formulação por programação linear, respeitando suas restrições quantitativas e maximizando o valor esperado total da atribuição. Justifique a corretude e complexidade de tempo do seu algoritmo.

17. Apresente algoritmos para resolver os seguintes problemas de transporte. Para cada caso, justifique a corretude e complexidade de tempo do seu algoritmo.

    - Resolva o Problema do Transporte: São dados um conjunto A de fornecedores e um conjunto B de consumidores, onde $A = \{1, \ldots, m\}$ e $B = \{1, \ldots, n\}$, todos relativos a um mesmo material. O fornecedor $i \in A$ produz a_i unidades e o consumidor $j \in B$ consome b_j unidades. O custo para transportar uma unidade do fornecedor i para o consumidor j é de c_{ij}, para todo $i \in A$ e $j \in B$. Considere que $T = \sum_i a_i = \sum_j b_j$. O problema consiste em determinar a forma mais barata para transportar T unidades dos fornecedores para os consumidores, satisfazendo a produção e consumo dos produtores e consumidores.
    - Resolva uma variação do Problema do Transporte (capacitado nas arestas) supondo que o fornecimento x_{ij} é no máximo um valor u_{ij}, dado na entrada, para todo $i \in A$ e $j \in B$.

- Resolva uma variação do Problema do Transporte supondo que cada aresta tem uma capacidade máxima, $\sum_{i \in A} a_i \geq \sum_{j \in B} b_j$ (não necessariamente valendo na igualdade) devemos atender todos os consumidores, mas os fornecedores só precisam respeitar capacidade de produção (não necessariamente, a produção de um fornecedor i precisa ser consumida).
- b-matching: Dados grafo bipartido $G = (A, B, E)$ com função de peso nas arestas $w : E \to \mathbb{Q}$, e função limitadora de grau $b : A \cup B \to Z^+$, encontrar um conjunto de arestas M tal que o número de arestas de M incidentes a um vértice i é no máximo b_i e o peso total das arestas de M é máximo.

18. Considere um grafo bipartido G. Reduza o problema de encontrar uma cobertura por vértices de cardinalidade mínima em G para o problema de emparelhamento de cardinalidade máxima em G.

19. O problema do conjunto independente em grafo bipartido é definido como: Dados grafo bipartido $G = (V, E)$ com função de peso nos vértices $w : V \to \mathbb{Q}$, encontrar um conjunto de vértices $I \subseteq V$ que não tem arestas incidentes em comum (conjunto independente). Apresente uma formulação em programação linear inteira para resolver este problema, cuja relaxação apresente apenas vértices inteiros. Com isso, projete um algoritmo de tempo polinomial para encontrar um conjunto independente em G de peso máximo. Justifique a corretude e complexidade de tempo do seu algoritmo.

20. O problema da cobertura por vértices em grafo bipartido é definido como: Dados grafo bipartido $G = (V, E)$ com função de peso nos vértices $w : V \to \mathbb{Q}$, encontrar um conjunto de vértices $C \subseteq V$ tal que toda aresta $e \in E$ tem pelo menos um dos extremos em C. Faça como no exercício anterior e projete um algoritmo de tempo polinomial para encontrar uma cobertura por vértices em G de peso mínimo. Outra maneira de resolver o problema da cobertura por vértices, é reduzindo este problema para o anterior. Mostre como isto pode ser feito. Justifique a corretude e complexidade de tempo dos algoritmos.

21. Dados grafo bipartido $G = (V, E)$ com função de peso nas arestas $w : E \to \mathbb{Q}$, encontrar um conjunto de arestas $F \subseteq E$, de peso mínimo, tal que para todo vértice $v \in V$, há pelo menos uma aresta de F incidente em v (cobertura por arestas). Faça como no exercício anterior e projete um algoritmo de tempo polinomial para encontrar uma cobertura por vértices em G de peso mínimo. Justifique a corretude e complexidade de tempo do algoritmo.

22. Considere o seguinte problema, que chamaremos de VAGAS:

    Problema VAGAS: Um certo país tem n cursos de Computação em universidades públicas, cada curso i (onde i é um inteiro em $\{1, \ldots, n\}$) tem v_i vagas. Há m pessoas que são candidatas a estas vagas, cada candidato j (onde j é um inteiro em $\{1, \ldots, m\}$) fez várias provas dentro de um exame, que chamaremos de EN (Exame Nacional). Como cada curso usa uma ponderação diferente do EN, cada candidato pode ter pesos diferentes para cursos diferentes. Vamos denotar a nota do candidato j para o curso i pelo peso p_{ij} (quanto maior o peso melhor).

    O governo do país deseja encontrar uma atribuição de candidatos a vagas, de maneira que: *(i)* Cada pessoa j é atribuida a no máximo 1 curso. *(ii)* Cada curso i recebe a atribuição de no máximo v_i candidatos. *(iii)* A soma total dos pesos das pessoas atribuidas aos correspondentes cursos é máxima.

    Faça uma redução de tempo linear do problema VAGAS para o problema de Programação Linear e prove porque esta redução está correta, justificando claramente porque é possível usar um resolvedor de programação linear (sem restrições de integralidade) para resolver corretamente o problema VAGAS.

23. Considere uma empresa que deseja estabelecer uma planta de centralização de produtos. Os produtos são produzidos em n locais. Cada local $i = 1, \ldots, n$ produz p_i unidades de produto. Os produtos podem ser levados para um conjunto de m centros, cada centro $j = 1, \ldots, m$ pode receber no máximo k_j unidades de produto. Há um custo c_{ij} para transportar cada unidade de produto de um produtor i para um centro j. Deve-se estabelecer um padrão de transporte de produtos para os centros de tal forma que as capacidades dos centros sejam respeitadas, e o total produzido por cada produtor seja levado para os centros. Deve-se minimizar o custo total para transportar os itens. Apresente uma formulação em programação linear para este problema e justifique sua corretude e sua resolução em tempo polinomial no número de itens tranportados. Justifique a corretude e complexidade de tempo do algoritmo.

24. Suponha que foi inaugurado um sambódromo em Campinas e foram construídas m camarotes, dado pelo conjunto $C = \{1, \ldots, m\}$. O camarote $j \in C$ tem uma capacidade máxima para d_j pessoas que poderão usá-lo para assistir o desfile de carnaval (o camarote pode ser ocupado por menos que d_j pessoas). Há muitas pessoas interessadas em assistir o carnaval, dado por $P = \{1, \ldots, n\}$. Com isso, a empresa pede que as pessoas deem lances do quanto querem pagar por estar em um certo dia em algum camarote. Cada pessoa $i \in P$ deseja ser atribuído a até d_i camarotes (pode ser atribuído a menos camarotes, mas não quer mais que d_i camarotes). Sabendo que a pessoa $i \in P$ deu um lance* de a_{ijk} reais para estar no camarote $j \in C$ no dia $k \in D$, faça um programa de tempo polinomial que encontra uma atribuição que maximiza o valor a ser recebido pela empresa, respeitando as restrições de capacidade e de dias que uma pessoa pode ir. Considere duas variações para este problema:

    - Variante 1: uma pessoas pode ter vários camarotes no mesmo dia.
    - Variante 2: uma pessoa pode ter no máximo um camarote por dia.

    Para cada variante, apresente seu algoritmo e justifique porque ele está correto e porque é resolvido em tempo polinomial. Este tipo de problema ocorre muito na alocação de recursos, seja computacionais ou não. Por exemplo, o caso de propagandas com restrições, como de horários.

25. Para algumas formulações em Programação Linear Inteira (PLI) envolvendo conectividade, teremos que resolver alguns problemas de corte mínimo, seja em grafos dirigidos ou não-dirigidos. Assim, apresente algoritmos de tempo polinomial para resolver os seguintes problemas abaixo. Para cada caso, justifique a corretude e complexidade de tempo do algoritmo.

    - São dados: um grafo não-dirigido $G = (V, E)$, uma função $c : E \to \mathbb{Q}^+$ e um vértice $r \in V$. O objetivo é encontrar uma partição do conjunto de vértices V em duas partes, $(S, \overline{S})$, tal que $r \in S$ e $\overline{S} \neq \emptyset$, e $\sum_{e \in \delta(S)} c_e$ é mínimo.
    - São dados: um grafo não-dirigido $G = (V, E)$, uma função $c : E \to \mathbb{Q}^+$ e dois vértices distintos $s, t \in V$. O objetivo é encontrar uma partição do conjunto de vértices V em duas partes, $(S, \overline{S})$, tal que $s \in S$ e $t \in \overline{S}$, e $\sum_{e \in \delta(S)} c_e$ é mínimo.
    - São dados: um grafo não-dirigido $G = (V, E)$ e uma função $c : E \to \mathbb{Q}^+$. O objetivo é encontrar uma partição do conjunto de vértices V em duas partes, $(S, \overline{S})$, tal que $S \neq \emptyset$ e $S \neq V$, e $\sum_{e \in \delta(S)} c_e$ é mínimo.
    - São dados: um grafo dirigido $G = (N, A)$, uma função $c : A \to \mathbb{Q}^+$ e um vértice $r \in V$. O objetivo é encontrar uma partição do conjunto de vértices V em duas partes, $(S, \overline{S})$, tal que $r \in S$ e $\overline{S} \neq \emptyset$, e $\sum_{e \in \delta^+(S)} c_e$ é mínimo.

*O lance a_{ijk} pode ser 0 caso a pessoa i não puder/quiser ir no dia/camarote j.

- São dados: um grafo dirigido $G = (N, A)$, uma função $c : A \to \mathbb{Q}^+$ e dois vértices distintos $s, t \in V$. O objetivo é encontrar uma partição do conjunto de vértices V em duas partes, $(S, \overline{S})$, tal que $s \in S$ e $t \in \overline{S}$, e $\sum_{e \in \delta^+(S)} c_e$ é mínimo.
- São dados: um grafo dirigido $G = (N, A)$ e uma função $c : E \to \mathbb{Q}^+$. O objetivo é encontrar uma partição do conjunto de vértices V em duas partes, $(S, \overline{S})$, tal que $S \neq \emptyset$ e $S \neq V$, e $\sum_{e \in \delta^+(S)} c_e$ é mínimo.