Universidade Estadual de Campinas

Faculdade de Engenharia Elétrica e de Computação

Otimização Inteira e Combinatoria – Turma A – Prova | **1S/2025**

Nome

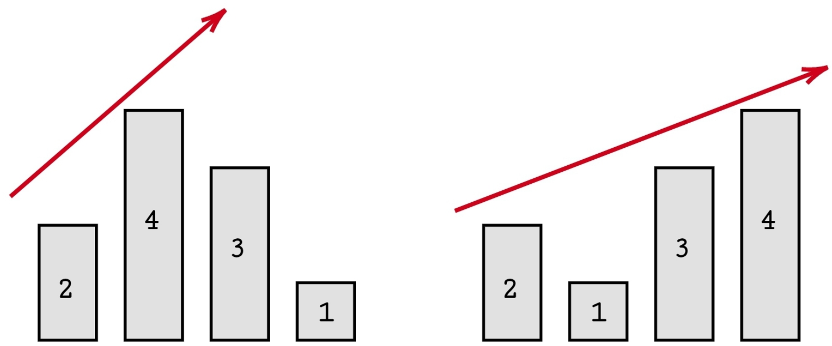
RA

Instruções

- Esta prova tem **05 questões**, das quais **apenas 04 devem ser resolvidas**.
- Cada questão deve ser resolvida, de forma **organizada**, **clara** e **formal**. Estes critérios fazem parte da avaliação.
- É permitida a consulta a livros e outros materiais didáticos.
- Não é permitida a comunicação com qualquer pessoa que não faça parte da equipe de ensino.
- É permitido o uso de softwares apenas para fins de validação, como indicado nas questões. Todas as resoluções devem independer do uso destes.
- Qualquer tentativa de fraude, se detectada durante ou após a realização da prova, implicará em nota **zero** para todos os envolvidos, além das penalidades disciplinares previstas no Regimento Geral da Unicamp.
- Indique, em cada questão, o tempo utilizado na sua resolução.

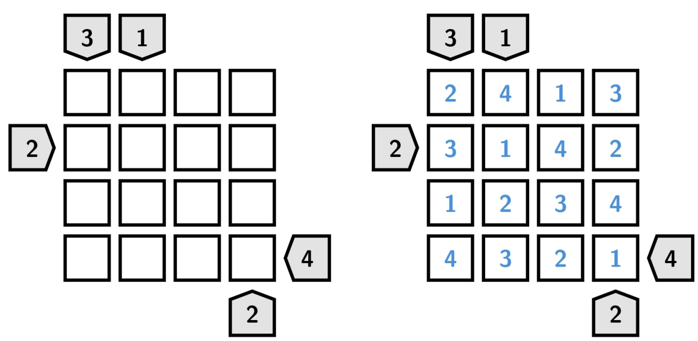
► **Questão 1: (Arranha-céus!)** O jogo "Arranha-Céus" é um desafio matemático que se desenvolve sobre uma grade quadrada $n \times n$. O objetivo do jogo é preencher essa grade com números inteiros consecutivos de 1 até n de forma que cada número represente a altura de um arranha-céu, respeitando duas condições principais:

- **Quadrado latino:** cada linha e cada coluna deve conter todos os números de 1 até n exatamente uma vez, sem repetições. Uma matriz $n \times n$ que satisfaz essa propriedade é chamada de *quadrado latino*.
- **Pistas visuais nas bordas:** ao redor da grade, são fornecidos números que representam quantos arranha-céus podem ser vistos a partir daquela posição específica (esquerda, direita, cima ou baixo). Um prédio mais alto esconde da vista os menores que estiverem atrás dele, assim como acontece com arranha-céus reais em uma fileira. Veja os exemplos nas figuras a seguir.



Prédios vistos: 2 **Prédios vistos: 3**

A figura a seguir inclui uma instância inicial de um jogo 4×4 e sua solução. Observe como as condições descritas acima são respeitadas. Novas instancias com números de dicas e dimensões variadas podem ser geradas em `https://www.puzzle-skyscrapers.com`.



Nosso objetivo nesta questão é modelar esse jogo como um problema de otimização linear inteira para automatizar a sua resolução. Para tanto, iremos modelar dois subproblemas separadamente e depois juntá-los para obter o modelo final para o jogo Arranha-Céus.

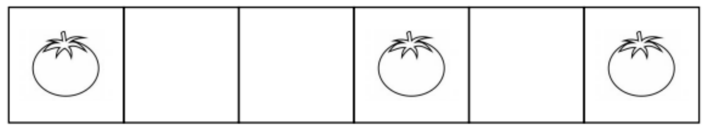
(a) **Quadrados latinos.** Proponha um modelo de otimização linear inteira para resolver o jogo *quadrados latinos* $n \times n$. Suponha, para o seu modelo, que algumas posições do quadrado estão preenchidas inicialmente. Indique claramente as variáveis de decisão adotadas e justifique as restrições utilizadas.

(b) **Prédios vistos em uma fileira.** Suponha que uma fileira de n prédios de alturas inteiras e distintas de 1 até n seja descrita pela sequência $h_1, \cdots, h_n$ das suas alturas. Proponha um modelo de otimização inteira cuja solução forneça o número de prédios que seriam vistos por uma pessoa próxima ao primeiro prédio da fileira. Indique as variáveis adotadas e justifique as restrições utilizadas.

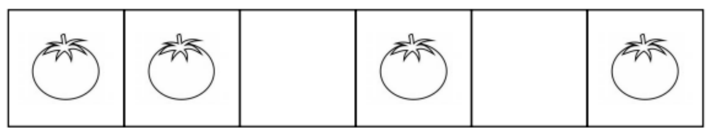
(c) **Modelo Final.** Adapte as ideias usadas nos itens anteriores para propor o modelo de otimização inteira que resolve o desafio dos Arranha-Céus. Valide o seu modelo com ao menos 3 exemplos gerados na `url` sugerida.

► **Questão 2: (Plantação de tomates!)** Você está plantando pés de tomate em um jardim, que tem n espaços dispostos em uma linha. Diferentes locais do jardim resultam em tomates de diferentes qualidades: assuma que cada espaço i do jardim resulta em um tomate com **deliciosidade** $T(i)$, sendo $T(i)$ um inteiro positivo, $i \in \{1, \cdots, n\}$. Além disso, você não pode plantar dois pés de tomate em locais adjacentes, pois eles competirão por recursos e irão murchar. Seu objetivo é plantar os tomates para a fim de obter a maior **deliciosidade total** (somando a deliciosidade de todos os pés de tomate plantados).

Por exemplo, se os índices de deliciosidade da linha disponível para plantação for $T = [21, 4, 6, 20, 2, 5]$, os tomates devem ser plantados no padrão



para fornecer a deliciosidade total de $21 + 20 + 5 = 46$. Neste mesmo exemplo, você **não** pode plantar os tomates no padrão



porque existem dois pés de tomate em posições adjacentes.

(a) **Estratégia gulosa.** Considere a seguinte estratégia gulosa para resolver esse problema: *"enquanto houver espaço disponível para plantar um pé de tomate, plante naquele que oferece a maior deliciosidade"*. Lembre-se de que um espaço só está disponível para plantio se seus vizinhos imediatos não estiverem ocupados. Construa um exemplo em que essa estratégia é ótima e outro em que ela não produz a solução ótima.

(b) **Programação dinâmica.** Construa um algoritmo recursivo que resolva esse problema, isto é, forneça a deliciosidade ótima total e o padrão ótimo de plantio para um dado vetor de deliciosidades T. Explique como a subestrutura ótima é explorada pelo seu algoritmo. A solução pode ser resolvida de forma iterativa e ordenada para melhorar a eficiência? Justifique.

(c) **Programação linear inteira.** Modele esse problema como um problema de otimização linear inteira (PLI). Detalhe a escolha das variáveis de decisão e das restrições.

(d) Valide as duas abordagens ótimas desenvolvidas para os vetores

$$T_1 = [5,\ 12,\ 10,\ 7,\ 15,\ 10,\ 11,\ 5,\ 8,\ 10]$$

e

$$T_2 = [10,\ 12,\ 5,\ 12,\ 20,\ 18,\ 5,\ 3,\ 2,\ 8].$$

► **Questão 3: (Dois problemas fundamentais em grafos bipartidos.)** Lembre que um grafo bipartido $G = (V, E)$ é um grafo cujos vértices podem ser divididos em dois conjuntos disjuntos V_1 e V_2 de forma que toda aresta a de E seja da forma $a = \{u, w\}$ com $u \in V_1$ e $w \in V_2$, isto é, toda aresta conecta nós de partições diferentes. Grafos bipartidos surgem em diversos problemas de interesse prático, como é o caso do problema de atribuição. A seguir, apresentamos dois tipos de conjuntos importantes em grafos bipartidos.

Definição. *Um emparelhamento de G é um subconjunto de arestas em E que são disjuntas, isto é, um conjunto de arestas em E que não têm vértices em comum. Uma cobertura de G é um conjunto de nós que juntos tocam todas as suas arestas.*

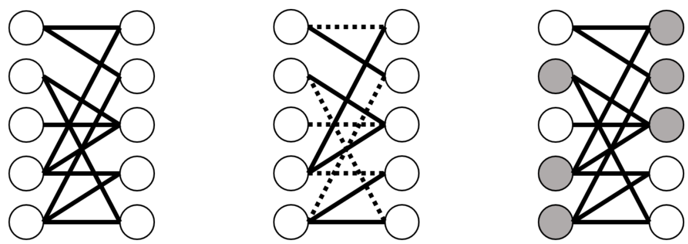


Figura 1: Um exemplo de grafo bipartido (à esquerda), com um emparelhamento (linhas tracejadas, ao centro) e uma cobertura (nós cinzas, à direita) destacados.

Com esta definição, podemos definir dois problemas de grande interesse de otimização combinatória:

- **Problema da Cobertura Mínima:** Encontrar uma cobertura $X \subset V$ tal que não exita uma cobertura $S \subset V$ tal que $|S| < |X|$.
- **Problema do Emparelhamento Máximo:** Encontrar um emparelhamento $X \subset E$ tal que não exista um emparelhamento $Y \subset E$ tal que $|Y| > |X|$.

Nesta questão, provaremos o importante Teorema de König:

Teorema. Se G for um grafo bipartido, a cardinalidade do emparelhamento máximo é igual à cardinalidade da cobertura mínima.

Os itens a seguir servem de guia nesta tarefa.

(a) Com base nos conjuntos definidos no problema, justifique a seguinte afirmação: "a cardinalidade de qualquer emparelhamento é menor do que ou igual à cardinalidade de qualquer cobertura".

(b) Formule um PLI que modele o problema de emparelhamento máximo. Mostre que este problema é um caso particular do problema de atribuição. Como a matriz associada às restrições do problema de atribuição é totalmente unimodular, conclua que as restrições de integralidade podem ser removidas sem alterar a solução ótima. Chame estes problemas de $\mathcal{P}_M$ (emparelhamento máximo) e $\mathcal{P}_{MR}$ (relaxação PL do emparelhamento máximo).

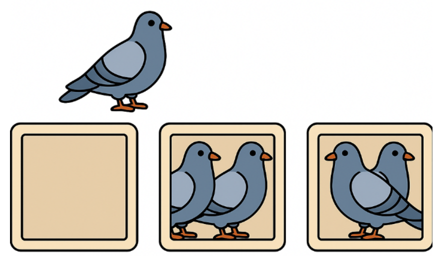
(c) Formule um PLI para o problema de cobertura mínima. Mostre que este problema também possui uma matriz totalmente unimodular. Conclua que as restrições de integralidade podem ser removidas sem alterar a solução ótima. Estes problemas serão $\mathcal{P}_C$ (cobertura mínima) e $\mathcal{P}_{CR}$ (relaxação PL do problema de cobertura mínima).

(d) Mostre que $\mathcal{P}_{CR}$ é o problema dual (ou seu equivalente) associado ao problema $\mathcal{P}_{MR}$. Conclua, por dualidade forte que os valores ótimos são iguais e que o Teorema de König é válido.

► **Questão 4: (O Princípio da Casa dos Pombos.)** O **Princípio da Casa dos Pombos**, também conhecido como **Princípio de Dirichlet**, é uma ideia fundamental da matemática discreta, com amplas aplicações em problemas de contagem e de lógica. Ele estabelece que, ao distribuir mais objetos do que recipientes disponíveis, pelo menos um recipiente conterá mais de um objeto. Esse raciocínio pode ser formalizado da seguinte maneira: o problema

(P) *Alocar (n + 1) pombos em n casas de forma que nenhuma casa seja compartilhada por dois pombos*

não tem solução. Ou seja, é impossível distribuir $(n+1)$ pombos entre n casas sem que ao menos uma delas receba dois ou mais pombos.



Embora pareça trivial à primeira vista, esse princípio (proposto por Peter Gustav Lejeune Dirichlet no século XIX) é uma poderosa ferramenta em matemática, frequentemente usada em argumentos por contradição ou para garantir a existência de repetições, coincidências ou padrões.

Nessa questão, nosso objetivo é provar o princípio de Dirichlet usando técnicas de otimização.

(a) Formule **(P)** como um problema de otimização linear inteira com dois tipos de restrições:

(r1) aquelas que expressam a condição de que cada pombo deve ser alocado a uma casa;

(r2) aquelas que expressam a condição de que cada par de pombos deve estar alocado em casas diferentes.

Para tanto, use as variáveis binárias:

$$x_{ij} = \begin{cases} 1, & \text{se pombo } i \text{ está alocado à casa } j \\ 0, & \text{c.c.} \end{cases}$$

(b) Podemos mostrar que esse problema é infactível por meio de uma sequência de desigualdades válidas, baseadas nas restrições **(r2)**. De fato, cada uma das restrições **(r2)** está associada a um par (i, k) de pombos e a uma casa j, sendo da forma:

$$x_{ij} + x_{kj} \leqslant 1.$$

Se somarmos as restrições associadas aos pares de pombos (i, k), (k, ℓ) e (i, ℓ), temos que

$$x_{ij} + x_{kj} + x_{\ell j} \leqslant 1$$

é uma desigualdade válida para **(P)**. Além disso, essa desigualdade torna as desigualdades $x_{ij} + x_{kj} \leqslant 1$, $x_{kj} + x_{\ell j} \leqslant 1$ e $x_{ij} + x_{\ell j} \leqslant 1$ redundantes. Justifique essas duas afirmações. Generalize esse raciocínio para mostrar que

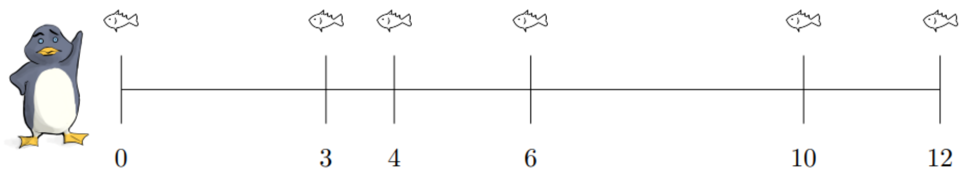
$$\sum_{i=1}^{n+1} x_{ij} \leqslant 1, \quad j = 1, \cdots, n,$$

são desigualdades válidas para **(P)**.

(c) Organize as variáveis de decisão em uma matriz $X = (x_{ij})$, com $(n+1)$ linhas e n colunas. Interprete as restrições **(r1)** e as restrições válidas construídas ao final do exercício anterior como condições sobre os elementos da matriz X. Conclua que esse conjunto de restrições é incompatível no caso inteiro.

▶ **Questão 5: (Pudim, o pinguim comilão.)** Pudim, o pinguim comilão, está fazendo uma caminhada de t km pela Antártica. Ele precisa fazer lanchinhos ao longo do caminho, mas só pode comer quando há um buraco de pesca disponível para pescar. Pudim consegue andar, no máximo, m km entre as refeições e há n buracos de pesca ao longo da rota.

Ao começar sua caminhada, Pudim recebe um vetor F, em que $F(i)$ indica a distância desde o ponto de partida até o i-ésimo buraco de pesca. Há exatamente n buracos de pesca ao longo da rota, sendo que o primeiro está no início ($F(1) = 0$) e o último no final do percurso ($F(n) = t$). Por exemplo, o vetor $F = [0, 3, 4, 6, 10, 12]$ representa os seguinte cenário:



Pudim deseja parar o menor número de vezes possível, dado que ele pode andar até m km sem comer. Como ele é comilão, ele já começa a caminhada comendo e também sempre finaliza a caminhada com um bom lanchinho. No exemplo acima, tomando-se $m = 4$ km, temos que Pudim fará 5 lanches em sua caminhada nos pontos 0, 4, 6, 10 e 12 km.

(a) Estabeleça uma condição básica para que esse problema admita solução, considerando o vetor F e a distância máxima m que Pudim consegue caminhar sem se alimentar. Essa hipótese será assumida nos itens a seguir.

(b) Modele uma instância genérica desse problema como um problema de otimização linear inteira. Suponha que sejam dados o vetor F e a distância máxima entre paradas m. Descreva as variáveis de decisão e as restrições adotadas.

(c) Pudim, além de comilão, é especialista em algoritmos gulosos. A estratégia adotada por Pudim é a seguinte: *Pudim irá aguentar o máximo que puder, e só vai parar para pescar se perceber que não conseguirá chegar até o próximo buraco de pesca.* Mostre que essa escolha é ótima. Dica: prove por indução.