# Trabalho 03 - Final

Pedro Hokama

XMCO02 2024s2

**Data escolha problema: 21/11/2024**
**Data entrega: 06/12/2024 (por e-mail)**
**Data apresentação (presencial): (09 ou 12)/12/2024 (a definir)**

Neste trabalho você deverá escolher um problema, que seja NP-Difícil e que não seja muito simples de ser modelado. A ideia é que não seja um problema clássico (Mochila, Vertex-Cover, Conjunto Independente, etc) já que esses problemas já foram muito explorados. Você deverá enviar seu problema por e-mail até 21/11, para aprovação. (Dica: também não pode ser muito complexo senão não vai dar tempo de resolver)

Depois você deverá elaborar e implementar 2 desses 3 métodos de resolução:

- Um algoritmo branch-and-bound
- Uma formulação em Programação linear inteira usando um resolvedor como o SCIP do OR-Tools.
- Uma formulação em Programação por restrições usando um resolvedor como o CP-SAT do OR-Tools.

O uso do OR-Tools não é mandatório, você pode utilizar outro resolvedor, como a versão acadêmica (e não a community) do CPLEX, ou o Gurobi. Em qualquer deles o docente não é capaz de oferecer suporte técnico sobre o uso e instalação dos mesmos.

Depois você deverá obter ou elaborar instâncias para o problema escolhido, e executar os seus algoritmos de forma a responder as seguintes perguntas:

- Para as instâncias que os dois métodos conseguem resolver. Qual é mais rápido?
- Quais os maiores tamanhos de instâncias que cada método consegue resolver em 600 segundos.
- Caso eles não resolvam de maneira ótima até 600 segundos, eles retornam uma solução viável?
- Como se comparam as soluções obtidas pelos dois métodos quando eles não chegam na solução ótima.
- Se possível, podemos dizer o quão longe estamos da solução ótima?

Você deverá elaborar gráficos e tabelas que ajudem a evidenciar as respostas para essas perguntas. Por fim você deve elabora um relatório de no máximo 4 páginas, com, pelo menos:

- Descrição do problema, de maneira precisa e formal.
- Revisão bibliográfica, mostrando artigos onde esse problema aparece e como já foi resolvido.
- Métodos e modelos elaborados.
- Resultados.

Abaixo uma sugestão de alguns problemas que podem ser escolhidos, mas preferencialmente deve ser um problema fora dessa lista, ou se preferir, modificações deles.

# 1 Problema dos Convidados na Festa

Suponha que você queira dar uma festa interessante, e precisa escolher quem serão seus convidados. Você sabe que duas pessoas que se conhecem tem uma certa afinidade mensurável, que pode ser inclusive negativa. Você precisa escolher $k$ convidados de forma que: obtenha o máximo de afinidade, para quaisquer duas pessoas na sua festa, tenha algum caminho de conhecidos entre elas. Formalmente:

**Problema do Subgrafo $k$-Induzido conexo de Peso Máximo** Dado um grafo $G = (V, E)$, cada aresta $e \in E$ tem um peso $w : E \to \mathbb{R}$ e um inteiro $k \in \{1, \ldots, |V|\}$. Encontrar um subconjunto $V' \subseteq V$, com $|V'| = k$, tal que o subgrafo induzido seja conexo e o peso das arestas seja maximizado.

Note que o Grafo não é Completo, então duas pessoas que não se conhecem não tem uma aresta entre elas.

Referência: https://www.algorithmics.informatik.uni-mainz.de/files/2016/05/Algorithms-for-the-Maximum-Weight-Connected-k-Induced-Subgraph-Problem.pdf

# 2 Problema do Escalonamento de Horários com Distanciamento Social

Suponha que em um determinado laboratório existam diversas bancadas, cada bancada possui equipamentos específicos e está em uma determinada coordenada $(x, y)$. Os pesquisadores desse laboratória precisam usar equipamentos em momentos específicos e você precisa atribuir a cada um deles uma bancada que possa atende-los, porém que dois pesquisadores que estejam trabalhando no mesmo momento não estejam a menos de $z$ de distância.

Por exemplo: "José precisa usar alguma das bancadas A, B, E ou F das 13hs até as 17hs, Maria precisa usar alguma das bancadas A, C, F ou H das 16hs até as 18hs", etc. Como alocar os pesquisadores para respeitem o distanciamento social.

# 3 Problema do Bombeiro

Suponha que um incêndio começou em um local na floresta, e existem $D$ equipes de bombeiros, a cada instante de tempo cada equipe de bombeiro consegue defender um local diferente mas o fogo se alastra para os locais adjacentes. O fogo para quando todos os locais adjacentes ao fogo já estiverem defendidos ou queimados. Seu objetivo é descobrir quais vértices defender para minimizar o número de locais queimados.

Referência: https://www.ic.unicamp.br/~cid/Problem-instances/Firefighter-in-Graphs/
https://doi.org/10.1111/itor.12638

# 4 Problema da Cobertura por MultiConjuntos

Dado uma coleção de Multiconjuntos $\mathcal{C}$, cada multiconjunto $C \in \mathcal{C}$ com um custo $V_C$ e uma quantidade máxima $d_C$, e um multiconjunto objetivo $O$. Uma solução $\mathcal{S}$ são algumas cópias de alguns multiconjunto de $\mathcal{C}$, tal que: a quantidade de cópias de cada multiconjunto $C$ não ultrapasse $d_C$, e que $O \subseteq \bigcup_{S \in \mathcal{S}} S$.

# 5 Problema de Alocação de portfólio

O problema de alocação de portfólio consiste em determinar como distribuir um orçamento fixo entre $N$ ativos disponíveis para maximizar o retorno esperado do portfólio, sujeito a restrições operacionais e de risco. Cada ativo possui um retorno esperado e um risco associado, medido por seu desvio-padrão. Além disso, os investimentos em cada ativo são limitados por valores mínimos e máximos pré-definidos. Para promover diversificação, não mais do que $M$ ativos podem ser selecionados, e o montante investido em cada ativo é permitido somente se ele for incluído no portfólio.

Além das restrições de diversificação e orçamento, é necessário controlar o risco do portfólio. A soma ponderada dos riscos individuais dos ativos não pode exceder um limite máximo definido pelo gestor.

Em algumas situações, podem ser impostas condições adicionais, como exclusões mútuas entre ativos ou limites mínimos de investimento por setor econômico. O objetivo final é encontrar uma alocação que maximize o retorno esperado enquanto atende a todas as restrições operacionais e estratégicas.