Problemas de Coberturas Justas e Máximas

Ana Paula S. Dantas
Cid C. de Souza
Zanoni Dias

Roteiro

Introdução

Introdução

- Algoritmos são poderosas ferramentas para a tomada de decisão
- Largamente utilizados nos últimos anos
  - ↳ concessão de liberdade condicional [1]
  - ↳ seleção de alunos em universidades norte-americanas [2]
  - ↳ alocação de recursos [3]
  - ↳ predição de emergências médicas em UTIs [4]
- Contudo, essas ferramentas não estão isentas de problemas como a discriminação

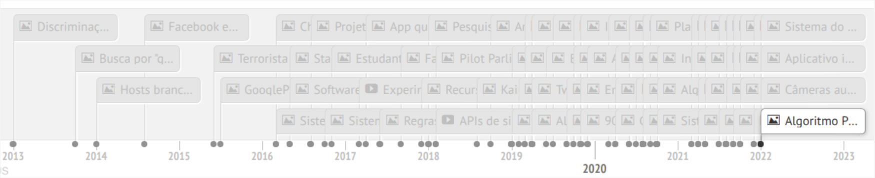
Introdução

Exemplos de casos onde algoritmos demonstraram um viés preconceituoso

- Concessão de liberdade condicional
  - ↳ sistema usado para atribuir riscos de reincidência criminal era duas vezes mais provável de indicar um réu negro com alto risco
- Alocação de recursos
  - ↳ oferecimento de serviços de entrega no mesmo dia em plataforma online de vendas excluiu bairros cuja população era predominante negra

Introdução

- Casos conhecidos como *Injustiça Algorítmica* ou *Racismo Algorítmico*
- Linha do tempo com notícias denunciando casos de racismo algorítmico [5]



- Casos recentes:
  - Google não permite que patrocinadores busquem por público alvo no YouTube com o tema do movimento *Black Lives Matter*
  - Twitter reconheceu que o algoritmo de recorte imagens postadas na plataforma privilegiava pessoas brancas

Introdução

- Em resposta, a comunidade científica vem estudando como esses casos podem ser evitados
- A ideia de que os algoritmos devem continuar ignorantes a características como raça e gênero foi contestada em um trabalho de Kleinberg *et al.*[6]
  - ↳ modelos de predição de sucesso usados para seleção de alunos em universidades norte-americanas
  - ↳ mostrou que o desempenho foi melhor quando o algoritmo usou características de raça explicitamente
- Na mesma linha, Lin *et al.* [7] desenvolveram métodos para encontrar regiões de uma base de dados com pouca cobertura
  - ↳ mostraram através de experimentos, que a inclusão explícita de amostras pertencentes às áreas sub-representadas melhorou o modelo

Introdução

- Considerando a ideia de que para obtermos modelos mais justos, precisamos incluir amostras de todas as classes, Asudeh *et al.* [8] definiram um problema de otimização que considera critérios de justiça
  - ↳ onde devem ser selecionadas amostras de modo que a soma dos pesos seja máxima e que todas as classes de características sejam igualmente representadas
  - ↳ esse problema foi chamado de Cobertura Justa Máxima (ou *Fair Maximum Coverage* – FMC)
  - ↳ além de integração de dados, também possui aplicações em localização de facilidades

Definições

Notação

- $\mathcal{U}$ é o conjunto universo, formado por elementos u_j
- $\mathcal{S}$ é a família de subconjuntos, formada por elementos $\mathcal{S}_\ell \in \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_n\}$
- uma cobertura é um subconjunto $Z \subseteq \mathcal{S}$ tal que, $\bigcup_{\mathcal{S}_\ell \in Z} \mathcal{S}_\ell = \mathcal{U}$
- k-cobertura é um subconjunto de tamanho k de $\mathcal{S}$
  - um elemento u_j é coberto se existe um subconjunto $\mathcal{S}_\ell$ na k-cobertura tal que $u_j \in \mathcal{S}_\ell$
  - note que, em uma k-cobertura não é necessário que todos os elementos sejam cobertos

Notação

- $\mathcal{C}$ é um conjunto de χ cores distintas, representadas pelos inteiros $\{1, 2, \ldots, \chi\}$
- uma coloração C é uma função $C : \mathcal{U} \to \mathcal{C}$ sobre os elementos do conjunto universo
- uma k-cobertura é dita justa, se o número de elementos da cor c cobertos é igual ao número de elementos da cor d cobertos, para todo $c, d \in \mathcal{C}$

Definição Formal do Problema

Tratamos o FMC como um problema de cobertura por conjuntos:

Problema de Cobertura Justa e Máxima

ENTRADA: Conjunto universo $\mathcal{U}$, função de coloração C, família de subconjuntos $\mathcal{S}$, função de pesos w para os elementos de $\mathcal{U}$ e um inteiro k.

OBJETIVO: Encontrar uma k-cobertura justa, tal que a soma dos pesos dos elementos cobertos é máxima.

Exemplo de uma Instância

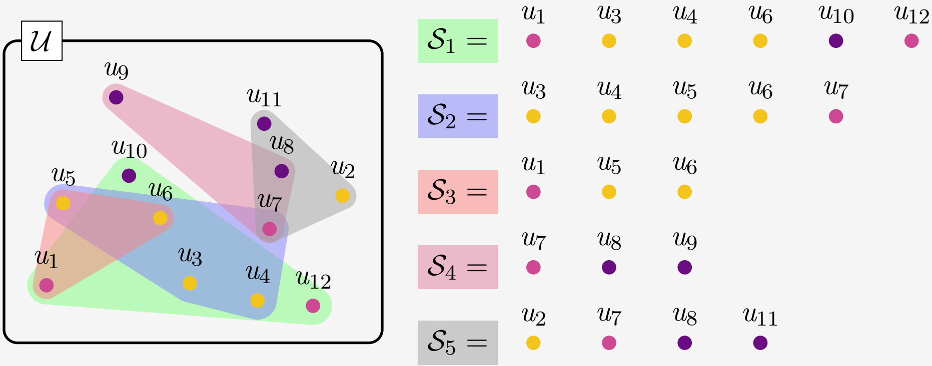


Figura: Exemplo de conjunto universo $\mathcal{U}$ (esquerda) e família de subconjuntos $\mathcal{S}$ (direita). Os elementos de $\mathcal{U}$ são coloridos com três cores e os elementos de $\mathcal{S}$ são subconjuntos de $\mathcal{U}$.

Exemplo de uma Instância

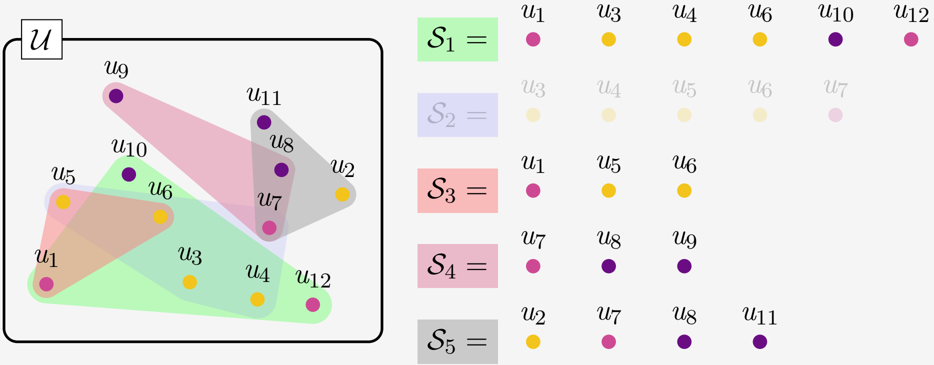


Figura: Exemplo de conjunto universo $\mathcal{U}$ (esquerda) e família de subconjuntos $\mathcal{S}$ (direita). Os elementos de $\mathcal{U}$ são coloridos com três cores e os elementos de $\mathcal{S}$ são subconjuntos de $\mathcal{U}$.

Exemplo de uma Instância

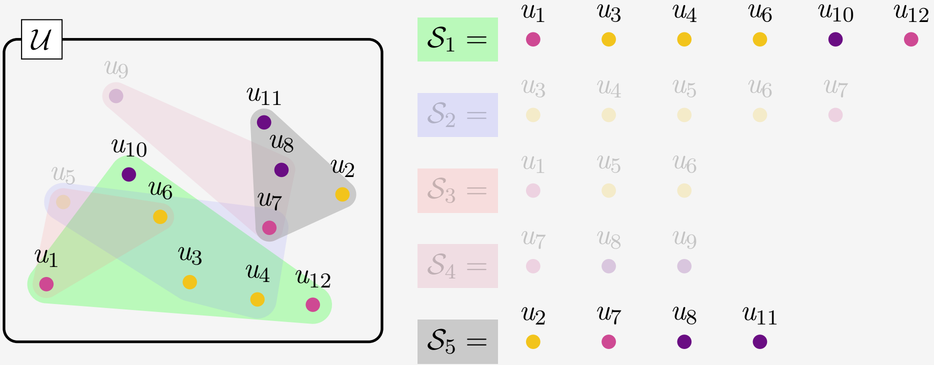


Figura: Exemplo de conjunto universo $\mathcal{U}$ (esquerda) e família de subconjuntos $\mathcal{S}$ (direita). Os elementos de $\mathcal{U}$ são coloridos com três cores e os elementos de $\mathcal{S}$ são subconjuntos de $\mathcal{U}$.

Exemplo de uma Instância

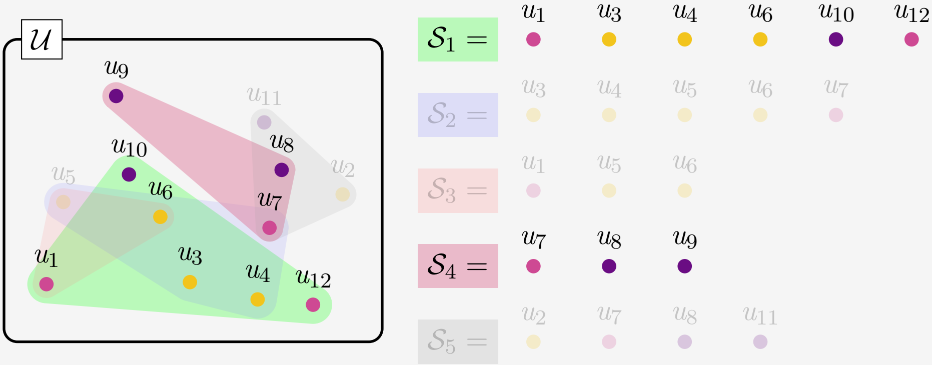


Figura: Exemplo de conjunto universo $\mathcal{U}$ (esquerda) e família de subconjuntos $\mathcal{S}$ (direita). Os elementos de $\mathcal{U}$ são coloridos com três cores e os elementos de $\mathcal{S}$ são subconjuntos de $\mathcal{U}$.

Revisão Bibliográfica

Revisão Bibliográfica – FMC

- NP-difícil [8]
- possui algoritmo com fator de aproximação $(1 - 1/f)^f$
  - f é a frequência máxima de um elemento do conjunto universo nos subconjuntos

Revisão Bibliográfica – Trabalhos Relacionados

- Problemas de Cobertura por Conjuntos
  - Selecionar uma subfamília de modo que todos os elementos são cobertos e o custo é mínimo
  - *NP*-difícil [9]
  - A versão de otimização, onde a soma dos pesos dos elementos cobertos deve ser minimizada, não pode ser aproximado com um fator logarítmico, a menos que $P = NP$ [10]
  - Abordagens recentes: Meta-heurísticas bio-inspiradas como Colônia de Formigas [11] e Algoritmo de Otimização baseado em Nuvens de Gafanhotos [12]
  - Modelo matemático que tem como objetivo remover a preocupação com tratamentos de soluções inviáveis e soluções redundantes [13]

Revisão Bibliográfica – Trabalhos Relacionados

- Cobertura Máxima por k-Conjuntos
  - Selecionar k subconjuntos de modo a maximizar os ganhos
  - NP-difícil [9]
  - Algoritmo de H_k-aproximação, com $H_k = \sum_{i=1}^{k} \frac{1}{i}$ [14]
  - Melhorias no fator de aproximação:
    - $H_k - \frac{k-1}{8k^9}$, versão com funções de peso arbitrárias [15]
    - $H_k - \frac{196}{390}$, versão sem pesos [16]

Revisão Bibliográfica – Trabalhos Relacionados

- Problema de Cobertura Justa e Robusta em Grafos [17]
  - Um subconjunto de vértices é escolhido para cobrir os demais
  - O número de vértices cobertos de um mesmo grupo deve ser pelo menos uma fração Z do tamanho do grupo no pior caso
  - O objetivo é cobrir o maior número de vértices, sob a hipótese de que alguns vértices na cobertura podem falhar
  - Principais Resultados: NP-difícil; análises do custo de inclusão dos critérios de justiça; método baseado em MILP com resultados competitivos em experimentos com bases reais

Modelagens para o Problema

Modelo de Programação Linear Inteira

Justiça como Igualdade

$$
\begin{aligned}
\max \quad & \sum_{j=1}^{m} w(u_j) x_j && && (1)\\
\text{sujeito a} \quad & \sum_{u_j \in \mathcal{S}_\ell} y_\ell \geq x_j && \forall\, u_j \in \mathcal{U} && (2)\\
& y_\ell \leq x_j && \forall\, u_j \in \mathcal{S}_\ell, \forall\, \mathcal{S}_\ell \in \mathcal{S} && (3)\\
& \sum_{\ell=1}^{n} y_\ell = k && && (4)\\
& \sum_{u_j \in C_c} x_j = \sum_{u_i \in C_d} x_i && \forall\, c, d \in \mathcal{C},\ \ c < d && (5)\\
& x, y \in \{0, 1\} && && (6)
\end{aligned}
$$

Relaxando Restrições

- Relaxando a restrição de orçamento

$$\sum_{\ell=1}^{n} y_\ell = k \tag{4}$$

- Relaxando as restrições de justiça

$$\sum_{u_j \in C_c} x_j = \sum_{u_i \in C_d} x_i \qquad \forall\, c, d \in \mathcal{C},\ \ c < d \tag{5}$$

- Tratando a justiça como parte do objetivo
- Justiça como proporcionalidade

Modelo de Programação Linear Inteira

Justiça como Igualdade – Relaxando a restrição de orçamento

$$
\begin{aligned}
\max \quad & \sum_{j=1}^{m} w(u_j)x_j - \alpha D && (7)\\
\text{sujeito a} \quad & x_j \leq \sum_{u_j \in \mathcal{S}_\ell} y_\ell \quad \forall\, u_j \in \mathcal{U} && (8)\\
& y_\ell \leq x_j \quad \forall\, u_j \in \mathcal{S}_\ell, \forall\, \mathcal{S}_\ell \in \mathcal{S} && (9)\\
& \sum_{\ell=1}^{n} y_\ell - k \leq D && (10)\\
& k - \sum_{\ell=1}^{n} y_\ell \leq D && (11)\\
& \sum_{u_j \in C_c} x_j = \sum_{u_i \in C_d} x_i \quad \forall\, c, d \in \mathcal{C},\ c < d && (12)\\
& x, y \in \{0, 1\} && (13)
\end{aligned}
$$

Modelo de Programação Linear Inteira

Justiça como Igualdade – Relaxando as restrições de justiça

$$\max \quad \sum_{j=1}^{m} w(u_j)x_j \tag{14}$$

$$\text{sujeito a} \quad x_j \leq \sum_{u_j \in \mathcal{S}_\ell} y_\ell \quad \forall\, u_j \in \mathcal{U} \tag{15}$$

$$y_\ell \leq x_j \quad \forall\, u_j \in \mathcal{S}_\ell, \forall\, \mathcal{S}_\ell \in \mathcal{S} \tag{16}$$

$$\sum_{\ell=1}^{n} y_\ell = k \tag{17}$$

$$\sum_{u_j \in C_{c'}} x_j = \beta \tag{18}$$

Modelo de Programação Linear Inteira

Justiça como Igualdade – Relaxando as restrições de justiça

$$\sum_{u_j \in C_d} x_j \leq \beta + \frac{\tau}{2} \quad \forall\, d \in \mathcal{C},\ \ d \neq c' \tag{19}$$

$$\sum_{u_j \in C_d} x_j \geq \beta - \frac{\tau}{2} \quad \forall\, d \in \mathcal{C},\ \ d \neq c' \tag{20}$$

$$x, y \in \{0, 1\} \tag{21}$$

Modelo de Programação Linear Inteira

Justiça como Igualdade – Justiça como parte do objetivo

$$
\begin{aligned}
\max \quad & \sum_{j=1}^{m} w(u_j)x_j - \sum_{c=1}^{\chi}\sum_{d=1}^{\chi} z_{c,d} && && (22)\\
\text{sujeito a} \quad & x_j \leq \sum_{u_j \in \mathcal{S}_\ell} y_\ell && \forall\, j \in \{1, 2, \ldots, m\} && (23)\\
& y_\ell \leq x_j && \forall\, u_j \in \mathcal{S}_\ell, \forall\, \mathcal{S}_\ell \in \mathcal{S} && (24)\\
& \sum_{\ell=1}^{n} y_\ell = k && && (25)\\
& \sum_{u_j \in C_c} x_j - \sum_{u_i \in C_d} x_i \leq z_{c,d} \quad \forall\, c, d \in \mathcal{C} && && (26)\\
& x, y \in \{0, 1\} && && (27)\\
& z_{c,d} \geq 0 && \forall\, c, d \in \mathcal{C} && (28)
\end{aligned}
$$

Modelo de Programação Linear Inteira

Justiça como Proporcionalidade

$$\max \quad \sum_{j=1}^{m} w(u_j) x_j \tag{29}$$

$$\text{sujeito a} \quad \sum_{u_j \in \mathcal{S}_\ell} y_\ell \geq x_j \qquad \forall\, u_j \in \mathcal{U} \tag{30}$$

$$y_\ell \leq x_j \qquad \forall\, u_j \in \mathcal{S}_\ell, \forall\, \mathcal{S}_\ell \in \mathcal{S} \tag{31}$$

$$\sum_{\ell=1}^{n} y_\ell = k \tag{32}$$

$$\sum_{u_j \in C_c} x_j = q_c \sum_{j=1}^{m} x_j \qquad \forall\, c \in \mathcal{C} \tag{33}$$

$$x, y \in \{0, 1\} \tag{34}$$

Objetivos

Proposta de Trabalho

Objetivo Geral

Desenvolver ferramentas eficientes que possam ser usadas para mitigar problemas de injustiça algorítmica.

Abordagem Experimental

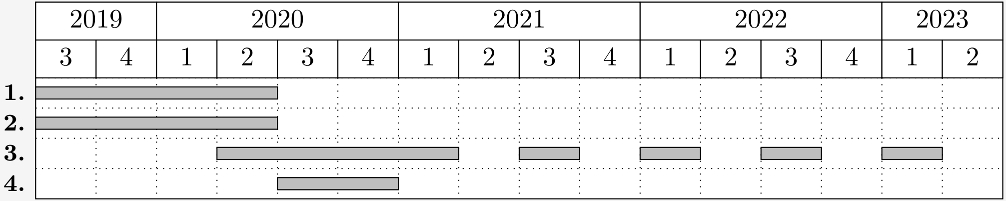
- Modelos de Programação Linear Inteira para os problemas
- Heurísticas e Meta-heurísticas

Abordagem Teórica

- Estudo de complexidade das diferentes versões do problema
- Estudo de limitantes

Cronograma

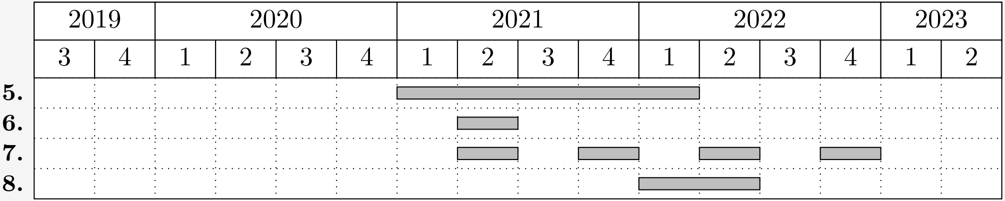
Cronograma



Cronograma de atividades (divididas em trimestres).

1. Obtenção dos créditos obrigatórios em disciplinas do Programa de Pós-Graduação.
2. Participação no Programa de Estágio Didático (PED).
3. Revisão bibliográfica.
4. Escrita do projeto e experimentos iniciais com o modelo de programação linear inteira da literatura.

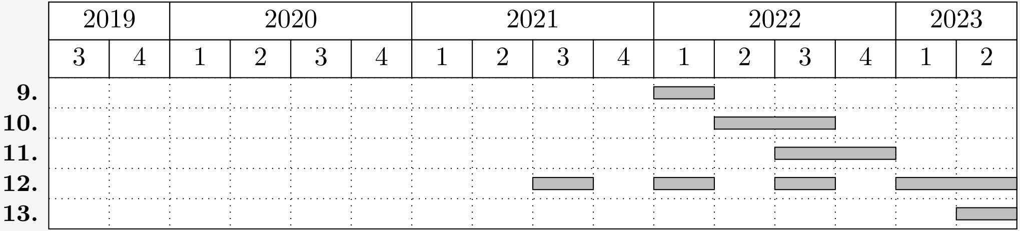
Cronograma



Cronograma de atividades (divididas em trimestres).

5. Obtenção de soluções exatas através de modelos matemáticos para o FMC, FPMC e QFMC, e suas respectivas versões em grafos.
6. Análise de bases de dados reais para as aplicações em integração de dados e localização de instalações.
7. Documentação dos resultados.
8. Análise da complexidade algorítmica dos problemas FPMC e QFMC.

Cronograma



Cronograma de atividades (divididas em trimestres).

9. Exame de Qualificação Específico (EQE).
10. Estudo e desenvolvimento de algoritmos de aproximação para os problemas.
11. Desenvolvimento de abordagens heurísticas para os problemas.
12. Escrita do documento da tese.
13. Defesa da tese de doutorado.

Resultados Preliminares

Experimentos com o Modelo PLI – Instâncias

- Características básicas de uma instância:
  - ↳ família de subconjuntos
  - ↳ tamanho da cobertura definido pelo parâmetro k
  - ↳ coloração dos elementos
- Objetivo: criar instâncias que tenham características controláveis
- Sobre as famílias de subconjuntos:
  - ↳ grafos aleatórios Erdős–Rényi
  - ↳ arestas do grafo representam os elementos do conjunto $\mathcal{U}$
  - ↳ vértices representam os subconjuntos da família $\mathcal{S}$
  - ↳ cada elemento pertence a exatamente dois subconjuntos

Experimentos com o Modelo PLI – Instâncias

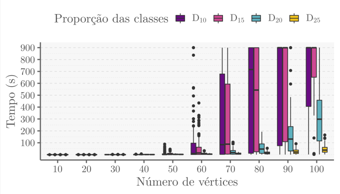
- Sobre o valor do parâmetro k:
  - ↳ número aleatório entre 1 e o número de vértices da instância
    - ↳ maioria das instâncias geradas são inviáveis
  - ↳ execução de um subproblema de otimização que maximiza o número de vértices na cobertura, mantendo a restrição de justiça

Experimentos com o Modelo PLI – Instâncias

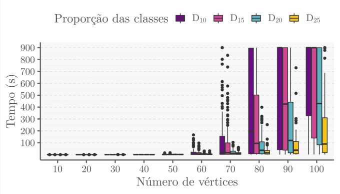
- Sobre a coloração dos elementos (arestas):
  - duas classes de cor
  - arestas igualmente distribuídas entre as duas classes:

    ↳ maioria das instâncias resolvidas antes do limite de tempo de execução
  - desbalanceamento nos tamanhos das classes de cores:
    - $(55, 45), (57.5, 42.5), (60, 40), (62.5, 37.5)$
  - distribuição das arestas
    - uniformemente aleatórias (tipo uniforme)
    - extraindo cliques maximais (tipo cliques)
    - usando uma busca em largura (tipo BFS)

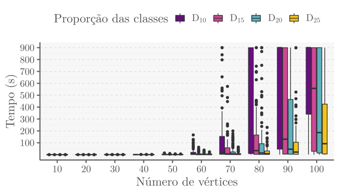
Experimentos – Instâncias



(a) uniforme



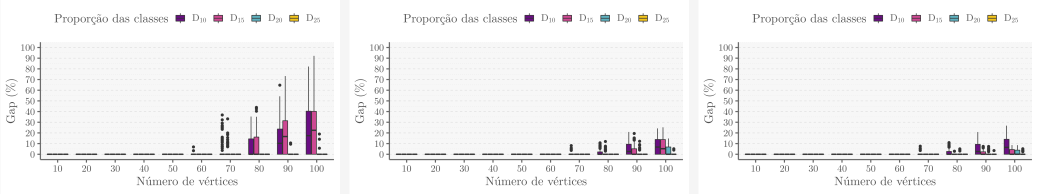
(b) cliques



(c) BFS

Distribuição de tempo de execução para os três tipos de instâncias com o modelo FMC-K.

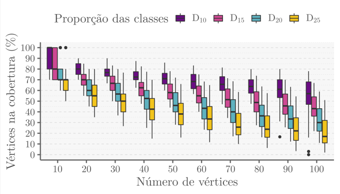
Experimentos – Instâncias



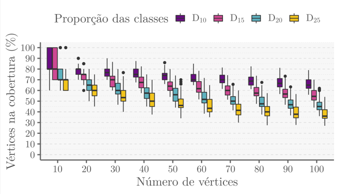
(a) uniforme (b) cliques (c) BFS

Distribuição de gaps para os três tipos de instâncias com o modelo FMC-K

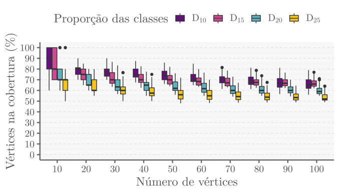
Experimentos – Instâncias



(a) uniforme



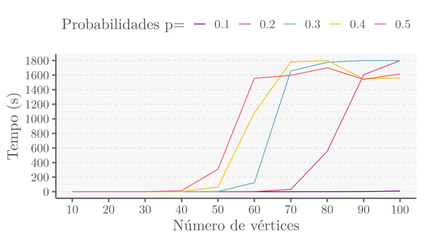
(b) cliques



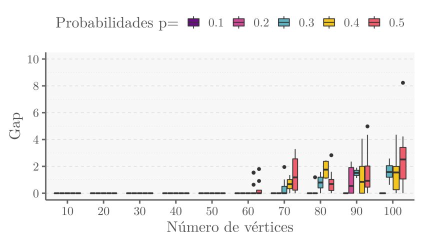
(c) BFS

Distribuição do valor objetivo para os três tipos de instâncias com o modelo FMC-K

Experimentos – Justiça como Igualdade



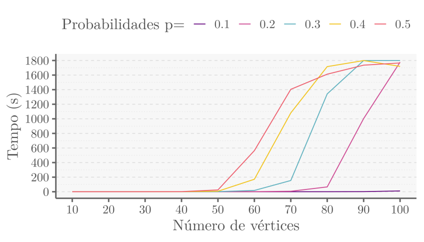
(a) tempo



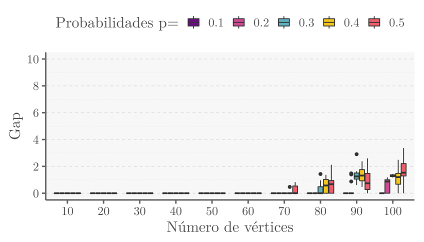
(b) gap

Média de tempo de execução e distribuição de gap para as instâncias do tipo uniforme

Experimentos – Justiça como Igualdade



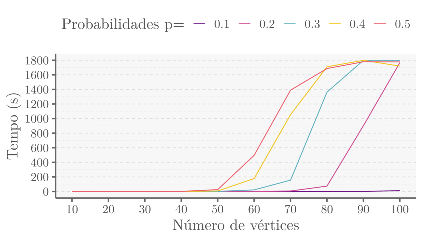
(a) tempo



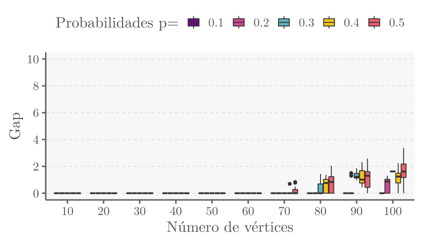
(b) gap

Média de tempo de execução e distribuição de gap para as instâncias do tipo cliques

Experimentos – Justiça como Igualdade



(a) tempo



(b) gap

Média de tempo de execução e distribuição de gap para as instâncias do tipo BFS

Referências I

- A. Christin, A. Rosenblat, and D. Boyd, "Courts and predictive algorithms," in *Data & CivilRight*, (Washington, DC), 10 2015.
- A. Waters and R. Miikkulainen, "GRADE: Machine learning support for graduate admissions," *AI Magazine*, vol. 35, no. 1, pp. 64–64, 2014.
- D. Ingold and S. Soper, "Amazon doesn't consider the race of its customers. should it?," *Bloomberg*, 2016.
- F. Ben Rejab, K. Nouira, and A. Trabelsi, *Health Monitoring Systems Using Machine Learning Techniques*, pp. 423–440. 2014.

Referências II

- T. Silva, "Linha do tempo do racismo algorítmico," *Blog do Tarcízio Silva*, 2020.
- J. Kleinberg, J. Ludwig, S. Mullainathan, and A. Rambachan, "Algorithmic fairness," *AEA Papers and Proceedings*, vol. 108, pp. 22–27, 2018.
- Y. Lin, Y. Guan, A. Asudeh, and H. V. J. Jagadish, "Identifying insufficient data coverage in databases with multiple relations," *Proceedings of the VLDB Endowment*, vol. 13, no. 12, p. 2229–2242, 2020.
- A. Asudeh, T. Berger-Wolf, B. DasGupta, and A. Sidiropoulos, "Maximizing coverage while ensuring fairness: a tale of conflicting objective," pp. 1–38, 2020.

Referências III

- R. M. Karp, *Reducibility among Combinatorial Problems*, pp. 85–103. Springer US, 1972.
- R. Raz and S. Safra, "A sub-constant error-probability low-degree test, and a sub-constant error-probability pcp characterization of np," in *Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC'1997)*, pp. 475–484, ACM, 1997.
- Z.-G. Ren, Z.-R. Feng, L.-J. Ke, and Z.-J. Zhang, "New ideas for applying ant colony optimization to the set covering problem," *Computers & Industrial Engineering*, vol. 58, no. 4, pp. 774 – 784, 2010.

Referências IV

- G. Villavicencio, M. Valenzuela, F. Altimiras, P. Moraga, and H. Pinto, "A k-means grasshopper optimisation algorithm applied to the set covering problem," in *Proceedings of the 10th CSOC: Artificial Intelligence and Bioinspired Computational Methods (CSOC'2020)*, pp. 312–323, Springer International Publishing, 2020.
- N. Bilal, P. Galinier, and F. Guibault, "A new formulation of the set covering problem for metaheuristic approaches," *International Scholarly Research Notices*, vol. 2013.
  10 pages, Article ID 203032, 2013.
- V. Chvatal, "A greedy heuristic for the set-covering problem," *Mathematics of Operations Research*, vol. 4, no. 3, pp. 233–235, 1979.

Referências V

R. Hassin and A. Levin, "A better-than-greedy approximation algorithm for the minimum set cover problem," *SIAM Journal on Computing*, vol. 35, no. 1, pp. 189–200, 2005.

A. Levin, "Approximating the unweighted k-set cover problem: greedy meets local search," *SIAM Journal on Discrete Mathematics*, vol. 23, no. 1, pp. 251–264, 2009.

A. Rahmattalabi, P. Vayanos, A. Fulginiti, E. Rice, B. Wilder, A. Yadav, and M. Tambe, "Exploring algorithmic fairness in robust graph covering problems," vol. 32, Curran Associates, Inc., 2019.

Problemas de Coberturas Justas e Máximas

Ana Paula S. Dantas
Cid C. de Souza
Zanoni Dias