

Avaliação de Desempenho de Filas de Prioridades para Manutenção de Rotas no OLSRd: Um Estudo de Caso da Relação Academia-Comunidade

Diogo M. Gonçalves¹, Saulo Queiroz²

¹Universidade Estadual de Campinas (UNICAMP)

²Universidade Tecnológica Federal do Paraná
(UTFPR-Ponta Grossa)

Roteiro

- Introdução
- Trabalhos relacionados
- Proposta e resultados
- Conclusão e contribuições

Introdução



Comumente
dependente
de trabalho
voluntariado

- Correção de *bugs*
- Novas funcionalidades
- Melhoria de eficiência
- Documentação de código

Fig. esq. adaptada de <http://tinytm.sourceforge.net/>

Fonte Fig. dir.: <https://web.liferay.com/web/paul.hinz/blog/-/blogs/open-source:-the-future-or-a-trend->

Introdução



Comumente
dependente
de trabalho
voluntariado

- Correção de *bugs*
- Novas funcionalidades
- Melhoria de eficiência
- Documentação de código

Atividade
típica de
cursos de
computação

Fig. esq. adaptada de <http://tinytm.sourceforge.net/>

Fonte Fig. dir.: <https://web.liferay.com/web/paul.hinz/blog/-/blogs/open-source:-the-future-or-a-trend->

Introdução

Oportunidade para
fortalecer relação
academia-comunidade.

Mas como?

Comumente
dependente
de trabalho
voluntariado

- Correção de *bugs*
- Melhorias de interface
- Melhoria de eficiência
- Documentação de código

Atividade
típica de
cursos de
computação

Ambiente acadêmico

Sobrevivência
FLOSS

Academia-Comunidade: Iniciativas

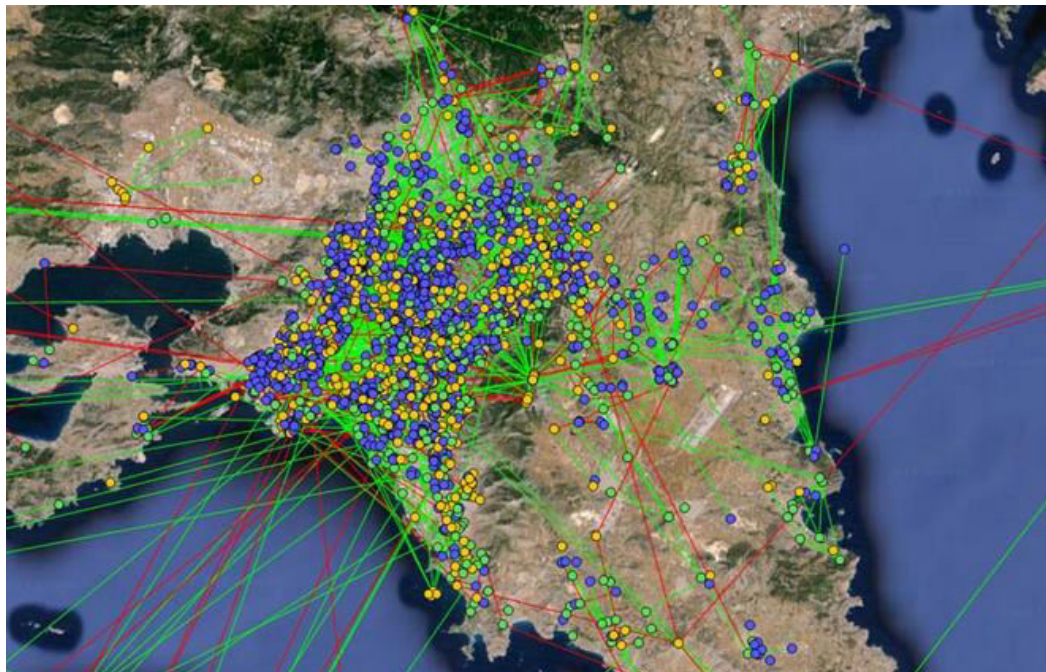
- [Patterson 2006] :
 - uso de problemas FLOSS como atividade em sala de aula.
Ex.: desafios de correção de *bugs* e exercícios de documentação de grandes sistemas.
- [Meneely et al. 2008]
 - repositório com dados sobre projetos FLOSS
 - Objetivo: ajudar a elaboração de atividades acadêmicas com base em FLOSS.
- [Cărbune and Vasilescu 2013]
 - Disciplina visando diminuir o tempo necessário à primeira contribuição de um aluno a uma comunidade FLOSS

Proposta

- Fortalecimento da relação academia-comunidade
- Relato de um estudo de caso nesse sentido
 - TCC Diogo Gonçalves
 - Comunidade OLSRd

Contextualização

- *Firmware* para APs Wi-Fi + protocolo de roteamento OLSRd = Redes em malha sem fio comunitárias (WMNs)



A Perspectiva da Comunidade

Desafio: (Re)Calcular rotas com CPUs “fracas” em grandes topologias dinâmicas!



A Perspectiva (Histórica) da Comunidade

- “o OLSR não é projetado para rotear grandes redes” [Tonnesen 2004].
 - Algoritmo de Dijkstra + listas como Fila de Prioridades (FP)
 - Tempo de busca em lista $O(n)$ \Rightarrow Dijkstra = $O(n^2)$ para n APs
- WMN com 450 APs [Aichele 2004~~2014~~]
 - Tempo de cálculo: Dezenas de segundos
 - Vazão da rede altamente penalizada!
- OLSRd 0.5.6 (2008):
 - Árvores AVL no lugar de listas
 - Melhoria esperada no Dijkstra: $O(n \log_2 n)$

Perspectiva da Academia

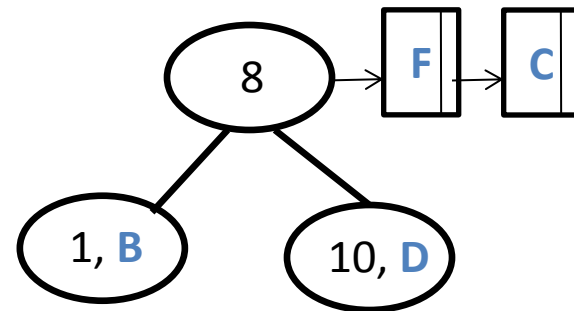
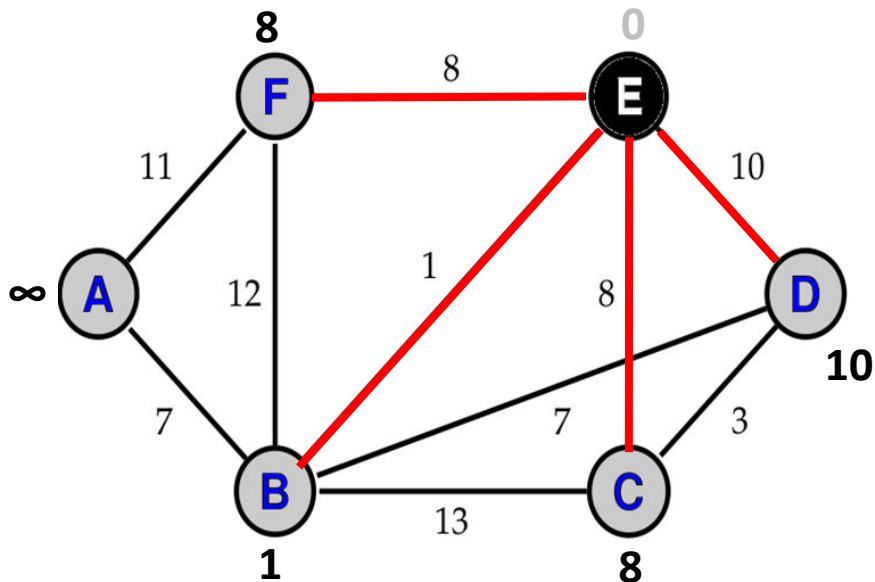
- AVLs garantem $O(n \log_2 n)$ assumindo valores chaves **distintos**
- Em FPs o campo chave é a prioridade cujo valor pode repetir-se para APs distintos
 - Em WMNs isso depende do peso das arestas.
- O que ocorre quando os valores se repetem?

Perspectiva da Academia

- AVLs garantem $O(n \log_2 n)$ assumindo valores chaves **distintos**
- Em FPs o campo chave é a prioridade cujo valor pode repetir-se para APs distintos
 - Em WMNs isso depende do peso das arestas.
- O que ocorre quando os valores se repetem?
 - APs com mesma prioridade são colocados em uma lista!

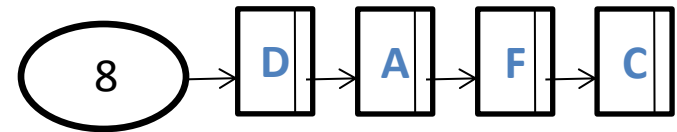
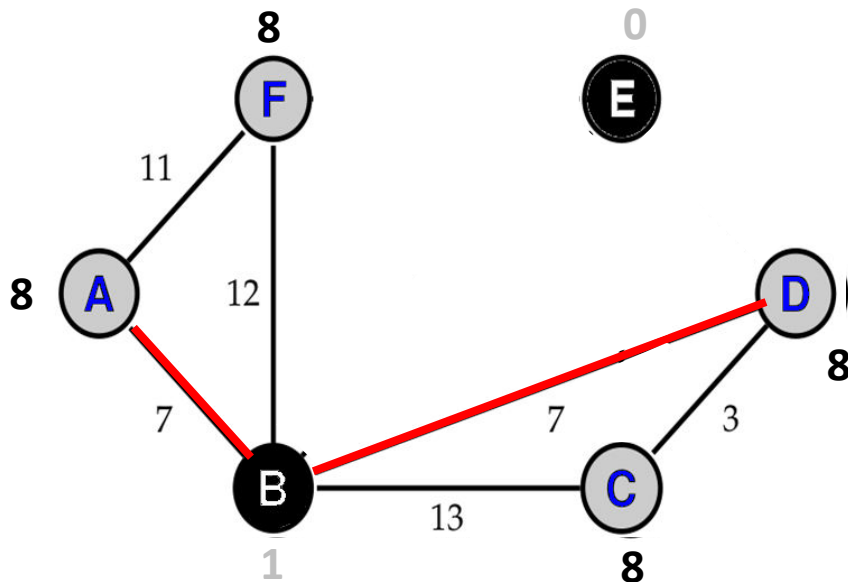
AVL como FPs

- Exemplo após 1ª iteração do Dijkstra



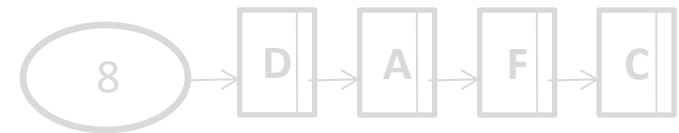
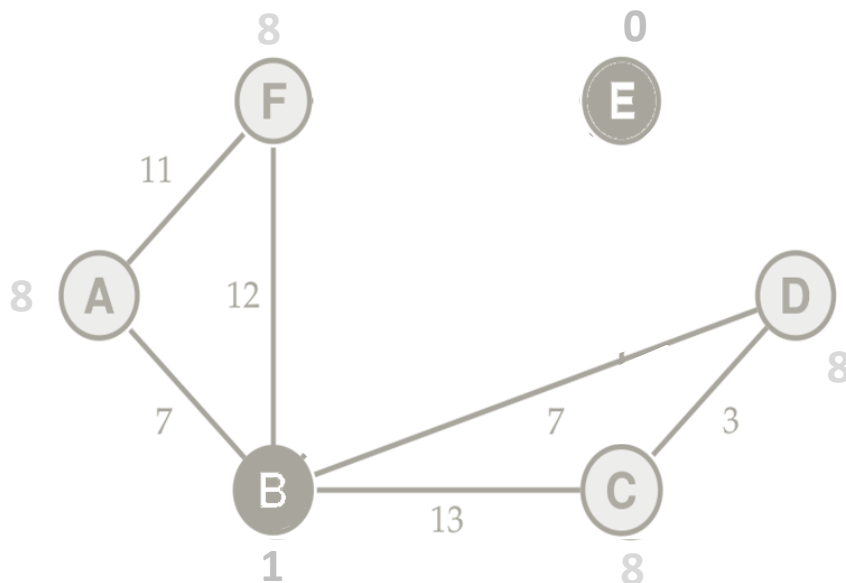
AVL como FPs

- Exemplo após 2ª iteração do Dijkstra



AVL como FPs

- Exemplo após 2ª iteração do Dijkstra



CASOS EXTREMOS:

Todos repetidos:

Dijkstra = $O(n^2)$

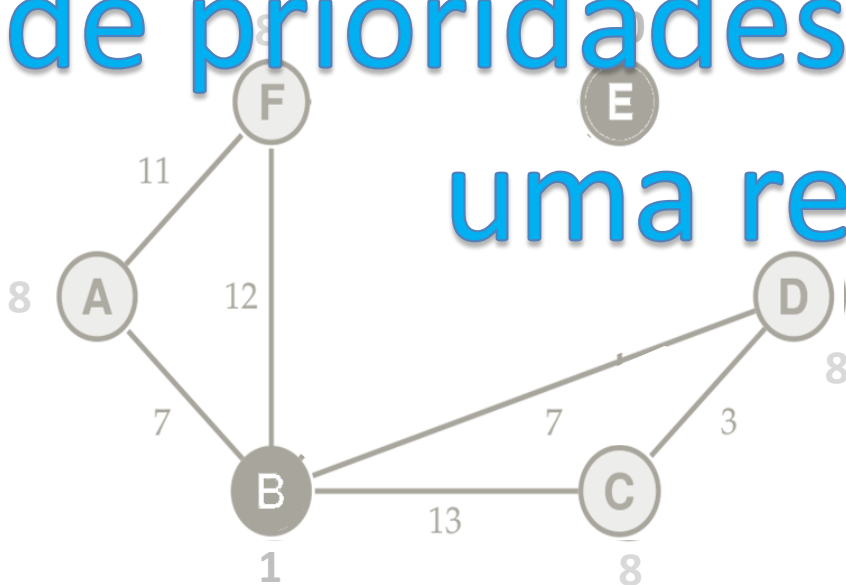
Todos distintos:

Dijkstra = $O(n \log_2 n)$

AVL como FPs

Exemplo após 2ª iteração do Dijkstra

Quão frequentemente os valores de prioridades se repetem em uma rede real?



CASOS EXTREMOS:

Todos repetidos:

Dijkstra = $O(n^2)$

Todos distintos:

Dijkstra = $O(n \log_2 n)$

Estudo de Caso: Dilema Oportunizador

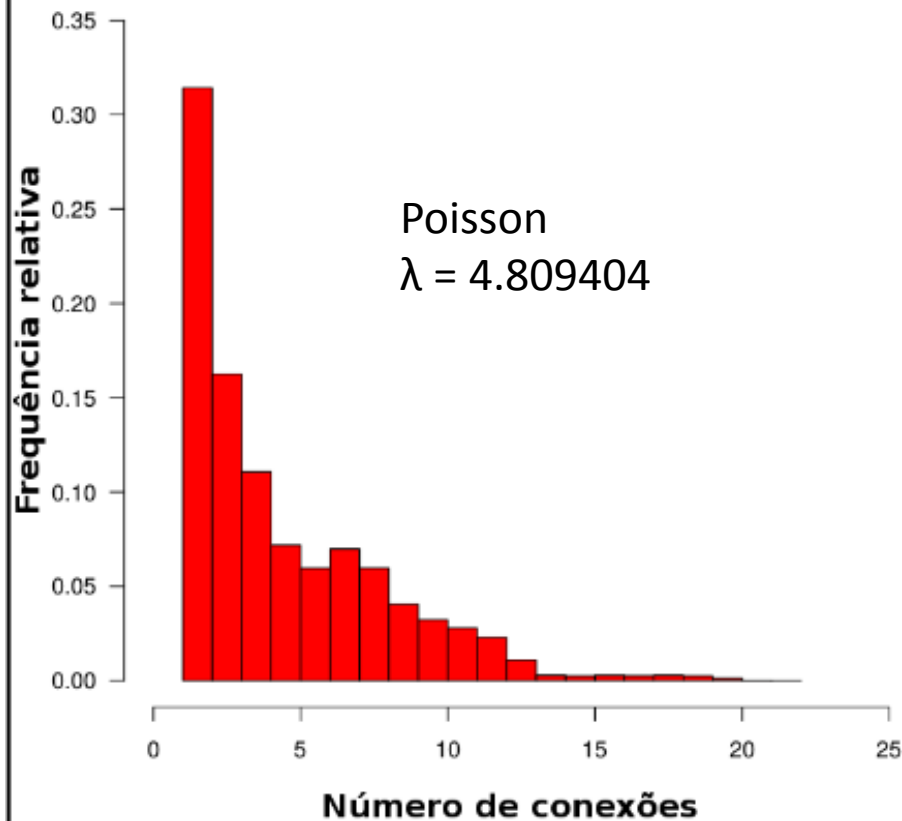
Ator	Interesse	Limitação
Comunidade FLOSS	Assegurar cálculo de rotas $O(n \log_2 n)$ em WMNs reais	Escassez de RH e tempo para estudos empírico
Academia	Estudo empírico do desempenho do Dijkstra sob diferentes FPs em WMNs	Geração de dados topológicos de WMNS reais em larga escala

Rede Opennet: Dados da Comunidade

- WMN OLSRd em Rostock, Alemanha
- Tempo de coleta [Wollenberg 2012]
 - 5 meses, 1/1 a 31/05 de 2011, 3664 horas
- Dados globais da rede
 - Roteadores cadastrados: ~300
 - Roteadores ativos: 259 (max.), média 175
 - Enlaces ativos em toda rede (média): 848
- **Dados topológicos coletados**
 - Quantidade de enlaces de um roteador
 - Peso de cada aresta (métrica de roteamento ETX)

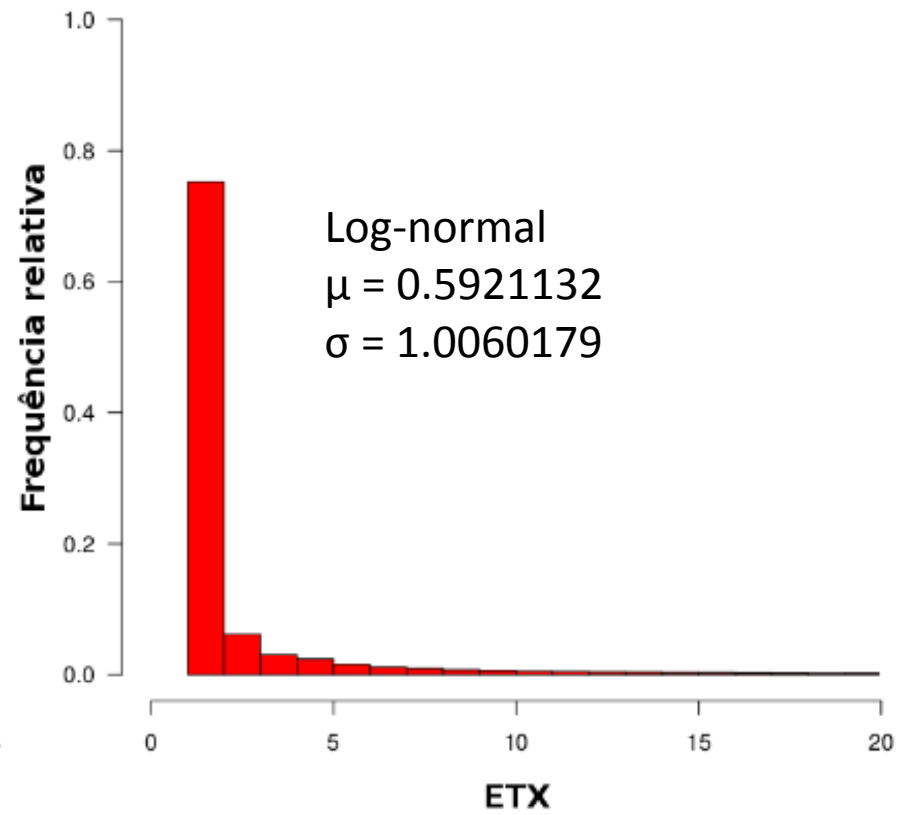
Análise dos dados: Histogramas

Número de conexões por roteador



(a)

Qualidade das conexões baseadas na métrica ETX

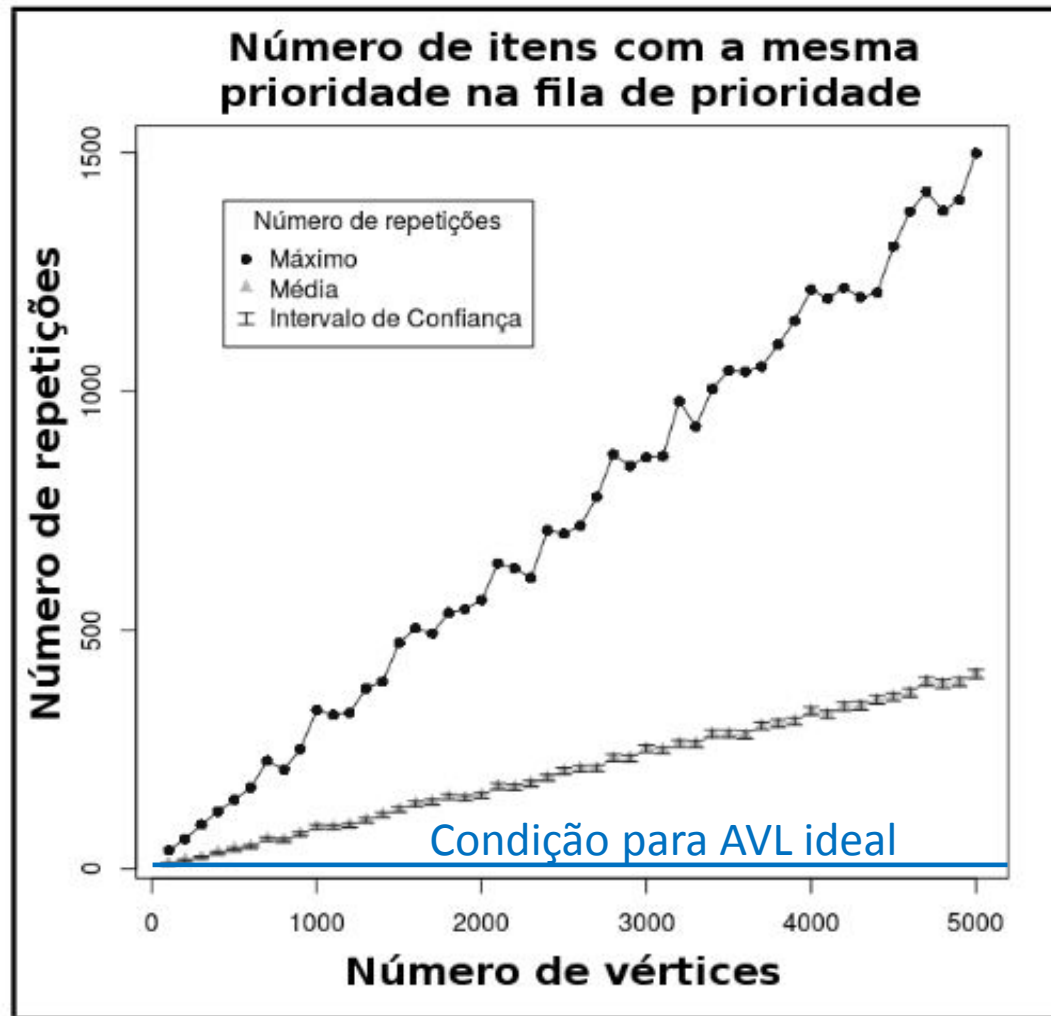


(b)

AVL como FP: Árvore ou Lista?



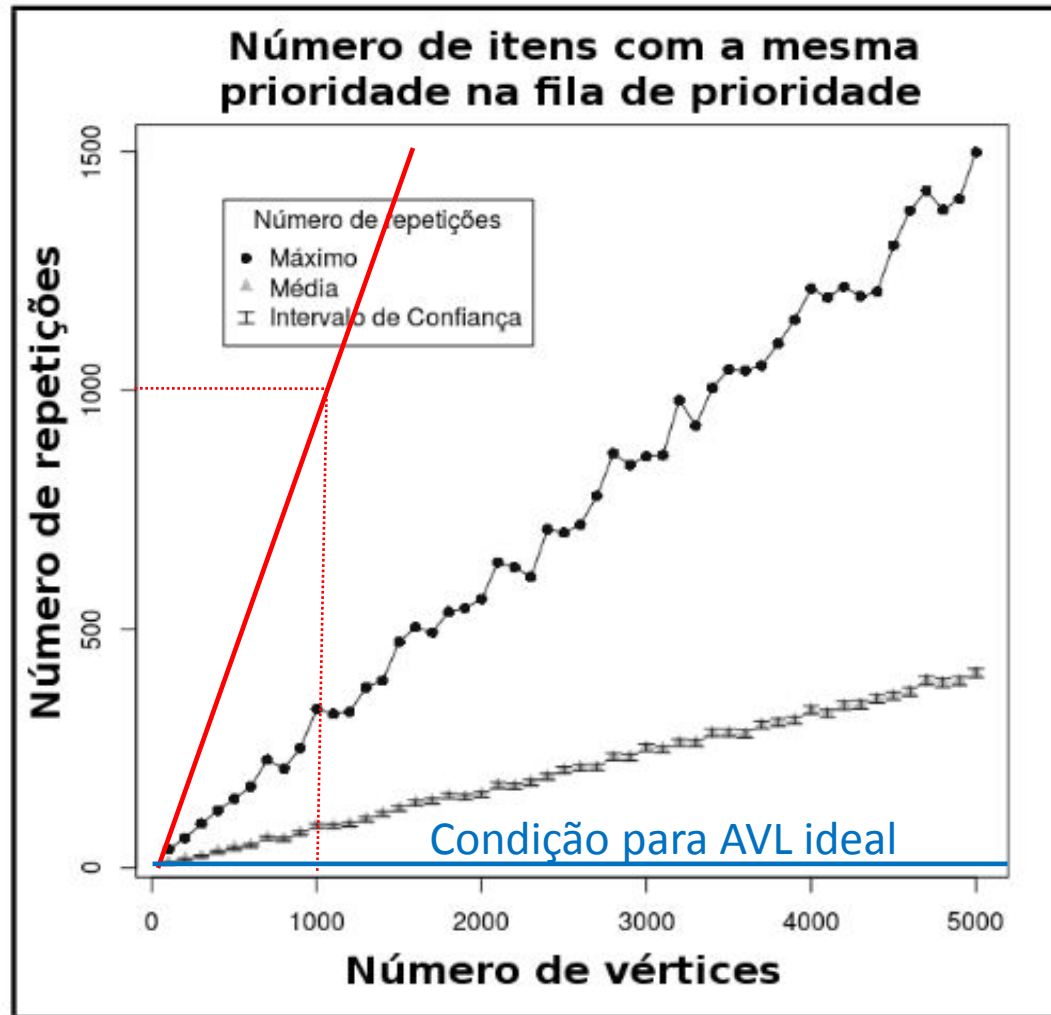
AVL como FP: Árvore ou Lista?



Indício:
 $AVL = \omega(\log_2 n)$

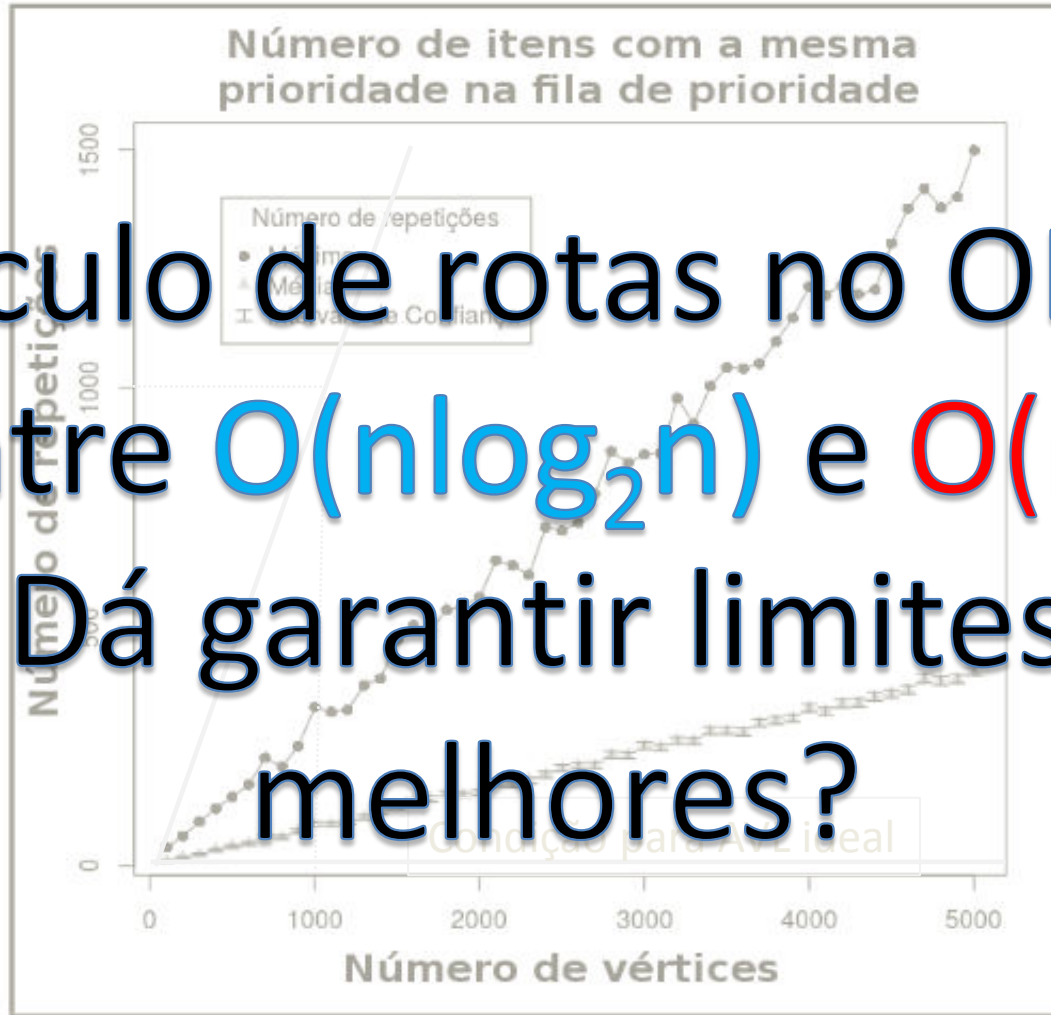
AVL como FP: Árvore ou Lista?

Condição para
AVL péssima



Indício:
 $AVL = \omega(\log_2 n)$
 $AVL = o(n)$

AVL como FP: Árvore ou Lista?



Cálculo de rotas no OLSRd
entre $O(n \log_2 n)$ e $O(n^2)$.
Dá garantir limites
melhores?

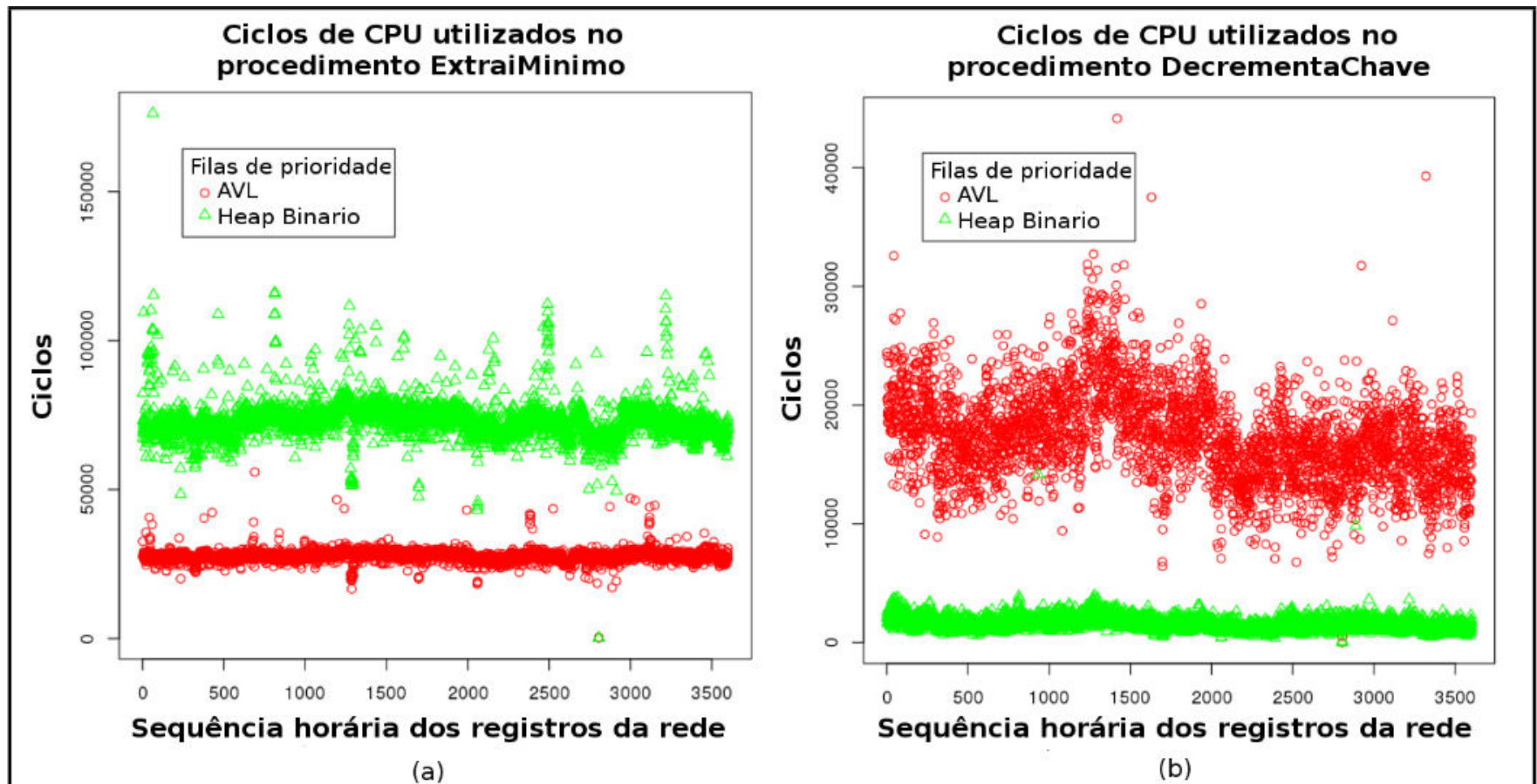
AVL = $O(\log_2 n)$
AVL = $O(n)$

Condição para AVL péssima

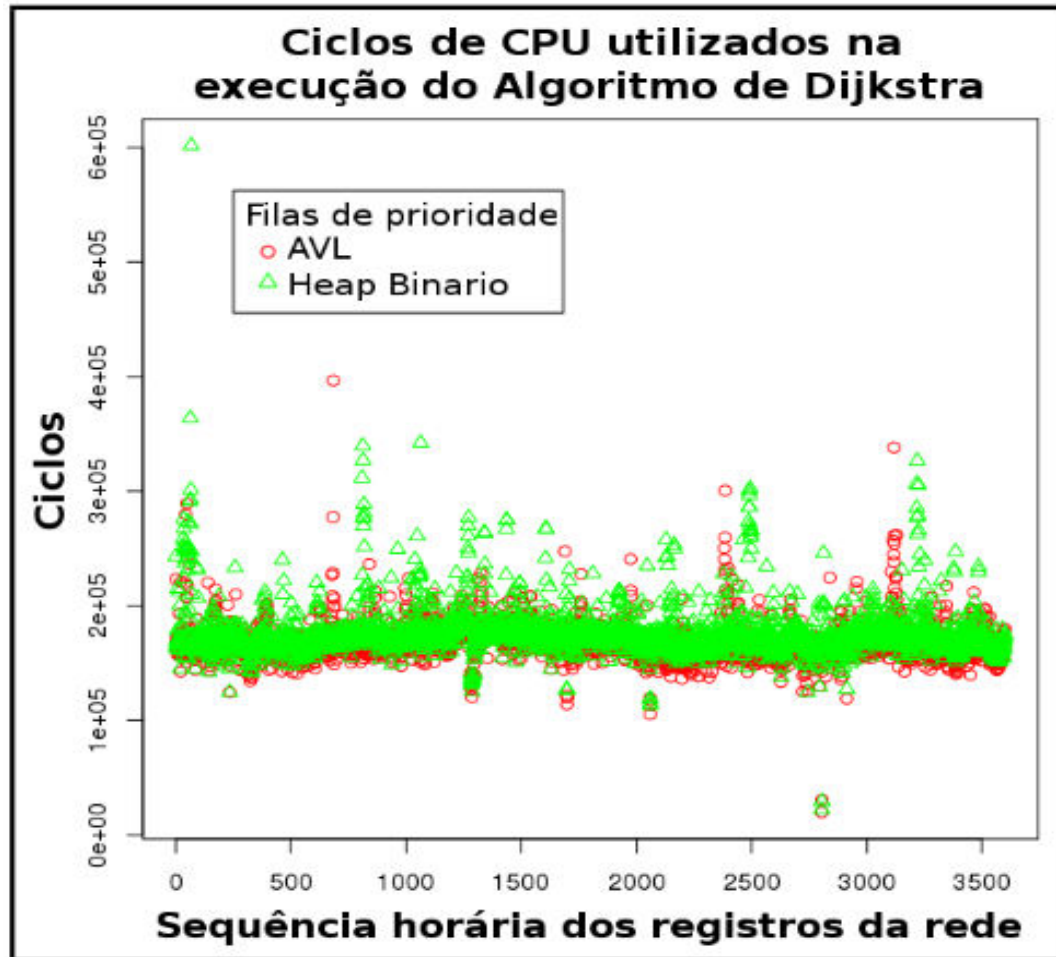
Avaliação de Desempenho

- FPs alternativas:
 - *Heap* (por ter complexidade logarítmica)
 - Van Emde Boas: $O(\log_2 \log_2 n)$
 - Não discutido por restrições de espaço em Aps
- Variável observada: ciclos de CPU consumidos
- Cenários:
 - Topologia real: ~200 roteadores
 - Topologia sintética: escalabilidade

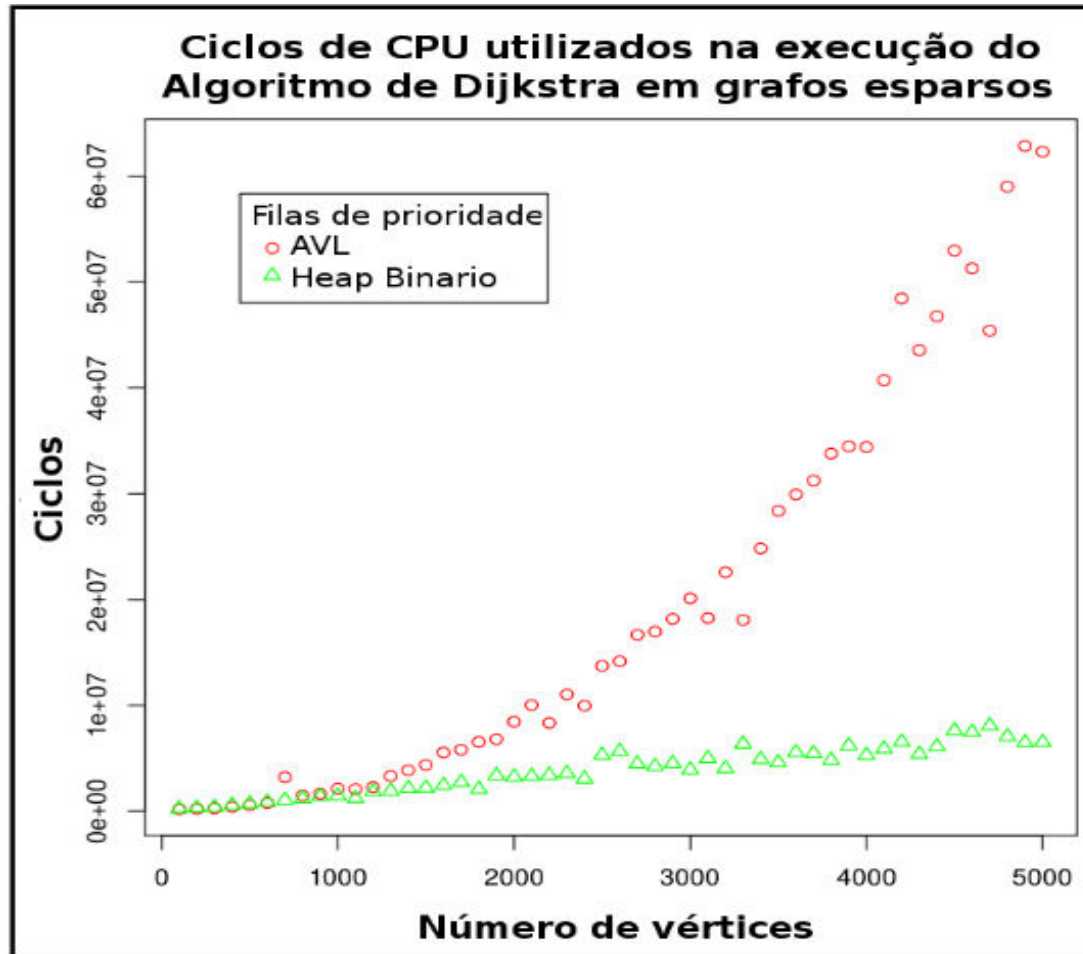
AVL vs *Heap* em Topologia Real: Extrai_Minimo e Decrementa_Chave



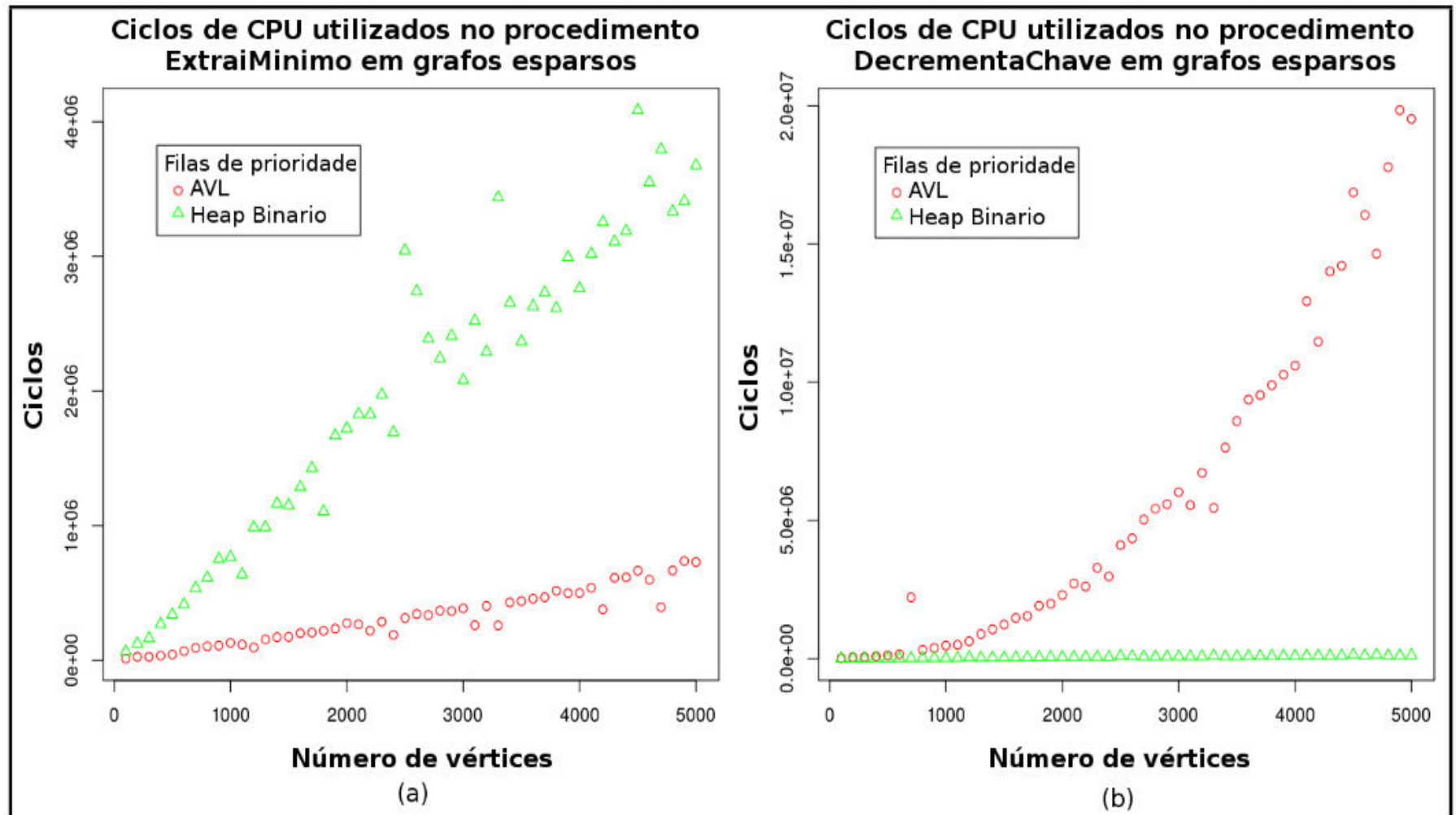
AVL vs. *Heap* em Topologia Real: Impacto no Algoritmo de Dijkstra



AVL vs. *Heap* em Topologia Sintética: Impacto no Algoritmo de Dijkstra



AVL vs *Heap* em Topologia Sintética: Extrai_Minimo e Decrementa_Chave



Conclusão e Contribuições

- O presente trabalho apresenta um estudo de caso promissor da relação academia-comunidade
 - A academia supriu a escassez de tempo e RH da comunidade
 - *“There were so much code to rewrite that I did not dive into the AVL one”* [Gonçalves 2016]
 - A comunidade supriu a limitação acadêmica de produzir dados em larga escala
- Contribuições à academia
 - Novo *branch* “dijkstra-binary-heap” no repo. oficial
 - Otimização na inserção da AVL: $O(n)$ para $O(1)$