UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

LEONARDO PERDOMO

Fluxo $s - t$ máximo
Ford-Fulkerson com estratégia *Fattest Path*

Porto Alegre, 2017

1.1 Tarefa

O objetivo deste trabalho foi a implementação prática em laboratório e a análise experimental da complexidade do algoritmo de Ford-Fulkerson [1] utilizando para aprimoramento do mesmo a estratégia de *fattest path*.

1.2 Solução

Para implementar a solução do problema proposto, foi utilizado o algoritmo de Dijkstra com alterações para busca do fluxo $s-t$ mais curto com maior capacidade. Para permitir um desenvolvimento mais organizado e compreensivo, o protótipo com fila de prioridade de *heaps* n-ários do primeiro trabalho foi reimplementado orientado à objetos com fila em *max heap* binário, ordenando sempre pela maior capacidade (de *deletemin* para *deletemax*), enquanto as alterações no algoritmo de Dijkstra consistiram na mudança de soma da distância para seleção da maior capacidade adjacente.

O algoritmo Ford-Fulkerson desenvolvido obtém sucessivamente o fluxo $s-t$ com Dijkstra, verifica o gargalo do mesmo, e monta os arcos residuais no grafo. O processo deve terminar quando Dijkstra não encontrar mais fluxo $s-t$ viável em no máximo $O(mlog_m U)$ passos, não havendo fluxo e não terminando caso contrário, sendo U o limite superior da capacidade dos arcos. A complexidade esperada para o algoritmo é de $O((mlogn)mlog_m C)$, para um limitante superior C do fluxo máximo, considerando o uso de Dijkstra com *heap* binário.

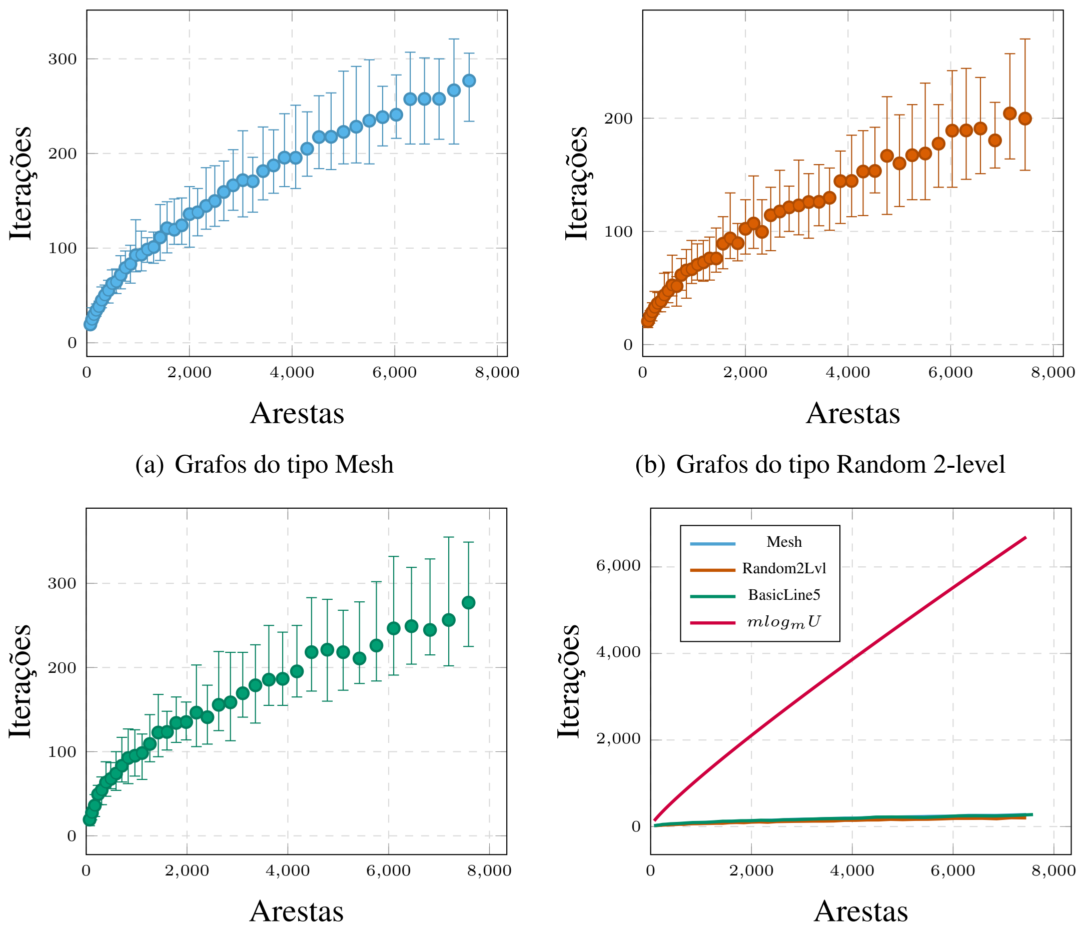
1.3 Ambiente de teste

O dispositivo utilizado para realização dos testes foi o *notebook* pessoal do autor deste trabalho, um ASUS X451CAP com Intel i3 3217U de 1.8GHz, 4GB de RAM DDR3 de 798MHz com Ubuntu 16.04 64bit LTS.

1.4 Resultados

Os experimentos realizados consistiram na utilização de grafos gerados aleatoriamente, dos tipos grade *Mesh* e *Random 2-level*, e linha *Basic Line*, pelo script *new-washington.c* disponibilizado pelo professor em aula. Para as grades, os parâmetros de linha r e coluna c da geração ($n = rc + 2$ e $m = 3r(c-1)$) foram aumentando incrementalmente de 5 à 50 a cada lote, enquanto para o tipo linha, o grau D foi fixado em 5, e os parâmetros de vértices n e arestas m ($n = nm + 2$ e $m = nmD + 2m$) também aumentaram incrementalmente de 5 à 50. Para cada configuração, foram gerados os grafos e resolvidos 20 vezes em repetição pelo algoritmo implementado.

O número de iterações por execução do algoritmo Ford-Fulkerson de acordo com o aumento de arestas foi coletado para verificação do cumprimento da condição esperada $O(mlog_m U)$ máxima para passos. As médias dos números de iterações por tipo de grafo e suas respectivas variações são apresentadas nas figuras 1.1.a, 1.1.b e 1.1.c, com as amplitudes indicadas pelos traços verticais. Estes resultados obtidos respeitam o limite máximo esperado, como pode ser observado comparativamente na figura 1.1.d, com as iterações coletadas dos três tipos de grafos na região inferior do gráfico estando significativamente abaixo em relação ao limite teórico de iterações, em destaque, devido à grandeza teórica do mesmo discutida em aula.



(a) Grafos do tipo Mesh

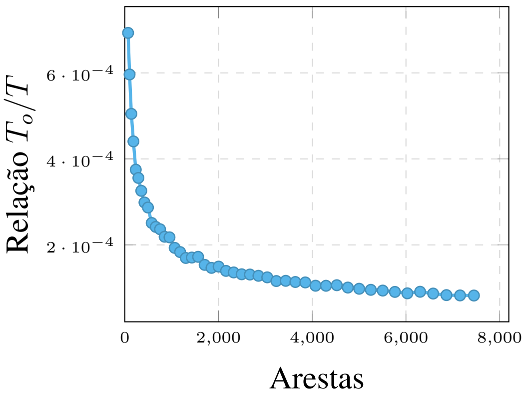
(b) Grafos do tipo Random 2-level

(c) Grafos do tipo Basic Line com grau 5

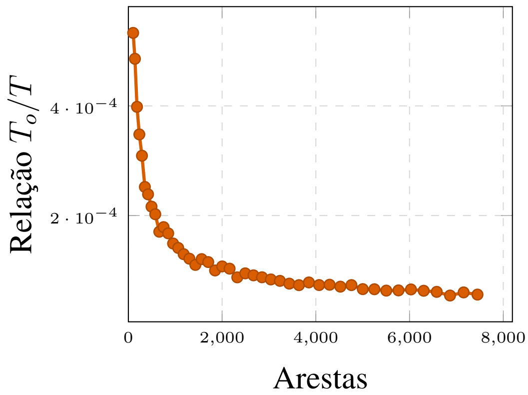
(d) Iterações em relação ao limite esperado

Figura 1.1: Crescimento no número de iterações conforme número de arestas dos grafos.

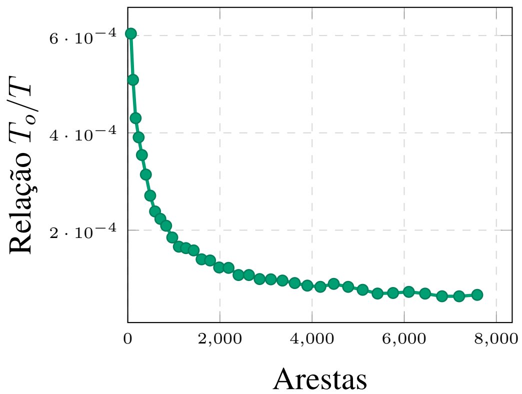
Para a análise da relação entre tempo observado T_o e teórico T, também foram coletados os tempos de execução do algoritmo implementado durante os experimentos. O tempo médio de execução para cada tipo de grafo foi dividido pela complexidade esperada $O((mlogn)mlog_mC)$, podendo ser observado nas figuras 1.2.a, para *Mesh*, 1.2.b, para *Random 2-level*, e 1.2.c, para *Basic Line* com grau 5, que são desenhadas curvas aproximando-se ao eixo com o aumento no número de arestas, conforme comportamento descrito em aula, demonstrando o respeito à complexidade esperada.



(a) Grafos do tipo Mesh



(b) Grafos do tipo Random 2-level



(c) Grafos do tipo Basic Line com grau 5

Figura 1.2: Relação entre tempo observado ao tempo esperado conforme número de arestas.

1.5 Conclusão

Através da implementação prática e experimentação conduzidas neste trabalho, foi possível observar que a estratégia *fattest path* para o algoritmo Ford-Fulkerson respeita a complexidade teórica e o número limite de passos. Para os três tipos de grafos analisados (*Mesh*, *Random 2-level* e *Basic Line* com grau 5) o número de iterações esteve significativamente abaixo do limite máximo previsto, o que já era esperado devido à grandeza do mesmo, enquanto o tempo de execução observado respeita o tempo teórico de $O((mlogn)mlog_mC)$.

REFERÊNCIAS

[1] J. L. R. Ford and D. R. Fulkerson, "Maximal flow through a network," *Canadian Journal of Mathematics*, vol. 8, pp. 399–404, 1956.