

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

LEONARDO PERDOMO

Emparelhamento
Algoritmo de Hopcroft-Karp

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 Hopcroft-Karp [1] para solução do problema de emparelhamento máximo em grafos bi-partidos. Em seguida, foi realizada a redução para um problema de fluxo, usando o algoritmo Ford-Fulkerson [2] com estratégia *Fattest Path* implementado no último trabalho, a fim de observar experimentalmente qual abordagem apresenta maior eficiência.

1.2 Solução

Para implementar a solução do problema proposto, foram utilizadas estruturas do tipo *map* para registro dos emparelhamentos entre vértices. O algoritmo obtém sucessivamente o caminho mais curto aumentante entre dois vértices não emparelhados por meio da busca em largura, navegando entre as duas partes do grafo alternando entre arestas emparelhadas e marcando com incrementos de 1 (garantindo também sua desconsideração nos próximos caminhos). Em seguida, por meio de busca em profundidade de vértices não emparelhados, o caminho incrementado é seguido verificando se é o mais curto para emparelhamento. A complexidade esperada para o algoritmo é de $O(\sqrt{n}(n + m))$.

Para redução a problema de fluxo máximo, a implementação *Fattest Path* do algoritmo Ford-Fulkerson realizada no último trabalho foi adaptada, adicionando um vértice s conectado a uma parte do grafo bi-partido, um vértice t à outra parte, e atribuindo pesos 1 nas arestas. O fluxo máximo resultante corresponde à cardinalidade máxima buscada no problema de emparelhamento.

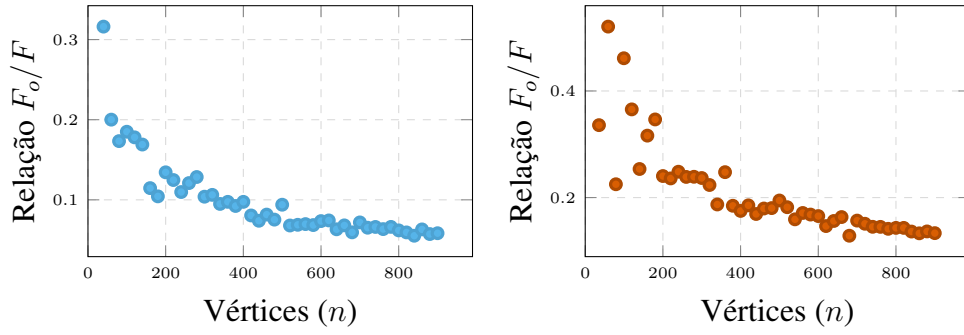
1.3 Ambiente de teste

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.

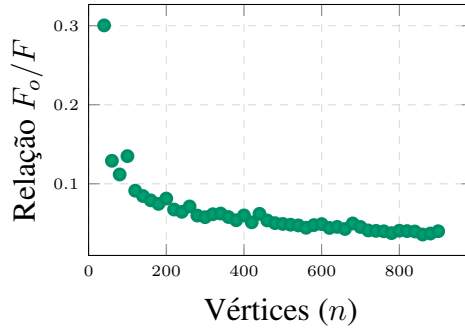
1.4 Resultados

Os experimentos realizados consistiram na utilização de grafos bi-partidos randômicos criados com o gerador disponibilizado pelo professor em aula. Foram utilizados grafos com aumento progressivo do número de vértices de 20 à 900, e distribuições de probabilidade $p = 0.1$, $p = 0.5$ e $p = 0.9$ na criação de arestas, atingindo ≈ 10000 arestas em instâncias de 900 vértices. Para cada configuração, foram gerados os grafos e resolvidos 20 vezes em repetição pelo algoritmo implementado. Os números de fases e de extrações de caminhos aumentantes por execução do algoritmo Hopcroft-Karp foi coletado para verificação do cumprimento dos respectivos limites esperados \sqrt{n} e $n + m$.

Por meio da relação entre fases observadas e limite esperado apresentada na figura 1.1, é possível observar por meio da curva em convergência que o limite teórico foi respeitado. O número de fases durante os experimentos manteve-se baixo, registrando no máximo 6 fases em algumas instâncias de grafos com $p = 0.1$ arestas, não passando de 2 fases para grafos com $p = 0.9$. O respeito ao limite teórico também é notado para as extrações de caminhos aumentantes, apresentadas na figura 1.2, em maior número para instâncias com menor quantidade de arestas.

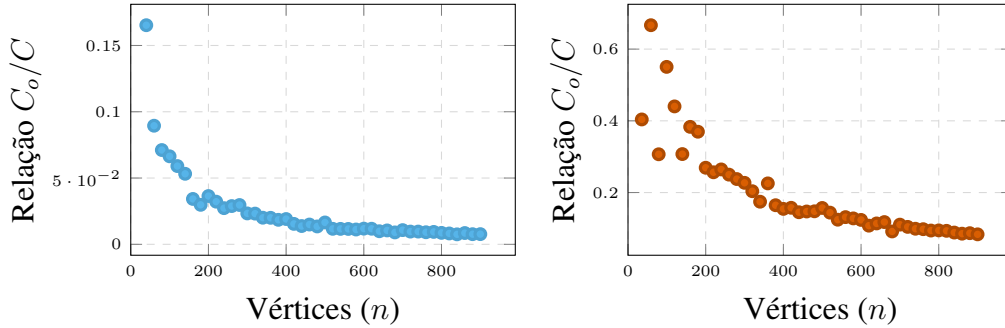


(a) Grafos com distrib. $p = 0.5$ de arestas. (b) Grafos com distrib. $p = 0.1$ de arestas.

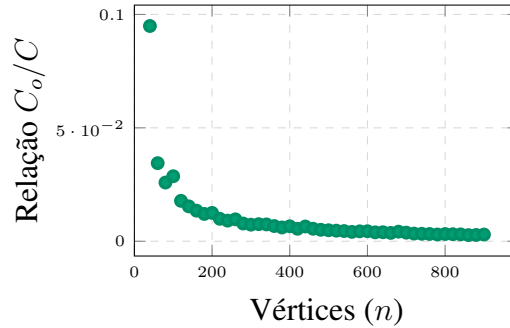


(c) Grafos com distrib. $p = 0.9$ de arestas.

Figura 1.1: Relação entre número fases observadas (F_o) e esperadas ($F = \sqrt{n}$).

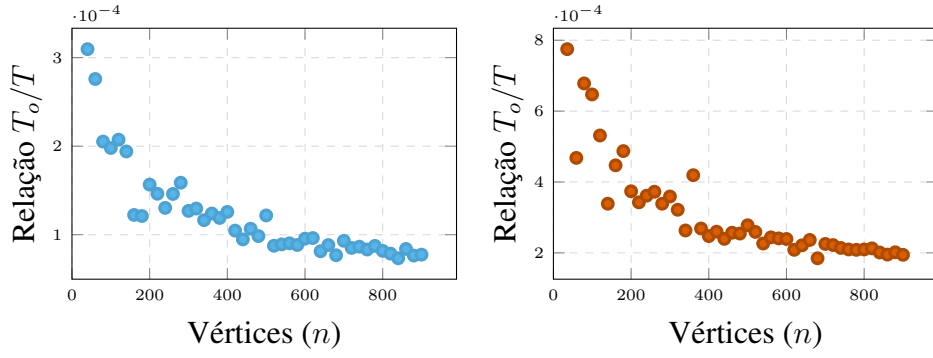


(a) Grafos com distrib. $p = 0.5$ de arestas. (b) Grafos com distrib. $p = 0.1$ de arestas.

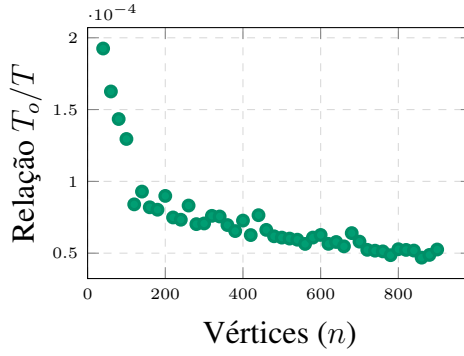


(c) Grafos com distrib. $p = 0.9$ de arestas.

Figura 1.2: Relação entre número de extrações de conj. maximal de caminhos aumentantes observados (C_o) e esperados ($C = n + m$).



(a) Grafos com distribuição $p = 0.5$ de arestas. (b) Grafos com distribuição $p = 0.1$ de arestas.



(c) Grafos com distribuição $p = 0.9$ de arestas.

Figura 1.3: Relação entre tempo observado (T_o) e esperado ($T = (\sqrt{n})(n + m)$).

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 $O(\sqrt{n}(n + m))$, podendo ser observado na figura 1.3, demonstrando respeito ao limite teórico.

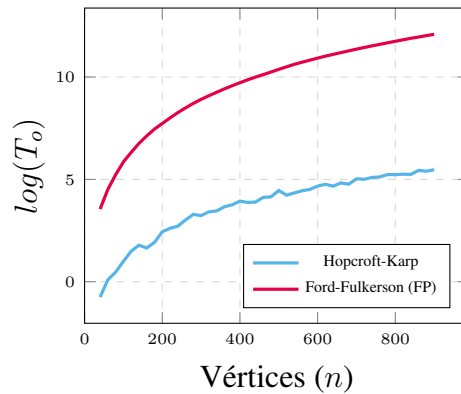


Figura 1.4: Comparação em escala logarítmica entre tempos observados (T_o) em ms para obtenção da cardinalidade máxima pelos algoritmos Hopcroft-Karp e Ford-Fulkerson com Fattest Path (FP) para grafos com distribuição $p = 0.5$ de arestas.

Com a redução do emparelhamento máximo para um problema de fluxo máximo, os tempos de execução dos dois algoritmos para grafos com $p = 0.5$ arestas são apresentados em escala logarítmica na figura 1.4. O custo em tempo foi significativamente maior para a variante *Fattest Path* do algoritmo Ford-Fulkerson em relação à implementação, mais eficiente, de Hopcroft-Karp, justificando a escolha da escala para melhor leitura da diferença entre ambos.

1.5 Conclusão

Através da implementação prática e experimentação conduzidas neste trabalho, foi possível observar que o algoritmo Hopcroft-Karp para emparelhamento respeita a complexidade teórica de $O(\sqrt{n}(n+m))$, e os limites de fases \sqrt{n} e de extrações de caminhos aumentantes $n+m$. Notadamente, para as três configurações randômicas de grafos com até 900 vértices e ≈ 10000 arestas o número de fases esteve significativamente abaixo do limite máximo previsto.

Por fim, a redução a um problema de fluxo máximo com a variante *Fattest Path* do algoritmo Ford-Fulkerson implementada no último trabalho permitiu a comparação de eficiência por tempo de execução com a implementação de Hopcroft-Karp, sendo concluído que a solução com Ford-Fulkerson é significativamente mais custosa.

REFERÊNCIAS

- [1] J. J. E. Hopcroft and R. M. Karp, “An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs,” *SIAM Journal on Computing*, vol. 2, pp. 225–231, 1973.
- [2] J. L. R. Ford and D. R. Fulkerson, “Maximal flow through a network,” *Canadian Journal of Mathematics*, vol. 8, pp. 399–404, 1956.