**Descrição do Problema e da Solução**

O problema apresentado foi modelado como um grafo dirigido e pesado: **um** vértice fonte que representa o processador **X**, **um** vértice destino que representa o processador **Y** e **n** vértices intermédios, **Pi**, que representam cada um dos processos. Quanto aos arcos, estes existem: de **X** para cada processo **Pi**, com peso igual ao custo de correr **Pi** em **X**, **Xi**; de cada **Pi** para **Y**, com peso **Yi**, o custo de correr **Pi** em **Y**; de **Pi** para **Pj** e de **Pj** para **Pi**, com peso igual ao custo de comunicação entre esses 2 processos, se esse custo existir.

Para chegar à solução pretendida, era necessário ver os arcos que levariam ao menor custo de correr os **n** processos de **P** em **X** e **Y**. Estes arcos seriam os que estrangulariam o fluxo, pois são os que têm menor custo, logo, estariam presentes no corte mínimo do grafo. Pelo Teorema do Corte Mínimo-Fluxo Máximo, o corte mínimo é igual do fluxo máximo, que é calculado após aplicar o Edmonds-Karp. Assim, obtemos o custo mínimo de correr um programa **P** com **n** processos a executar em 2 processadores, **X** e **Y**.

A implementação usada é baseada em:

* <https://www.geeksforgeeks.org/minimum-cut-in-a-directed-graph/>

**Análise Teórica**

* Leitura dos custos de execução com ciclo a depender de N: Θ(N).
* Leitura dos custos de comunicação com ciclo a depender de K: Θ(E).
* Aplicação do Edmonds-Karp ao grafo: O() ou O(E|f\*|).
  + Número de ciclos a executar BFS: O(VE).
  + Aplicação da BFS na rede residual do grafo: O(V+E) = O(E).
* O algoritmo Edmonds-Karp é uma especialização do método genérico Ford-Fulkerson. Assim, é limitado ou pela aplicação de uma BFS, O(E), tendo em conta os caminhos de aumento conseguidos, O(VE) ou pelo método genérico, com um custo O(E|f\*|). Como na modelação apresentada as capacidades dos arcos são reduzidas e o grafo possui poucos arcos, estes serão o fator limitante na aplicação do algoritmo, o que nos leva a concluir que a complexidade total será limitada pelo fluxo máximo, logo, O(E|f\*|).

Complexidade global da solução: O(E|f\*|)

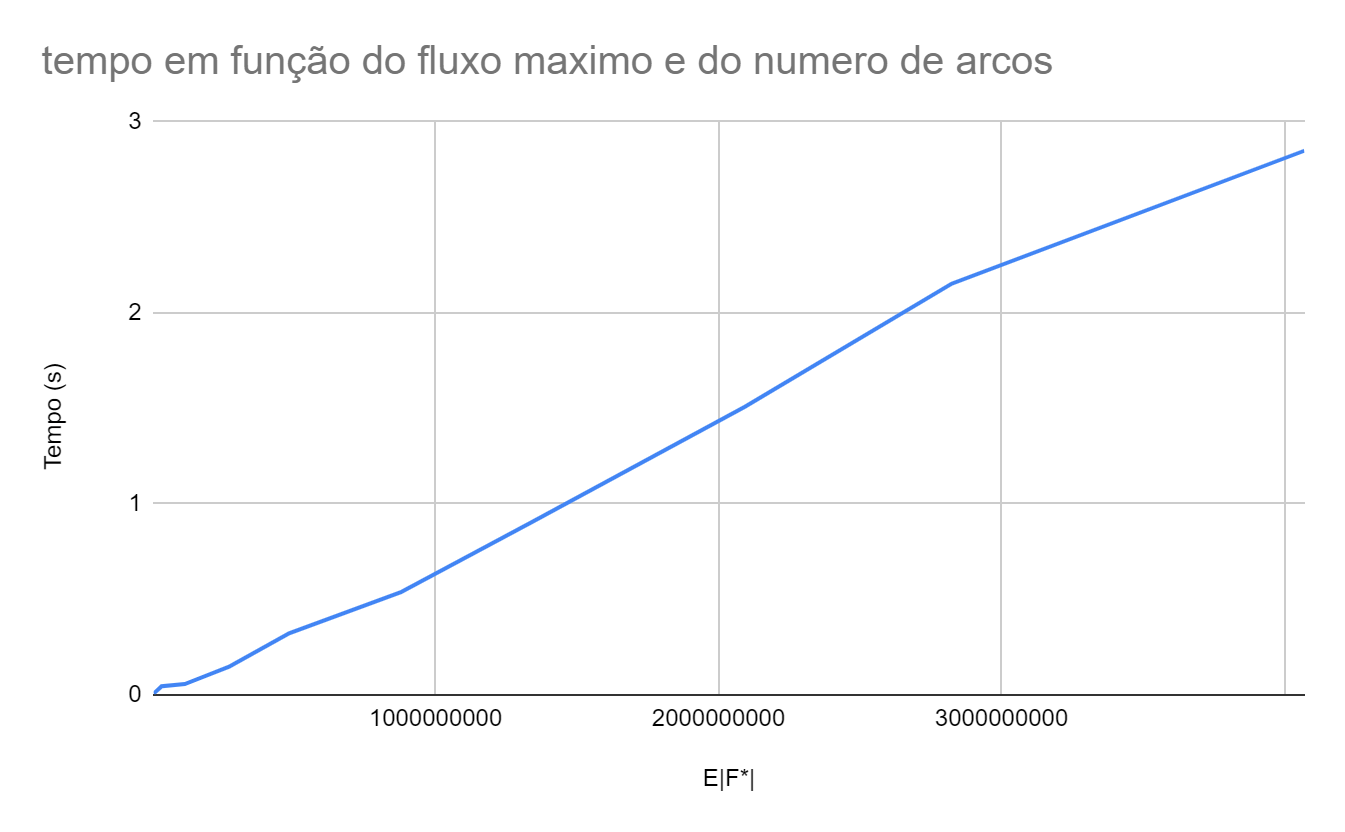
**Avaliação Experimental dos Resultados**

Foram gerados ficheiros de teste correspondendo a um número variável de processos entre 100 e 1000, totalizando V+2 vértices e um custo de comunicação máximo fixo de 20. A tabela seguinte apresenta os resultados obtidos.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **V+2** | **E** | **VE^2** | **E|F\*|** | **Time** |
| 102 | 3890 | 1543474200 | 3823870 | 0,004 |
| 202 | 16144 | 52647004672 | 31787536 | 0,043 |
| 302 | 37100 | 415675820000 | 114230900 | 0,054 |
| 402 | 66224 | 1763018506752 | 270988608 | 0,145 |
| 502 | 102388 | 5262617877088 | 482452256 | 0,32 |
| 602 | 145344 | 12717176758272 | 877587072 | 0,535 |
| 702 | 196826 | 27195812941752 | 1342746972 | 0,904 |
| 802 | 256992 | 52968000227328 | 2091914880 | 1,504 |
| 902 | 318402 | 91444609910808 | 2823270534 | 2,15 |
| 1002 | 403104 | 162817820485632 | 4070141088 | 2,846 |

Após a criação de vários grafos com base nos dados obtidos, obtiveram-se os seguintes resultados:

* Um grafo exponencial, comparando o tempo em função de V
* Um grafo exponencial, comparando o tempo em função de VE^2
* Um grafo linear, comparando o tempo em função de E|f\*|



Assim, concluímos que a análise teórica prevista está de acordo com os testes realizados. A complexidade total do algoritmo usado na modelação do problema é então de O(E|f\*|).