**Homework Assignment Week 8: Linear Programming and Network Flow**

Hyuntaek Oh

ohhyun@oregonstate.edu

CS 514\_400 Algorithm

Nov. 25, 2024

1. **(a) Implement Ford-Fulkerson algorithm (Fat, Short pipes) for maximum flow.**

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**Fig 1.** Fat pipes’ function

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명 텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**Fig 2.** Short pipes’ function

텍스트, 스크린샷, 소프트웨어, 폰트이(가) 표시된 사진

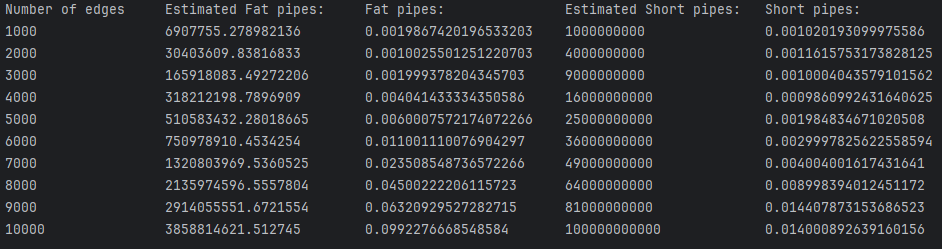
자동 생성된 설명 텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

**Fig 3.** Helper functions (Dijkstra, BFS)

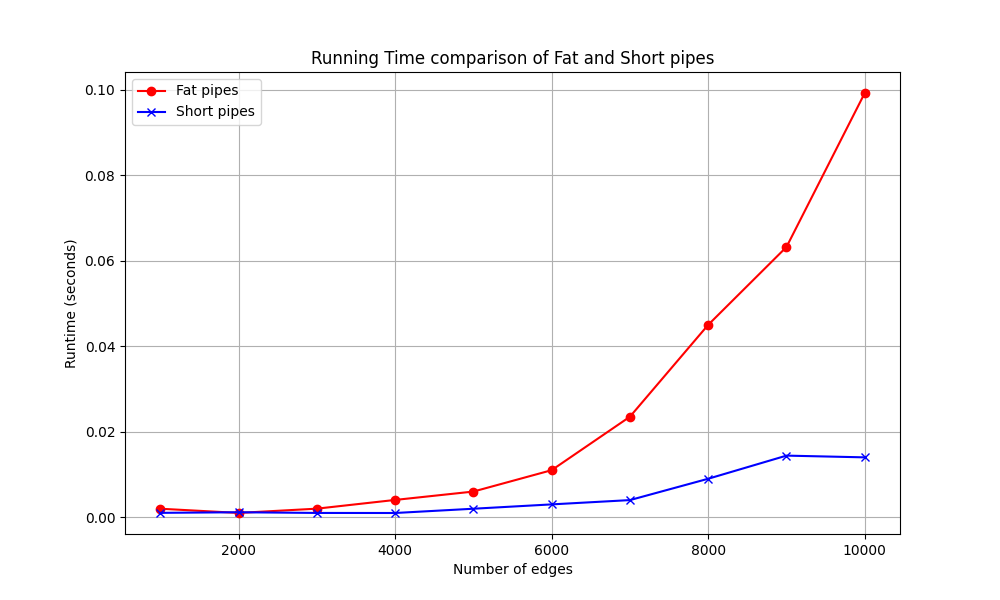
**(b) Generate a table and a graph showing the execution time vs. edges, and then Analyze the results.**

The experiment is conducted under the condition that the fixed number of nodes, which is 1,000, is used. Based on the fixed number of nodes, the number of edges increases 1,000 at each step. The result is below.



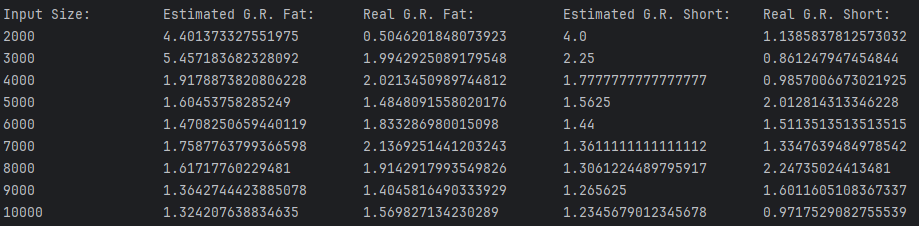
**Table. 1** Fat and Short pipes time table

Table 1 shows that the real time taken with two different approaches, which are Fat and Short pipes respectively. The time taken with Fat pipe approach takes , where is the number of edges and is the maximum flow since we use priority queue using heap dictionary per each edge, which takes time, and the number of finding maximum flow, which takes . On the other hand, according to the Introduction to Algorithm CRLS, the time taken with Short pipe approach takes , where is the number of vertices. As can be seen in Table 1, Short pipes approach is relatively superior to Fat pipes based on the real time taken. For visualization, a graph of two different approaches is below.



**Fig. 4** A graph comparing execution time of Fat and Short pipes

As shown in Fig 4, we can guess that Short pipes takes less time than Fat pipes. In particular, the real time taken of Short pipes is 0.014 when the number of edges reach 10,000 whereas the time of Fat pipes is nearly 0.09. The trend of this graph indicates that as the number of edges increases, the time of running Fat pipes takes exponentially increase. Otherwise, it seems that the number of edges do not significantly affect the performance of Short pipes even though the time complexity is influenced by the number of edges.



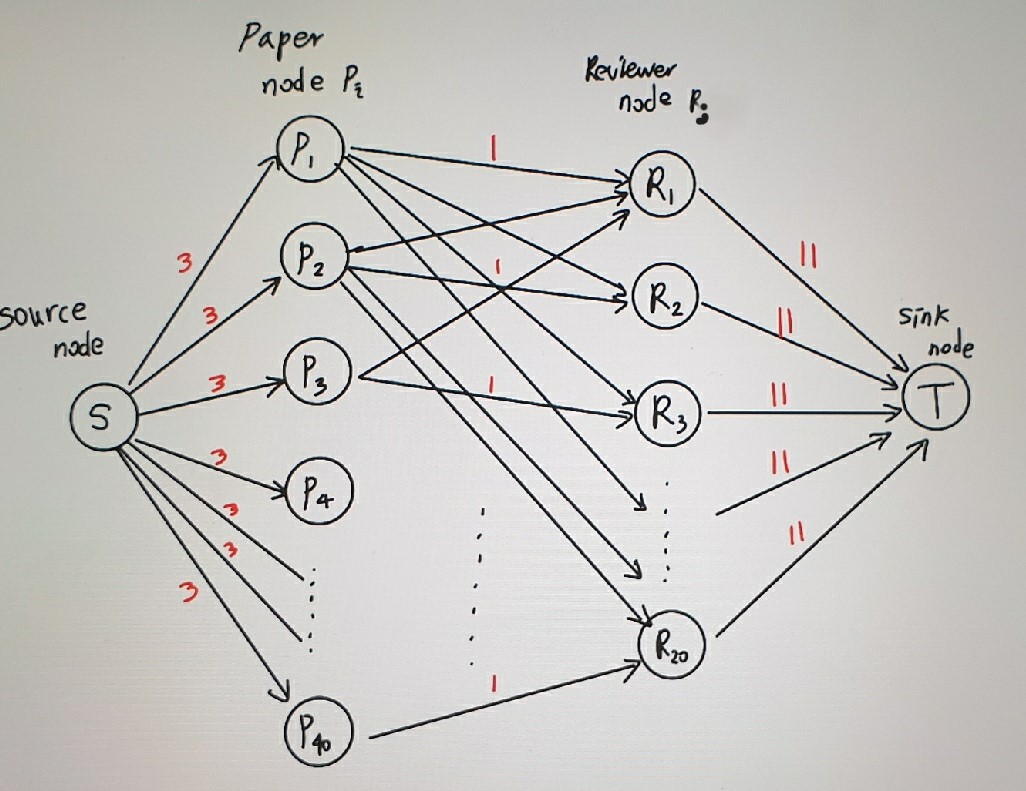
**Table. 2** Fat and Short pipes growth rate table

Table 2 shows the growth rate of each approach based on the input size, which is the number of edges. Despite the estimation growth rates of Fat and Short pipes, the real growth rate is different from the estimation of them. This may possibly be derived from the randomly generated graph where the edges might be arbitrarily distributed or skewed. The magnitude of the growth rate of Fat pipes is relatively larger than one of the growth rates of Short pipes, meaning that the changes of the numbers in Fat pipes are greater than the changes of numbers in Short pipes.

1. **(a) Formulate the problem as a max-flow problem, i.e., describe the architecture of the network and the capacities of the edges.**

To formulate the problem as a max-flow problem, we need to set up an architecture of the network and the capacities of the edges. The architecture is:

* Source node : The starting point to cover all research paper.
* Papers nodes (): connected to with edge capacities 3.
* Reviewer nodes (): connected to with edge capacities 1
* Sink node : The end point where all are collected with edge capacities 11.



**Fig. 5** Network flow architecture

**(b) What is the maximum flow?**

The maximum flow is 120.

We can find the maximum flow from to in this network. All edge capacities:

:

:

:

The minimum capacity across these layers determines the maximum flow. The bottleneck is the source node to paper nodes layer. Thus, 120 is the maximum flow.

1. **(Hall’s Theorem) Show that there is a perfect match if and only if every subset of of boys is connected to at least girls.**

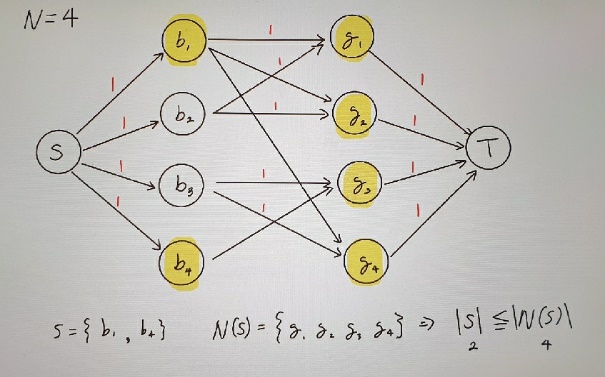
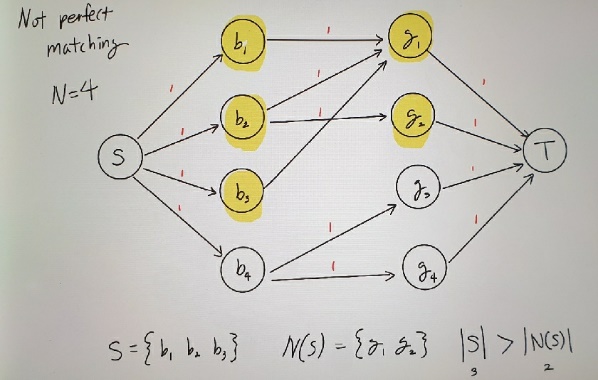
Let’s assume that is a set of boys and is a set of girls .

Hall’s Theorem:

A subset is connected to , where is the set of neighbors of in . The condition guarantees that can be matched to without conflict.

The maximum flow in the network is limited by the minimum cut, which partitions the network into two disjoint sets such that no additional flow can cross from ource to sink . The cut capacity is determined by three flows: flow from to boys in , flow from boys to girls in , and flow from girls in to . For any subset , the cut capacity includes units from to , at most units from to , and the sink edge capacities from to . Therefore, no cut can have a capacity less than , meaning that the maximum flow is also .

If a perfect matching exists, every boy is matched to a unique girl . For any subset , the corresponding matched girls can form a subset . This subset is equal to . Since every matched girl is also a neighbor of the subset , it follows the condition .

**Fig. 6** Perfect match and Not perfect match cases

1. **(a) Write the shortest path problem as a linear program**

To find the shortest path from node to in a directed graph where edge () has length , we need to define the problem as a linear programming. The variable denotes existing edges from node to on the shortest path from node to .

According to the CLRS book, the shortest path means the minimum cost of the flow from to . So, we can express it as:

For each node , which is except node and on the path, the total flow into equals the flow out of :

From the source node , it ensures that one unit of flow leaves since the minimum value of the flow is 1 under the condition that . This means:

From the sink node , it ensures that one unit of flow enters :

For variable , the value of it should be equal to or greater than 0 for all edges from node to , and be binary (0 or 1).

**(b) Show that the dual of the problem can be written as , where in E.**

According to the CLRS book, we can express (a) as a generic primal linear programming form:

In dual problem, the form above can be converted to:

The variable can be expressed as:

A formula for calculating the amount of flows at each single node :

Amount of leaving flow from Amount of entering flow to

Based on the constraints, we can assume that:

Therefore, the dual problem’s objective function is:

From the constraints, we get:

We conclude: