UNIVERSITE CATHOLIQUE DE LOUVAIN

ECOLE POLYTECHNIQUE DE LOUVAIN

Study and Analysis of Networks Flows

Supervisor: Yves Deville
Readers: François Aubry
Jean-Charles Delvenne

Thesis submitted for the Master's degree
in computer science and engineering
options: Artificial Intelligence
by Denis Genon & Victor Velghe

Louvain-la-Neuve
Academic year 2015-2016

Abstract

Acknowledgment

Contents

Chapter 1

The Maximum Flow Problem

Chapter 2

Existing Algorithms

2.1 Augmenting path algorithms

2.1.1 Introduction

The idea behind the augmenting path algorithms is as follows : As long as there is a path from the source to the sink, we send flow along this path. And so on until there is no more path from the source to the sink.

An available path from the source to the sink is called *augmenting path* and to find it, we use the *residual graph*. A *residual graph* is a double oriented graph with the available capacities. For instance here is a graph with its *residual graph*:

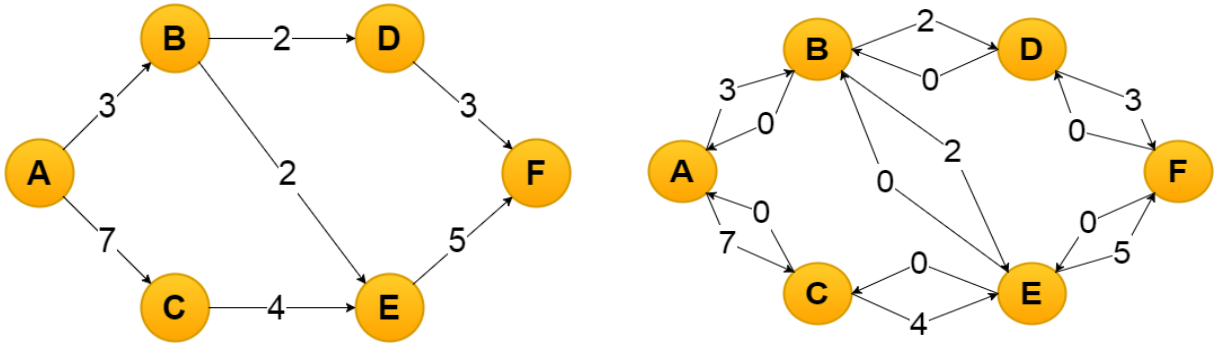

Figure 2.1: A graph with its residual graph

When an *augmenting path* is found, we send a flow equivalent to the minimum capacity of the edges of this path. We update the *residual graph* by decreasing capacities in forward edges and increasing capacities in backward edges. Then we look for a new augmenting path.

Here is the *residual graph* after sending 4 units of flow through the *augmenting path* A-C-E-F :

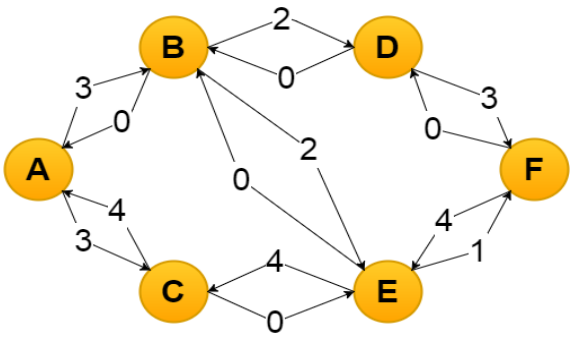

Figure 2.2: Residual graph

The pseudo-code of the augmenting path algorithm is given here :

Look for an augmenting path;
while *There is an augmenting path* **do**
    Send flow through this path;
    Update the residual graph;
    Look for an new augmenting path;
end

2.1.2 Ford-Fulkerson and Edmonds-Karp

There are two main augmenting path algorithms, Ford-Fulkerson (published in 1956) and Edmonds-Karp (published in 1972). The second being a variant of the first one. Indeed, the unique difference between both is the way of looking for an *augmenting path* in the *residual graph*.

Ford-Fulkerson uses a depth-first search while Edmonds-Karp uses a breadth-first search.

2.1.3 Complexities

The max flow problem, being a problem of complexity class P, can be solved at polynomial time. When the capacities are integers, Ford-Fulkerson is bounded by $O(E * f)$ and Edmonds-Karp by $O(V * E^2)$, where E is the number of edges in the graph, V is the number of vertices and f is the maximum flow.

Ford-Fulkerson is in $O(E * f)$ because each augmenting path can be found in $O(E)$ and in the worst case, the flow will increase by 1.

Edmonds-Karp is in $O(V * E^2)$ because the breadth-first search assures us that after each iteration, the length of the augmenting path can't decrease. Futhermore, the length of the augmenting path can stay the same for at most E iterations before increasing. We also know that the length of the augmenting path is between 1 and $V - 1$. Thus there are at most $V * E$ iterations and each augmenting path can be found in $O(E)$.

2.2 Pre-flow algorithms

Chapter 3

Data Structures

3.1 Introduction

Jéxplique que cést pour representer le graphe en memoire. Avec un tableau de structure de donnees ou chaque noeud est une ligne

3.2 Data structures

3.2.1 Hash Map

Jéxplique la structure, avec les operations principales

3.2.2 Tree Map

Jéxplique la structure, avec les operations principales

3.2.3 Simple Linked List

Jéxplique la structure, avec les operations principales

3.2.4 Split Array

Jéxplique la structure, avec les operations principales

3.2.5 Complexities

Je met un tableau avec les complexitees

Chapter 4

Improvements of Existing Algorithms

Chapter 5

Implementation

Chapter 6

Experimental Analysis

Chapter 7

Conclusion

List of Figures

List of Tables