

# Electrical Flows, Laplacian Systems, and Faster Approximation of Maximum Flow in Undirected Graphs

Cristiano, Kelner , et al

Karthik Abinav

## 1 Introduction

This paper gives a laplacian solver based algorithm to the max flow problem. Using electrical flows and fast laplacian solvers, the running time to the  $(1-\epsilon)$ -approximate s-t flow problem in  $\tilde{O}(mn^{\frac{1}{3}}\epsilon^{-\frac{11}{3}})$ . The paper starts of by giving a simpler algorithm which runs in time  $\tilde{O}(m^{\frac{3}{2}}\epsilon^{\frac{5}{2}})$ . Then some key observation in this algorithm is used to get the modified algorithm that runs in time  $\tilde{O}(m^{\frac{4}{3}}\epsilon^{-3})$ . Then using the smoothing and sampling techniques given by [insert bibtex link to karger's paper]. The main drawback with this algorithm is that, it can get the approximate value of the flow. But, there is no technique to map the flow in the sparse graph to the original graph. The paper also discusses an algorithm to dual of this problem that is the minimum s-t cut. It gives a  $(1+\epsilon)$ - algorithm to the minimum s-t cut problem running in time  $\tilde{O}(m + n^{\frac{4}{3}}\epsilon^{-\frac{8}{3}})$ . The advantage of this algorithm is that, it can get the approximate cut along with the value of the minimum cut. Hence, a gap exists between the primal problem and the dual problem. This issue is addressed in a further work by Kelner, Orecchia, et. al.

## 2 Notations

In this section, some notations that will be used throughout this summary will be discussed.

$u_e$  - This defines the capacity associated with the edge  $e$ .

$U$  - This defines the ratio between the highest capacity to the lowest capacity in the graph i.e.  $\frac{\max_e u_e}{\min_e u_e}$

Every edge in the graph is arbitrarily oriented for notational purposes. An edge that comes into a vertex  $v$  is denoted by  $E^+(v)$  and every edge that leaves the vertex  $v$  is defined as  $E^-(v)$ .

A vector  $f$  is a vector of flow. The  $i^{th}$  coordinate of the vector denotes the flow in the  $i^{th}$  edge for a given ordering of the edges

Every valid flow of value  $F$  has the following properties:

- Flow Conservation:

$$\sum_{e \in E^+(v)} f(e) - \sum_{e \in E^-(v)} f(e) = 0 \quad \forall v \in V \setminus \{s, t\}$$

- Net flow value:

$$|f| = F = \sum_{e \in E^+(s)} f(e) - \sum_{e \in E^-(s)} f(e)$$

- Feasible flow :

$$f(e) \leq u_e \quad \forall e \in E$$

**The maximum flow problem:** The goal of this problem is to find a valid flow of value  $F^*$  such that the maximum flow that can be routed from  $s$  to  $t$  in the graph is  $F^*$ .

A  $(1-\epsilon)$ -approximation is an algorithm that returns a valid s-t flow, whose value is atleast  $(1-\epsilon) * F^*$

## 2.1 Electrical flows

Every edge of a graph  $G$ , we visualize an edge as a resistor with resistance  $r_e$ . We can then define an energy associated with a flow  $f$  as follows:

$$\epsilon_r(f) = \sum_e r_e f_e^2$$

**Lemma:**

Among all valid flows of value  $F$ , the one that minimizes the energy is the one that follows kirchoff's laws of current and voltage

**Proof:**

Consider the case of two resistors in series and two resistors in parallel. If we prove the lemma for these two cases, it applies for any general circuit.

*Series:* In this case, there is just one possible flow, which is the electrical flow. Hence, statement trivially holds true.

*Parallel:* Let us start with an electrical flow. We will show that any change to the flows will only increase the energy. Let  $i_1$  and  $i_2$  be the electrical flow through resistors  $r_1$  and  $r_2$ . From Kirchoff's law we have  $i_1 * r_1 = i_2 * r_2$ . The energy now is  $\epsilon = r_1 * i_1^2 + r_2 * i_2^2$ . Let us now redistribute the flow as  $i_1 + \delta$  and  $i_2 - \delta$ . The value of the flow remains the same. The energy now becomes  $\epsilon' = \epsilon + \delta^2(r_1 + r_2) \geq \epsilon$ . Hence the lemma.

**Incidence Matrix:** The incidence matrix gives the details of the directed edges in a graph. The incidence matrix is denoted by  $B$  a  $n * m$  matrix. The  $(i,j)$  entry is 1, if the edge  $j$  leaves vertex  $i$ . It is defined as -1, if the edge  $j$  is incident vertex  $i$ .

Hence, the max flow problem can now be stated as calculating a vector  $f$ , such that.

$$Bf = \chi_{s,t}$$

Here,  $\chi_{s,t}$  is a vector which is 1 at the  $s^{th}$  coordinate, -1 at the  $t^{th}$  coordinate and 0 elsewhere.

**Weighted Laplacian:** Weighted Laplacian  $L$  is defined as a  $n * n$  matrix