Lecture 16

Gidon Rosalki

2024-12-22

1 Vertex Cover

Input: An undirected $G = (V, E)$, and a weight function $w : V \to \mathbb{R}^+$ **Output:** A subset $S^* \subset V$ that contains all the edges of the graph, with a minimal weight for the subset:

$$w\left(S^*\right) = \sum_{x \in S^*} w\left(x\right)$$

What is the relative approximation between this problem, and the algorithm from the unweighted version? Unconstrained. For example, on a graph with two nodes, one of weight ε, and one of weight 1, the above algorithm will choose both nodes, when the optimal solution is just the node ε.

1.0.1 Strategy to create an approximate solution through linear programming

1. We will formalise the problem as a linear programming problem, with integer requirements (ILP)
2. We will remove the requirements of integer on the variables and switch them to linear. We have got a problem of linear programming, which we can solve in polynomial time.
3. We will find the optimal solution to the problem from 2
4. We will build a correct approximate solution to the problem through the partial solution that we found in step 3.

ILP:

- For every node $v \in V$ we will define a variable

$$x\left(v\right) = \begin{cases} 1, & \text{if } v \in S \\ 0, & \text{if } v \notin S \end{cases}$$

- We will check that every edge $e = (u, v) \in E$ is contained within the requirement:

$$x\left(u\right) + x\left(v\right) \geq 1$$

- We want to minimise the weight of the chosen solution:

$$w\left(S\right) = \sum_{v \in S} w\left(v\right) = \sum_{v \in V} x\left(v\right) \cdot w\left(v\right)$$

Let us formally represent the requirements for the ILP:

$$\min_{v \in V : x(u) + x(v) \geq 1 \wedge x(v) \in \{0,1\}} \left\{x\left(v\right) \cdot w\left(v\right)\right\}$$

Note this is not the standard form, since we are not maximising, but rather minimising.

Remove the integer requirements:

$$\min_{v \in V : x(u) + x(v) \geq 1 \wedge 0 \leq x(v) \leq 1} \left\{x\left(v\right) \cdot w\left(v\right)\right\}$$

Find optimal solution to step 2:
The ellipsoid algorithm find the optimal solution (if there is one) in polynomial time.

Theorem 1. *There always exists an optimal solution.*

Proof . There are a finite number of nodes ☐

1.0.2 Connection between the optimal solution to LP and ILP

Let x^*_{LP} be the optimal solution to the linear programming problem, and x^*_{ILP} be the optimal solution to the integer linear programming problem. So

$$\sum_{v\in V} x^*_{LP}(v)\cdot w(v) \le \sum_{v\in V} x^*_{ILP}(v)\cdot w(v)$$

Let there be S_{ILP} the set of correct solutions to the ILP problem, and S_{LP} the set of correct solutions to the LP problem. We shall note that for every integer solution it is also a solution to LP, and therefore $S_{ILP} \subset S_{LP}$. From here:

$$\begin{aligned} w\left(x^*_{LP}\right) &= \sum_{v\in V} x^*_{LP}(v)\cdot w(v) \\ &= \min_{x\in S_{LP}} \left\{\sum_{v\in V} x(v)\cdot w(v)\right\} \\ &\le \min_{x\in S_{ILP}} \left\{\sum_{v\in V} x(v)\cdot w(v)\right\} \\ &= \sum_{v\in V} x^*_{ILP}(v)\cdot w(v) \\ &= w\left(x^*_{ILP}\right) \end{aligned}$$

1.0.3 We will build an approximate solution from the partial solution in step 3

Let there be x^*_{LP} the optimal solution to the linear programming problem. For all $v \in V$ we will define

$$x(v) = \begin{cases} 1, & \text{if } x^*_{LP} \ge \frac{1}{2} \\ 0, & \text{if } x^*_{LP} < \frac{1}{2} \end{cases}$$

Theorem 2 (Lemma)**.** *The rounded solution is a correct solution to the problem.*

Proof . Let there be $(u, v) \in E$ an edge in the graph. Since X^*_{LP} is a correct solution $x^*_{LP}(u) + x^*_{LP}(v) \ge 1$, from here $x^*_{LP}(u) \ge \frac{1}{2} \vee x^*_{LP}(v) \ge \frac{1}{2}$. This implies that $x(u) = 1 \vee x(v) = 1$. Therefore (u, v) is covered, and x must be a covering. □

Theorem 3. *The algorithm is a* 2*-approximation algorithm to the weighted vertex cover problem.*

Proof . Let there be x^*_{LP} the optimal partial solution, and x the rounded solution. Let $v \in V$, if $x^*_{LP}(v) < \frac{1}{2}$ then $x(v) = 0 \le x^*_{LP}(v) \le 2x^*_{LP}(v)$. If $x^*_{LP}(v) \ge \frac{1}{2}$ then $x(v) = 1 \le 2x^*_{LP}(v)$, and therefore, for all $v \in V$, $x(v) \le 2x^*_{LP}(v)$.

$$\begin{aligned} \sum_{v\in V} x(v)\cdot w(v) &\le \sum_{v\in V} 2x^*_{LP}(v)\cdot w(v) \\ &= 2\sum_{v\in V} x^*_{LP}(v)\cdot w(v) \\ &\le 2\sum_{v\in V} x^*_{LP}(v)\cdot w(v) \\ &= 2w\left(S^*\right) \end{aligned}$$

□

2 Flow

2.0.1 Informal network flow

Input: A directed graph, the amounts that each edge can have pass through it, the origin node s, and destination node t

Legal flow:

1. Conservation of flow: What enters is what leaves

2. Capacity limit: What flows on an edge is not more than its capacity

Flux: The amount of flow that leaves the origin node s.
Objective: Find the flow with the maximised flux.

2.0.2 Formal network flow

Definition 2.1 (Network flow)**.** $N = (V, E, c, s, t)$ *where*

- $G = (V, E)$ *the directed graph*
- $c : E \to \mathbb{R}^+$*, the capacity function on the edges*
- $s \in V$*, the origin node*
- $t \in V$*, the destination node*

A correct flow in a flow network N is the function $f : E \to \mathbb{R}^+$ that satisfies:

- Capacity limit: $\forall e \in E \; 0 \leq f(e) \leq c(e)$
- Conservation of flow: $\forall v \in V \setminus \{s, t\} \; \sum_{(u,v) \in E} f(u, v) = \sum_{(v,u) \in E} f(v, u)$

The flux for a given flow f, which is written $|f|$ is the amount of flow that leaves the end node t:

$$\sum_{x \in V \wedge (s,x) \in E} f(s, x)$$

The objective: Given a network flow N, find the flow with the maximised flux. We may solve this problem in polynomial time using Linear Programming. The Ford-Fulkerson algorithm solves this problem in **less time**, and ensures that integer flow in the case that there are integer capacities on the edges.
There is always a feasible solution to the problem, since the flow can be zero.
If we look at two correct flows f, g, then $q = \frac{f + g}{2}$ is also a correct flow. Since the set is closed, and convex, and there exists a feasible solution, then there is an optimal solution.