

## Lecture 15: October 31

Lecturer: Micah Adler

Scribe: Yuandong Yang Zhaojun Wo

## 15.1 Review: Karger's Algorithm

**Theorem 15.1** Let  $C$  be a minimum cut of graph  $G = (V, E)$ , then

$$\Pr(C \text{ returned by Karger's algorithm}) \geq \frac{2}{n^2},$$

where  $n = |V|$

## 15.2 Revised Karger's Algorithm

We can simply repeat Karger's algorithm  $m$  times and return the smallest cut found. Given  $\epsilon > 0$ , we can ensure that the probability that this algorithm fails is less than  $\epsilon$  by adjusting  $m$  accordingly. In fact, if we let

$$m = \frac{n^2}{2} \cdot \ln \frac{1}{\epsilon},$$

then

$$\begin{aligned} \Pr(C \text{ is never returned}) &\leq \left(1 - \frac{2}{n^2}\right)^m \\ &= \left(1 - \frac{2}{n^2}\right)^{\frac{n^2}{2} \cdot \ln \frac{1}{\epsilon}} \\ &\leq \left(\frac{1}{e}\right)^{\ln \frac{1}{\epsilon}} \\ &= \epsilon. \end{aligned}$$

The second inequality is from that fact that

$$\left(1 - \frac{1}{k}\right)^k \leq \frac{1}{e}. \quad (15.1)$$

To get a sense of what this result means, consider  $\epsilon = e^{-100}$ , then we only need to make

$$m = \frac{n^2}{2} \ln e^{100} = 50n^2.$$

How small is this probability  $\epsilon$ ? Well, the probability of a person getting hit by a meteor is about  $e^{-60}$

Using a more involved technique with probability of being wrong  $\epsilon$ , the running time of the algorithm is

$$O\left(n^2 \log n \log \frac{1}{\epsilon}\right).$$

## 15.3 Verifying Polynomial Identities

**Input:**  $n$ -variable polynomial  $Q(X_1, X_2, \dots, X_n)$  of degree  $d$ , with a representation that is “easy” to evaluate at any point.

**Output:** The answer to the question “Is  $Q(X_1, X_2, \dots, X_n) \equiv 0$  ?”

The  $\equiv$  sign means *equivalent*, i.e., no matter what values the variables take, the polynomial is always zero. For example,

$$Q = X_1^2 X_2^3 + X_3 \not\equiv 0, \quad (15.2)$$

here  $n=3$ ,  $d=5$ . while

$$Q = (X_1 + X_2)(X_1 - X_2) + (X_2 + X_3)(X_2 - X_3) + (X_3 + X_1)(X_3 - X_1) \equiv 0. \quad (15.3)$$

Polynomial 15.2 can be easily seen to not be equivalent to zero. To see that 15.3 is equivalent to zero, we can simply expand the polynomial and all the terms cancel each other out.

An equivalent problem to this is to test if two polynomials, possibly in different representations, are actually the same. We can simply test if the difference of the two polynomials is equivalent to zero.

In general, there is no known polynomial time deterministic algorithm for this problem. However, a Monte Carlo type of algorithm works well for this problem.

### 15.3.1 Schwartz’s Algorithm

1. Let  $S$  = a set of distinct real numbers
2. For  $i = 1$  to  $n$  do
3.     pick  $z_i \in S$  uniformly random
4. if  $Q(z_1, z_2, \dots, z_n) = 0$  then output “YES”
5. else output “NO”

We observe that a “No” answer from Schwartz’s algorithm is always correct, since we know for sure there is a set of numbers that makes the polynomial not equal to zero. The “Yes” answer from the algorithm might be wrong. However, the following theorem states that the probability that it’s wrong can be bounded.

**Theorem 15.2** *If  $Q \equiv 0$ , then the output of Schwartz’s algorithm is always correct. If  $Q \not\equiv 0$ , then the probability that the output is wrong is less than  $d/|S|$ , i.e.,*

$$Pr(Q(z_1, z_2, \dots, z_n) = 0) \leq \frac{d}{|S|}. \quad (15.4)$$

**Proof:** The first half of the theorem is obvious. The second half of the theorem is proved by induction on  $n$ .

**Base case:** If  $n = 1$ ,  $Q$  is a single variable polynomial of degree  $d$ . Then there are at most  $d$  real numbers that makes  $Q$  evaluate to 0. Thus

$$Pr(Q(z_1) = 0) \leq \frac{d}{|S|}.$$

**Inductive step:** Suppose that for all polynomials with less than  $n$  variables, the claim of the theorem is true. Now consider a degree  $d$  polynomial with  $n$  variables,  $Q(X_1, X_2, \dots, X_n)$ . We can rewrite  $Q$  as a

polynomial of  $X_1$ , (treating other variables as constants):

$$Q(X_1, X_2, \dots, X_n) = \sum_{i=0}^k Q_i(X_2, \dots, X_n) X_1^i,$$

where  $k$  is the maximum degree of  $X_1$ . Obviously  $Q_k(X_2, \dots, X_n) \neq 0$  because otherwise there will be no term of  $X_1$  with degree  $k$ . Also note that the degree of  $Q_k$  can be no larger than  $d-k$ . Now pick  $z_2, z_3, \dots, z_n$  randomly from  $S$ . Since  $Q_k(X_2, \dots, X_n)$  is a  $n-1$  variable polynomial, by induction,

$$\Pr(Q_k(z_2, \dots, z_n) = 0) \leq \frac{d-k}{|S|}.$$

On the other hand, if  $Q_k(z_2, \dots, z_n) \neq 0$ , then

$$Q(X_1, z_2, z_3, \dots, z_n) = \sum_{i=0}^k Q_i(z_2, \dots, z_n) X_1^i$$

is a one variable ( $X_1$ ) polynomial with degree  $k$ . Thus given that  $Q_k(z_2, \dots, z_n) \neq 0$ ,

$$\Pr(Q(z_1, z_2, z_3, \dots, z_n) = 0) \leq \frac{k}{|S|},$$

which is equivalent to saying that

$$\Pr(Q(z_1, z_2, z_3, \dots, z_n) = 0 \mid Q_k(z_2, \dots, z_n) \neq 0) \leq \frac{k}{|S|}.$$

Thus,

$$\begin{aligned} & \Pr(Q(z_1, \dots, z_n) = 0) \\ &= \Pr(Q(z_1, \dots, z_n) = 0 \mid Q_k(z_2, \dots, z_n) = 0) + \Pr(Q(z_1, \dots, z_n) = 0 \mid Q_k(z_2, \dots, z_n) \neq 0) \\ &\leq \Pr(Q(z_1, \dots, z_n) = 0 \mid Q_k(z_2, \dots, z_n) = 0) + \Pr(Q_k(z_2, \dots, z_n) \neq 0) \\ &= \frac{k}{|S|} + \frac{d-k}{|S|} \\ &= \frac{d}{|S|}. \end{aligned}$$

■

### 15.3.2 Discussion

If we set  $|S| = 2d$ , then the probability that Schwartz's algorithm returns the wrong answer is less than  $1/2$ , given that  $Q \neq 0$ . If we repeat the algorithm for  $\log_2(1/\epsilon)$  times, then the probability that algorithm returns the wrong answer all the time is less than:

$$\left(\frac{1}{2}\right)^{\log_2\left(\frac{1}{\epsilon}\right)} = \epsilon.$$

### 15.3.3 Boolean Satisfiability

**Input:** Boolean formula  $\phi(X_1, X_2, \dots, X_n)$ .

**Question:** Is  $\phi \equiv FALSE$ ?

This problem looks similar to the polynomial identity verification problem, but actually it's not. A random algorithm can not solve this problem with high enough probability of success within a reasonable time period. For example, let  $\phi = X_1 \wedge X_2 \wedge \dots \wedge X_n$ . We know that  $\phi \neq FALSE$ . However,

$$Pr\left(\phi(z_1, z_2, \dots, z_n) = TRUE\right) = \frac{1}{2^n},$$

thus,

$$Pr\left(\phi(z_1, z_2, \dots, z_n) = FALSE\right) = 1 - \frac{1}{2^n}.$$

That is, if  $\phi \neq FALSE$ , the probability that the algorithm gives a wrong answer is  $1 - 1/2^n$ . We will need to run the algorithm an exponential number of times to bring down the error probability.

### 15.3.4 Application of Schwartz's Algorithms

**Input:** A bipartite graph  $G = (U, V, E)$ , where  $|U| = |V| = n$ .

**Question:** Does  $G$  contain a perfect matching (i.e., a matching of size  $n$ )?

### 15.3.5 Network Flow Solution

We can use Network Flow to solve this problem, and we know that the Even-Tarjan[ET75] algorithm can solve it in time  $O(m\sqrt{n})$ , where  $m$  is the number of edges.

Let  $M[G]$  be the  $n \times n$  adjacency matrix for  $G = (U, V, E)$ , defined as follows:

$$M[G](i, j) = \begin{cases} 1 & \text{if } (u_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases}$$

Let  $A[G]$  be an  $n \times n$  matrix s.t.

$$A[G](i, j) = \begin{cases} x_{ij} & \text{if } M[G](i, j) = 1 \\ 0 & \text{otherwise} \end{cases}$$

**Example:**

Here is a bipartite graph  $G = (U, V, E)$ , where  $|U| = |V| = 4$ :

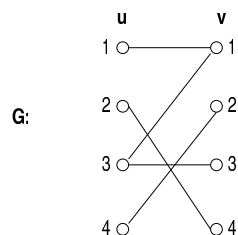


Figure 15.1: Example of a Bipartite Graph

From this graph,  $G$ , in Figure 14.1, we see that  $E = \{(u_1, v_1), (u_2, v_4), (u_3, v_1), (u_3, v_3), (u_4, v_2)\}$ , so:

$$A[G] = \begin{bmatrix} x_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{24} \\ x_{31} & 0 & x_{33} & 0 \\ 0 & x_{42} & 0 & 0 \end{bmatrix}$$

**Theorem 15.3 (Tutte)**  $G$  contains a perfect matching iff  $\det(A[G]) \neq 0$ .

## References

K74 A. KARZANOV, Determining the maximal flow in a network with the method of pre-flows, *Soviet Math. Dokl.* 15, 1974, pp. 434-437.

- WEI WEI, HUAN LI, subscribe-note of **Advanced Algorithm of Fall 2000**, lecture: *Micah Adler*.