18.415/6.854 Advanced Algorithms November 26, 2001

Lecture 19

Lecturer: Michel X. Goemans *Scribe: Shalini Agarwal, Shane Swenson*

Previously, we discussed solving a system of equations of the form:

$$x_i + x_j + x_k \equiv \left\{ \begin{array}{l} 0 \\ 1 \end{array} \right. \pmod 2 : x_i, x_j, x_k \in \{0, 1\}$$

The goal is to satisfy as many equations as possible. It is easy to satisfy at least half of them by assigning all $x_i = 0$ or all $x_i = 1$. In this lecture, we consider the related problem of satisfying a set of linear equations $\pmod 2$, each having exactly 2 variables. We call this problem Lin-2-Mod-2.

1 Lin-2-Mod-2

Given m equations on n variables of the form:

$$x_i + x_j \equiv \left\{ \begin{array}{l} 0 \\ 1 \end{array} \right. \pmod 2$$

Assign a value from {0,1} to each x_i so that the maximum number of equations are satisfied.

1.1 Simple 0.5-approximation algorithm

This 0.5-approximation algorithm finds a solution that satisfies a number of equations greater than or equal to half of the number satisfied by the optimal solution.

1.1.1 Randomized Algorithm

Select an assignment uniformly at random.

There are 2^n possible assignments so

$$Pr(x_i = a_i \ \forall i) = \frac{1}{2^n} \ \forall (a_1, a_2, ..., a_n).$$

In order to bound the expected number of equations satisfied by this assignment let

$$I_i = \left\{ \begin{array}{ll} 1 & \text{if equation } i \text{ is satisfied} \\ 0 & \text{otherwise} \end{array} \right.$$

Claim 1 $E[I_i] = \frac{1}{2}$

Proof of claim:

$$E[I_i] = Pr[\text{equation } i \text{ is satisfied}]$$

Equation i has the form $x_j + x_k \equiv c_i \pmod 2$ where $c_i \in \{0,1\}$. Since x_j is independent of x_k and $Pr[x_j = 1] = \frac{1}{2}$,

$$\begin{aligned} Pr[\text{equation } i \text{ is satisfied}] &= Pr[\text{equation } i \text{ is satisfied}|x_k] \\ &= \frac{1}{2}. \end{aligned}$$

$\square$

Now consider the expected number of equations satisfied.

$$\begin{aligned} E[\# \text{ equations satisfied}] &= E[\sum_{i=1}^{m} I_i] \\ &= \sum_{i=1}^{m} E[I_i] \\ &= \frac{1}{2}m \geq \frac{1}{2}OPT \end{aligned}$$

where OPT is the number of equations satisfied by the optimum assignment.

1.1.2 Derandomization

We now present two ways of derandomizing the above randomized algorithm. We can either use the method of conditional expectations to assign values to variables one-by-one or we can reduce the sample space to one of polynomial size and perform an exhaustive search.

- **Method of conditional expectations**
  Let $W = \#$ *equations satisfied*. Following the formula for conditional expectations:

  $$\begin{aligned} E[W] &= E[W|x_1 = 0] \cdot Pr[x_1 = 0] + E[W|x_1 = 1] \cdot Pr[x_1 = 1] \\ &\leq max(E[W|x_1 = 0], E[W|x_1 = 1]) \end{aligned}$$

  since $Pr[x_1 = 0] = Pr[x_1 = 1] = \frac{1}{2}$.
  Suppose $E[W|x_1 = 0]$ is the maximum in the above expression. Then we set $x_1 = 0$ and examine the next variable, x_2. Consider the equation:

  $$\begin{aligned} E[W|x_1 = 0] &= E[W|x_1 = 0, x_2 = 0] \cdot Pr[x_2 = 0] + E[W|x_1 = 0, x_2 = 1] \cdot Pr[x_2 = 1] \\ &\leq max(E[W|x_1 = 0, x_2 = 0], E[W|x_1 = 0, x_2 = 1]) \end{aligned}$$

  By choosing the variable assignment that yields the maximum conditional $E[W]$ in each step we have:

  $$\frac{1}{2}OPT \leq E[W] \leq E[W|x_1] \leq E[W|x_1, x_2] \leq ... \leq E[W|x_1, x_2, ..., x_n]$$

  so our assignment is a 0.5-approximation.

Now we show how to compute $E[W|x_1, x_2, ..., x_k]$. From above,

$$E[W|x_1, x_2, ..., x_t] = \sum_{i=1}^{m} E[I_i|x_1, x_2, ..., x_t]$$

and if equation i has the form $x_j + x_k \equiv c_i \pmod 2$ then

$$E[I_i|x_1, x_2, ..., x_t] = \begin{cases} \frac{1}{2} & \text{if } x_j > t \text{ } or \text{ } x_k > t \\ 1 & \text{if } i, j \leq t \text{ } and \text{ } x_i + x_j \equiv c_i \pmod 2 \\ 0 & \text{otherwise} \end{cases}$$

So to compute $E[W|x_1, x_2, ..., x_t]$ we simply add 1 for all equations already satisfied and $\frac{1}{2}$ for all equations not yet determined.

- **Reducing the sample space**
  In the analysis of our randomized algorithm, we used two properties of our random assignment.
  - $Pr[x_i = 1] = \frac{1}{2} \; \forall i$
  - Pair-wise independence: $Pr[(x_i = a_i) \wedge (x_j = a_j)] = Pr[x_i = a_i] \cdot Pr[x_j = a_j] \; \forall i, j, a_i, a_j$

  We will now construct a polynomial-size sample space that preserves these two properties. If we choose assignments uniformly from this sample space, our previous analysis of $E[W]$ remains valid and we have

$$E[W] \geq \frac{1}{2} OPT.$$

  Thus if we exhaustively search the sample space we are guaranteed to find an assignment such that

$$W \geq \frac{1}{2} OPT.$$

  Assume n has the form (power of 2) - 1. If not, round it up to the next such number. We can construct a set of "Hadamard" matrices recursively as follows:

$$H_1 = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

$$H_{i+1} = \begin{bmatrix} H_i & H_i \\ H_i & J - H_i \end{bmatrix}$$

  where J is the appropriately dimensioned matrix with every entry equal to 1.

  For example, here is H_2:

$$H_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

  Since n is (power of 2) - 1, let $n + 1 = q = 2^k$. We define our distribution of assignments as follows. Let $a_1, a_2, ..., a_n$ be the elements of any column of H_k excluding the top element of that column. That is, if we number the rows of H_k beginning with 0, a_1 is always taken from row 1, a_2 is always taken from row 2, etc. and all a_i are taken from the same column of H_k.

$$H_k = \begin{bmatrix} . & \cdots & . & \cdots & . \\ . & \cdots & a_1 & \cdots & . \\ . & \cdots & a_2 & \cdots & . \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ . & \cdots & a_n & \cdots & . \end{bmatrix}$$

Then let

$$Pr[x_i = a_i \ \forall i] = \frac{1}{q}.$$

Claim 2 *Using this distribution,* $Pr[x_i = 1] = \frac{1}{2} \ \forall i$.

Proof of claim: We use induction to show that each row, other than the top row, of H_n has an equal number of 1s and 0s for all n. Since we choose columns of H_k uniformly at random, this is sufficient to prove the claim.

- Base case:

$$H_1 = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

  The second row of H_1 has an equal number of 1s and 0s.

- Inductive step: Assume each row, other than the top row, of H_i has an equal number of 1s and 0s and consider the rows of H_{i+1}.

$$H_{i+1} = \begin{bmatrix} H_i & H_i \\ H_i & J - H_i \end{bmatrix}$$

  There are two cases to consider.

  - If the row is in the top half of H_{i+1}, but isn't the top row, it has identical left and right halves, each of which are comprised of an equal number of 1s and 0s by the inductive hypothesis.
  - If the row is in the bottom half of H_{i+1}, each 1 in the left half corresponds with a 0 in the same position in the right half and vice versa.

  Thus every row, except for the top row, of H_{i+1} has an equal number of 1s and 0s.

□

Claim 3 *Using this distribution,* x_i *and* x_j *are pairwise independent for all* i, j.

Proof of claim: We use induction on n to show that if we vertically match any two distinct rows, excluding the top row, of H_n we have an equal number of 00, 01, 10, and 11 pairs.

- Base case:

$$H_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

- Inductive step: Assume each pair of rows, excluding the top row, of H_i has an equal number of 00, 01, 10, and 11 pairs and consider the pairs of rows of H_{i+1}.

$$H_{i+1} = \begin{bmatrix} H_i & H_i \\ H_i & J - H_i \end{bmatrix}$$

  There are five cases to consider.

  - If both rows are in the top half of H_{i+1} the claim follows from the inductive hypothesis.
  - If both rows are in the bottom half of H_{i+1} and neither row is the top row of that half, the claim holds for the left halves of the pair by the inductive hypothesis. Since the right halves are merely inversions of the left halves, the claim also holds for the right halves and therefore applies to the pair of rows along their entire length.
  - If one row is taken from the top half of H_{i+1} and the other is taken from the bottom half, they correspond to different rows in H_i, and neither row corresponds to the top row of H_i, the claim holds for the left halves by the inductive hypothesis. For the right halves, apply the inductive hypothesis to the original rows and replace 00 with 01, 01 with 00, 10 with 11, and 11 with 10. This proves the claim for the pair of rows along their entire length.
  - If one row is taken from the top half of H_{i+1} and the other row is taken from the bottom half and they correspond to the same row in H_i, from the proof of the previous claim we find exactly 2^{i-1} 00 and 2^{i-1} 11 pairs in the left halves of the rows and 2^{i-1} 01 and 2^{i-1} 10 pairs in the right halves of the rows.
  - If one of the rows is in the bottom half of H_{i+1} and corresponds to the top row of H_i, we again use the previous claim to find 1 matched with an equal number of 1s and 0s in the left halves and 0 matched with an equal number of 1s and 0s in the right halves of the rows.

$\square$

We have now proved the necessary properties of this distribution to apply our previous reasoning and deduce that if we sample uniformly from this new sample space

$$E[W] = \frac{1}{2}m$$

and therefore there exists an assignment in this sample space such that

$$W \geq \frac{1}{2}m \geq \frac{1}{2}OPT.$$

Since the sample space contains less than $2n$ possibilities, we can exhaustively search it for such an assignment.

1.2 0.878-approximation algorithm based on "semidefinite" programming

In this algorithm, we convert the original Lin-2-Mod-2 problem into a slightly different, more familiar graph problem. We define the graph G to contain a vertex for each variable, a solid edge for each equation valued at 1, and a dashed edge for each equation valued at 0.

To solve this problem, we find a partition of G, such that the maximum number of solid edges are cut and the minimum number of dashed edges are cut. We call this partition MAXCUT.

1.2.1 MAXCUT

Given a graph $G = (V, E)$, find $S \subset V : \max d(S)$, where $d(S) = |\delta(S)| = \#$ of cut edges. More formally, we can rewrite MAXCUT as:

$$\max \sum_{(i,j)\in E} \frac{1 - x_i x_j}{2}$$

such that

$$x_i \in \{1, -1\} \; \forall i.$$

The constraint can be rewritten as $x_i \in \mathbb{R} : |x_i| = 1$.

1.2.2 Relaxation

Since the MAXCUT problem as stated above is hard to solve, we find a relaxation that can be solved in almost polynomial time. This relaxation is defined as:

$$\max \sum_{(i,j)\in E} \frac{1 - v_i v_j}{2}$$

such that

$$v_i \in \mathbb{R}^p \; and \; \|v_i\| = 1, \; \forall i.$$

This relaxation gives an upper bound on the MAXCUT as it is maximizing the same objective function subject to weaker constraints.

1.2.3 Example

For example consider C_5, a graph comprised of a 5-cycle. The max cut for this graph is 4. Figure 1 shows C_5 and the optimal solution to its relaxation for $p = 2$. In this case, $\angle v_i = \frac{4\pi}{5}(i-1)$. Thus

$$v_i v_j = \cos\frac{4\pi}{5} \; \forall (i,j) \in E$$

and

$$\sum_{(i,j)\in E} \frac{1 - v_i v_j}{2} = \frac{5(1 - \cos\frac{4\pi}{5})}{2} \approx 4.52$$

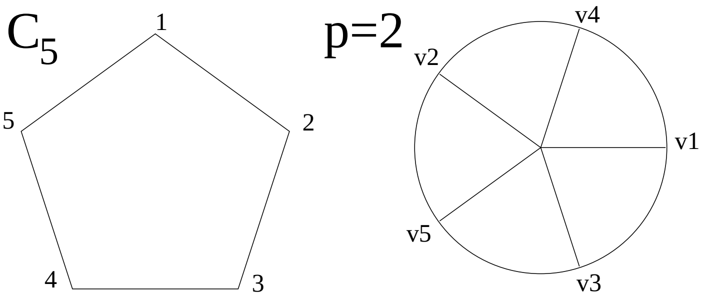

Figure 1: C_5 and the relaxation for $p = 2$

1.2.4 Randomized Algorithm

In order to convert our relaxed solution to a cut we choose a hyperplane in $\mathbb{R}^p$ and assign 1 to all vectors on one side of it and -1 to all vectors on the other.

Select vector r uniformly at random from $S_{p-1} = \{x \in \mathbb{R}^p : \|x\| = 1\}$. Then let

$$x_i = \begin{cases} 1 & \text{if } r \cdot v_i \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

and

$$S = \{i : x_i = 1\}.$$

Theorem 4 *If we choose S in this way,*

$$E[d(S)] \geq 0.878UB \geq 0.878OPT.$$