6.893 Approximability of Optimization Problems	November 29, 1999
Lecture 22	
Lecturer: Madhu Sudan	*Scribe: Matthew Frank*

1 Hardness of Approximation

We will start with Håstad's PCP and use it to generate a bound for approximations to the MAX-CUT problem. In the process we will generalize the result to a large class of problems that have similar gadgets.

Håstad's PCP has a verifier that tosses $O(\log n)$ coins and looks at the proof in 3 bits, π_{i_1}, π_{i_2}, and π_{i_3}. In addition it generates a bit b and then accepts if $\pi_{i_1} \oplus \pi_{i_2} \oplus \pi_{i_3} = b$.

The MAX-3LIN problem is a natural optimization problem corresponding to this PCP.

Instance: n boolean variables
m linear equations, each with 3 variables
(e.g. $\pi_{i_1} \oplus \pi_{i_2} \oplus \pi_{i_3} = b$)

Goal: Find an assignment that satisfies the maximum number of equations.

MAX-3LIN maps naturally to Håstad's PCP, with the result that finding the maximum number of satisfied constraints to within a factor of $2-\epsilon$ is NP-Hard. Even better, this demonstrates that deciding whether an instance has more than $m(1-\epsilon)$ or less than $\frac{m}{2}(1-\epsilon)$ satisfiable constraints is NP-Hard.

By typical (Garey and Johnson type) NP-reductions we can show that

$$\text{MAX-3LIN} \leq \text{MAX-3SAT} \leq \text{MAX-2SAT} \leq \text{MAX-CUT}$$

This gives us the following harness results:

MAX-3LIN		MAX-3SAT		MAX-2SAT		MAX-CUT
hard w/in	$\Rightarrow$	hard w/in	$\Rightarrow$	hard w/in	$\Rightarrow$	hard w/in
$2-\epsilon$		$\frac{8}{7}-\epsilon$		$\frac{71}{70}-\epsilon$		$\frac{141}{140}-\epsilon$

Note that all of these problems are of a similar nature: MAX $\mathcal{F}$, where $\mathcal{F}$ is a family of boolean functions.

Instance: n boolean vars
m constraints of form $f = b, f \in \mathcal{F}$

Goal: Find assignment that satisfies max number of constraints.

The distinction between the problems has to do with the nature of the constraints. That is, the boolean functions in the constraints are of different types for each problem. For example:

Problem:	Constraints look like:
3LIN	$\{x \oplus y \oplus z = 0$ or $x \oplus y \oplus z = 1\}$
3SAT	$\{x \vee y \vee z, x \vee y \vee \bar{z}, x \vee \bar{y} \vee \bar{z}, \bar{x} \vee \bar{y} \vee \bar{z}\}$
2SAT	$\{x \vee y, x \vee \bar{y}, \bar{x} \vee \bar{y}\}$
CUT	$\{x \oplus y = 1\}$

2 Gadget Reductions

Our goal is to reduce between function families. For example, mapping between 3LIN and 3SAT uses the following gadget (the typical Garey and Johnson thing):

$$\begin{aligned} x \oplus y \oplus z = 0 \quad &\rightarrow \quad \bar{x} \vee \bar{y} \vee \bar{z} \\ & \quad x \vee y \vee \bar{z} \\ & \quad x \vee \bar{y} \vee z \\ & \quad \bar{x} \vee y \vee z \\ x \oplus y \oplus z = 1 \quad &\rightarrow \quad \ldots \end{aligned}$$

3SAT to 2SAT is a very similar gadget reduction with each 3SAT being assigned some auxiliary variables and something like 10 2SAT constraints.

To generalize, we want to reduce a function f, (*e.g.* $x \oplus y \oplus z = 0$), to a function family $\mathcal{F}$, (*e.g.* the 3SAT constraints).

Given $f(x_1, \ldots, x_k)$, create auxiliary variables $x_{k+1}, \ldots, x_n$. We'd like our gadget to create a series of m weighted functions in $\mathcal{F}$:

$$\begin{array}{c} f_1(x_{i_{1,1}}, x_{i_{2,1}}, \ldots, x_{i_{k,1}}) = 1 \\ \vdots \\ f_m(x_{i_{1,m}}, x_{i_{2,m}}, \ldots, x_{i_{k,m}}) = 1 \end{array} \qquad f_j \in \mathcal{F}$$

Completeness of Gadgets

1. If $f(a_1, \ldots, a_k) = 1$ then $\exists b_{k+1}, \ldots, b_N$ such that at least α of the constraints $f_j(a_1, \ldots, a_k, b_{k+1}, \ldots, b_N)$ are satisfied.

Soundness of Gadgets

2. If $f(a_1, \ldots, a_k) = 1$ then $\forall b_{k+1}, \ldots, b_N$ at most α of the constraints are satisfied.

3. If $f(a_1, \ldots, a_k) = 0$ then $\forall b_{k+1}, \ldots, b_N$ at most $\alpha - 1$ of the constraints are satisfied.

Condition 2 is required because without it, we'd be able to reduce MAX-3SAT to MAX-1SAT with $\alpha = 1$ using the following gadget:

$$\begin{array}{ccc} x \vee y \vee z & \rightarrow & x \\ & & y \\ & & z \end{array}$$

A gadget satisfying these completeness and soundness conditions with parameter α is called an *α-gadget*. The definition can also be easily extended to deal with weighted clauses.

For example, in the standard Garey and Johnson gadget for 3SAT $\leq$ 2SAT, each $f \in$ 3SAT constraint gets mapped to 10 2SAT constraints. Then if $f = 1$ *exactly* 7 of the 2SAT clauses are satisfied and if $f = 0$ then 6 (< 7) of the 2SAT clauses are satisfied. So this is a 7-gadget.

Given an α_0-gadget reducing $x \oplus y \oplus z = 0$ to $\mathcal{F}$ and an α_1-gadget reducting $x \oplus y \oplus z = 1$ to $\mathcal{F}$ then MAX-$\mathcal{F}$ is hard to approximate to within

$$\frac{1}{1 - \frac{1}{2\max\{\alpha_0, \alpha_1\}}}$$

And given an α-gadget reducing every function $f \in \mathcal{F}_1$ to $\mathcal{F}_2$ and MAX-$\mathcal{F}_1$ is hard to approximate to $\frac{1}{1-\frac{1}{\beta}}$ then MAX-$\mathcal{F}_2$ is hard to approximate to $\frac{1}{1-\frac{1}{\alpha\beta}}$.

3 Finding the Best Gadget

We are trying to find a reduction directly from MAX-3LIN $\leq$ MAX-CUT. That is, reducing functions $x \oplus y \oplus z = 1$ and $x \oplus y \oplus z = 0$ both to the function family $\{x \oplus y = 1\}$. There are many gadgets that might work, each with a different α parameter. The question is how do we find the *best* gadget?

The reason that this problem is hard is that it is not obvious how to bound the number of auxiliary variables we are adding. The completeness condition is really what makes things hard, because it has an existential qualifier ($\exists$), while the soundness conditions have universal qualifiers ($\forall$). None the less, [Trevisan, Sorkin, Sudan and Williamson] have found a bound for a large class of problems which includes MAX-3LIN $\leq$ MAX-CUT.

Let $x_{k+1}, \ldots, x_n$ and $f_1, \ldots, f_m$ form an α-gadget reducting f to $\mathcal{F}$. Then for every assignment $f(a_1, \ldots, a_k) = 1, \exists b_{k+1}, \ldots, b_n$ such that α of the clauses are satisfied. Let $\mathcal{S} = \{(a_1, \ldots, a_k) | f(a_1, \ldots, a_k) = 1\}$. Now treat $\vec{b}$ as a function from $\mathcal{S} \rightarrow \{0,1\}^{n-k}$.

Suppose that $b_{i_1}(\vec{a}) \equiv b_{i_2}(\vec{a})$ for all $\vec{a} \in \mathcal{S}$. Then i_1 and i_2 can be identified.

As an example, let's look at what might happen if we have gadget mapping from a constraint in 3LIN to some hypothetical function family of 8 variables:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
0	0	0	1	0	1	1	1
0	1	1	0	1	0	1	1
1	0	1	0	1	0	0	1
1	1	0	1	0	1	1	0

Each row of the table corresponds to one of the satisfying assignments to the 3LIN constraint (in this case, those where x_1, x_2, x_3 have even parity). These are the members of $\mathcal{S}$ and in general $s = |\mathcal{S}|$ will be $O(2^k)$. Note that the columns corresponding to auxiliary variables x_4 and x_6 are the same.

So the nice property we are looking for is that $\mathcal{F}$ is closed under replication of variables. For example MAX-3SAT allows constraints of the form $x \vee x \vee z$. Given this property then we can limit the number of possible auxiliary variables, and each auxiliary variable can be given a unique $s = O(2^k)$ bit vector. This vector is simply its column in the table. Let N be the number of auxiliary variables. We have demonstrated that $N \leq 2^s = 2^{2^k}$.

Note that [Karloff and Zwick] have shown that there is no finite reduction for at least one function family where variables can not be replicated, $f = \{x = 1\}$ and $\mathcal{F} = \{x \oplus y \oplus z = 1\}$. So the condition that $\mathcal{F}$ is closed under replication of variables is necessary.

Once we have limited the number of auxiliary variables we can calculate the number of possible constraints we could generate. It will be something on the order of $m = \begin{pmatrix} N \\ k \end{pmatrix} k!|\mathcal{F}|$. There will then be order of $2^m \approx 2^{(2^{2^k})^k}$ possible gadgets to search.

We can improve this search considerably by formulating it as a linear program. We'd like to set up a system of constraints for our three gadget completeness and soundness conditions. For example, suppose we are given weights $w_1, \ldots, w_m$ associated with the m constraints, and an assignment $a_1, \ldots, a_k, b_{k+1}, \ldots, b_n$. Now suppose $f(a_1, \ldots, a_n) = 1$. We'd like

$$\sum_{\substack{j \in \text{ constraints satisified by} \\ a_1, \ldots, a_k, b_{k+1}, \ldots b_N}} w_j \leq \alpha$$

This is condition 2 (soundness for true constraints). There are similar LP constraints for completeness and the other soundness condition. This gives us a linear program which can be solved in a reasonable amount of time. Better yet, the resulting gadgets have provably optimal α values by LP Duality.

Here are some results that have been produced by this technique:

f	$\mathcal{F}$	α
$x \oplus y \oplus z = 0$	$\{x \oplus y = 1\}$ (MAXCUT)	8-gadget (5 aux vars)
3LIN	CUT	9-gadget
3SAT	2SAT	3.5-gadget
kSAT	3SAT	(k-2)-gadget

By these results finding a $\frac{17}{16}$-approximation for MAX-CUT is NP-Hard.

Several similar cut problems have tighter bounds. MAX-DICUT (the problem of cutting the maximum number of links going left to right in a directed graph) is hard within $\frac{13}{12}$. MAX-CUT with "anti-edges", represented by $\mathcal{F} = \{x \oplus y = 0, x \oplus y = 1\}$ is hard to approximate to within $\frac{12}{11}$.

4 Classification

Let's now look at characterizing maximization problems. [Creignou], and [Khanna, Sudan, and Williamson] have developed the following results for maximization problems over families of boolean functions.

$\forall \mathcal{F} \exists \alpha$ such that MAX-$\mathcal{F}$ is either polynomial time solvable (exactly) or is NP-Hard to approximate to within some factor $\alpha > 1$.

Further, $\mathcal{F}$ is poly-time solvable under only the following conditions:

1. $\forall f \in \mathcal{F}, f(\vec{0}) = 1$.
2. $\forall f \in \mathcal{F}, f(\vec{1}) = 1$.
3. $\forall f \in \mathcal{F}, f$ is representable as $f(x_1, \ldots, x_k) = (x_{i_1} \wedge x_{i_2} \wedge \ldots \wedge x_{i_p}) \vee (\bar{x_{j_1}} \wedge \bar{x_{j_2}} \wedge \ldots \wedge \bar{x_{j_q}})$. (*e.g.* we can formulate min-cut with $s \vee \bar{t}$ directed edges).

For *every* other function family we can find a gadget that proves that $\mathcal{F}$ is hard to approximate. This classification is similar to [Schaefer]'s demonstration that exact problems on boolean function families fall either into P or NP-complete.