An instance of subset sum hard for cutting planes

Yuval Filmus

November 20, 2015

Abstract

Proof complexity started its life as an attempt to prove that NP is different from coNP. The main goal of the area is to show that no proof system can prove that all negative instances of 3SAT are unsatisfiable using polynomial size proofs. However, this goal has proved too difficult, and nowadays the area concentrates on proving the same for specific proof systems.

The focus of this talk is the proof system *cutting planes*, which is the proof complexity analog of the popular *branch-and-bound* algorithm for integer programming. A proof in this system consists of a sequence of inequalities over Boolean variables. We will describe a pair of such inequalities (encoding an unsatisfiable subset sum instance) that have no common Boolean solutions, but proving this in cutting planes requires a proof of exponential size.

No familiarity with proof complexity will be assumed. Joint work with Pavel Hrubeš and Massimo Lauria.

1 Proof complexity

The fundamental problem in theoretical computer science is showing that P is different from NP. In order to do that, we have to show that no polynomial time algorithm can solve 3SAT. A few decades ago, Cook suggested a line of attack, nowadays known as *Cook's program*: show that there is no efficient way to "get convinced" that a 3SAT instance is unsatisfiable (in contrast, it is easy to "get convinced" that a 3SAT instane is satisfiable: all one needs is a satisfying assignment). This will show that NP differs from coNP, and in particular P differs from NP.

As a starting point, several specific proof systems were studies, the goal being to show that these proof systems don't have polynomial size refutations of all unsatisfiable 3SAT instances. The first proof system which was seriously studied was Resolution, for which Haken proved exponential lower bounds. Other proof systems were then suggested, and strong lower bounds were proved for some of them. However, eventually it became clear that the study of weak proof systems would not lead to a resolution of Cook's program any more than the study of restricted circuit classes would lead to a separation of P from NP. Proof complexity is now being studied as a respected discipline in its own right.

Another motivation for the study of specific proof systems, especially Resolution, is their (implicit) use in SAT solvers: a SAT solver functions in effect as a refutation system for unsatisfiable 3SAT instances. Most popular SAT solvers use a variant of the DPLL algorithm, and it turns out that their runs can always be converted to Resolution refutations. Hence a lower bound on Resolution is a limitation on SAT solvers.

2 Cutting planes

Cutting planes was introduced by Gomory as a technique for solving integer programs, and was further studied by Chvátal. The main idea of the method is that if x is an integer and $x \geq r$ then in fact $x \geq \lceil r \rceil$. Setting its use in optimization aside, we can use this idea as a refutation system for SAT instances. Consider a SAT instance $\varphi = \varphi_1 \wedge \cdots \wedge \varphi_m$, where $\varphi_1, \ldots, \varphi_m$ are clauses over the Boolean variables $x_1, \ldots, x_n$. The first step is to convert each clause into an inequality. We explain how to do this by example:

$$x \vee \bar{y} \vee z \Longrightarrow x + (1-y) + z \geq 1 \Longrightarrow x - y + z \geq 0.$$

We think of these translations as *axioms*, m in total. We also have the $2n$ Boolean axioms $x_i \geq 0$ and $-x_i \geq -1$. A *derivation* in cutting planes consists of a sequence of lines, in which each line is either an axiom or follows from earlier lines through the use of one of the following derivation rules:

Linear combination From $\sum_i a_i x_i \geq A$ and $\sum_i b_i x_i \geq B$, deduce $\sum_i (\alpha a_i + \beta b_i) x_i \geq \alpha A + \beta B$, where $\alpha, \beta \geq 0$ are integers.

Rounding From $\sum_i d a_i x_i \geq A$ (where $d \geq 0$ and $a_1, \ldots, a_n$ are all integers), deduce $\sum_i a_i x_i \geq \lceil A/d \rceil$.

A *refutation* is a derivation ending with an inequality $0 \geq p$ for some positive integer p. Cutting planes is a complete refutation system: every unsatisfiable 3SAT formula can be refuted using cutting planes. But at what cost?

2.1 Example — Pigeonhole principle

As an example, here is a refutation of the pigeonhole principle. The pigeonhole principle states that $f \colon [n+1] \to [n]$ is an injection. We encode f as a set of variables x_{ij}, where $i \in [n+1]$ and $j \in [n]$. Our axioms are:

- For all i, $\sum_j x_{ij} \geq 1$.
- For all $i_1 \neq i_2, j$, $-x_{i_1 j} - x_{i_2 j} \geq -1$.

Fix some j. Adding $-x_{1j} - x_{2j} \geq -1$, $-x_{2j} - x_{3j} \geq -1$ and $-x_{-1j} - x_{3j} \geq -1$, we deduce

$$-2x_{1j} - 2x_{2j} - 2x_{3j} \geq -3 \Longrightarrow -x_{1j} - x_{2j} - x_{3j} \geq -1.$$

Similarly $-x_{2j} - x_{3j} - x_{4j} \geq -1$. Adding the previous two inequalities together with $-x_{1j} - x_{4j} \geq -1$, we deduce

$$-2x_{1j} - 2x_{2j} - 2x_{3j} - 2x_{4j} \geq -3 \Longrightarrow -x_{1j} - x_{2j} - x_{3j} - x_{4j} \geq -1.$$

Continuing this way, we deduce $-\sum_i x_{ij} \geq -1$. Summing this over all j, we deduce

$$-\sum_{ij} x_{ij} \geq -n.$$

On the other hand, if we sum all axioms of the first type then we obtain

$$\sum_{ij} x_{ij} \geq n + 1.$$

Summing the last two inequalities gives the desired contradiction $0 \geq 1$.

2.2 Clique vs. coloring

We chose to illustrate cutting planes using the pigeonhole principle, since this is a contradiction which is hard for many other proof system, including Resolution. A contradiction which does turn out to be difficult is the one known as *clique vs. coloring*, which states that a graph G has a clique of size k and a valid coloring with $k-1$ colors. One way to translate this to a SAT instance is using the variables y_{iv} (vertex v is the ith vertex of the clique) and z_{jv} (vertex v gets color j) and the following clauses:

- For all i: $\bigvee_v y_{iv}$ (some vertex is the ith vertex of the clique).
- For all $i \neq j, v$: $\neg y_{iv} \vee \neg y_{jv}$ (v is not both the ith vertex and the jth vertex).
- For all v: $\bigvee_j z_{jv}$ (vertex v gets some color).
- For all $i \neq j, k, u \neq v$: $\neg y_{iu} \vee \neg y_{jv} \vee \neg z_{ku} \vee \neg z_{kv}$ (two vertices getting the same color cannot belong to a clique).

3 Feasible interpolation

The only technique we know to prove lower bounds on cutting planes is *feasible interpolation*, due to Krajíček and to Bonet, Pitassi and Raz. The idea is that a refutation of the (say) clique-coloring principle can be turned into a *monotone* circuit which given a graph, distinguishes the case in which the graph has a k-clique and the case in which it is $(k-1)$-colorable. Alon and Boppana proved (unconditionally!) that monotone circuits accomplishing this task must have exponential size (for an appropriate value of k, say $\sqrt[3]{n}$), hence the same is true for cutting planes refutations of the clique-coloring principle!

How does feasible interpolation work? The idea is to consider a game between two players, the *clique* player and the *coloring* player. The clique player claims that the graph has a k-clique, and the coloring player claims that the graph is $(k-1)$-colorable. The game starts at the last line $0 \geq 1$ of the proof. At each round of the game, we go from the current line to one of the lines which implied it. Eventually we reach an axiom. If it is a clique-axiom, the coloring player wins. If it is a coloring-axiom, the clique player wins. If it is an axiom involving the edge (u,v), the clique player wins if the edge is in the graph.

Roughly speaking, the rules of the game are arranged so that for each setting of the variables, the line currently in play is false. This can always be ensures since if a line is false then one of the lines implying it was false. If the clique player chooses his variables to encode some k-clique appearing in the graph, then the game cannot end at an axiom involving some edge (u,v) not in the graph, since such an axiom is satisfies by his variables. Thus he wins the game. An analogous argument works for the other player. Thus the game can be used to decide whether a graph has a k-clique or is $(k-1)$-colorable.

All we need to do in order to convert this game into a monotone circuit is to replace the turns in which the clique player speaks with $\bigvee$ gates and the turns in which the coloring player speaks with $\bigwedge$ gates. At the leaves we have variables x_{uv}, and so the resulting circuit is monotone!

This is the argument of Beame, Pitassi and Raz, and it works as long as all lines in the proof have small (polynomial) weights. Curiously enough, this argument never uses the fact that the specific derivation rules listed above are the ones used. It would go through for *any* valid set of derivation rules! In particular, it gives a lower bound for *semantic cutting planes*, in which given every two inequalities ℓ_1, ℓ_2 we can derive *any* inequality ℓ semantically following from them, that is any inequality which is true for all 0/1-assignments satisfying ℓ_1, ℓ_2. This curious phenomenon, that lower bound proofs work even for semantic versions of the proof systems, occurs in several places in proof complexity.

The main problem with this lower bound is that it needs all weights to be small. Pudlák came up with a different proof which does not suffer from this restriction, but only works for the standard version of cutting planes. Is the semantic version stronger than the standard version? Can it refute any unsatisfiable 3SAT using polynomial size proofs? This doesn't imply that NP equals coNP, since it is difficult to verify the validity of a proof in semantic cutting planes proof. Indeed, consider a derivation of $0 \geq 1$ from the pair of inequalities $\sum_i w_i x_i \geq T$, $-\sum_i w_i x_i \geq -T$. This derivation is valid as long as the subset sum problem with weights $w_1, \ldots, w_n$ and target T has no solution. So verifying a semantic cutting planes proof is coNP-complete!

It turns out that Pudlák's proof can be modified to work for semantic cutting planes: Hrubeš showed that the clique-coloring principle is indeed hard for semantic cutting planes. This leaves one question open: Is semantic cutting planes stronger than standard cutting planes?

4 Separating semantic cutting planes from standard cutting planes

The *equational version* of the clique-coloring principle is obtained by replacing each inequality with an equation involving auxiliary variables. For example, we replace $-y_{iv} - y_{jv} \geq -1$, the translation of $\neg y_{iv} \vee \neg y_{jv}$, with the equation $-y_{iv} - y_{jv} + t_{ijv} = 0$. The inequality and the equation are completely equivalent: if only one of y_{iv}, y_{jv} is false, we set $t_{ijv} = 1$, and otherwise we set $t_{ijv} = 0$.

We actually think of this equation as two inequalities: $-y_{iv} - y_{jv} + t_{ijv} \geq 0$ and $y_{iv} + y_{jv} - t_{ijv} \leq 0$. We do this for all axioms of the clique-coloring principle to obtain the *equational clique-coloring principle.* How hard is this principle? It turns out that in terms of the standard cutting planes proof system, this principle is as hard as the original one: any proof of the new principle can be turned into proof of the original principle by using the substitution $t_{ijv} = y_{iv} + y_{jv}$.

What about semantic cutting planes? Here the situation is radically different: this principle is easy for semantic cutting planes! Why is that? Let $e_1 = 0, \ldots, e_m = 0$ be the various equations in the new principle. For a large enough value of M, these equations are equivalent to the single equations $\sum_{t=1}^{m} M^{t-1} e_t = 0$, and in fact we can derive this equation from $e_1 = 0, \ldots, e_m = 0$ already in standard cutting planes, by deriving the two constituent inequalities separately. At this point, we get a pair of inequalities which have no Boolean solution, so we can immediately conclude that $0 \geq 1$ in semantic cutting planes. However, the same conclusion takes standard cutting planes an exponential number of steps, due to the lower bound. We have reached our goal!