Information Processing Letters 136 (2018) 70–75

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

On semantic cutting planes with very small coefficients

Massimo Lauria [a,*], Neil Thapen [b]

[a] Dipartimento di Scienze Statistiche — Sapienza Università di Roma, Italy
[b] Institute of Mathematics — Czech Academy of Sciences, Czech Republic

ARTICLE INFO

Article history:
Received 24 December 2017
Received in revised form 12 March 2018
Accepted 10 April 2018
Available online 12 April 2018
Communicated by Marcin Pilipczuk

Keywords:
Theory of computation
Proof complexity
Cutting planes

ABSTRACT

Cutting planes proofs for integer programs can naturally be defined both in a syntactic and in a semantic fashion. Filmus et al. (STACS 2016) proved that semantic cutting planes proofs may be exponentially stronger than syntactic ones, even if they use the semantic rule only once. We show that when semantic cutting planes proofs are restricted to have coefficients bounded by a function growing slowly enough, syntactic cutting planes can simulate them efficiently. Furthermore if we strengthen the restriction to a constant bound, then the simulating syntactic proof even has polynomially small coefficients.

1. Introduction

The field of *proof complexity* studies the length of proofs for propositional unsatisfiability, also called refutations. The historical motivation was the P vs NP problem. If there are unsatisfiable formulas without short refutations, then it must be that NP is different from co-NP, and therefore that P is different from NP [11]. In this context a proof must be efficiently verifiable and therefore written in some clear format, in some specific *proof system*. If this format is simple enough, we can sometimes show strong lower bounds on the length of such proofs. As in circuit complexity, proving lower bounds is hard even for some apparently simple proof systems.

There are other good reasons to study proof systems. Algorithms which solve unsatisfiability implicitly produce refutations in a relatively simple proof system. See for example the well known connection between DPLL algorithms, decision trees and treelike resolution proofs [14,13,4,5]. Another classic example, more relevant for this paper, is the use of Gomory cuts to solve integer programs [22]. Algorithms that mix branch and bound techniques, linear programming and Gomory cuts can often be formalized as proofs in the *cutting planes* proof system [9,10].

Despite the importance of the system, the only method we know to lower-bound the length of cutting planes proofs is *interpolation* [24], which was used to prove the first lower bounds [26]. Recently a variant of this method has been applied to random k-CNFs, with $k = \omega(1)$, as well [20,23].

Most systems studied in proof complexity, including cutting planes, are actually inference systems. A proof is developed line by line, and each line is either an axiom of the system or is derived from some previous lines according to a specific inference rule. Nevertheless it turns out that the specifics of the inference rules are not important for many results in the area, and the main factor in the power of the proof system is the expressivity of the proof lines. Thus it makes sense to study both *syntactic* and *se-*

* Corresponding author.
E-mail addresses: massimo.lauria@uniroma1.it (M. Lauria), thapen@math.cas.cz (N. Thapen).

https://doi.org/10.1016/j.ipl.2018.04.007
0020-0190/

mantic proofs. In the former a specific set of inference rules are available to derive a new proof line from lines derived before. In the latter a new line can be an arbitrary logical consequence of a constant number of previously derived lines.[1]

A similar (but more powerful) form of semantic proofs naturally occur in the study of proof space [15,1]. In this framework a proof is seen as a sequence of memory configurations, each consisting of a set of proof lines, and each configuration is semantically implied by the previous one. This approach can be used to study the memory usage of proof verification algorithms. Most successful lower-bound techniques related to proof space, either based on connection to other complexity measures [2,18], to pebbling games [25,21,8], or to matching games [19,3,6,17], work against this type of semantic proofs. Only limited results are known for the space complexity of CP proofs, though (see [27]).

If we study proof length, the appropriate semantic version of cutting planes is the one that infers any new inequality ℓ which logically follows (over $\{0,1\}^n$) from two previously derived inequalities. Observe that this is not a proof system in the technical sense, because there is no known efficient algorithm to verify whether an inference step is sound. Indeed, even to check whether the two linear inequalities $\sum_i a_i x_i \leq b$ and $\sum_i a_i x_i \geq b$ are simultaneously satisfiable over $x_i \in \{0,1\}$ is NP-complete if the coefficients have exponential magnitude with respect to the number of variables (it is the Subset Sum problem). The situation is different with small coefficients – see the discussion at the end of this note.

Semantic cutting planes seems to be a much stronger proof system than syntactic cutting planes, and indeed even allowing just one application of the semantic rule (together with the usual syntactic rules) gives an exponential advantage over purely syntactic cutting planes [16]. Still, the same paper shows that the formula that [26] proved to be hard for syntactic CP is hard for semantic CP as well. If semantic CP is stronger in general, is there any condition under which syntactic CP efficiently simulates semantic CP? In this paper we show that

Theorem 1 *(Informal). A semantic cutting planes proof in which all coefficients have very small size can be transformed into a syntactic cutting planes proof with at most a polynomial blowup in size. If the coefficients in the semantic proof are constant, the coefficients in the syntactic proof can be made polynomial.*

The idea of the proof is to realize that if the coefficients have small size, then the linear inequalities involved in the inference must have a lot of symmetries, hence the argument can be viewed as proving the soundness of an inference rule with a small number of variables. The main contribution of this paper is to show that this can be done in syntactic CP. Compare this result with the separation in [16]. They exhibit a short semantic CP refutation for a CNF which is hard for syntactic CP. Such a refutation uses exponential magnitude coefficients.

The paper is organized as follows. In Section 2 we give the necessary definitions and notation. In Section 3 we discuss implicational completeness of CP and prove some upper bounds. Finally in Section 4 we show our main result, namely that semantic proofs with very small coefficients can be simulated by syntactic proofs. We conclude the paper with some open problems.

2. Preliminaries

We consider *cutting planes (CP)* [9,12], a proof system based on manipulation of inequalities over variables $x_1, \dots, x_n$. Each line in the proof is an inequality of the form $\sum_i a_i x_i \geq b$ where $a_i, b \in \mathbb{Z}$. Variables $x_1, \dots, x_n$ are understood to take integer values.

A *syntactic CP derivation* of an inequality ℓ_τ from a set of inequalities $\mathcal{S}$ is denoted as $\mathcal{S} \vdash \ell_\tau$ and is a sequence of inequalities $(\ell_1, \dots, \ell_\tau)$ such that for $1 \leq i \leq \tau$ the inequality ℓ_i is either in $\mathcal{S}$ or is obtained by one of the following rules.

- **Sum:** We can add two earlier inequalities.
- **Multiplication:** We can multiply an inequality by a positive integer.
- **Division:** From an inequality $\sum_i a_i x_i \geq b$ we can derive

  $$\sum_i (a_i/c)x_i \geq \lceil b/c \rceil$$

  if c is a positive integer which divides all coefficients a_i.

When used as a propositional proof system a syntactic CP derivation may also include

- **Boolean axioms:** We can introduce inequalities $x_i \geq 0$ and $-x_i \geq -1$.

A *semantic CP derivation* of an inequality ℓ_τ from a set of inequalities $\mathcal{S}$ is a sequence of inequalities $(\ell_1, \dots, \ell_\tau)$ such that for $1 \leq i \leq \tau$ the inequality ℓ_i is either in $\mathcal{S}$ or follows semantically from two earlier inequalities ℓ_j and ℓ_k, in the sense that ℓ_i holds for every point in $\{0,1\}^n$ where ℓ_j and ℓ_k both hold. We will also consider semantic entailment over $\mathbb{Z}^n$ rather than $\{0,1\}^n$, but we do not need a formal definition of derivations of this kind.

A syntactic (resp. semantic) CP refutation of $\mathcal{S}$ is a syntactic (resp. semantic) CP derivation of $0 \geq 1$ from $\mathcal{S}$.

If we do not care to specify the coefficients, we abbreviate $\sum_i a_i x_i \geq b$ as $A\bar{x} \geq b$. For our convenience we sometimes write $A\bar{x} \leq b$ as an alias for $-A\bar{x} \geq -b$ and $A\bar{x} = b$ as a shorthand for the conjunction of the inequalities $A\bar{x} \geq b$ and $A\bar{x} \leq b$. The *length* of a CP derivation is the number of steps. The *magnitude* of a CP derivation is the maximum absolute value among the coefficients and constants in all its inequalities. The *size* of a CP derivation is the sum, over all inequalities, of the binary length of all coefficients and the constant of each inequality. Clearly the size is at most polynomial in the length times $\log_2$ of the magnitude.

[1] The limitation to a constant number of premises keeps the proof systems from being trivial.