Information Processing Letters 34 (1990) 81–85
North-Holland

16 March 1990

A FEASIBLY CONSTRUCTIVE LOWER BOUND FOR RESOLUTION PROOFS *

Stephen COOK and Toniann PITASSI

Department of Computer Science, University of Toronto, Sandford Fleming Bldg 2303 C, Toronto, Canada M5S1A4

Communicated by T. Lengauer
Received 2 May 1989
Revised 2 October 1989

Keywords: Constructive proof, resolution, lower bound, polynomial time

1. Introduction

Haken [4] first proved the intractability of resolution by showing that a family of propositional formulas encoding the pigeon-hole principle require superpolynomial-sized resolution proofs. Later, several authors [6,1,2] extended Haken's techniques to obtain exponential lower bounds on the size of resolution proofs. All of these results are based on the counting method used by Haken and later reformulated as a probabilistic method by Urquhart [6].

In this paper we describe a simpler and more direct method which can replace the counting and probabilistic methods. Our method is feasibly constructive in the sense discussed in [3]. Informally, this means that all concepts in the proof are polynomial time. We present a polynomial time greedy algorithm which, when presented with a candidate resolution proof that is too short, produces a mistake in the proof. Further, we present a feasibly constructive correctness proof for the algorithm. In contrast, the counting arguments show the existence of the mistake indirectly by bounding the sizes of exponentially large sets.

In [3] it is shown (Theorem 10.16) that an "extended Frege system" (a kind of propositional proof system stronger than resolution) cannot feasibly constructively be proved not polynomially bounded, in the precise sense that a certain formula $\neg$PB(EF) is not a theorem of the system IPV$^\omega$. In contrast, the present paper shows in outline that the formula $\neg$PB(RES) is indeed a theorem of IPV$^\omega$, where $\neg$PB(RES) is a formula similar to $\neg$PB(EF), but stating that resolution (instead of an extended Frege system) is not polynomially bounded. This shows that proving intractability of extended Frege systems requires new techniques.

In this paper, we first review how the counting and probabilistic methods are used to obtain superpolynomial lower bounds on resolution proofs. Second, we present a similar lower bound using the greedy method. Third, by extending the analysis of this new method to weaker forms of the pigeon-hole principle formulas, we obtain lower bounds similar to those obtained by Buss and Turán [1]. Finally, we briefly discuss how the greedy method can be extended and shown to apply to the results of Urquhart [6] and Chvátal and Szemerédi [2].

2. Background

The negation of the pigeon-hole principle can be encoded by a family of unsatisfiable propositional formulas $\{\mathrm{PHP}_n \mid n \in \mathbb{N}\}$ in conjunctive normal form. PHP_n has $n(n+1)$ variables $\{x_{ij} \mid 1 \leqslant i$

* Research supported by the Natural Sciences and Engineering Research Council of Canada.

$\leqslant n,\ 1 \leqslant j \leqslant n+1\}$ where variable x_{ij} represents the condition that pigeon j is sitting in hole i. PHP_n is defined to be

$$\left(\bigwedge_{j=1}^{n+1} \bigvee_{i=1}^{n} x_{ij}\right) \wedge \left(\bigwedge_{i=1}^{n} \bigwedge_{1 \leqslant j_1 < j_2 \leqslant n+1} (\bar{x}_{ij_1} \vee \bar{x}_{ij_2})\right). \quad (1)$$

Note that the size of the above formula is $O(n^3)$. By completeness of resolution, there must be a resolution refutation (henceforth called a resolution proof) of PHP_n. A resolution proof is a rooted, directed acyclic graph where the root vertex represents the empty clause, leaf vertices represent input clauses, and all other clauses are represented by vertices of indegree two. The size of a resolution proof is defined to be equal to the number of clauses (vertices) in the proof tree as a function of n.

We visualize truth assignments and clauses relating to PHP_n by an $n \times (n+1)$ matrix. A *clause* is represented by a matrix of $\oplus$'s, $\ominus$'s and blanks where a $\ominus$ in a matrix position x_{ij} corresponds to $\bar{x}_{ij}$; a $\oplus$ in position x_{ij} corresponds to x_{ij}; and a blank in x_{ij} corresponds to no appearance of x_{ij} in the clause. A *truth assignment* is represented by the same matrix filled with 0's and 1's; a *partial truth assignment* is represented by a matrix filled with 0's, 1's and blanks. (Here 1 represents *true* and 0 represents *false*.)

Assume for simplicity that n is divisible by 4. Let PTA be the set of those partial truth assignments with exactly $\frac{1}{4}n$ 1's, no two of which are in the same row or column; each row or column with a 1 has all remaining positions filled with 0's, and all positions are blank which are not in the same row or column with a 1. Let a *critical truth assignment* be a total truth assignment with exactly n 1's, no two in the same row or column. Thus each pta in PTA can be extended to a critical truth assignment by selecting any positions for the remaining $\frac{3}{4}n$ 1's, provided the new 1's do not conflict with each other or with the 1's in pta.

Let P be a resolution proof of PHP_n. A clause in P is said to be *complex* if (1) every column has at most one $\ominus$ and (2) it has at least $\frac{1}{4}n+1$ "good columns": columns containing exactly one $\ominus$ or at least $\frac{1}{2}n$ $\oplus$'s. We say that a partial truth assignment pta *verifies* a complex clause cc iff either some 1 in pta is in the same position as a $\oplus$ in cc or some 0 in pta is in the same position as a $\ominus$ in cc. Thus pta verifies cc iff every total extension of pta makes cc true.

Haken defined a map γ from PTA to the complex clauses in P which satisfies the following condition:

No pta in PTA verifies its image $\gamma(\text{pta})$. (2)

(In fact, Haken showed the stronger condition that some critical truth assignment extending pta makes γ(pta) false, but (2) suffices for our purposes.)

Haken proved a lower bound on the number of complex clauses in the proof P by deriving a lower bound l on the cardinality of PTA, and an upper bound u on the number of elements pta in PTA which fail to verify any fixed complex clause cc. By (2), the proof P must contain at least l/u complex clauses. Haken concluded that any resolution proof for PHP_n must have size at least c^n, with $c = 1.49^{0.01}$ for $n > 200$.

Urquhart [6] showed how to give the argument a probabilistic cast. If p is an upper bound on the probability that a random element pta in PTA fails to verify any given complex clause cc, then by (2) the proof P must have at least $1/p$ complex clauses. To obtain the estimate p, we may assume as the worst case that the given complex clause has exactly $\frac{1}{4}n+1$ good columns each containing exactly $\frac{1}{2}n$ $\oplus$'s and no $\ominus$. A random member of PTA can be selected by first randomly selecting $\frac{1}{4}n$ columns and then, for each column, selecting a distinct row. Let X be the number of selected columns that are good columns of our complex clause. Then X has a hypergeometric distribution with mean approximately $\frac{1}{16}n$, and known bounds on the tails of the hypergeometric distribution show that $\Pr[X \leqslant \frac{1}{32}n]$ decays exponentially in n. On the other hand, a simple calculation shows that the conditional probability that a random member of PTA fails to verify the complex clause given $X > \frac{1}{32}n$ also decays exponentially. This two facts combine to give an exponentially small upper bound p.

3. The greedy method

Neither Haken's nor Urquhart's argument is feasible as it stands, since both involve estimating cardinalities of sets of exponential size. Our feasible proof proceeds as follows. Suppose $B < c^n$ for suitable $c > 1$. We will give a greedy (see for example [5]) algorithm G which does the following:

Given a candidate proof $\hat{P}$ for PHP_n with at most B complex clauses, return pta* in PTA which verifies every complex clause in $\hat{P}$. (3)

The algorithm G runs in time polynomial in n and the length of $\hat{P}$, and has a feasible correctness proof. Further the map γ is computable in polynomial time in the length of $\hat{P}$, and the property (2) has a feasible proof, under the assumption that $\hat{P}$ is a resolution proof. Since pta* verifies its image $\gamma(\text{pta}^*)$, (2) is violated and the computation of $\gamma(\text{pta}^*)$ leads to a point in $\hat{P}$ which violates the definition of a resolution of PHP_n.

We note that the existence of pta* satisfying (3) can be inferred from both Haken's counting argument and Urquhart's probabilistic argument, but neither of these yields a polynomial time algorithm for finding pta*.

It remains to describe and analyze the greedy algorithm G. First we put the complex clauses in a standard form.

For each complex clause cc we construct a complex clause cc^+ as follows. First, select the first $\frac{1}{4}n+1$ good columns of cc and delete all entries in all other columns. Next, for each column containing a $\ominus$, delete the $\ominus$ in that column and place a $\oplus$ in all positions of that column except the one that contained the deleted $\ominus$. Now each column is either empty or contains no $\ominus$'s and at least $\frac{1}{2}n$ $\oplus$'s. The final step is to delete all but the first $\frac{1}{2}n$ $\oplus$'s from each of these nonempty columns. The resulting clause cc^+ is a complex clause with exactly $\frac{1}{4}n+1$ nonempty columns each with exactly $\frac{1}{2}n$ $\oplus$'s and no $\ominus$'s. The following fact is easy to check:

For each pta in PTA and each complex clause cc, if pta verifies cc^+ then pta verifies cc. (4)

The input to the algorithm G is the set S_0 of all cc^+ such that cc is a complex clause in the candidate proof $\hat{P}$. Assuming $|S_0| \leqslant B$ (B is specified below), the output is a partial truth assignment pta* which verifies each clause in S_0 (and hence each complex clause in $\hat{P}$). The algorithm proceeds by successively choosing 1's in pta* so as to verify as many remaining members of S_0 as possible. After i steps, $S_i \subseteq S_0$ is the set of clauses left to be satisfied and E_i is the set of all blank positions in pta*.

Greedy algorithm G

(Input is a set S_0 of complex clauses cc^+.)

pta* is initially the empty partial truth assignment (all blanks).

E_0 is the set of all positions (in the truth assignment matrix).

for $i = 1$ **to** $\frac{1}{4}n$

Find a position p in E_{i-1} which maximizes the number of clauses in S_{i-1} with a $\oplus$ in position p.

Augment pta* by placing a 1 in position p and 0's elsewhere in the row and column of p.

(*) S_i is S_{i-1} with all clauses verified by (the augmented) pta* deleted.

E_i is E_{i-1} with all positions in the row and column of p deleted.

end for

The algorithm clearly runs in time bounded by a polynomial in $|S_0|$ and n, and by line (*) the final value of pta* clearly verifies all members of $S_0 - S_{n/4}$. Hence in view of (4), it suffices to prove that $S_{n/4}$ is empty in order to verify (3).

If s_0 is the number of positions of any cc^+ which have $\oplus$, then the average over all positions p of E_0 of the number of clauses in S_0 with a $\oplus$ in position p is

$$a_0 = \frac{|S_0| s_0}{|E_0|} = \frac{|S_0| \frac{1}{2}n\left(\frac{1}{4}n+1\right)}{n(n+1)}.$$

Since some position p hits at least a_0 clauses, $|S_1| \leqslant |S_0| - a_0$.

In general, if we prune each clause in S_i by deleting all $\oplus$'s not in E_i, then the number of $\oplus$'s deleted is at most $\frac{1}{2}n + (\frac{1}{4}n + 1) - 1 = \frac{3}{4}n$. Thus each pruned clause has at least s_i $\oplus$'s remaining, where

$$s_i = s_0 - i\tfrac{3}{4}n = \tfrac{1}{2}n\left(\tfrac{1}{4}n + 1\right) - \tfrac{3}{4}in.$$

The average over positions p of E_i of the number of clauses in S_i with a $\oplus$ in position p is

$$a_i \geqslant \frac{|S_i|s_i}{|E_i|} \geqslant \frac{|S_i|s_i}{|E_0|}.$$

For any ϵ, $0 < \epsilon < 1$, if we run the algorithm for $t = \lfloor \frac{1}{6}\epsilon n \rfloor$ steps (so $i \leqslant t$) we have

$$\begin{aligned} a_{i-1} &\geqslant \frac{|S_{i-1}|s_{t-1}}{|E_0|} \\ &\geqslant \frac{\frac{n}{2}\left(\frac{n}{4}+1\right) - \frac{3n\epsilon n}{4\cdot 6}}{n(n+1)}|S_{i-1}| \\ &= \frac{\frac{n}{8}(1-\epsilon)+\frac{1}{2}}{n+1}|S_{i-1}| \\ &> \frac{1-\epsilon}{8}|S_{i-1}| \quad \text{(assuming } S_{i-1} \neq 0\text{)}. \end{aligned}$$

Thus

$$|S_i| \leqslant |S_{i-1}| - a_{i-1} < \frac{7+\epsilon}{8}|S_{i-1}|$$

so

$$|S_{n/4}| \leqslant |S_t| < \left(\frac{7+\epsilon}{8}\right)^{\epsilon n/6}|S_0|.$$

Setting $B = (8/(7+\epsilon))^{\epsilon n/6}$, if $|S_0| \leqslant B$ then $|S_{n/4}| < 1$ so $S_{n/4} = 0$, as required.

We note that if $\epsilon = \frac{1}{2}$, then our bound B is slightly better than Haken's bound.

4. Generalization to weaker forms of PHP_n

A more general form of the pigeon-hole principle, PHP_n^m, states that there cannot be a one-to-one mapping from m pigeons to n holes, for $m \geqslant n+1$. The formulation of this principle as a family of propositional formulas is identical to equation (1), but with m replacing $n+1$. As m increases, the statement becomes weaker and therefore may have shorter resolution proofs. By extending Haken's argument, Buss and Turán [1] showed that if $m = o(n^2/\log n)$, then the technique still yields superpolynomial lower bounds. More precisely, they proved that every resolution proof of PHP_n^m has length at least

$$\tfrac{1}{2}\cdot\left(\tfrac{3}{2}\right)^{(1/50)\cdot(n^2/m)}.$$

We will show a similar result using the greedy method.

Let P be a resolution proof of PHP_n^m. The definition of the set PTA of partial truth assignments and the definition of complex clause are exactly the same as before. Buss and Turán show the existence of a mapping γ from PTA to the complex clause in P satisfying the condition (2), just as in the previous case. We can apply the same greedy algorithm to a candidate proof $\hat{P}$ and analyze it as follows. The only difference is that now $|E_0|$ is nm instead of $n(n+1)$, which leads to the estimate

$$a_{i-1} \geqslant \frac{1-\epsilon}{8}\cdot\frac{n}{m}|S_{i-1}| = x|S_{i-1}|$$

where $x = (1-\epsilon)/8\cdot(n/m)$. Thus

$$\begin{aligned} |S_{n/4}| &\leqslant (1-x)^{\epsilon n/6}|S_0| = (1-x)^{(1/x)\cdot(\epsilon n x/6)}|S_0| \\ &< \left(\frac{1}{e}\right)^{\frac{\epsilon n x}{6}}|S_0| \\ &= e^{-(\epsilon(1-\epsilon)n^2/48m)}|S_0|. \end{aligned}$$

Setting $\epsilon = \frac{1}{2}$ we obtain a bound

$$B = e^{(1/192)\cdot(n^2/m)} > \left(\frac{3}{2}\right)^{(1/78)\cdot(n^2/m)},$$

a little worse than the Buss-Turán bound.

5. Other applications

Urquhart [6] obtained an exponential lower bound on the size of resolution proofs for a family of graph-based clauses. A straightforward application of the greedy method can also yield this

result, and the resulting proof can be made feasibly constructive, assuming that the necessary expander graphs can be constructed and verified in polynomial time. Chvátal and Szemerédi [2] obtain an exponential lower bound with probability approaching 1 as n approaches infinity for certain sets of randomly chosen clauses. We must generalize the greedy algorithm to apply our method to obtain this lower bound. (In this case the whole proof is still not feasibly constructive because even the statement involves probabilities.)

For the pigeon-hole formulas, we exploit the uniform structure of each pta $\in$ PTA to select a covering set which is also a valid pta. At each step, an element e^* is chosen that covers a constant fraction of the remaining sets according to the selection strategy. All elements in the same row or column as e^* are then eliminated to ensure that we obtain a valid partial truth assignment.

The set PTA defined by Chvátal and Szemerédi is not structurally defined but instead is based on a combinatorial property. We therefore need a new mechanism to obtain a valid partial truth assignment. In brief, we use a double averaging argument where the selection strategy maintains two conditions at each step:

(1) As before, the chosen element must cover at least a constant fraction of the remaining sets.

(2) The second condition concerns the size of the valid PTA set at each step. At step i, the covering set pta* contains i elements. The valid partial truth assignments in PTA at step i, PTA_i, are those which contain pta*. We must ensure that the size of PTA_i remains large, i.e., that it is at least a constant fraction of the size of PTA_{i-1}.

It can be shown that each of the two conditions is satisfied on average; they can then be combined to guarantee the existence of an element satisfying both conditions at each step.

References

[1] S. Buss and G. Turán, Resolution proofs of generalized pigeonhole principles, *Theoret. Comput. Sci.* **62** (3) (1988) 311–317.

[2] V. Chvátal and E. Szemerédi, Many hard examples for resolution, *J. ACM* **35** (4) (1988) 759–768.

[3] S. Cook and A. Urquhart, Functional interpretations of feasibly constructive arithmetic, Tech. Rep. 210/88, University of Toronto, Toronto (1988); also Extended Abstract, in: *Proc. 21st ACM Symposium on Theory of Computing* (1989) 107–112.

[4] A. Haken, The intractability of resolution, *Theoret. Comput. Sci.* **39** (1985) 297–308.

[5] C. Papadimitriou and K. Steiglitz, *Combinatorial Optimization* (Prentice-Hall, Englewood Cliffs, NJ, 1982).

[6] A. Urquhart, Hard examples for resolution, *J. ACM* **34** (1) (1987) 209–219.