Study of the Probabilistic Polynomial Method in Circuit Complexity

Sahil Grover Nirbhay Modhe

Department of Computer Science and Engineering
Indian Institute of Technology Kanpur

Advised by Prof. Raghunath Tewari

Abstract

The polynomial method in circuit complexity is a way of obtaining better lower bounds for running time of algorithms by exploiting the efficient translation of low-depth circuits to polynomials. Ryan Williams first proposed this method to solve the All Pairs Shortest Path (APSP) problem in sub-cubic time using a randomized algorithm, and subsequently used the same technique for solving Boolean Orthogonal Detection in sub-quadratic time. We present an analysis of the probabilistic polynomial construction, and how it led to better lower bounds in the case of the aforementioned problems. We then try to approach the problem of fast min-plus multiplication of two matrices given their tensor product decomposition.

1 Introduction

The probabilistic polynomial method is a general framework for using circuit complexity to solve a certain class problems. It was first introduced by [Wil14] for the obtaining a subcubic time algorithm for the All Pairs Shortest Path problem, using a randomized algorithm. He then generalized this to identify a class of problems [CW16] for which a faster algorithm may be obtained.

The problems of this class have the property that their time complexity is constrained by a subroutine that is executed several times on different inputs. If this subroutine can be modelled by a low-complexity circuit, then the circuit can be randomly converted to a sparse polynomial. This sparse polynomial's algebraic properties are such that its repeated execution can be sped up.

In this project, we present our analysis of the tools used to obtain a faster algorithm using the above methodology. We will focus on the tools used to prove the lower bounds for two problems which R. Williams tackled - All Pairs Shortest Path and Boolean Orthogonal Detection, in sections 2 and 3 respectively.

2 Faster All Pairs Shortest Path

Since the 1970s,[Mun71] it has been known that the search for faster algorithms for APSP is equivalent to the search for faster algorithms for the min-plus (or max-plus) matrix product defined as:

$$(A * B)[i, j] = min_k(A[i, k] + B[k, j])$$

Time	Author(s)	Year(s)
n^3	Floyd/Warshall	1962
$n^3/\log^{1/3} n$	Fredman	1975
$n^3/\log n$	Zwick/Takaoka/Chan	2004/2005/2005
$n^3/\log^2 n$	Chan/Han-Takaoka	2007/2012
$n^3/2^{\Omega(\log n)^{1/2}}$	Williams	2014/2016*

*Deterministic time bound was proven.

Table 1: Running times for general APSP (Table by [Wil14])

A $T(n)$-time algorithm exists for this product if and only if there is an $O(T(n))$-time algorithm for APSP.

To improve the min-plus matrix product, Ryan Williams used block division. Let A be an $n \times d$ matrix with entries from $W := ([0, M] \cap Z) \cup \infty$, and let B be an $d \times n$ matrix with entries from W . So, we wish to compute

$$C[i, j] = min_{k=1}^{d}(A[i][k] + B[k][j])$$

We transform the entries such that after the transformation, there is a unique k, say k^* that is achieving the minimum $A[i][k] + B[k][j]$.

$$A[i, j] \leftarrow A[i, j] * (n + 1) + j$$

$$B[i, j] \leftarrow B[i, j] * (n + 1)$$

i.e. $C[i][j] = min_k(A[i, k] + B[k, j])(n + 1) + k^*$

2.1 Fredman's Trick

$$A[i, k] - A[i, k'] \leq B[k', j] - B[k, j]$$

$$\Leftrightarrow$$

$$A[i, k] + B[k, j] \leq A[i, k'] + B[k', j]$$

R. Williams used Fredman's Trick[Fre76] and contructs two matrices A' and B' which are $n \times d^2$ and $d^2 \times n$. The columns of A' and rows of B' are indexed by pairs (k, k') from $[d]^2$.

$$A'[i, (k, k')] := A[i, k] - A[i, k'] \; and \; B'[(k, k'), j] := B[k', j] - B[k, j]$$

Construct a set,

$$S_{(k,k')} = \{A'[i, (k, k')], B'[(k, k'), i] \mid i = 1, ..., n\}$$

Replace the entries in A' and B' and replace them with their ranks in $S_{(k,k')}$, giving priority to entries of A' for breaking ties. Let these new matrices be A'' and B''
This replacement takes $\widehat{O}(nd^2 logM)$ time on a word RAM model.

By this reduction all the entries of the new matrices have only $O(\log n)$ bits instead of $O(\log M)$ bits.

2.2 Polynomial Construction

Let,

$$P(i, j, l) = \bigvee_{k=1,...,d}^{l^{th} \; bit \; of \; k \; is \; 1} \bigwedge_{k' \in \{1,...,d\}} \langle A''[i, (k, k')] \leq B''[(k, k'), j] \rangle$$

$P(i, j, l)$ equals the l^{th} bit of the smallest k^* such that

$$min_k A[i, k] + B[k, j] = A[i, k^*] + B[k^*, j]$$

Since, only one input of output or is true, we can replace outer or with xor gate. Xor gates are computationally less expensive as they account for addition whereas 'or' and 'and' gates accounts for multiplication.

$$P(i, j, l) = \bigoplus_{k=1,...,d}^{l^{th} \; bit \; of \; k \; is \; 1} \bigwedge_{k' \in \{1,...,d\}} \langle A''[i, (k, k')] \leq B''[(k, k'), j] \rangle$$

Also,

$$LEQ(a,b) = \left(\bigwedge_{i=1}^{t} (1 + a_i + b_i) \oplus \bigoplus_{i=1}^{t} \left((1 + a_i) \wedge b_i \wedge \bigwedge_{j=1}^{i-1} (1 + a_j + b_j) \right) \right)$$

where $t = 2 + \log n$. This circuit outputs one when $a \leq b$

2.3 Razborov-Smolensky Polynomial

Razborov [Raz87] and Smolensky[Smo87] proposed the following construction to reduce the number of 'and' gates:

$$E(y_1, \ldots, y_d) = \bigwedge_{i=1}^{e} \left(1 + \bigoplus_{j=1}^{d} r_{i,j} \cdot (y_j + 1) \right)$$

where $r_{i,j}$ are ed random bits.
Claim: *For every fixed* $(y_1, \ldots, y_d) \in \{0,1\}^d$,

$$Pr_{r_{i,j}}[E(y_1, \ldots, y_d) = y_1 \wedge \cdots \wedge y_d] \geq 1 - 1/2^e$$

Proof: If all the y_i are oness. Then $E(y_1, \ldots, y_d)$ is one as y_i+1 is zero modulo two. If $y_1 \wedge \cdots \wedge y_d = 0$, then there is a subset S of y_j's which are 0, and hence a subset S of $(y_j + 1)$'s that are 1. . The probability we choose $r_{i,j} = 1$ for an odd number of the y_j's in S is at exactly 1/2. So, the probability that the circuit outputs one is the product of $1/2$ e times(due to the outer and). Hence the claim is proved.

$$P'(i,j,l) = \bigoplus E([A''[i,(k,1)] \leq B''[(k,1),j]], \ldots [A''[i,(k,k')] \leq B''[(k,k'),j]]$$

Set $e = 2 + \log d$, so that E fails on a point y with probability at most $1/(4d)$. By the union bound, the probability that the (randomly generated) expression P' differs from P on a given row $A''[i,:]$ and column $B''[:,j]$ is at most 1/4.

R. Williams also used Razborov-Smolensky construction on the LEQ circuit which reduces to

$$\bigoplus_{t+1} \left[\bigwedge_{e'} \left[\bigoplus_{\leq t} [2 \oplus \ gates] \right] \right]$$

Also, each term of the form $\oplus_{\leq t}[2 \oplus gates]$ can be viewed an XOR of three quantities: an XOR of a subset of $O(\log n)$ variables a_i (from the matrix A''), another XOR of a subset of $O(\log n)$ variables b_j (from the matrix B''), and a constant (0 or 1). This preprocessing requires $\widehat{O}(nd^2(t+1)e')$ time and the new circuit becomes:

$$\bigoplus_{t+1} \left[\bigwedge_{e'} [2 \oplus gates] \right]$$

Applying the distributive law of F_2 we get LEQ' as a polynomial of degree e' and at most $m = (t+1)3^{e'}$ monomials. By the union bound, since the original circuit for $LEQ(a,b)$ contains only $t+1$ AND gates, and the probability of error of E' is at most $1/2^{e'}$, we have that for a fixed pair of strings (a,b), $LEQ(a,b) = LEQ'(a,b)$ with probability at least $1 - (t+1)/2^{e'}$. Set $e' = 3 + 2\log d + \log t$. This ensures that the the d^2 copies of LEQ' gives the same output as LEQ in P' with probability of at least 3/4.

Let,

$$m' = (t+1)3^{3+2\log d + \log t}$$

Plugging polynomial for LEQ', say Q, we get $P''(i,j,l)$ which now has the form:
An XOR of $\leq$ d fan-in,
ANDs of $1 + \log d$ fan-in,
XORs of $\leq d+1$ fan-in,
XORs of $\leq m'$ fan-in,
ANDs of e' variables.
Applying distributive property again, we get number of monomials as:

$$d((d+1)m')^{1+\log d} \leq d((d+1)(t+1)3^{3+2\log d+\log t})^{1+\log d}$$

2.4 Coppersmith Lemma

Theorem 1 *[Cop82] For all sufficiently large N, multiplication of an $N \times N^{.172}$ matrix with an $N^{.172} \times N$ matrix can be done in $O(N^2 \log^2 N)$ arithmetic operations.*

R. Williams extended Coppersmith Lemma and uses that to evaluate his polynomials

Theorem 2 *[Wil14] Let p be a $2k$-variate polynomial over the integers (in its monomial representation) with $m \leq n^{0.1}$ monomials, along with $A, B \subseteq \{0,1\}^k$ such that $|A| = |B| = n$. The polynomial $p(a_1, ..., a_k, b_1, ..., b_k)$ can be evaluated over all points $(a_1, ..., a_k, b_1, ..., b_k) \in A \times B$ in $n^2 poly(\log n)$ arithmetic operations.*

To use the above theorem, we need $\#monomials \leq n^{0.1}$ .Optimal d satisfying the inequality (for sufficiently small δ) comes out to be :

$$d = 2^{\delta(logn)^{1/2}}$$

2.5 Improving Accuracy

$$Pr\big[D_l[i,j] = P(i,j,l)\big] = Pr\Big[D_l[i,j] \text{ is the lth bit of the smallest } k^* \text{ such that } A[i,k^*] + B[k^*,j] = min_k(A[i,k] + B[k,j])\Big] \geq 3/4$$

To improve accuracy, R.Williams used majority amplification trick. For every $l = 1, ..., \log d$, he chose $c \log n$ independent random polynomials $Q'(i,j,l)$ according to the above process and compute:

$$C_l[i,j] = MAJ(D_{l,1}[i,j], ..., D_{l,c\log n}[i,j]$$

where $(i,j) \in [n]^2, l \in [\log d],$ *and* $k = 1, ..., c \log n$

So, we have

$$Pr[D_{l,k}[i,j] = P(i,j,l)] \geq 3/4$$

Consider a random variable $X := \sum_{k=1}^{c\log n}[D_{l,k}[i,j] = P(i,j,l)]$, so the expectation of X is

$$E[X] \geq (3c \log n)/4$$

. In order for the event $MAJ(D_{l,1}[i,j], ..., D_{l,c\log n}[i,j]) \neq P(i,j,l)$ to happen, we must have that $X < (c \log n)/2$

Using

$$Pr[Y < (1-\varepsilon)E[Y]] \leq e^{-\varepsilon^2 E[Y]/2}$$

We get ,

$$Pr[C_l(i,j) \neq P(i,j,l)] \leq e^{-4E[X]/18}$$

Choosing $c = 18.$ and by a union bound over all pairs $(i,j) \in [n]^2$ and $l \in [\log d]$,

$$Pr[\textit{there are } i,j,l, C_l \neq P(i,j,l)] \leq (n^2 log d) e^{-4\log n} \leq (\log d)/n^2$$

Therefore for $d = 2^{\delta(logn)^{1/2}}$, the algorithm outputs the min-plus product of an $n \times d$ and $d \times n$ matrix with probability at least $1 - (\log n)/n^2$ and complexity of order $n^3/2^{\Omega(\log n)^{1/2}}$ on real RAM model.

3 Boolean Orthogonal Detection

The Boolean Orthogonal Detection problem is to detect for two sets $A, B \subseteq \{0,1\}^d$ of size n if there is an $x \in A$ and $y \in B$ such that $\langle u, v \rangle = 0$. Theorem 3 was given Williams for the time complexity of this problem. Note that $c : \mathbb{N} \to \mathbb{N}$ such that $c(n) < n^{\varepsilon}$.

Theorem 3 *[AWY15] For vectors of dimension $d = c(n) log\ n$,* Boolean Orthogonal Detection *can be solved in $n^{2-1/O(\log c(n))}$ time by a randomized algorithm that is correct with high probability.*

Although the randomized algorithm proposed for Theorem 3 was later derandomized by [CW16], we will focus on the circuit construction, translation and how the Razborov-Smolensky polynomial was used in the algorithm.

3.1 Circuit Construction

The low-complexity circuit to be constructed will take two sets A, B and output whether there is an orthogonal pair of vectors for this set. However, the sizes of the sets A, B must be constrained in order to obtain a small circuit. Hence, the sets $A, B \subseteq \{0,1\}^d$ of size n are first partitioned into $\lceil n/s \rceil$ groups of size at most s. A circuit C is constructed which takes a group A' from A and group B' from B and outputs 1 if and only if there is an orthogonal pair of vectors in (A', B'). Consider two vectors x_i, y_j from a group pair A', B':

$$E(x_i, x_j) = \bigwedge_{k=1}^{d} (\neg x_i[k] \vee \neg y_j[k])$$

Here, $E(x_i, x_j) = 1$ if and only if $\langle x_i, x_j \rangle = 0$. Now, the circuit C is effectively

$$\bigvee_{All\ s^2\ pairs} E(x_i, x_j)$$

$$\bigvee_{s^2} \bigwedge_{d} (\neg x_i[k] \vee \neg y_j[k])$$

Now, the steps that follow will be similar to the circuit translation shown in APSP earlier i.e. the number of AND gates will be reduced in order to create a short polynomial. Using the Razborov-Smolensky construction for randomized reduction of number of AND gates, we obtain

$$\bigvee_{s^2} \bigwedge_{d} (\neg x_i[k] \vee \neg y_j[k])$$

$$\bigwedge_{2} \bigoplus_{s^2+1} \bigwedge_{3\log s} \bigoplus_{d+1} (1 + a \cdot b)$$

This probability that this circuit will give the correct answer is at least 2/3 (the derivation of this probability has been skipped). Now, to increase the probability of correctness, $O(\log n)$ polynomials are sampled from a distribution of polynomials, and a majority is taken over all pairs of groups. The Coppersmith lemma[Wil14][Cop82] is used to evaluate the polynomial on all pairs of s-sized groups of sets A and B. The final probability of correctness for these two sets will be at least $1 - 1/n$. However, it is interesting to note that the constraint on s for obtaining the time bound of $n^{2-1/O(\log c(n))}$ turns out to be $s < n^{1/400}$.

4 Conclusion

In this report, we presented our understanding of the probabilistic polynomial method proposed by R. Williams [Wil14]. We briefly explained some of the tools used in faster APSP and Boolean Orthogonal Detection for obtaining an efficient circuit and translating it using the Razborov-Smolensky randomized construction.

5 Future Directions

We approached a few problems which were related to min-plus product of matrices, and tried to use the scheme laid down by R. Williams[Wil14]. One of the problems was min-plus product of two $n \times n$ matrices M_1, M_2, given the tensor decomposition of both as $M_1 = A_1 \otimes B_1$ and $M_2 = A_2 \otimes B_2$. A possible approach to this problem can be exploiting pre-computed min-plus products of rows of B_1 with columns of B_2.

References

[AWY15] Amir Abboud, Ryan Williams, and Huacheng Yu. More applications of the polynomial method to algorithm design. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 218–230. SIAM, 2015.

[Cop82] Don Coppersmith. Rapid multiplication of rectangular matrices. *SIAM Journal on Computing*, 11(3):467–471, 1982.

[CW16] Timothy M Chan and Ryan Williams. Deterministic apsp, orthogonal vectors, and more: Quickly derandomizing razborov-smolensky. SODA, 2016.

[Fre76] Michael L Fredman. New bounds on the complexity of the shortest path problem. *SIAM Journal on Computing*, 5(1):83–89, 1976.

[Mun71] Ian Munro. Efficient determination of the transitive closure of a directed graph. *Information Processing Letters*, 1(2):56–58, 1971.

[Raz87] Alexander A Razborov. Lower bounds for the size of circuits of bounded depth with basis f^; g. *Math. Notes Acad. Sci. USSR*, 41(4):333–338, 1987.

[Smo87] Roman Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 77–82. ACM, 1987.

[Wil14] Ryan Williams. Faster all-pairs shortest paths via circuit complexity. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 664–673. ACM, 2014.