Quantum Algorithms for Equation Solving and Optimization over Finite Fields and Applications in Cryptanalysis

Yu-Ao Chen

Academy of Mathematics and Systems Science
Chinese Academy of Sciences

May 19, 2019

Joint work with Xiao-Shan Gao and Chun-Ming Yuan

What have we done?

- (Main contribution) We present a quantum algorithm to find a Boolean solution of $\mathcal{F} = \mathbf{0}$ for $\mathcal{F} \subset \mathbb{C}[\mathbb{X}]$ if existing,

What have we done?

- (Main contribution) We present a quantum algorithm to find a Boolean solution of $\mathcal{F} = \mathbf{0}$ for $\mathcal{F} \subset \mathbb{C}[\mathbb{X}]$ if existing,
  (Boolean: each x_i is 0 or 1);

What have we done?

- (Main contribution) We present a quantum algorithm to find a Boolean solution of $\mathcal{F} = \mathbf{0}$ for $\mathcal{F} \subset \mathbb{C}[\mathbb{X}]$ if existing,
  (Boolean: each x_i is 0 or 1);
- 'finding a solution of $\mathcal{F} = \mathbf{0}$ for $\mathcal{F} \subset \mathbb{F}_q[\mathbb{X}]$' $\Longrightarrow_{\text{poly}}$ 'finding a Boolean solution of $\mathcal{F}' = \mathbf{0}$ where $\mathcal{F}' \subset \mathbb{C}[\mathbb{X}]$';

What have we done?

- (Main contribution) We present a quantum algorithm to find a Boolean solution of $\mathcal{F} = \mathbf{0}$ for $\mathcal{F} \subset \mathbb{C}[\mathbb{X}]$ if existing, (Boolean: each x_i is 0 or 1);
- 'finding a solution of $\mathcal{F} = \mathbf{0}$ for $\mathcal{F} \subset \mathbb{F}_q[\mathbb{X}]$' $\Longrightarrow_{\text{poly}}$ 'finding a Boolean solution of $\mathcal{F}' = \mathbf{0}$ where $\mathcal{F}' \subset \mathbb{C}[\mathbb{X}]$';
- 'solving many NP problems' $\Longrightarrow_{\text{poly}}$ 'equation solving over $\mathbb{F}_2$';
- 'optimization over $\mathbb{F}_p$' $\Longrightarrow_{\text{poly}}$ 'equation solving over $\mathbb{F}_p$';

What have we done?

- (Main contribution) We present a quantum algorithm to find a Boolean solution of $\mathcal{F} = \mathbf{0}$ for $\mathcal{F} \subset \mathbb{C}[\mathbb{X}]$ if existing,
  (Boolean: each x_i is 0 or 1);
- 'finding a solution of $\mathcal{F} = \mathbf{0}$ for $\mathcal{F} \subset \mathbb{F}_q[\mathbb{X}]$' $\Longrightarrow_{\text{poly}}$ 'finding a Boolean solution of $\mathcal{F}' = \mathbf{0}$ where $\mathcal{F}' \subset \mathbb{C}[\mathbb{X}]$';
- 'solving many NP problems' $\Longrightarrow_{\text{poly}}$ 'equation solving over $\mathbb{F}_2$';
- 'optimization over $\mathbb{F}_p$' $\Longrightarrow_{\text{poly}}$ 'equation solving over $\mathbb{F}_p$';
- 'algebraic attack on many cryptosystems' $\Longleftrightarrow$ 'equation solving over $\mathbb{F}_2$'.

Inspiration

- Quantum algorithm for linear equation solving, HHL algorithm (Harrow, Hassidim and Lloyd, 2009);

Inspiration

- Quantum algorithm for linear equation solving, HHL algorithm (Harrow, Hassidim and Lloyd, 2009);
- Gröbner basis, F4 algorithm (Faugère, 1999) or the XL algorithm (Courtois et al, 2000): ‘Gröbner basis computing’ $\Longrightarrow$ ‘linear equation solving’

Inspiration

- Quantum algorithm for linear equation solving, HHL algorithm (Harrow, Hassidim and Lloyd, 2009);
- Gröbner basis, F4 algorithm (Faugère, 1999) or the XL algorithm (Courtois et al, 2000): ‘Gröbner basis computing’ $\Longrightarrow$ ‘linear equation solving’
- ‘Gröbner basis computing’ $\Longleftrightarrow$ ‘polynomial equation symbolic (algebraic) solving ’

HHL algorithm:

Quantum algorithm for linear equation solving

Quantum Algorithms for Linear Equation Solving

$A \in \mathbb{C}^{N\times N}$, s-sparse matrix, condition number κ (the quotient of the maximal and minimal nonzero singular value).

Let $\epsilon \in (0,1)$ be an error bound.

Theorem (HHL Algorithm, 2009)

For linear system $A|x\rangle = |b\rangle$, there exist quantum algorithms computing a solution state $|\widehat{x}\rangle$ with error bounded by ϵ and in time $\tilde{O}(\log(N)s\kappa^2/\epsilon)$ using the best known Hamiltonian simulation.

$|x\rangle = \sum x_i|i\rangle \in \mathbb{C}^N$.

Quantum Algorithms for Linear Equation Solving

$A \in \mathbb{C}^{N\times N}$, s-sparse matrix, condition number κ (the quotient of the maximal and minimal nonzero singular value).

Let $\epsilon \in (0,1)$ be an error bound.

Theorem (HHL Algorithm, 2009)

For linear system $A|x\rangle = |b\rangle$, there exist quantum algorithms computing a solution state $|\widehat{x}\rangle$ with error bounded by ϵ and in time $\tilde{O}(\log(N)s\kappa^2/\epsilon)$ using the best known Hamiltonian simulation.

$|x\rangle = \sum x_i|i\rangle \in \mathbb{C}^N$.

If κ and s are small, HHL algorithm achieves exponential speedup comparing to traditional methods,

for instance, comparing to the conjugate gradient method which has complexity $\widetilde{O}(N\sqrt{\kappa}/\epsilon)$.

F4 algorithm:

Gröbner basis, Macaulay matrix, Macaulay linear system and solving degree

Macaulay Linear System and Macaulay Matrix

Polynomial System: $\mathcal{F} = \{f_1, \ldots, f_r\} \subset \mathbb{C}[\mathbb{X}]$ with $d_i = \deg f_i$, where $\mathbb{X} = \{x_1, \ldots, x_n\}$.

Macaulay Linear System and Macaulay Matrix

Polynomial System: $\mathcal{F} = \{f_1, \ldots, f_r\} \subset \mathbb{C}[\mathbb{X}]$ with $d_i = \deg f_i$, where $\mathbb{X} = \{x_1, \ldots, x_n\}$.

Let $D \in \mathbb{N}$ such that $D \geq \max_{i=1}^{r} d_i$.

For each monomial m in $\mathbb{X}$ with $\deg(m) \leq D - d_i$,

$$mf_i = c_* m_0 + c_* m_1 + \cdots + c_* m_{Q_D - 1}$$

is a linear combination of monomials in $\mathbb{X}$ of degree up to D

Macaulay Linear System and Macaulay Matrix

Polynomial System: $\mathcal{F} = \{f_1, \dots, f_r\} \subset \mathbb{C}[\mathbb{X}]$ with $d_i = \deg f_i$, where $\mathbb{X} = \{x_1, \dots, x_n\}$.

Let $D \in \mathbb{N}$ such that $D \geq \max_{i=1}^{r} d_i$.

For each monomial m in $\mathbb{X}$ with $\deg(m) \leq D - d_i$,

$$mf_i = c_* m_0 + c_* m_1 + \cdots + c_* m_{Q_D - 1}$$

is a linear combination of monomials in $\mathbb{X}$ of degree up to D

Written as matrix form: $\mathscr{M}_{\mathcal{F},D}\mathbf{m}_D = \mathbf{b}_{\mathcal{F},D}$

$$\begin{array}{c} \\ m_0 f_1 \\ \vdots \\ \vdots \\ m_{Q_{D-d_r}-1} f_r \end{array} \begin{array}{c} m_1 < m_2 < \cdots < m_{Q_D - 1} \\ \begin{pmatrix} \cdots \\ \cdots \\ \cdots \\ \cdots \end{pmatrix} \end{array} \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_{Q_D - 1} \end{pmatrix} = \begin{array}{c} m_0 = 1 \\ \begin{pmatrix} -f_1(\mathbf{0}) \\ \vdots \\ -f_r(\mathbf{0}) \\ 0 \\ \vdots \\ 0 \end{pmatrix} \end{array}$$

Example

Let $f_1 = x_1^2 - x_2,\ f_2 = x_1 - 2,\ D = 2.$ *Then we have*

$$\begin{array}{c} f_1 \\ f_2 \\ x_1 f_2 \\ x_2 f_2 \end{array} \begin{pmatrix} 0 & -1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ -2 & 0 & 1 & 0 & 0 \\ 0 & -2 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_1^2 \\ x_1 x_2 \\ x_2^2 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \\ 0 \\ 0 \end{pmatrix}.$$

Sparseness of Macaulay Matrix

A matrix M is called ***s*-sparse** if each row and column of M have at most s nonzero entries.

Sparseness of Macaulay Matrix

A matrix M is called ***s*-sparse** if each row and column of M have at most s nonzero entries.

Polynomial System: $\mathcal{F} = \{f_1, \ldots, f_r\} \subset \mathbb{C}[\mathbb{X}]$ with $t_i = \# f_i$ the number of terms of f_i and $\mathbb{X} = \{x_1, \ldots, x_n\}$

Lemma

$\mathscr{M}_{\mathcal{F},D}$ has $(\sum_{i=1}^{r} t_i)$-sparseness and can be computed effectively.

Sparseness of Macaulay Matrix

A matrix M is called ***s*-sparse** if each row and column of M have at most s nonzero entries.

Polynomial System: $\mathcal{F} = \{f_1, \ldots, f_r\} \subset \mathbb{C}[\mathbb{X}]$ with $t_i = \# f_i$ the number of terms of f_i and $\mathbb{X} = \{x_1, \ldots, x_n\}$

Lemma

$\mathscr{M}_{\mathcal{F},D}$ has $(\sum_{i=1}^{r} t_i)$-sparseness and can be computed effectively.

Two often used cases

- **Classic Macaulay Matrix**: Let $r = n + 1$ and $D = \sum_{i=1}^{n+1} d_i - n$, $\mathscr{M}_{\mathcal{F},D}$ has dimension $\widetilde{O}(D^n)$ and $(\sum_{i=1}^{r} t_i)$-sparseness.
- **Multivariate Quadratic Polynomial System (MQ)**: $\mathscr{M}_{\mathcal{F},D}$ has dimension $\widetilde{O}(4^n)$ and $O(rn^2)$-sparseness.

Gröbner Basis

For a polynomial system $\mathcal{F} = \{f_1, \ldots, f_r\} \subset \mathbb{C}[\mathbb{X}]$, denote $(\mathcal{F})$ the ideal generated by $\mathcal{F}$.

For a given monomial order $\preceq$, we can define the leading (maximal) monomial of f (denoted by $\mathrm{lm}(f)$) for any polynomial $f \in \mathbb{C}[\mathbb{X}]$.

A subset $\mathbb{G} \subset (\mathcal{F})$ is called the Gröbner basis of $(\mathcal{F})$ under the monomial order $\preceq$, if

$$\forall f \in (\mathcal{F}), \exists g \in \mathbb{G}, \mathrm{lm}(g) \mid \mathrm{lm}(f).$$

Solving degree:

the relationship between Macaulay linear system and Gröbner basis

Solving Degree for Polynomial System

Polynomial System: $\mathcal{F} = \{f_1, \ldots, f_r\}$, $d_i = \deg f_i$, $d = \max_i d_i$

Solving Degree (minimal) D of $\mathcal{F}$: if the Gröbner basis of $(\mathcal{F})$ can be obtained by Gaussian elimination over $\mathbb{C}$ for linear system
$\mathscr{M}_{\mathcal{F},D}\mathbf{m}_D = \mathbf{b}_{\mathcal{F},D}$,
i.e. $\forall g \in \mathbb{G}, \exists h_i$, s.t. $g = \sum_{i=1}^{r} h_i f_i$ and $\deg(h_i f_i) \leq D$.

Solving Degree for Polynomial System

Polynomial System: $\mathcal{F} = \{f_1, \ldots, f_r\}$, $d_i = \deg f_i$, $d = \max_i d_i$

Solving Degree (minimal) D of $\mathcal{F}$: if the Gröbner basis of $(\mathcal{F})$ can be obtained by Gaussian elimination over $\mathbb{C}$ for linear system $\mathscr{M}_{\mathcal{F},D}\mathbf{m}_D = \mathbf{b}_{\mathcal{F},D}$,
i.e. $\forall g \in \mathbb{G}, \exists h_i$, s.t. $g = \sum_{i=1}^{r} h_i f_i$ and $\deg(h_i f_i) \leq D$.

Results about solving degree:

- **Hermann26**: $\mathrm{SDeg}(\mathcal{F}) \leq (2d)^{2^n}$.
- **Brownawell87**: $\mathrm{SDeg}(\mathcal{F}) \leq d^n$, if $1 \in (\mathcal{F})$.
- **Lazard83, Caminata-Gorla17**: For DRL monomial ordering, if $(\mathcal{F})$ is projective zero-dimension, then the solving degree:

  $\mathbf{SDeg}(\mathcal{F}) \leq D = d_1 + \cdots + d_{n+1} - n + 1 \leq nd + d - n + 1$.
- **MQ**: under Lazard's condition, $\mathrm{SDeg}(\mathcal{F}) \leq D = n + 3$

Complete solving degree

instead of solving degree:

the bridge between solution and solution

Complete Solving Degree for Polynomial System

Complete Solving Degree (minimal) D of $\mathcal{F} = \{f_1, \ldots, f_r\} \subset \mathbb{C}[\mathbb{X}]$:
$\mathbb{G}$ is the reduced Gröbner basis of $(\mathcal{F})$.
$\forall g \in \mathbb{G}$ and $\forall m$ a monomial satisfying $\deg(mg) \le D$, $\exists h_i$ s.t.
$mg = \sum_{i=1}^{r} h_i f_i$, and $\deg(h_i f_i) \le D$.

Complete Solving Degree for Polynomial System

Complete Solving Degree (minimal) D of $\mathcal{F} = \{f_1, \ldots, f_r\} \subset \mathbb{C}[\mathbb{X}]$:
$\mathbb{G}$ is the reduced Gröbner basis of $(\mathcal{F})$.
$\forall g \in \mathbb{G}$ and $\forall m$ a monomial satisfying $\deg(mg) \leq D$, $\exists h_i$ s.t.
$mg = \sum_{i=1}^{r} h_i f_i$, and $\deg(h_i f_i) \leq D$.

Solving Degree (minimal) D of $\mathcal{F}$: if the Gröbner basis of $(\mathcal{F})$ can be obtained by Gaussian elimination over $\mathbb{C}$ for linear system
$\mathscr{M}_{\mathcal{F},D}\mathbf{m}_D = \mathbf{b}_{\mathcal{F},D}$,
i.e. $\forall g \in \mathbb{G}, \exists h_i$, s.t. $g = \sum_{i=1}^{r} h_i f_i$ and $\deg(h_i f_i) \leq D$.

Complete Solving Degree for Polynomial System

Complete Solving Degree (minimal) D of $\mathcal{F} = \{f_1, \ldots, f_r\} \subset \mathbb{C}[\mathbb{X}]$:
$\mathbb{G}$ is the reduced Gröbner basis of $(\mathcal{F})$.
$\forall g \in \mathbb{G}$ and $\forall m$ a monomial satisfying $\deg(mg) \le D$, $\exists h_i$ s.t.
$mg = \sum_{i=1}^{r} h_i f_i$, and $\deg(h_i f_i) \le D$.

Lemma (Monomials are solvable in Macaulay linear system)

Let $\mathbb{V}_{\mathbb{C}}(\mathcal{F}) = \{\mathbf{a}_1, \ldots, \mathbf{a}_w\}$ and $(\mathcal{F})$ radical. For $D \ge \mathrm{CSdeg}(\mathcal{F})$, any solution of $\mathscr{M}_{\mathcal{F},D}\mathbf{m}_D = \mathbf{b}_{\mathcal{F},D}$ is of form $\widehat{\mathbf{m}}_D = \sum_{i=1}^{w} \eta_i \mathbf{m}_D(\mathbf{a}_i)$, where $\sum_{i=1}^{w} \eta_i = 1$.

Complete Solving Degree for Polynomial System

Complete Solving Degree (minimal) D of $\mathcal{F} = \{f_1, \ldots, f_r\} \subset \mathbb{C}[\mathbb{X}]$:
$\mathbb{G}$ is the reduced Gröbner basis of $(\mathcal{F})$.
$\forall g \in \mathbb{G}$ and $\forall m$ a monomial satisfying $\deg(mg) \leq D$, $\exists h_i$ s.t.
$mg = \sum_{i=1}^{r} h_i f_i$, and $\deg(h_i f_i) \leq D$.

Lemma (Monomials are solvable in Macaulay linear system)

Let $\mathbb{V}_{\mathbb{C}}(\mathcal{F}) = \{\mathbf{a}_1, \ldots, \mathbf{a}_w\}$ *and* $(\mathcal{F})$ *radical. For* $D \geq \mathrm{CSdeg}(\mathcal{F})$*, any solution of* $\mathscr{M}_{\mathcal{F},D}\mathbf{m}_D = \mathbf{b}_{\mathcal{F},D}$ *is of form* $\widehat{\mathbf{m}}_D = \sum_{i=1}^{w} \eta_i \mathbf{m}_D(\mathbf{a}_i)$*, where* $\sum_{i=1}^{w} \eta_i = 1$.

Lemma (Bound for CSdeg)

Let $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2 \subset \mathbb{C}[\mathbb{X}]$*, where* $\mathcal{F}_1 = \{g_1, \ldots, g_r\}$, $\mathcal{F}_2 = \{f_1, \ldots, f_n\}$ *satisfying* $\mathrm{lm}(f_i) = x_i^{d_i}$*. We have* $\mathrm{CSdeg}(\mathcal{F}) \leq d - 2n + 2\sum_{i=1}^{n} d_i$*, where* $d = \max \deg(g_i)$.

Use HHL on Macaulay Linear System

Polynomial System: $\mathcal{F} = \{f_1, \ldots, f_r\}$, $d = \max_i \deg f_i$, $t_i = \# f_i$

Let the ideal $(\mathcal{F})$ is radical and projective dimension zero,

also $\mathbb{V}(\mathcal{F}) = \{\mathbf{a}_1, \ldots, \mathbf{a}_w\}$,

Use HHL on Macaulay Linear System

Polynomial System: $\mathcal{F} = \{f_1, \ldots, f_r\}$, $d = \max_i \deg f_i$, $t_i = \# f_i$

Let the ideal $(\mathcal{F})$ is radical and projective dimension zero,

also $\mathbb{V}(\mathcal{F}) = \{\mathbf{a}_1, \ldots, \mathbf{a}_w\}$,

Theorem (Quantum Pseudo Solving for Polynomial System)

For $D \geq CSdeg(\mathcal{F})$ using the HHL algorithms to $\mathscr{M}_{\mathcal{F},D}\mathbf{m}_D = \mathbf{b}_{\mathcal{F},D}$, the solution is

$$|\widehat{\mathbf{m}}_D\rangle = \sum_{i=1}^{w} \eta_i \, |\mathbf{m}_D(\mathbf{a}_i)\rangle,$$

for certain $\eta_i \in \mathbb{C}$.

The complexity is $\widetilde{O}(\log(D) n T_{\mathcal{F}} \kappa^2)$

where $T_{\mathcal{F}} = \sum_{i=1}^{r} t_i$, κ condition number of $\mathscr{M}_{\mathcal{F},D}$.

We can measure $|\widehat{\mathbf{m}}_D\rangle$ to obtain a state $|k\rangle$ also the information:

$\exists \mathbf{a}_i$ **s.t.** $\mathbf{m}_D(\mathbf{a}_i)_k = m_k(\mathbf{a}_i) \neq 0$.

We can measure $|\widehat{\mathbf{m}}_D\rangle$ to obtain a state $|k\rangle$ also the information:

$\exists \mathbf{a}_i$ s.t. $\mathbf{m}_D(\mathbf{a}_i)_k = m_k(\mathbf{a}_i) \neq 0$.

If all $\mathbf{a}_i$ are assumed Boolean, we have $m_k(\mathbf{a}_i) = 1$ furtherly.

We can measure $|\widehat{\mathbf{m}}_D\rangle$ to obtain a state $|k\rangle$ also the information:

$\exists \mathbf{a}_i$ s.t. $\mathbf{m}_D(\mathbf{a}_i)_k = m_k(\mathbf{a}_i) \neq 0$.

If all $\mathbf{a}_i$ are assumed Boolean, we have $m_k(\mathbf{a}_i) = 1$ furtherly.

Thus we can reduce the equation system to another one with less variables by substituting $x_i = 1$ for those x_i appearing in m_k.

Quantum Algorithm to Find Boolean Solutions

Polynomial System: $\mathcal{F} = \{f_1, \ldots, f_r\} \subset \mathbb{C}[x_1, \ldots, x_n]$,

$T_{\mathcal{F}} = \sum_i \# f_i$, $\epsilon \in (0, 1)$, $\mathbb{V}_B(\mathcal{F}) = \mathbb{V}_{\mathbb{C}}(\mathcal{F}, x_1^2 - x_1, \ldots, x_n^2 - x_n)$,

Quantum Algorithm to Find Boolean Solutions

Polynomial System: $\mathcal{F} = \{f_1, \ldots, f_r\} \subset \mathbb{C}[x_1, \ldots, x_n]$,

$T_{\mathcal{F}} = \sum_i \# f_i$, $\epsilon \in (0, 1)$, $\mathbb{V}_B(\mathcal{F}) = \mathbb{V}_{\mathbb{C}}(\mathcal{F}, x_1^2 - x_1, \ldots, x_n^2 - x_n)$,

CSdeg $(\mathcal{F}, x_1^2 - x_1, \ldots, x_n^2 - x_n) \leq 3n$

Lemma (Bound for CSdeg)

Let $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2 \subset \mathbb{C}[\mathbb{X}]$, *where* $\mathcal{F}_1 = \{g_1, \ldots, g_r\}$, $\mathcal{F}_2 = \{f_1, \ldots, f_n\}$ *satisfying* $\mathrm{lm}(f_i) = x_i^{d_i}$. *We have* $\mathrm{CSdeg}(\mathcal{F}) \leq d - 2n + 2\sum_{i=1}^n d_i$, *where* $d = \max \deg(g_i)$.

Quantum Algorithm to Find Boolean Solutions

Polynomial System: $\mathcal{F} = \{f_1, \ldots, f_r\} \subset \mathbb{C}[x_1, \ldots, x_n]$,

$T_{\mathcal{F}} = \sum_i \# f_i$, $\epsilon \in (0, 1)$, $\mathbb{V}_B(\mathcal{F}) = \mathbb{V}_{\mathbb{C}}(\mathcal{F}, x_1^2 - x_1, \ldots, x_n^2 - x_n)$,

$\mathrm{CSdeg}\,(\mathcal{F}, x_1^2 - x_1, \ldots, x_n^2 - x_n) \leq 3n$

Theorem

- *If the algorithm returns a Boolean solution, then it is a solution of* $\mathcal{F} = 0$.

  Equivalently, if $\mathbb{V}_B(\mathcal{F}) = \emptyset$*, the algorithm finds no solution.*

Quantum Algorithm to Find Boolean Solutions

Polynomial System: $\mathcal{F} = \{f_1, \ldots, f_r\} \subset \mathbb{C}[x_1, \ldots, x_n]$,
$T_{\mathcal{F}} = \sum_i \# f_i$, $\epsilon \in (0, 1)$, $\mathbb{V}_B(\mathcal{F}) = \mathbb{V}_{\mathbb{C}}(\mathcal{F}, x_1^2 - x_1, \ldots, x_n^2 - x_n)$,
$\mathrm{CSdeg}\,(\mathcal{F}, x_1^2 - x_1, \ldots, x_n^2 - x_n) \leq 3n$

Theorem

- *If the algorithm returns a Boolean solution, then it is a solution of* $\mathcal{F} = 0$.

  Equivalently, if $\mathbb{V}_B(\mathcal{F}) = \emptyset$, *the algorithm finds no solution.*
- *If* $\mathbb{V}_B(\mathcal{F}) \neq \emptyset$, *the algorithm computes a Boolean solution of* $\mathcal{F} = 0$ *with probability* $\geq 1 - \epsilon$.

Quantum Algorithm to Find Boolean Solutions

Polynomial System: $\mathcal{F} = \{f_1, \ldots, f_r\} \subset \mathbb{C}[x_1, \ldots, x_n]$,

$T_{\mathcal{F}} = \sum_i \# f_i$, $\epsilon \in (0,1)$, $\mathbb{V}_B(\mathcal{F}) = \mathbb{V}_{\mathbb{C}}(\mathcal{F}, x_1^2 - x_1, \ldots, x_n^2 - x_n)$,

$\mathrm{CSdeg}\,(\mathcal{F}, x_1^2 - x_1, \ldots, x_n^2 - x_n) \leq 3n$

Theorem

- *If the algorithm returns a Boolean solution, then it is a solution of* $\mathcal{F} = 0$.

  Equivalently, if $\mathbb{V}_B(\mathcal{F}) = \emptyset$*, the algorithm finds no solution.*
- *If* $\mathbb{V}_B(\mathcal{F}) \neq \emptyset$*, the algorithm computes a Boolean solution of* $\mathcal{F} = 0$ *with probability* $\geq 1 - \epsilon$.
- *The complexity is* $\widetilde{O}(n^{2.5}(n + T_{\mathcal{F}})\kappa^2 \log 1/\epsilon)$

κ *is the maximal condition number of the Macaulay matrices.*

Comparing with HHL

$A \in \mathbb{C}^{M\times N}$, s-sparse matrix, condition number κ.

Let $\epsilon \in (0, 1)$ be an error bound.

Theorem (HHL Algorithm)

For linear system $A|x\rangle = |b\rangle$, there exist quantum algorithms computing a solution state $|\widehat{x}\rangle$ with error bounded by ϵ and in time $\tilde{O}(\log(N + M)s\kappa^2/\epsilon)$

Comparing with HHL

$A \in \mathbb{C}^{M\times N}$, s-sparse matrix, condition number κ.

Let $\epsilon \in (0,1)$ be an error bound.

Theorem (HHL Algorithm)

For linear system $A|x\rangle = |b\rangle$, there exist quantum algorithms computing a solution state $|\widehat{x}\rangle$ with error bounded by ϵ and in time $\tilde{O}(\log(N+M)s\kappa^2/\epsilon)$

Several Subtle Properties:

- The algorithm does not give a solution of $Ax = b$, but a state $|\widehat{x}\rangle$. Measure of $|\widehat{x}\rangle$ gives $|x_1| : |x_2| : \cdots : |x_n|$.

Comparing with HHL

$A \in \mathbb{C}^{M\times N}$, s-sparse matrix, condition number κ.

Let $\epsilon \in (0, 1)$ be an error bound.

Theorem (HHL Algorithm)

For linear system $A|x\rangle = |b\rangle$, there exist quantum algorithms computing a solution state $|\widehat{x}\rangle$ with error bounded by ϵ and in time $\tilde{O}(\log(N+M)s\kappa^2/\epsilon)$

Several Subtle Properties:

- The algorithm does not give a solution of $Ax = b$, but a state $|\widehat{x}\rangle$. Measure of $|\widehat{x}\rangle$ gives $|x_1| : |x_2| : \cdots : |x_n|$.
- The algorithm gives a state even if $Ax = b$ has no solution.

Comparing with HHL

$A \in \mathbb{C}^{M \times N}$, s-sparse matrix, condition number κ.

Let $\epsilon \in (0, 1)$ be an error bound.

Theorem (HHL Algorithm)

For linear system $A|x\rangle = |b\rangle$, there exist quantum algorithms computing a solution state $|\widehat{x}\rangle$ with error bounded by ϵ and in time $\tilde{O}(\log(N + M)s\kappa^2/\epsilon)$

Several Subtle Properties:

- The algorithm does not give a solution of $Ax = b$, but a state $|\widehat{x}\rangle$. Measure of $|\widehat{x}\rangle$ gives $|x_1| : |x_2| : \cdots : |x_n|$.
- The algorithm gives a state even if $Ax = b$ has no solution.
- The algorithm works over $\mathbb{C}$ not $\mathbb{F}_2$.

Comparing with HHL

$A \in \mathbb{C}^{M\times N}$, s-sparse matrix, condition number κ.

Let $\epsilon \in (0,1)$ be an error bound.

Theorem (HHL Algorithm)

For linear system $A|x\rangle = |b\rangle$, there exist quantum algorithms computing a solution state $|\widehat{x}\rangle$ with error bounded by ϵ and in time $\tilde{O}(\log(N+M)s\kappa^2/\epsilon)$

Several Subtle Properties:

- The algorithm does not give a solution of $Ax = b$, but a state $|\widehat{x}\rangle$. Measure of $|\widehat{x}\rangle$ gives $|x_1| : |x_2| : \cdots : |x_n|$.
- The algorithm gives a state even if $Ax = b$ has no solution.
- The algorithm works over $\mathbb{C}$ not $\mathbb{F}_2$.

The nice property of our algorithm mainly due to the Boolean solutions.

Equation Solving and Optimization over Finite Fields

Main Result

An optimization problem over $\mathbb{F}_p = \{0, \ldots, p-1\}$

$\min_{\mathbb{X} \in \mathbb{F}_p^n, \mathbb{Y} \in \mathbb{Z}^m} o(\mathbb{X}, \mathbb{Y})$, subject to

$f_j(\mathbb{X}) = 0, j = 1, \ldots, r$; over $\mathbb{F}_p$

$0 \leq g_i(\mathbb{X}, \mathbb{Y}) \leq b_i, i = 1, \ldots, s; 0 \leq y_l \leq u_l, l = 1, \ldots, m$

where $f_i \in \mathbb{F}_p[\mathbb{X}]$, $o, g_i \in \mathbb{Z}[\mathbb{X}, \mathbb{Y}]$, and $u, b_i, u_i \in \mathbb{N}$.

Main Result

An optimization problem over $\mathbb{F}_p = \{0, \ldots, p-1\}$

$\min_{\mathbb{X}\in\mathbb{F}_p^n, \mathbb{Y}\in\mathbb{Z}^m} o(\mathbb{X}, \mathbb{Y})$, subject to

$f_j(\mathbb{X}) = 0, j = 1, \ldots, r$; over $\mathbb{F}_p$

$0 \le g_i(\mathbb{X}, \mathbb{Y}) \le b_i, i = 1, \ldots, s; 0 \le y_l \le u_l, l = 1, \ldots, m$

where $f_i \in \mathbb{F}_p[\mathbb{X}]$, $o, g_i \in \mathbb{Z}[\mathbb{X}, \mathbb{Y}]$, and $u, b_i, u_i \in \mathbb{N}$.

- We give a quantum algorithm with **complexity polynomial in the input size and** κ (the condition number of certain matrix)

  Achieved exponential speedup if κ is small.

Main Result

An optimization problem over $\mathbb{F}_p = \{0, \ldots, p-1\}$

$\min_{\mathbb{X}\in\mathbb{F}_p^n, \mathbb{Y}\in\mathbb{Z}^m} o(\mathbb{X}, \mathbb{Y})$, subject to

$f_j(\mathbb{X}) = 0, j = 1, \ldots, r$; over $\mathbb{F}_p$

$0 \leq g_i(\mathbb{X}, \mathbb{Y}) \leq b_i, i = 1, \ldots, s; 0 \leq y_l \leq u_l, l = 1, \ldots, m$

where $f_i \in \mathbb{F}_p[\mathbb{X}]$, $o, g_i \in \mathbb{Z}[\mathbb{X}, \mathbb{Y}]$, and $u, b_i, u_i \in \mathbb{N}$.

- We give a quantum algorithm with **complexity polynomial in the input size and** κ (the condition number of certain matrix)

  Achieved exponential speedup if κ is small.
- **Include many NP hard problems as special cases**:
  Equation Solving over finite field, Boolean equation solving, SAT, Polynomial systems with noise, Short integer solution problem $(0,1)$-programming, knapsack problem

Solving Boolean Equations

Example (A Boolean equation may have no complex solution)

Let $f = x_1 + x_2 + 1$. *Then* $\mathbb{V}_{\mathbb{F}_2}(f) = \{(0,1),(1,0)\}$.

But Boolean solution $\mathbb{V}_B(f) = \emptyset$.

So we cannot use our QA over $\mathbb{C}$ *to solve Boolean equations directly.*

Solving Boolean Equations

Example (A Boolean equation may have no complex solution)

Let $f = x_1 + x_2 + 1$*. Then* $\mathbb{V}_{\mathbb{F}_2}(f) = \{(0,1),(1,0)\}$.

But Boolean solution $\mathbb{V}_B(f) = \emptyset$.

So we cannot use our QA over $\mathbb{C}$ *to solve Boolean equations directly.*

Reduce f *to an equivalent polynomial* $C(f) = f - 2 = x_1 + x_2 - 1$.

Then, Boolean solution $\mathbb{V}_B(C(f)) = \{(0,1),(1,0)\}$.

Reduction 1: Integer interval $\{0, 1, \ldots, b\}$

- $b \in \mathbb{N}_+$, $s = \lfloor \log_2(b) \rfloor$
- $\mathbb{B}_{\mathrm{bit}} = \{B_0, \ldots, B_s\}$ Boolean variables
- Inspired by $b = (2^s - 1) + (b + 1 - 2^s)$, we introduce

$$\theta_b(\mathbb{B}_{\mathrm{bit}}) = \sum_{i=0}^{s-1} 2^i B_i + (b + 1 - 2^s) B_s.$$

- $\theta_b : \{0, 1\}^{s+1} \rightrightarrows \{0, 1, \ldots, b\}$.

Reduction 2: Solving Equations over Finite Fields

Let $\mathcal{F} \subset \mathbb{F}_p[\mathbb{X}]$ and $\mathbb{X} = \{x_1, \ldots, x_n\}$

Reduce to poly in Boolean variables over $\mathbb{C}$: $C(\mathcal{F}) \subset \mathbb{C}[\mathbb{X}_{\text{bit}}]$:

Reduction 2: Solving Equations over Finite Fields

Let $\mathcal{F} \subset \mathbb{F}_p[\mathbb{X}]$ and $\mathbb{X} = \{x_1, \ldots, x_n\}$

Reduce to poly in Boolean variables over $\mathbb{C}$: $C(\mathcal{F}) \subset \mathbb{C}[\mathbb{X}_{\mathrm{bit}}]$:

- Reduce $\mathcal{F}$ to Multivariate Quadratic Polynomials (MQ) over $\mathbb{F}_p$
- Reduce MQ over $\mathbb{F}_p$ to MQ in Boolean variables over $\mathbb{F}_p$

  Since $x_i \in \mathbb{F}_p$,
  $x_i = \theta_{p-1}(x_{ij}) = \sum_{j=0}^{\lfloor \log_2 p \rfloor - 1} x_{ij} 2^j + (p - 2^{\lfloor \log_2 p \rfloor}) x_{i \lfloor \log_2 p \rfloor}$,
  x_{ij} Boolean variables

- Reduce MQ over $F_p \Rightarrow$ Equations over $\mathbb{C}$

  $f \Rightarrow f - p\theta_{\# f}(u_{fj})$ over $\mathbb{C}$

Reduction 2: Solving Equations over Finite Fields

Let $\mathcal{F} \subset \mathbb{F}_p[\mathbb{X}]$ and $\mathbb{X} = \{x_1, \ldots, x_n\}$

Reduce to poly in Boolean variables over $\mathbb{C}$: $C(\mathcal{F}) \subset \mathbb{C}[\mathbb{X}_{\mathrm{bit}}]$:

- Reduce $\mathcal{F}$ to Multivariate Quadratic Polynomials (MQ) over $\mathbb{F}_p$
- Reduce MQ over $\mathbb{F}_p$ to MQ in Boolean variables over $\mathbb{F}_p$

  Since $x_i \in \mathbb{F}_p$,
  $x_i = \theta_{p-1}(x_{ij}) = \sum_{j=0}^{\lfloor \log_2 p \rfloor - 1} x_{ij} 2^j + (p - 2^{\lfloor \log_2 p \rfloor}) x_{i \lfloor \log_2 p \rfloor}$,
  x_{ij} Boolean variables
- Reduce MQ over $F_p \Rightarrow$ Equations over $\mathbb{C}$

  $f \Rightarrow f - p\theta_{\#f}(u_{fj})$ over $\mathbb{C}$

Then, we have

$\mathbb{V}_{\mathbb{F}_p}(\mathcal{F}) = \mathbb{V}_B(C(\mathcal{F}))$
The size of $C(\mathcal{F})$ is nicely controlled.

We can solve $\mathcal{F} = 0$ in $\widetilde{O}(n^{3.5} T_{\mathcal{F}}^{3.5} \log^8 p \kappa_{\mathcal{F}}^2)$

Example

Example

For $f = x_1 + x_2 + 1$*, we consider*
$C(f) = f - 2\theta_3(u_{fj}) = (x_1 + x_2 + 1) - 2(u_{f1} + 2u_{f2})$.
$C(f) = 0$ *has only two Boolean solution* $x_1 = u_{f1} = 0, x_2 = u_{f1} = 1$ *and* $x_2 = u_{f1} = 0, x_1 = u_{f1} = 1$*, which is coincident to* $\mathbb{V}_{\mathbb{F}_2}(f) = \{(0,1),(1,0)\}$.

Reduction 3: Inequality over Finite Fields

Inequality: $\mathcal{I} : 0 \leq g \leq u, g \in \mathbb{Z}[y_1, \ldots, y_n]$, and $0 \leq y_i \leq b_i$

Reduce to poly in Boolean variables: $I(\mathcal{I}) \subset \mathbb{C}[Y_{\text{bit}}]$:

Reduction 3: Inequality over Finite Fields

Inequality: $\mathcal{I} : 0 \leq g \leq u, g \in \mathbb{Z}[y_1, \ldots, y_n]$, and $0 \leq y_i \leq b_i$

Reduce to poly in Boolean variables: $I(\mathcal{I}) \subset \mathbb{C}[Y_{\mathrm{bit}}]$:

- The inequality $\mathcal{I}$ is equivalent to

  an equation in Boolean variables:
  $g(\theta_{b_1}(y_{1j}), \ldots, \theta_{b_n}(y_{nj})) - \theta_u(u_j) = 0$

Reduction 4: The Objective Function

The objective function: $\min_{\mathbb{X}} o(\mathbb{X})$ under assumption $0 \leq o(\mathbb{X}) \leq h$

Reduce $o(\mathbb{X}) \in [\alpha, \alpha + 2^{\beta})$ to find Boolean solutions over $\mathbb{C}$

- Check $\exists \mathbb{X}$ s.t. $o(\mathbb{X}) \in [\alpha, \alpha + 2^{\beta})$ by solving the equation

  $E_{\alpha,\beta} = o(\mathbb{X}) - (\alpha + \sum_{i=0}^{\beta-1} E_i 2_i)$, where E_i are Boolean variables

Reduction 4: The Objective Function

The objective function: $\min_{\mathbb{X}} o(\mathbb{X})$ under assumption $0 \leq o(\mathbb{X}) \leq h$

Reduce $o(\mathbb{X}) \in [\alpha, \alpha + 2^{\beta})$ **to find Boolean solutions over** $\mathbb{C}$

- Check $\exists \mathbb{X}$ s.t. $o(\mathbb{X}) \in [\alpha, \alpha + 2^{\beta})$ by solving the equation

  $E_{\alpha,\beta} = o(\mathbb{X}) - (\alpha + \sum_{i=0}^{\beta-1} E_i 2_i)$, where E_i are Boolean variables

Find $\min_{\mathbb{X}} o(\mathbb{X})$ for $0 \leq o(\mathbb{X}) \leq h$

- Set $\alpha = 0$, $\beta = \lfloor \log(h) \rfloor - 1$.
  Solve $E_{\alpha,\beta} = 0$ to find a value of $o(\mathbb{X})$ in $[\alpha, \alpha + 2^{\beta})$.
- Obtain minimal value of $o(\mathbb{X})$ in $[0, h]$ by dividing $[0, h]$ into intervals of the form $[\alpha, \alpha + 2^{\beta})$ for at most $\log h$ times and by solving equations $E_{\alpha,\beta}$.

Summary

An optimization problem over $\mathbb{F}_p$

$\min_{\mathbb{X}\in\mathbb{F}_p^n, \mathbb{Y}\in\mathbb{Z}^m} o(\mathbb{X},\mathbb{Y})$, subject to $0 \le o(\mathbb{X}) \le h$

$f_j(\mathbb{X}) = 0, j = 1, \ldots, r$; over $\mathbb{F}_p$

$0 \le g_i(\mathbb{X},\mathbb{Y}) \le b_i, i = 1, \ldots, s; 0 \le y_k \le u_l, l = 1, \ldots, m$

Solution procedure to the optimization problem

- Reduce equality constraints $f_j(\mathbb{X}) = 0$ over $\mathbb{F}_q$ to a polynomial system $\mathcal{F}_1$ over $\mathbb{C}$.
- Reduce inequalities $0 \le g_i(\mathbb{X},\mathbb{Y}) \le b_i$ to a polynomial system $\mathcal{F}_2$ over $\mathbb{C}$.
- Divide $[0, h]$ into intervals of form $[\alpha, \alpha + 2^\beta)$ and compute the Boolean solutions of $\mathcal{F}_1 \cup \mathcal{F}_2 \cup \{E_{\alpha,\beta}\}$ for at most $\log h$ times to find the minimal value for $o(\mathbb{X},\mathbb{Y})$

Applications

Solving Boolean MQ and Cryptanalysis

Analysis of many cryptosystems $\Rightarrow$ solving Boolean MQ.

- Stream cipher Trivum (ISO/IEC standard)
- Block cipher AES (NIST standard)
- Hash function SHA-3/Keccak (NIST standard)

Solving Boolean MQ and Cryptanalysis

Analysis of many cryptosystems $\Rightarrow$ solving Boolean MQ.

Stream cipher Trivum (ISO/IEC standard)
Block cipher AES (NIST standard)
Hash function SHA-3/Keccak (NIST standard)

Ciphers	#Vars(n)	#Eqs	$T_{\mathcal{F}}$	Bezout B.	QComplexity
AES-128	4288	10616	252288	2^n	$2^{73.80}c\kappa^2$
AES-192	7488	18096	421248	2^n	$2^{76.44}c\kappa^2$
AES-256	11904	29520	696384	2^n	$2^{79.04}c\kappa^2$
Trivium	3543	4407	24339	2^n	$2^{57.08}c\kappa^2$
Trivium	6999	9015	49683	2^n	$2^{60.74}c\kappa^2$
Keccak	76800	77160	611023	2^n	$2^{78.04}c\kappa^2$
Keccak	76800	77288	611540	2^n	$2^{78.04}c\kappa^2$

These cryptosystems are secure under QA only if the condition numbers of their corresponding equation systems are large.

(0, 1)-Programming

$$\min_{x_i \in \{0,1\}} o(\mathbb{X}) = c_1 x_1 + \cdots + c_n x_n,$$

subject to

$$a_{i1} x_1 + \cdots + a_{in} x_n \leq b_i, i = 1, \ldots, s$$

(0, 1)-Programming

$\min_{x_i \in \{0,1\}} o(\mathbb{X}) = c_1 x_1 + \cdots + c_n x_n,$

subject to

$a_{i1} x_1 + \cdots + a_{in} x_n \leq b_i, i = 1, \ldots, s$

Quantum Complexity: $\widetilde{O}(n^{4.5} s^{3.5} \log^8 h\kappa^2)$, where $u = \max\{b_i, \sum_j c_j\}$

(0, 1)-Programming

$\min_{x_i \in \{0,1\}} o(\mathbb{X}) = c_1 x_1 + \cdots + c_n x_n$,

subject to

$a_{i1} x_1 + \cdots + a_{in} x_n \leq b_i, i = 1, \ldots, s$

Quantum Complexity: $\widetilde{O}(n^{4.5} s^{3.5} \log^8 h\kappa^2)$, where $u = \max\{b_i, \sum_j c_j\}$

Include many famous NP-hard optimization problems

- Subset sum problem: (0, 1) solution of one linear equation $\sum_i a_i x_i = b$.
  Complexity: $\widetilde{O}(n^{3.5}\kappa^2)$.
- **Knapsack problem** $(s = 1)$

Polynomial System Solving with Noise (PSWN)

For an over-determined polynomial system $\mathcal{F} = \{f_1, \ldots, f_r\} \subset \mathbb{F}_p[\mathbb{X}]$ ($r \gg n$), the PSWN problem is to find an $\mathbb{X} \in \mathbb{F}_p^n$ satisfies the maximal number of equations in $\mathcal{F}$.

Polynomial System Solving with Noise (PSWN)

For an over-determined polynomial system $\mathcal{F} = \{f_1, \ldots, f_r\} \subset \mathbb{F}_p[\mathbb{X}]$ ($r \gg n$), the PSWN problem is to find an $\mathbb{X} \in \mathbb{F}_p^n$ satisfies the maximal number of equations in $\mathcal{F}$.

- PSWN can be solved with $\widetilde{O}(r^{3.5} T_{\mathcal{F}}^{3.5} (\log p)^8 \kappa^2)$

Polynomial System Solving with Noise (PSWN)

For an over-determined polynomial system $\mathcal{F} = \{f_1, \ldots, f_r\} \subset \mathbb{F}_p[\mathbb{X}]$ ($r \gg n$), the PSWN problem is to find an $\mathbb{X} \in \mathbb{F}_p^n$ satisfies the maximal number of equations in $\mathcal{F}$.

- PSWN can be solved with $\widetilde{O}(r^{3.5} T_{\mathcal{F}}^{3.5} (\log p)^8 \kappa^2)$
- Linear System with Noise (LWN) $A\mathbb{X} = \mathbf{b}$ is NP-hard

  can be solved in time $\widetilde{O}((n + r\log p)^{2.5}(T_A + r\log^2 p)\log^{3.5} p\kappa^2)$, $A \in \mathbb{C}^{r\times n}$, T_A is the number of nonzero entries in A

Conclusion

1. We give a quantum algorithm to solve polynomial systems and optimization over finite fields, whose complexity is polynomial in the input size and the condition numbers of certain Macaulay matrices.
2. The quantum algorithm achieves exponential speedup if the condition number is small.
3. The quantum algorithm applies to cryptanalysis of important cryptosystems and show that these cryptosystems are safe only if their condition numbers are large.

Thanks !

Chen, Y.A. and Gao, X.S., Quantum Algorithms for Boolean Equation Solving and Quantum Algebraic Attack on Cryptosystems, ArXiv1712.06239v3, 2017.

Chen, Y.A., Gao, X.S., Yuan, C.M., Quantum Algorithms for Optimization and Polynomial Systems Solving over Finite Fields, ArXiv1802.03856v2, 2018.

How about the condition number κ?

$\kappa(n, D)$

For the field equations $H_{\mathbb{X}} = \{x_1^2 - x_1, \ldots, x_n^2 - x_n\}$, denote the condition number of $\mathscr{M}_{H_{\mathbb{X}},D}$ by $\kappa(n, D)$, and we have

$\kappa(n, D)$

For the field equations $H_{\mathbb{X}} = \{x_1^2 - x_1, \ldots, x_n^2 - x_n\}$, denote the condition number of $\mathscr{M}_{H_{\mathbb{X}},D}$ by $\kappa(n, D)$, and we have

Theorem

$\kappa(1, D) = \cot \frac{\pi}{2D} = O(D)$, $\kappa(2, D) \le 8D - 20 = O(D)$.

$\kappa(n, D)$

For the field equations $H_{\mathbb{X}} = \{x_1^2 - x_1, \ldots, x_n^2 - x_n\}$, denote the condition number of $\mathscr{M}_{H_{\mathbb{X}},D}$ by $\kappa(n, D)$, and we have

Theorem

$\kappa(1, D) = \cot \frac{\pi}{2D} = O(D)$, $\kappa(2, D) \leq 8D - 20 = O(D)$.

Theorem

For $n \geq D - 1$, $\kappa(n, D) = \kappa(D - 1, D)$.

$\kappa(n, D)$

For the field equations $H_{\mathbb{X}} = \{x_1^2 - x_1, \ldots, x_n^2 - x_n\}$, denote the condition number of $\mathscr{M}_{H_{\mathbb{X}},D}$ by $\kappa(n, D)$, and we have

Theorem

$\kappa(1, D) = \cot \frac{\pi}{2D} = O(D)$, $\kappa(2, D) \leq 8D - 20 = O(D)$.

Theorem

For $n \geq D - 1$, $\kappa(n, D) = \kappa(D - 1, D)$.

Conjecture

For fixed $n \geq 3$, $\kappa(n, D) = O(D^{1.5})$.

Pascal's simplex

$\kappa(n, D)$ is related to the algebraic connectivity of the graph Pascal's simplex.

Figure: $G(1,3)$

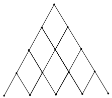

Figure: $G(2,4)$

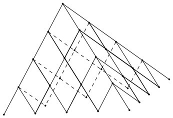

Figure: $G(3,4)$

To compute the algebraic connectivity for a random graph is NP-hard.

Thanks !

One more time!