Solving k-SAT on quantum machines

Alain Sarlette — QUANTIC — inria Paris

Q-feedback workshop, Mines Paris, June 2026

Co-authors

U.Berkeley

Tathagata Karmakar

Philippe Lewalle

Thilo Scharnhorst

Yipei Zhang

Birgitta Whaley (PI)

inria / ENS Paris

Artem Mamichev

Pierre Rouchon

Outline

I. Optimization / classical problem

II. Quantum annealing methods

III. Harnessing quantum dissipation

Outline

I. Optimization / classical problem

k-SAT : computational approach

II. Quantum annealing methods

standard quantum annealing

III. Harnessing quantum dissipation

Outline

I. Optimization / classical problem

k-SAT : computational approach

II. Quantum annealing methods

standard quantum annealing

new quantum annealing

III. Harnessing quantum dissipation

quantum Zeno dragging

Tutorial

New developments

Outline

I. Optimization / classical problem

k-SAT : computational approach

II. Quantum annealing methods

III. Harnessing quantum dissipation

Tutorial

New developments

Main result

k-SAT combinatorial optimization paradigm

x_j in $\{ 0, 1 \} = \{ \text{true}, \text{false} \}$ for $j=1,2, \ldots , n$

k-SAT combinatorial optimization paradigm

x_j in $\{ 0, 1 \} = \{ \text{ true, false} \}$ for $j=1,2, \dots , n$

Can we get « true » as an output of :

$$F(x) = (x_1 \text{ or } x_2 \text{ or } x_3) \text{ and}$$
$$(\overline{x}_1 \text{ or } x_4 \text{ or } x_5) \text{ and}$$
$$(\overline{x}_2 \text{ or } \overline{x}_4 \text{ or } \overline{x}_5) \text{ and } \dots \quad ?$$

k-SAT combinatorial optimization paradigm

x_j in $\{ 0, 1 \} = \{ \text{true}, \text{false} \}$ for $j=1,2, \ldots , n$

Can we get « true » as an output of :

$$F(x) = (x_1 \text{ or } x_2 \text{ or } x_3) \text{ and}$$
$$(\overline{x}_1 \text{ or } x_4 \text{ or } x_5) \text{ and}$$
$$(\overline{x}_2 \text{ or } \overline{x}_4 \text{ or } \overline{x}_5) \text{ and } \ldots \quad ?$$

$k=3$ literals per clause

k-SAT combinatorial optimization paradigm

x_j in $\{ 0, 1 \} = \{ \text{true}, \text{false} \}$ for $j=1,2, \dots , n$

Can we get « true » as an output of :

$$F(x) = (x_1 \text{ or } x_2 \text{ or } x_3) \text{ and}$$
$$(\overline{x}_1 \text{ or } x_4 \text{ or } x_5) \text{ and}$$
$$(\overline{x}_2 \text{ or } \overline{x}_4 \text{ or } \overline{x}_5) \text{ and } \dots \quad ?$$

m clauses

k=3 literals per clause

k-SAT combinatorial optimization paradigm

x_j in $\{ 0, 1 \} = \{ \text{true}, \text{false} \}$ for $j=1,2, \dots , n$

Can we get « true » as an output of :

$$F(x) = (x_1 \text{ or } x_2 \text{ or } x_3) \text{ and}$$
$$(\overline{x}_1 \text{ or } x_4 \text{ or } x_5) \text{ and}$$
$$(\overline{x}_2 \text{ or } \overline{x}_4 \text{ or } \overline{x}_5) \text{ and } \dots \quad ?$$

m clauses

k=3 literals per clause

Example: yes, take e.g.
$x_1=x_2=\text{false}$
$x_3=\text{true}$

k-SAT combinatorial optimization paradigm

x_j in $\{ 0, 1 \} = \{ \text{ true, false} \}$ for $j=1,2, \ldots , n$

Can we get « true » as an output of :

$$F(x) = (x_1 \text{ or } x_2 \text{ or } x_3) \text{ and}$$
$$(\overline{x}_1 \text{ or } x_4 \text{ or } x_5) \text{ and } \ldots \ ?$$

NP – complete : Hard to solve for some instances $(\sim 2^{\alpha n})$

Equivalent to whole class of (practical) problems

Solving k-SAT on classical computers

- **Brute force:** try all $\mathrm{T} = 2^{\mathrm{n}}$ possibilities

Solving k-SAT on classical computers

- **Brute force:** try all $\mathrm{T} = 2^{\mathrm{n}}$ possibilities
- **Local search:** still low success probability

  → retry with (exponentially) many initial states

Solving k-SAT on classical computers

- **Brute force:** try all $\mathrm{T} = 2^{\mathrm{n}}$ possibilities
- **Local search:** still low success probability
  → retry with (exponentially) many initial states

Schöning algorithm = repeatedly flip a random bad bit : $\mathrm{T} = (4/3)^{\mathrm{n}} = 2^{\,0.415\,\mathrm{n}}$

Solving k-SAT on classical computers

- **Brute force:** try all $T = 2^n$ possibilities
- **Local search:** still low success probability
  → retry with (exponentially) many initial states

Schöning algorithm = repeatedly flip a random bad bit : $T = (4/3)^n = 2^{0.415\,n}$

PPSZ algorithm = preprocess to build longer implied clauses : $T = 2^{0.386\,n}$

Solving k-SAT on classical computers

- **Brute force:** try all $T = 2^n$ possibilities
- **Local search:** still low success probability
  → retry with (exponentially) many initial states

  Schöning algorithm = repeatedly flip a random bad bit : $T = (4/3)^n = 2^{0.415\,n}$

  PPSZ algorithm = preprocess to build longer implied clauses : $T = 2^{0.386\,n}$

- **Backtracking tree:** keep track of implications of partial assignments ;
  heuristic for next variable choice

Solving k-SAT on classical computers

- **Brute force:** try all $T = 2^n$ possibilities
- **Local search:** still low success probability
  → retry with (exponentially) many initial states

Schöning algorithm = repeatedly flip a random bad bit : $T = (4/3)^n = 2^{0.415\,n}$

PPSZ algorithm = preprocess to build longer implied clauses : $T = 2^{0.386\,n}$

- **Backtracking tree:** keep track of implications of partial assignments ; heuristic for next variable choice

(best current guarantee)

Cutting runtime to $\sqrt{T}$ with Quantum Search (1)

Cutting runtime to $\sqrt{T}$ with Quantum Search (1)

Achieving quantum speedup with Grover search

= by using calls to $x \rightarrow G(x)=1$ or $G(x)=0$ in quantum superposition,
finds the M values of x satisfying $G(x)=1$, among 2^n, in runtime $\sqrt{2^n/M}$

Cutting runtime to $\sqrt{T}$ with Quantum Search (1)

Achieving quantum speedup with Grover search

= by using calls to $\mathrm{x} \rightarrow \mathrm{G(x)=1}$ or $\mathrm{G(x)=0}$ in quantum superposition, finds the M values of x satisfying $\mathrm{G(x)=1,}$ among 2^{n}, in runtime $\sqrt{2^{\mathrm{n}}/\mathrm{M}}$

- **brute force :** $\mathrm{G(x)=1}$ iff x is a k-SAT solution → $\mathrm{T}_{new} = 2^{\,0.5\,\mathrm{n}}$

Cutting runtime to $\sqrt{T}$ with Quantum Search (1)

Achieving quantum speedup with Grover search

= by using calls to $\mathrm{x} \rightarrow \mathrm{G(x)=1}$ or $\mathrm{G(x)=0}$ in quantum superposition,
finds the M values of x satisfying $\mathrm{G(x)=1,}$ among 2^{n}, in runtime $\sqrt{2^{\mathrm{n}}/\mathrm{M}}$

- **brute force :** $\mathrm{G(x)=1}$ iff x is a k-SAT solution → $\mathrm{T}_{new} = 2^{\,0.5\,\mathrm{n}}$
- **Schoening,… :** $\mathrm{G(x)=1}$ iff local search from x yields k-SAT solution → $\mathrm{T}_{new} = 2^{\,0.21\,\mathrm{n}}$

Cutting runtime to $\sqrt{T}$ with Quantum Search (1)

Achieving quantum speedup with Grover search

= by using calls to $\mathrm{x} \rightarrow \mathrm{G(x)=1}$ or $\mathrm{G(x)=0}$ in quantum superposition, finds the M values of x satisfying $\mathrm{G(x)=1,}$ among 2^{n}, in runtime $\sqrt{2^{\mathrm{n}}/\mathrm{M}}$

- **brute force :** $\mathrm{G(x)=1}$ iff x is a k-SAT solution → $\mathrm{T}_{new} = 2^{\,0.5\,\mathrm{n}}$
- **Schoening,… :** $\mathrm{G(x)=1}$ iff local search from x yields k-SAT solution → $\mathrm{T}_{new} = 2^{\,0.21\,\mathrm{n}}$
- **backtracking trees:** Grover-inspired quantum walk procedure → $\mathrm{T}_{new} \sim \sqrt{T}$

Cutting runtime to $\sqrt{T}$ with Quantum Search (2)

Limited power of Grover search

= proven optimal speedup, when using calls to $\mathrm{x} \rightarrow \mathrm{G(x)=1}$ or $\mathrm{G(x)=0}$

Cutting runtime to $\sqrt{T}$ with Quantum Search (2)

Limited power of Grover search

= proven optimal speedup, when using calls to $\mathrm{x} \rightarrow \mathrm{G(x)}=1$ or $\mathrm{G(x)}=0$

k-SAT cost functions could have more **structure**

e.g. $\mathrm{G(x)}$ = number of unsatisfied clauses

Open question: quantum algebraic acceleration exploiting this?

Cutting runtime to $\sqrt{T}$ with Quantum Search (2)

Limited power of Grover search

!! naïve nesting, not giving a speedup:

Cutting runtime to $\sqrt{T}$ with Quantum Search (2)

Limited power of Grover search

!! naïve nesting, not giving a speedup:

1. Grover search for satisfying only clause 1

Cutting runtime to $\sqrt{T}$ with Quantum Search (2)

Limited power of Grover search

!! naïve nesting, not giving a speedup:

1. Grover search for satisfying only clause 1
2. Among those, Grover search for satisfying clause 2

Cutting runtime to $\sqrt{T}$ with Quantum Search (2)

Limited power of Grover search

!! naïve nesting, not giving a speedup:

1. Grover search for satisfying only clause 1
2. Among those, Grover search for satisfying clause 2
3. …

Cutting runtime to $\sqrt{T}$ with Quantum Search (2)

Limited power of Grover search

!! naïve nesting, not giving a speedup:

1. Grover search for satisfying only clause 1
2. Among those, Grover search for satisfying clause 2
3. …

Hidden in this procedure: rising cost of uncomputing

Outline

I. Optimization / classical problem

II. Quantum annealing methods

standard quantum annealing

new quantum annealing

III. Harnessing quantum dissipation

The Quantum Annealing method

Idea: « drag along the eigenstate »

The Quantum Annealing method

$H_0 = \sum_x G(x)\ |x><x|$

(cost function)

in computational basis {x}

Idea: « drag along the eigenstate »

The Quantum Annealing method

$H_0 = \sum_x G(x)\ |x><x|$

(cost function)

in computational basis {x}

$H_1 = \sum_j \ I \otimes |+_j><+_j|$

has obvious ground state

$|\psi>$ = |--...->

Idea: « drag along the eigenstate »

The Quantum Annealing method

$H_0 = \sum_x G(x)\ |x><x|$

(cost function)

in computational basis {x}

$H_1 = \sum_j \ I \otimes |+_j><+_j|$

has obvious ground state

$|\psi>$ = |--...->

Idea: « drag along the eigenstate »

1. Initialize in $|\psi> = |$--...->
2. Propagate $\frac{d}{dt}|\psi> = -\,i\,H(t)\,|\psi>$

   where $H(t)$ slowly varies from H_1 to H_0 ;
3. Measure in computational basis

The Quantum Annealing method

Adiabatic propagation theorem:

$|\psi(t)>$ remains guaranteed close to the ground state of $H(t)$ if

$$\left\| \frac{d}{dt} H \right\| \ll \Delta^2$$

with $\Delta(t)$ the spectal gap of H(t)

1. Initialize in $|\psi> = |\text{--...->}$
2. Propagate $\frac{d}{dt}|\psi> = -i\, H(t)\, |\psi>$

   where $H(t)$ slowly varies from H_1 to H_0 ;
3. Measure in computational basis

The Quantum Annealing method

Adiabatic propagation theorem:

$|\psi(t)>$ remains guaranteed close to the ground state of $H(t)$ if

$$\left\| \frac{d}{dt} H \right\| \ll \Delta^2$$

with $\Delta(t)$ the spectal gap of H(t)

→ runtime $T \sim \|H'\| / \Delta^2$

H_1 induces avoided crossings

1. Initialize in $|\psi> = |\text{--...->}$
2. Propagate $\frac{d}{dt}|\psi> = -i\, H(t)\, |\psi>$

   where $H(t)$ slowly varies from H_1 to H_0 ;
3. Measure in computational basis

k-SAT with standard Quantum Annealing

Key issue: maximize the gap $\Delta(t)$

k-SAT with standard Quantum Annealing

Key issue: maximize the gap $\Delta(t)$

- Grover-like cost: $G(x) = 0$ for k-SAT solution, otherwise $G(x) = 1$

  Same speedup as Grover, by refined analysis: $\Delta \sim 1/\sqrt{2^n}$ but $T \sim 1/\Delta$

k-SAT with standard Quantum Annealing

Key issue: maximize the gap $\Delta(t)$

- Grover-like cost: $G(x) = 0$ for k-SAT solution, otherwise $G(x) = 1$

  Same speedup as Grover, by refined analysis: $\Delta \sim 1/\sqrt{2^n}$ but $T \sim 1/\Delta$

  ! moreover, building this $G(x)$ from local interactions may be hard

k-SAT with standard Quantum Annealing

Key issue: maximize the gap $\Delta(t)$

- Grover-like cost: $G(x) = 0$ for k-SAT solution, otherwise $G(x) = 1$

  Same speedup as Grover, by refined analysis: $\Delta \sim 1/\sqrt{2^n}$ but $T \sim 1/\Delta$

- Local cost: $G(x) =$ number of clauses violated by x $= \sum_m P_{(\text{failing clause m})}(x)$

  In principle contains more structure; but no conclusive (refined) $\Delta(t)$

k-SAT with standard Quantum Annealing

Key issue: maximize the gap $\Delta(t)$

- **Grover-like cost:** $G(x) = 0$ for k-SAT solution, otherwise $G(x) = 1$

  Same speedup as Grover, by refined analysis: $\Delta \sim 1/\sqrt{2^n}$ but $T \sim 1/\Delta$

- **Local cost:** $G(x) =$ number of clauses violated by x $= \sum_m P_{(\text{failing clause m})}(x)$

  In principle contains more structure; but no conclusive (refined) $\Delta(t)$

Example: for $\{x_1 \text{ or } \overline{x}_2 \text{ or } x_4\}$, take

$P_m = |1><1| \otimes |0><0| \otimes I \otimes |1><1| \otimes_{n-4} I$

k-SAT with standard Quantum Annealing

Key issue: maximize the gap $\Delta(t)$

- **Grover-like cost:** $G(x) = 0$ for k-SAT solution, otherwise $G(x) = 1$

  Same speedup as Grover, by refined analysis: $\Delta \sim 1/\sqrt{2^n}$ but $T \sim 1/\Delta$

- **Local cost:** $G(x) =$ number of clauses violated by x $= \sum_m P_{(\text{failing clause m})}(x)$

  In principle contains more structure; but no conclusive (refined) $\Delta(t)$

  ! also more natural implementation, local couplings

k-SAT with standard Quantum Annealing

Key issue: maximize the gap $\Delta(t)$

- **Grover-like cost:** $G(x) = 0$ for k-SAT solution, otherwise $G(x) = 1$

  Same speedup as Grover, by refined analysis: $\Delta \sim 1/\sqrt{2^n}$ but $T \sim 1/\Delta$

- **Local cost:** $G(x) =$ number of clauses violated by x $= \sum_m P_{(\text{failing clause m})}(x)$

  In principle contains more structure; but no conclusive (refined) $\Delta(t)$

- Gap amplification (details later): 1 ancilla level per clause gives $\Delta_{new} \sim \sqrt{\Delta_{old}}$ but now with only guarantee $T \sim \frac{1}{\Delta_{new}^2} \sim 1/\Delta_{old}$

Outline

I. Optimization / classical problem

Tutorial

II. Quantum annealing methods

standard quantum annealing

new quantum annealing

New developments

III. Harnessing quantum dissipation

A different take on H(t) in Quantum Annealing

Idea: « drag along the eigenstate »

1. Initialize in $|\psi> = |\text{--...->}$
2. Propagate $\frac{d}{dt}|\psi> = -\,\mathrm{i}\,\mathrm{H(t)}\,|\psi>$

   as $\mathrm{H(t)}$ slowly varies from $\mathrm{H}_{s=0}$ to $\mathrm{H}_{s=1}$;
3. Measure in computational basis

A different take on H(t) in Quantum Annealing

Idea: « drag along the eigenstate »

1. Initialize in $|\psi> = |\text{--...->}$
2. Propagate $\frac{d}{dt}|\psi> = -\,\mathrm{i}\,\mathrm{H(t)}\,|\psi>$

   as $\mathrm{H(t)}$ slowly varies from $\mathrm{H}_{s=0}$ to $\mathrm{H}_{s=1}$;
3. Measure in computational basis

A different take on H(t) in Quantum Annealing

$H(t) = \sum_x G(x)\ |x(s)><x(s)|$

$\{\ |x(1)>\ \}$: computational basis

Idea: « drag along the eigenstate »

1. Initialize in $|\psi> = |\text{--...->}$
2. Propagate $\frac{d}{dt}|\psi> = -\,i\,H(t)\ |\psi>$

   as $H(t)$ slowly varies from $H_{s=0}$ to $H_{s=1}$;
3. Measure in computational basis

A different take on H(t) in Quantum Annealing

$H(t) = \sum_x G(x)\ |x(s)><x(s)|$

$\{\ |x(1)>\ \}$: computational basis

$\{\ |x(0)> = |{+}{+}\ldots{+}>$ for all $x\ \}$:

H(0) : trivial Hamiltonian,
highly degenerate

Idea: « drag along the eigenstate »

1. Initialize in $|\psi> = |{-}{-}\ldots{-}>$
2. Propagate $\frac{d}{dt}|\psi> = -\,i\,H(t)\,|\psi>$

   as $H(t)$ slowly varies from $H_{s=0}$ to $H_{s=1}$;
3. Measure in computational basis

A different take on H(t) in Quantum Annealing

$H(t) = \sum_x G(x)\ |x(s)><x(s)|$

{ $|x(1)>$ } : computational basis

{ $|x(0)> = |{+}{+}\ldots{+}>$ for all x }

$|x(s)> = \cos(s\pi/4)\ |+> \pm \sin(s\pi/4)\ |->$

on each qubit

$|0_{logic}>$

$|1_{logic}>$

almost degenerate for $s \ll 1$

Idea: « drag along the eigenstate »

1. Initialize in $|\psi> = |{-}{-}\ldots{-}>$
2. Propagate $\frac{d}{dt}|\psi> = -\,i\,H(t)\ |\psi>$

   as $H(t)$ slowly varies from $H_{s=0}$ to $H_{s=1}$;
3. Measure in computational basis

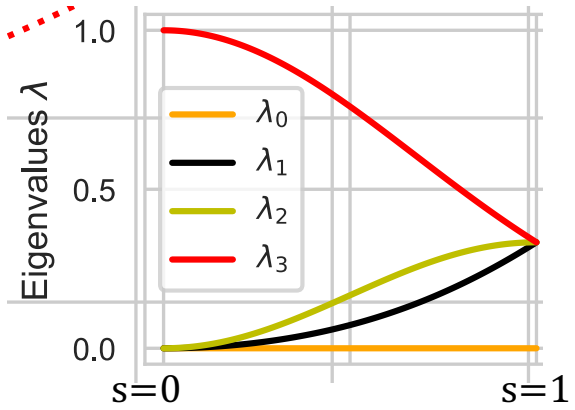
The new viewpoint drastically simplifies analysis

Start analysis of $|\psi(t)>$ by jumping (= quenching) to $s_1 \ll 1$

The new viewpoint drastically simplifies analysis

Start analysis of $|\psi(t)>$ by jumping (= quenching) to $s_1 \ll 1$

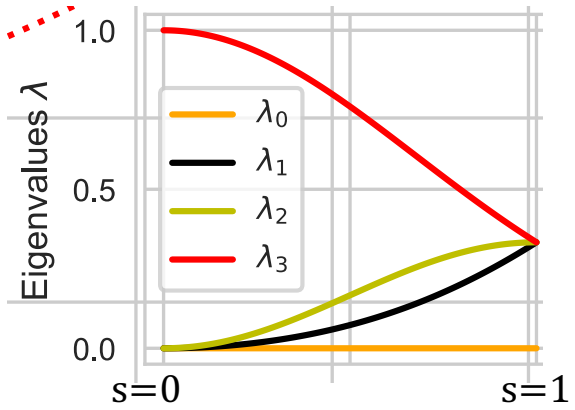
- $\Delta(s)$ monotonically increases from $s=0$ to $s=1$ $\rightarrow$ $\Delta(s_1)$ is bottleneck



The new viewpoint drastically simplifies analysis

Start analysis of $|\psi(t)>$ by jumping (= quenching) to $s_1 \ll 1$

- $\Delta(s)$ monotonically increases from $s=0$ to $s=1$ $\rightarrow$ $\Delta(s_1)$ is bottleneck
- Initial loss of fidelity: $|<\psi_{\text{ground}}(s_1)\,|\,\psi(0)>|$



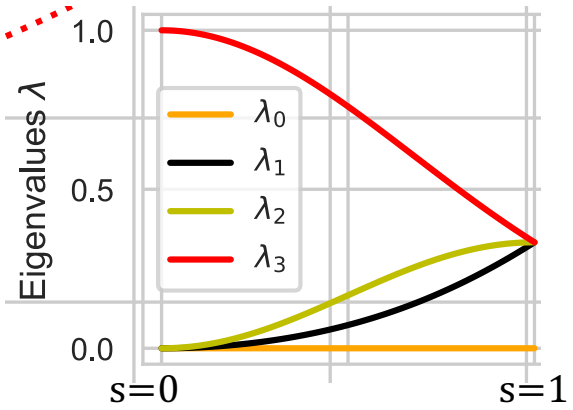
The new viewpoint drastically simplifies analysis

Start analysis of $|\psi(t)>$ by jumping (= quenching) to $s_1 \ll 1$

- $\Delta(s)$ monotonically increases from $s=0$ to $s=1$ $\rightarrow$ $\Delta(s_1)$ is bottleneck
- Initial loss of fidelity: $| < \psi_{\text{ground}}(s_1) \,|\, \psi(0)> |$

  $= |$ <011 … 101 $_{\text{logic}}(s_1) \,|$ --…-> $|$ $\sim \cos(s_1)^n$

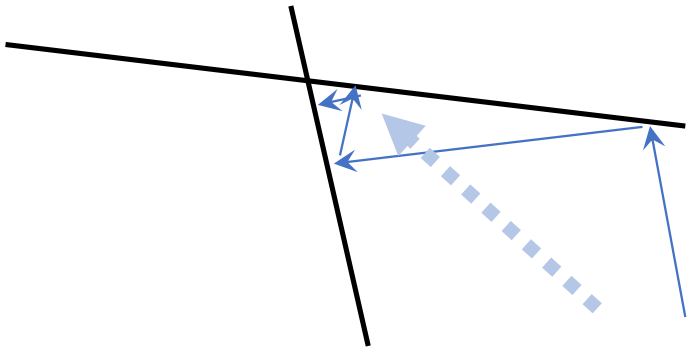
Runtime T follows this tradeoff in the choice of s_1

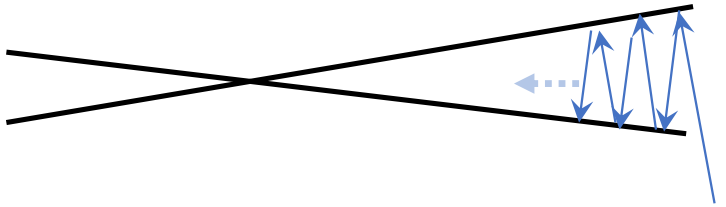


(weak) relation to alternated projections method

To find intersection between subspaces: project alternatively onto each

- Almost parallel subspaces imply slow convergence

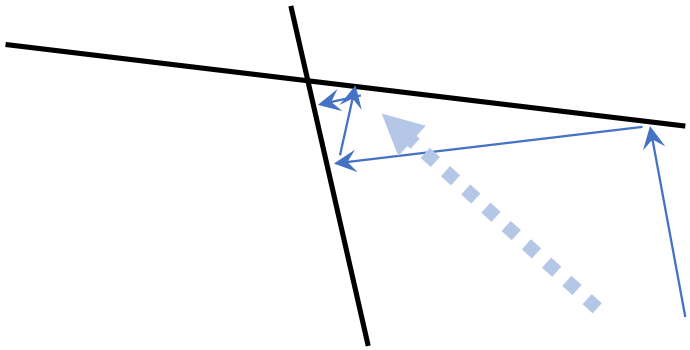


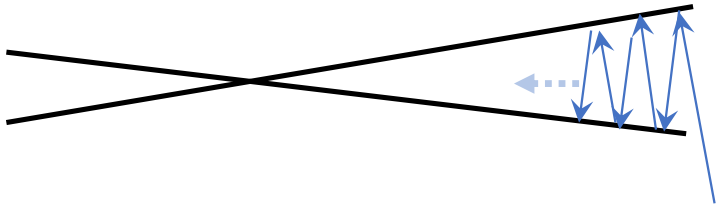


(weak) relation to alternated projections method

To find intersection between subspaces: project alternatively onto each

- Almost parallel subspaces imply slow convergence
- Here, track slowly opening intersection, until it encodes k-SAT solution

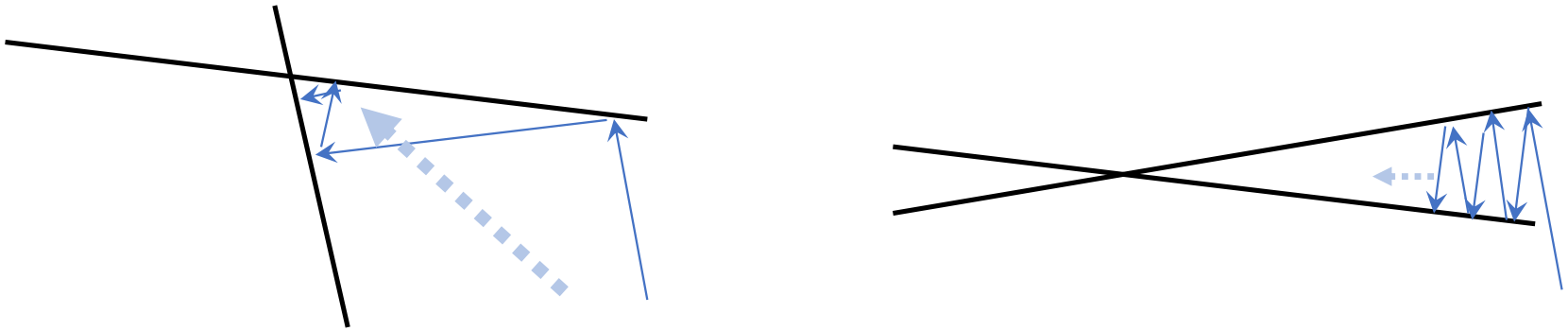




(weak) relation to alternated projections method

Detectability Lemma: Let $P_1, P_2, \ldots, P_m$ quasi-local with $\leq g$ pairwise overlaps

then $\| \prod_k (I - P_k) |\psi_\perp\rangle \|^2 \leq 1 / (1 + \Delta/g^2)$



Recap: k-SAT annealing with t-dependent basis

$$|x(t)> = \cos(s_t\,\pi/4)\,|+> \pm \sin(s_t\,\pi/4)\,|->$$ on each qubit

true

false

Recap: k-SAT annealing with t-dependent basis

$$|x(t)> = \cos(s_t\,\pi/4)\,|+> \pm \sin(s_t\,\pi/4)\,|->$$ on each qubit

true

false

$$H = \sum_m P_{(\text{failing clause } m)}(x(t))$$

Example: for $\{x_1 \text{ or } \overline{x_2} \text{ or } x_4\}$, take

$$P_m = |1><1| \otimes |0><0| \otimes I \otimes |1><1| \otimes_{n-4} I$$

Recap: k-SAT annealing with t-dependent basis

$$|x(t)> = \cos(s_t\,\pi/4)\,|+> \pm \sin(s_t\,\pi/4)\,|->$$ on each qubit

true / false

$$H = \sum_m P_{(\text{failing clause } m)}(x(t))$$

- **Initial loss of fidelity**: sub-exponential with $t_1 \sim 1/\sqrt{n}$
- **With gap amplification**: runtime $T \sim 1/\sqrt{\Delta(s_1)}^2 \sim 1/\Delta(s_1)$
- **Bound on $\Delta(s_1)$**: still work in progress…

Outline

I. Optimization / classical problem

II. Quantum annealing methods

Tutorial

III. Harnessing quantum dissipation

quantum Zeno dragging

New developments

quantum Zeno dragging

= measure in slowly varying basis (quantum Zeno effect)

quantum Zeno dragging

= measure in slowly varying basis (quantum Zeno effect)

For k-SAT: measure the $\mathrm{P}_{(\text{failing clause m})}(\mathrm{x}(\mathrm{t}))$

fast in random order

quantum Zeno dragging

= measure in slowly varying basis (quantum Zeno effect)

For k-SAT: measure the $\mathrm{P}_{(\text{failing clause m})}(\mathrm{x}(\mathrm{t}))$

fast in random order

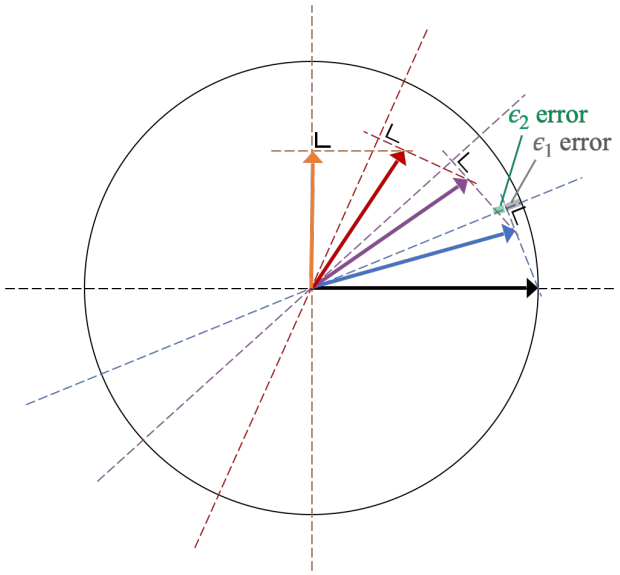
Equivalent Lindbladian:

$$\frac{d}{dt}\rho = \frac{1}{\mathrm{m}} \sum_{\mathrm{m}} \mathrm{P}_{\mathrm{m}}\, \rho\, \mathrm{P}_{\mathrm{m}} - \frac{1}{2}\mathrm{P}_{\mathrm{m}}\, \rho - \frac{1}{2}\, \rho\, \mathrm{P}_{\mathrm{m}}$$

Straightforward analysis with discrete steps $\mathbf{x(t)}$

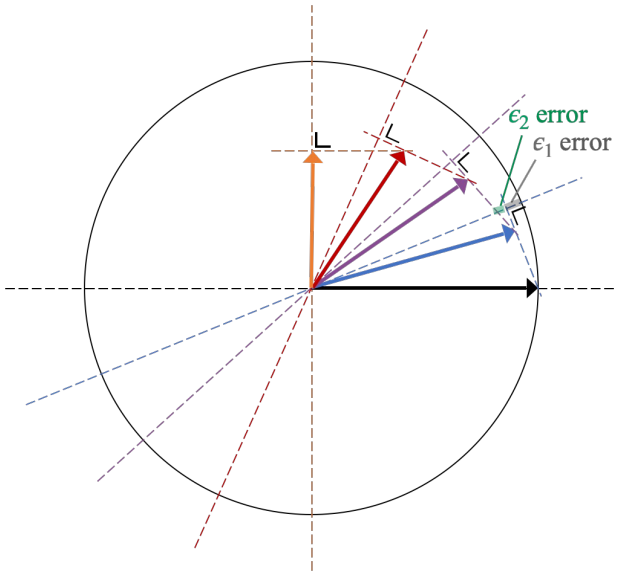
Straightforward analysis with discrete steps $\mathbf{x(t)}$

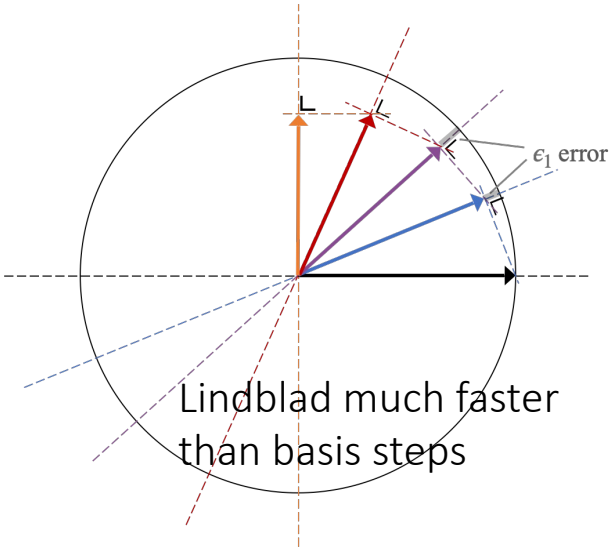
- Two types of error: - fidelity between old and new basis
  - incomplete projection onto basis



Straightforward analysis with discrete steps $\mathbf{x(t)}$

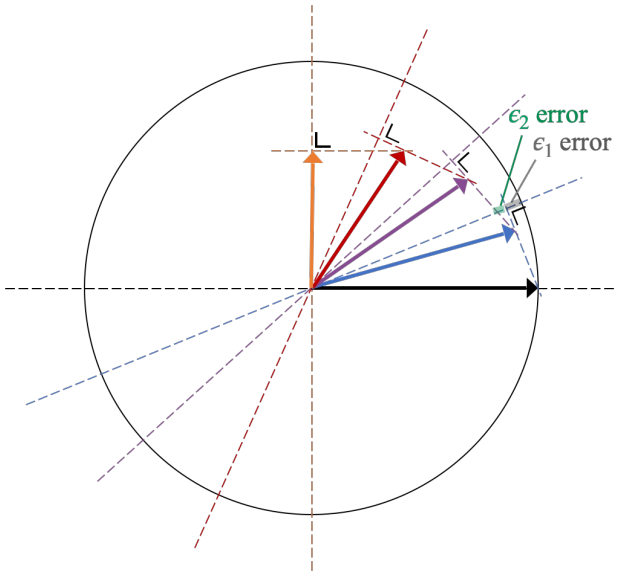
- Two types of error: - fidelity between old and new basis
  - incomplete projection onto basis

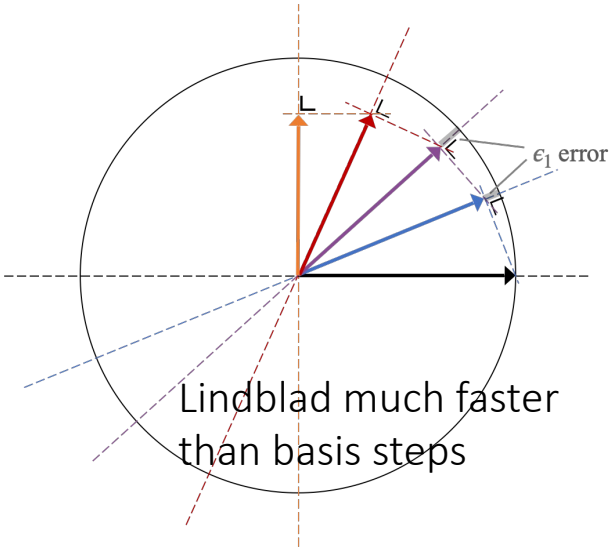


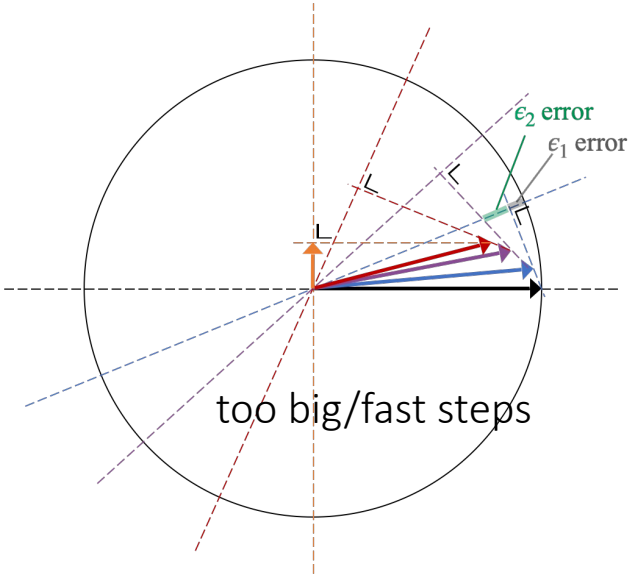


Straightforward analysis with discrete steps $\mathbf{x(t)}$

- Two types of error: - fidelity between old and new basis
  - incomplete projection onto basis



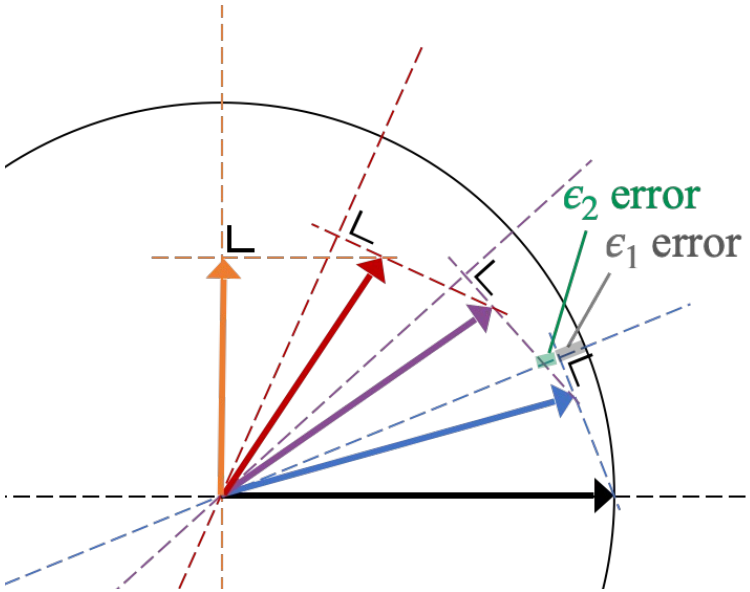




Bottleneck: still spectral gap of H_0

Lindbladian convergence properties at fixed x(s) :

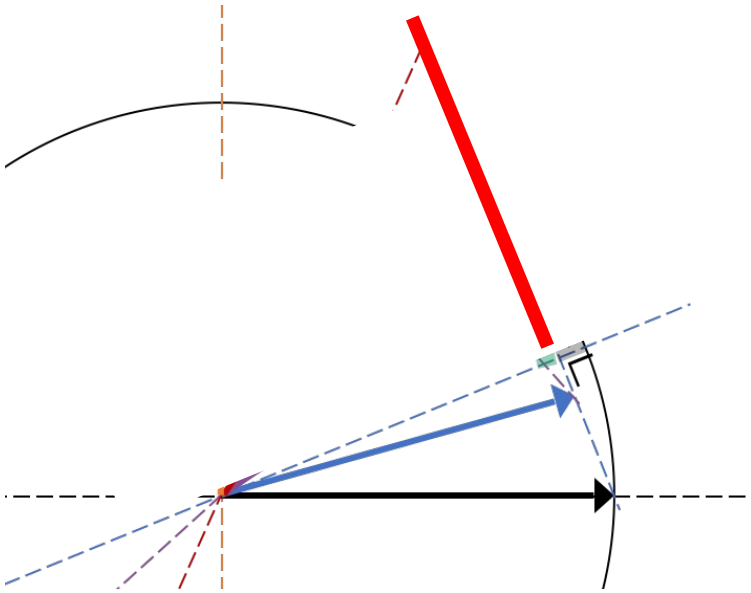
- Fidelity to solution space = constant



Bottleneck: still spectral gap of H_0

Lindbladian convergence properties at fixed $x(s)$:

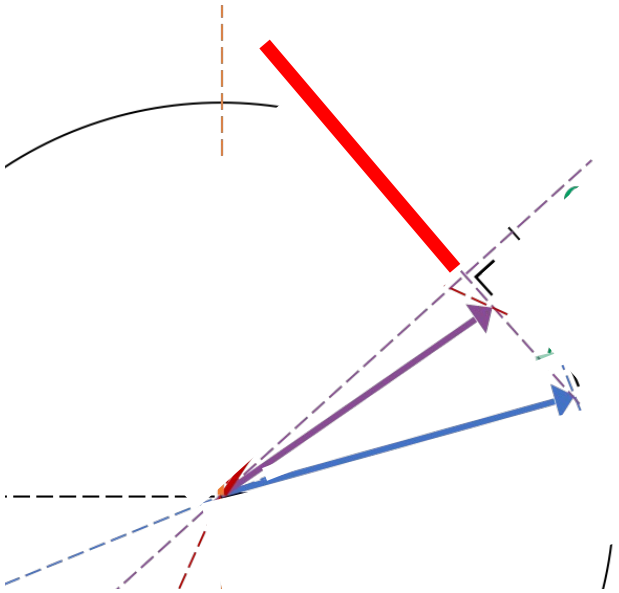
- Fidelity to solution space = constant



Bottleneck: still spectral gap of H_0

Lindbladian convergence properties at fixed $x(s)$:

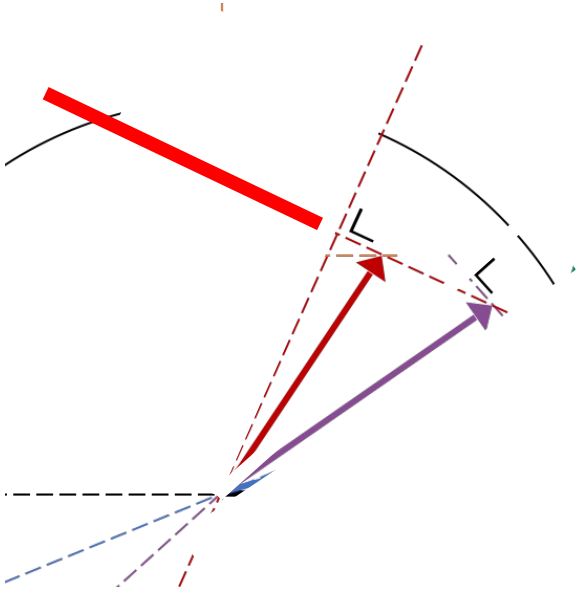
- Fidelity to solution space = constant



Bottleneck: still spectral gap of H_0

Lindbladian convergence properties at fixed $x(s)$:

- Fidelity to solution space = constant



Bottleneck: still spectral gap of H_0

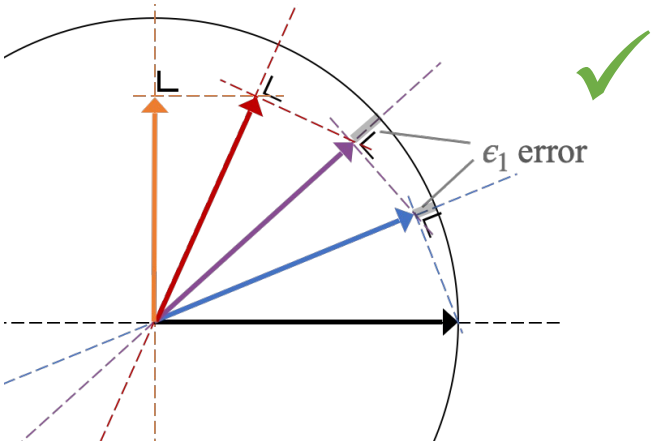
Lindbladian convergence properties at fixed $x(s)$:

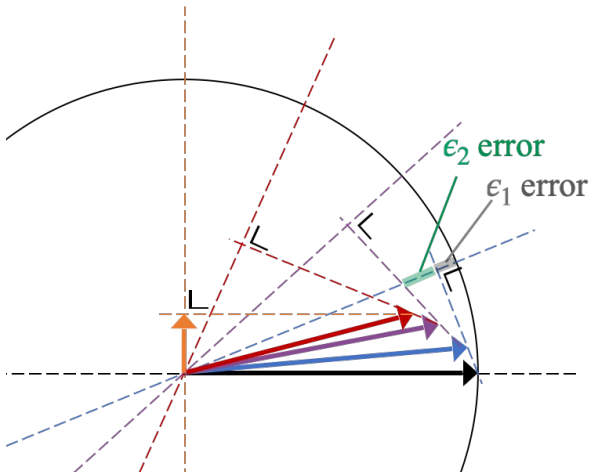
- Fidelity to solution space = constant
- Task = suppress coherences {solution} vs. {orthogonal Hilbert subspace}

Bottleneck: still spectral gap of H_0

Lindbladian convergence properties at fixed x(s) :

- Fidelity to solution space = constant
- Task = suppress coherences {solution} vs. {orthogonal Hilbert subspace}



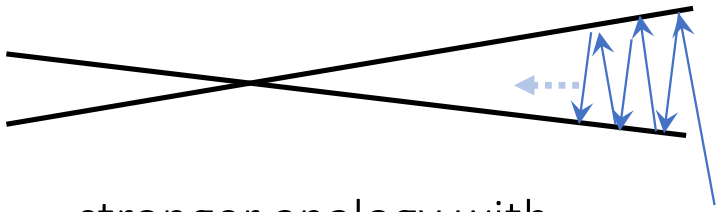


Bottleneck: still spectral gap of H_0

Lindbladian convergence properties at fixed $x(s)$:

- Fidelity to solution space = constant
- Task = suppress coherences {solution} vs. {orthogonal Hilbert subspace}

!! decay rate = $\Delta(s_1)$ of H_0



stronger analogy with
alternated projections method

Bottleneck: still spectral gap of H_0

Lindbladian convergence properties at fixed $x(s)$:

- Fidelity to solution space = constant
- Task = suppress coherences {solution} vs. {orthogonal Hilbert subspace}

!! decay rate = $\Delta(s_1)$ of H_0

runtime $T \sim 1/\Delta(t_1)$

« As fast as amplified annealing, but with simpler **open quantum system** »

Outline

I. Optimization / classical problem

II. Quantum annealing methods

Tutorial

New developments

III. Harnessing quantum dissipation

quantum Zeno dragging

[Last part of the talk still unpublished — end of public slides]

Future work

- Spectral gap: estimation per k-SAT instance
  is just a bound: go beyond (with Artem Mamichev)

- Clause-by-clause scheduling

- Introducing feedback: algorithmic (e.g. Schoening)
  for Quantum Error Correction / Fault-tolerance

Co-authors

U.Berkeley

Tathagata Karmakar

Philippe Lewalle

Thilo Scharnhorst

Yipei Zhang

Birgitta Whaley (PI)

inria / ENS Paris

Artem Mamichev

(Pierre Rouchon)