Linear time algorithm for Quantum 2SAT

Aarthi Sundaram

Center for Quantum Technologies, National University of Singapore

ICALP 2016

Rome, Italy

This work done jointly with:

Itai Arad	Miklos Santha	Shengyu Zhang
Centre for Quant. Tech.	CNRS, Univ. Paris Diderot	Dept. of Comp. Sci.
Nat. Univ. of Singapore	CQT, Nat. Univ. of Singapore	Chinese Univ. of Hong Kong

Background

Classical 2SAT

A classical 2SAT instance Φ is a Boolean formula defined on

- a set of n variables: $\{x_1, \ldots, x_n\}$
- as a conjunction of m clauses: $\{C_1, \ldots, C_n\}$ where
- each clause is an `OR` of at most 2 literals (i.e. x_i and $\overline{x}_i$)

Goal: Find an assignment to the variables so that Φ evaluates to `true`.

Example (2SAT)

An instance: $\Phi = (x_1 \vee x_2) \wedge (\overline{x}_1 \vee \overline{x}_2) \wedge (x_1 \vee \overline{x}_4) \wedge (x_4) \wedge (x_2 \vee x_3)$
Satisfying assignment: $a = 1011$

Classical 2SAT

A classical 2SAT instance Φ is a Boolean formula defined on

- a set of n variables: $\{x_1, \ldots, x_n\}$
- as a conjunction of m clauses: $\{C_1, \ldots, C_n\}$ where
- each clause is an `OR` of at most 2 literals (i.e. x_i and $\overline{x}_i$)

Goal: Find an assignment to the variables so that Φ evaluates to `true`.

Example (2SAT)

An instance: $\Phi = (x_1 \vee x_2) \wedge (\overline{x}_1 \vee \overline{x}_2) \wedge (x_1 \vee \overline{x}_4) \wedge (x_4) \wedge (x_2 \vee x_3)$
Satisfying assignment: $a = 1011$

Algorithms for 2SAT

1. Even, Itai & Shamir (1976): A backtracking, resolution based search of possible assignments.
2. Apsvall, Plass & Tarjan (1979): Finds strongly connected components in the implication graph of the instance.

Both algorithms have an optimal $O(n+m)$ running time.

Quantum 2SAT (2-QSAT)

A quantum 2SAT instance $\mathcal{H}$ is a 2-local Hamiltonian defined on

- n qubits: $\{x_1, \ldots, x_n\}$
- as a sum of m local terms: $\mathcal{H} = \sum_{uv} \Pi_{uv}$ where
- each Π_{uv} is a projector acting non-trivially on qubits (u, v);

Example (Q2SAT)

A 2-QSAT *instance:* $\mathcal{H} = \Pi_{12} + \Pi_{23} + \Pi_{34}$ *with*

$\Pi_{12} = |00\rangle\langle 00| + |11\rangle\langle 11|;$ $\Pi_{34} = |\Psi^-\rangle\langle\Psi^-|;$ $\Pi_{23} = |01\rangle\langle 01|;$

Quantum 2SAT (2-QSAT)

A quantum 2SAT instance $\mathcal{H}$ is a 2-local Hamiltonian defined on

- n qubits: $\{x_1, \ldots, x_n\}$
- as a sum of m local terms: $\mathcal{H} = \sum_{uv} \Pi_{uv}$ where
- each Π_{uv} is a projector acting non-trivially on qubits (u, v);

Example (Q2SAT)

A 2-QSAT *instance:* $\mathcal{H} = \Pi_{12} + \Pi_{23} + \Pi_{34}$ *with*

$\Pi_{12} = |00\rangle\langle 00| + |11\rangle\langle 11|;$ $\Pi_{34} = |\Psi^{-}\rangle\langle\Psi^{-}|;$ $\Pi_{23} = |01\rangle\langle 01|;$

entangled

Quantum 2SAT (2-QSAT)

A quantum 2SAT instance $\mathcal{H}$ is a 2-local Hamiltonian defined on

- n qubits: $\{x_1, \ldots, x_n\}$
- as a sum of m local terms: $\mathcal{H} = \sum_{uv} \Pi_{uv}$ where
- each Π_{uv} is a projector acting non-trivially on qubits (u, v);

Example (Q2SAT)

A 2-QSAT *instance:* $\mathcal{H} = \Pi_{12} + \Pi_{23} + \Pi_{34}$ *with*

$\Pi_{12} = |00\rangle\langle 00| + |11\rangle\langle 11|$; $\Pi_{34} = |\Psi^-\rangle\langle\Psi^-|$; $\Pi_{23} = |01\rangle\langle 01|$;

entangled *product*

Quantum 2SAT (2-QSAT)

A quantum 2SAT instance $\mathcal{H}$ is a 2-local Hamiltonian defined on

- n qubits: $\{x_1, \dots, x_n\}$
- as a sum of m local terms: $\mathcal{H} = \sum_{uv} \Pi_{uv}$ where
- each Π_{uv} is a projector acting non-trivially on qubits (u, v);
- the smallest eigenvalue of $\mathcal{H}$ is its ground energy and
- the corresponding eigenvector $|\psi\rangle$ is the ground state of $\mathcal{H}$.

Goal: Given $\mathcal{H}$, output a ground state if the ground energy is 0 or "Unsatisfiable" otherwise.

Example (Q2SAT)

A 2-QSAT *instance:* $\mathcal{H} = \Pi_{12} + \Pi_{23} + \Pi_{34}$ *with*

$\Pi_{12} = \|00\rangle\langle 00\| + \|11\rangle\langle 11\|;$	$\Pi_{34} = \|\Psi^-\rangle\langle\Psi^-\|;$	$\Pi_{23} = \|01\rangle\langle 01\|;$
	entangled	*product*

Solving 2-QSAT

Prior Work

An $O(n^4)$ 2-QSAT algorithm by Bravyi (2006) based on finding the transitive closure of a directed graph.

- For $k \geq 3$, k-QSAT is QMA_1-complete

Solving 2-QSAT

Prior Work

An $O(n^4)$ 2-QSAT algorithm by Bravyi (2006) based on finding the transitive closure of a directed graph.

- For $k \geq 3$, k-QSAT is QMA_1-complete

Our Result

An optimal $O(n+m)$ time algorithms for 2-QSAT
(w.r.t $\#$ of operations over complex numbers).

Solving 2-QSAT

Prior Work

An $O(n^4)$ 2-QSAT algorithm by Bravyi (2006) based on finding the transitive closure of a directed graph.

- For $k \geq 3$, k-QSAT is QMA_1-complete

Our Result

An optimal $O(n+m)$ time algorithms for 2-QSAT
(w.r.t $\#$ of operations over complex numbers).

- Quantum variant of the EIS Algorithm
  - Infers a qubit assignment and propagates it throughout the system.

Solving 2-QSAT

Prior Work

An $O(n^4)$ 2-QSAT algorithm by Bravyi (2006) based on finding the transitive closure of a directed graph.

- For $k \geq 3$, k-QSAT is QMA_1-complete

Our Result

An optimal $O(n+m)$ time algorithms for 2-QSAT
(w.r.t $\#$ of operations over complex numbers).

- Quantum variant of the EIS Algorithm
  - Infers a qubit assignment and propagates it throughout the system.

Related Work

- Quantum analogue of the APT algorithm [dBG]
  - Uses the notion of transfer matrices to mirror the implications in Boolean Formulae.

Improvements in run-time

Bravyi's algorithm uses the following approach:

- For every qubit triple (i, j, k), if there is a constraint on (i, j) and (j, k), add an implied constraint acting on (i, k).
- Takes $O(n^3)$ time to add all implied constraints
- Requires globally affecting the instance and manipulating it.

Improvements in run-time

Bravyi's algorithm uses the following approach:

- For every qubit triple (i, j, k), if there is a constraint on (i, j) and (j, k), add an implied constraint acting on (i, k).
- Takes $O(n^3)$ time to add all implied constraints
- Requires globally affecting the instance and manipulating it.

The linear time algorithms approach an instance by:

- Analyzing only a part of the instance and manipulating it with local operations.
- Local sections of the instance on being solved are decoupled from the rest of instance.
- Governed by graph traversals that can be executed in linear time.

Algorithm Building Blocks

Preliminaries

Assume WLOG that $\mathcal{H}$ contains rank-1 and rank-3 projectors only.

Preliminaries

Assume WLOG that $\mathcal{H}$ contains rank-1 and rank-3 projectors only.

The constraint graph, $G(\mathcal{H})$ is defined as having

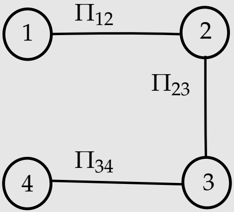
- n vertices representing each qubit and
- labeled edges Π_{ij} between (i, j) for each term in $\mathcal{H}$

Example (Constraint Graph)

$\mathcal{H} = \Pi_{12} + \Pi_{23} + \Pi_{34}$ *with*

$\Pi_{12} = |00\rangle\langle 00| + |11\rangle\langle 11| + |\Psi^-\rangle\langle\Psi^-|$

$\Pi_{23} = |11\rangle\langle 11|; \qquad \Pi_{34} = |\Phi^+\rangle\langle\Phi^+|$



Preliminaries

Assume WLOG that $\mathcal{H}$ contains rank-1 and rank-3 projectors only.

The constraint graph, $G(\mathcal{H})$ is defined as having

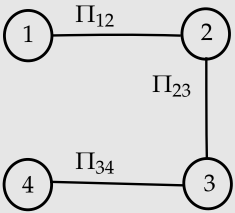
- n vertices representing each qubit and
- labeled edges Π_{ij} between (i, j) for each term in $\mathcal{H}$

Example (Constraint Graph)

$\mathcal{H} = \Pi_{12} + \Pi_{23} + \Pi_{34}$ *with*

$\Pi_{12} = |00\rangle\langle 00| + |11\rangle\langle 11| + |\Psi^-\rangle\langle\Psi^-|$

$\Pi_{23} = |11\rangle\langle 11|; \qquad \Pi_{34} = |\Phi^+\rangle\langle\Phi^+|$



Theorem (Product State Theorem [CCD+11,ASSZ15])

Any satisfiable 2-QSAT *instance has a ground state which is a tensor product of one qubit and two-qubit states, where two-qubit states only appear in the support of rank-3 projectors.*

Preliminaries

Assume WLOG that $\mathcal{H}$ contains rank-1 and rank-3 projectors only.

The constraint graph, $G(\mathcal{H})$ is defined as having

- n vertices representing each qubit and
- labeled edges Π_{ij} between (i, j) for each term in $\mathcal{H}$

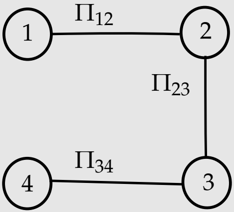
Example (Constraint Graph)

$$\mathcal{H} = \Pi_{12} + \Pi_{23} + \Pi_{34} \textit{ with}$$

$$\Pi_{12} = |00\rangle\langle 00| + |11\rangle\langle 11| + |\Psi^-\rangle\langle\Psi^-|$$

$$\Pi_{23} = |11\rangle\langle 11|; \qquad \Pi_{34} = |\Phi^+\rangle\langle\Phi^+|$$

$$\textit{Ground State} = |\Psi^+\rangle_{12} \otimes |0\rangle_3 \otimes |1\rangle_4$$



Theorem (Product State Theorem [CCD+11,ASSZ15])

Any satisfiable 2-QSAT *instance has a ground state which is a tensor product of one qubit and two-qubit states, where two-qubit states only appear in the support of rank-*3 *projectors.*

Propagation

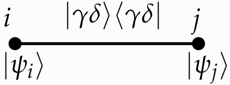
Definition (Propagation)

Let $\Pi_{ij} = |\psi\rangle\langle\psi|$ *be a rank-*1 *projector and* $|\alpha\rangle$ *be the state assigned to* i*. Then,* Π_{ij} *propagates* $|\alpha\rangle$ *if, up to a phase, there exists a unique* 1*-qubit state* $|\beta\rangle$ *such that* $\langle\psi|(|\alpha\rangle_i \otimes |\beta\rangle_j) = 0$*.*

Propagation

Definition (Propagation)

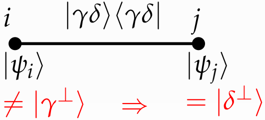
Let $\Pi_{ij} = |\psi\rangle\langle\psi|$ *be a rank-*1 *projector and* $|\alpha\rangle$ *be the state assigned to* i*. Then,* Π_{ij} *propagates* $|\alpha\rangle$ *if, up to a phase, there exists a unique* 1*-qubit state* $|\beta\rangle$ *such that* $\langle\psi|(|\alpha\rangle_i \otimes |\beta\rangle_j) = 0$*.*



Propagation

Definition (Propagation)

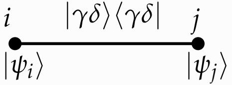
Let $\Pi_{ij} = |\psi\rangle\langle\psi|$ *be a rank-*1 *projector and* $|\alpha\rangle$ *be the state assigned to* i. *Then,* Π_{ij} *propagates* $|\alpha\rangle$ *if, up to a phase, there exists a unique* 1*-qubit state* $|\beta\rangle$ *such that* $\langle\psi|(|\alpha\rangle_i \otimes |\beta\rangle_j) = 0$.



Propagation

Definition (Propagation)

Let $\Pi_{ij} = |\psi\rangle\langle\psi|$ *be a rank-*1 *projector and* $|\alpha\rangle$ *be the state assigned to* i*. Then,* Π_{ij} *propagates* $|\alpha\rangle$ *if, up to a phase, there exists a unique* 1*-qubit state* $|\beta\rangle$ *such that* $\langle\psi|(|\alpha\rangle_i \otimes |\beta\rangle_j) = 0$.



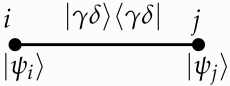
$\neq |\gamma^{\perp}\rangle \quad \Rightarrow \quad = |\delta^{\perp}\rangle$

$= |\gamma^{\perp}\rangle \quad \not\Rightarrow$ No propagation

Propagation

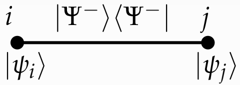
Definition (Propagation)

Let $\Pi_{ij} = |\psi\rangle\langle\psi|$ *be a rank-*1 *projector and* $|\alpha\rangle$ *be the state assigned to* i*. Then,* Π_{ij} *propagates* $|\alpha\rangle$ *if, up to a phase, there exists a unique* 1*-qubit state* $|\beta\rangle$ *such that* $\langle\psi|(|\alpha\rangle_i \otimes |\beta\rangle_j) = 0$*.*



$\neq |\gamma^\perp\rangle \quad \Rightarrow \quad = |\delta^\perp\rangle$

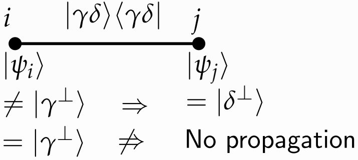
$= |\gamma^\perp\rangle \quad \not\Rightarrow \quad$ No propagation



Propagation

Definition (Propagation)

Let $\Pi_{ij} = |\psi\rangle\langle\psi|$ *be a rank-*1 *projector and* $|\alpha\rangle$ *be the state assigned to* i. *Then,* Π_{ij} *propagates* $|\alpha\rangle$ *if, up to a phase, there exists a unique* 1*-qubit state* $|\beta\rangle$ *such that* $\langle\psi|(|\alpha\rangle_i \otimes |\beta\rangle_j) = 0$.

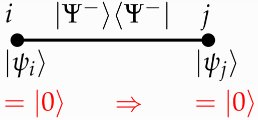


i $\quad |\gamma\delta\rangle\langle\gamma\delta| \quad$ j

$|\psi_i\rangle \qquad |\psi_j\rangle$

$\neq |\gamma^\perp\rangle \quad \Rightarrow \quad = |\delta^\perp\rangle$

$= |\gamma^\perp\rangle \quad \not\Rightarrow$ No propagation



i $\quad |\Psi^-\rangle\langle\Psi^-| \quad$ j

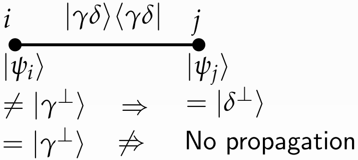
$|\psi_i\rangle \qquad |\psi_j\rangle$

$= |0\rangle \quad \Rightarrow \quad = |0\rangle$

Propagation

Definition (Propagation)

Let $\Pi_{ij} = |\psi\rangle\langle\psi|$ *be a rank-*1 *projector and* $|\alpha\rangle$ *be the state assigned to* i*. Then,* Π_{ij} *propagates* $|\alpha\rangle$ *if, up to a phase, there exists a unique* 1*-qubit state* $|\beta\rangle$ *such that* $\langle\psi|(|\alpha\rangle_i \otimes |\beta\rangle_j) = 0$*.*

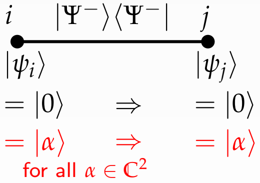


i $\quad |\gamma\delta\rangle\langle\gamma\delta|$ $\quad j$

$|\psi_i\rangle$ $\qquad$ $|\psi_j\rangle$

$\neq |\gamma^\perp\rangle \quad \Rightarrow \quad = |\delta^\perp\rangle$

$= |\gamma^\perp\rangle \quad \not\Rightarrow$ $\quad$ No propagation



i $\quad |\Psi^-\rangle\langle\Psi^-|$ $\quad j$

$|\psi_i\rangle$ $\qquad$ $|\psi_j\rangle$

$= |0\rangle \quad \Rightarrow \quad = |0\rangle$

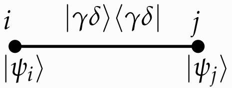
$= |\alpha\rangle \quad \Rightarrow \quad = |\alpha\rangle$

for all $\alpha \in \mathbb{C}^2$

Propagation

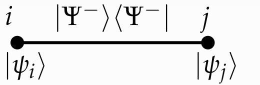
Definition (Propagation)

Let $\Pi_{ij} = |\psi\rangle\langle\psi|$ *be a rank-*1 *projector and* $|\alpha\rangle$ *be the state assigned to* i*. Then,* Π_{ij} *propagates* $|\alpha\rangle$ *if, up to a phase, there exists a unique* 1*-qubit state* $|\beta\rangle$ *such that* $\langle\psi|(|\alpha\rangle_i \otimes |\beta\rangle_j) = 0$*.*



$\neq |\gamma^\perp\rangle \quad \Rightarrow \quad = |\delta^\perp\rangle$

$= |\gamma^\perp\rangle \quad \not\Rightarrow$ No propagation



$= |0\rangle \quad \Rightarrow \quad = |0\rangle$

$= |\alpha\rangle \quad \Rightarrow \quad = |\alpha\rangle$

for all $\alpha \in \mathbb{C}^2$

A product constraint will not propagate a state when already satisfied.

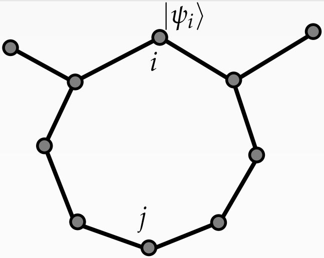
An entangled constraint always propagates every state.

Multi-qubit Propagation

1. Assign $|\psi_i\rangle$ to qubit i.
2. Propagate via a breadth first traversal of the constraint graph.
3. Stop if no propagation is possible or a contradiction is found.

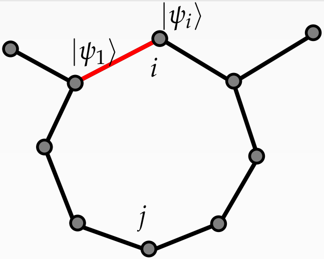
Multi-qubit Propagation

1. Assign $|\psi_i\rangle$ to qubit i.
2. Propagate via a breadth first traversal of the constraint graph.
3. Stop if no propagation is possible or a contradiction is found.



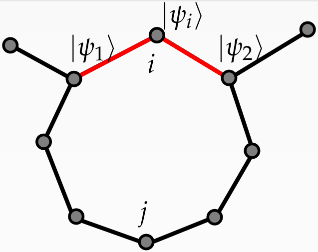
Multi-qubit Propagation

1. Assign $|\psi_i\rangle$ to qubit i.
2. Propagate via a breadth first traversal of the constraint graph.
3. Stop if no propagation is possible or a contradiction is found.



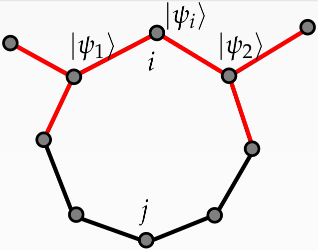
Multi-qubit Propagation

1. Assign $|\psi_i\rangle$ to qubit i.
2. Propagate via a breadth first traversal of the constraint graph.
3. Stop if no propagation is possible or a contradiction is found.



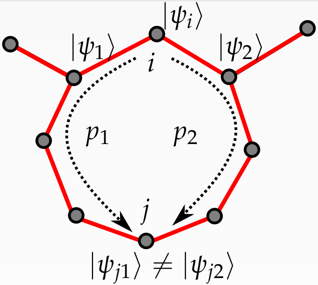
Multi-qubit Propagation

1. Assign $|\psi_i\rangle$ to qubit i.
2. Propagate via a breadth first traversal of the constraint graph.
3. Stop if no propagation is possible or a contradiction is found.



Multi-qubit Propagation

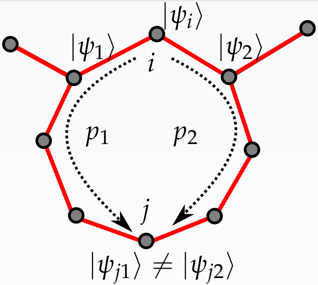
1. Assign $|\psi_i\rangle$ to qubit i.
2. Propagate via a breadth first traversal of the constraint graph.
3. Stop if no propagation is possible or a contradiction is found.



Contradiction: When a qubit j is assigned different states while propagating along different paths from i to j.

Multi-qubit Propagation

1. Assign $|\psi_i\rangle$ to qubit i.
2. Propagate via a breadth first traversal of the constraint graph.
3. Stop if no propagation is possible or a contradiction is found.



Contradiction: When a qubit j is assigned different states while propagating along different paths from i to j.

Lemma (Propagation Lemma, Informal Statement)

Let the propagation $(i, |\psi_i\rangle)$ on $G(\mathcal{H})$ extend the assignment to $|\psi_i\rangle \otimes |\Phi\rangle$. If the propagation is:

1. *"Unsuccessful", there is no solution of the form $|\psi_i\rangle \otimes |rest\rangle$.*
2. *Otherwise, there exists a solution of the form $|\psi_i\rangle \otimes |\Phi\rangle \otimes |rest\rangle$.*

Algorithm Sketch

Part A: Rank-3 and Rank-1 Product Constraints

2-QSAT`Solver`$(G(\mathcal{H}))$

Step 1 For all rank-3 constraints Π_{ij} in $(G(\mathcal{H}))$

- a. Assign the unique state orthogonal to Π_{ij} to qubits i, j.

Step 2 Propagate the previous assignments and if a contradiction is found return "Unsuccessful".

Part A: Rank-3 and Rank-1 Product Constraints

2-QSAT`Solver`$(G(\mathcal{H}))$

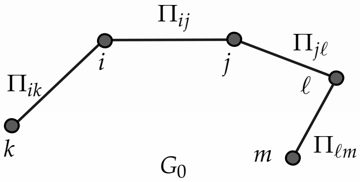
Step 1 For all rank-3 constraints Π_{ij} in $(G(\mathcal{H}))$

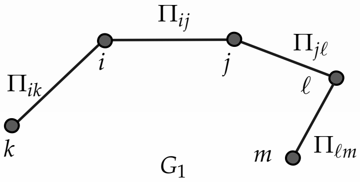
  a. Assign the unique state orthogonal to Π_{ij} to qubits i, j.

Step 2 Propagate the previous assignments and if a contradiction is found return "Unsuccessful".

Step 3 For all rank-1 product constraints $\Pi_{ij} = |\alpha_i \beta_j\rangle\langle\alpha_i \beta_j|$ in $(G(\mathcal{H}))$

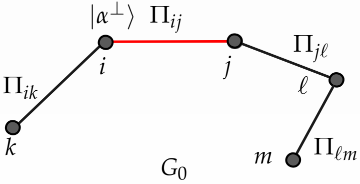
  a. Propagate in parallel the assignments $(i, |\alpha_i^\perp\rangle)$ and $(j, |\beta_j^\perp\rangle)$.

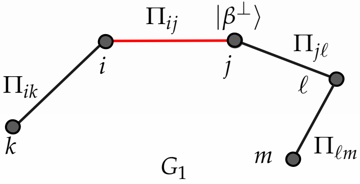




Part A: Rank-3 and Rank-1 Product Constraints

```
2-QSATSolver$(G(\mathcal{H}))$
Step 1 For all rank-3 constraints $\Pi_{ij}$ in $(G(\mathcal{H}))$
    a. Assign the unique state orthogonal to $\Pi_{ij}$ to qubits $i, j$.
Step 2 Propagate the previous assignments and if a contradiction is
       found return "Unsuccessful".
Step 3 For all rank-1 product constraints $\Pi_{ij} = |\alpha_i\beta_j\rangle\langle\alpha_i\beta_j|$ in $(G(\mathcal{H}))$
    a. Propagate in parallel the assignments $(i, |\alpha_i^\perp\rangle)$ and $(j, |\beta_j^\perp\rangle)$.
```





Part A: Rank-3 and Rank-1 Product Constraints

2-QSAT`Solver`$(G(\mathcal{H}))$

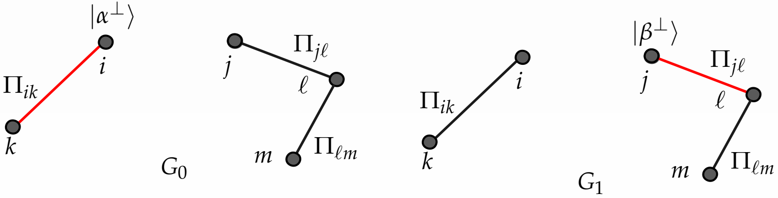
Step 1 For all rank-3 constraints Π_{ij} in $(G(\mathcal{H}))$

- a. Assign the unique state orthogonal to Π_{ij} to qubits i, j.

Step 2 Propagate the previous assignments and if a contradiction is found return "Unsuccessful".

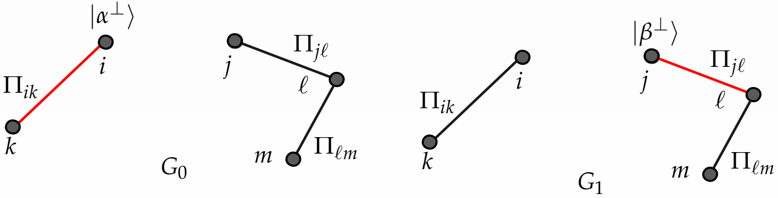
Step 3 For all rank-1 product constraints $\Pi_{ij} = |\alpha_i\beta_j\rangle\langle\alpha_i\beta_j|$ in $(G(\mathcal{H}))$

- a. Propagate in parallel the assignments $(i, |\alpha_i^{\perp}\rangle)$ and $(j, |\beta_j^{\perp}\rangle)$.



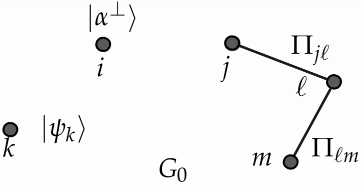
Part A: Rank-3 and Rank-1 Product Constraints

```
2-QSAT`Solver`$(G(\mathcal{H}))$
Step 1  For all rank-3 constraints $\Pi_{ij}$ in $(G(\mathcal{H}))$
        a. Assign the unique state orthogonal to $\Pi_{ij}$ to qubits $i, j$.
Step 2  Propagate the previous assignments and if a contradiction is
        found return "Unsuccessful".
Step 3  For all rank-1 product constraints $\Pi_{ij} = |\alpha_i\beta_j\rangle\langle\alpha_i\beta_j|$ in $(G(\mathcal{H}))$
        a. Propagate in parallel the assignments $(i, |\alpha_i^\perp\rangle)$ and $(j, |\beta_j^\perp\rangle)$.
        b. If both are unsuccessful, return "Unsuccessful".
        c. Accept the assignments of the first successful propagation.
```



Part A: Rank-3 and Rank-1 Product Constraints

```
2-QSATSolver($G(\mathcal{H})$)
Step 1 For all rank-3 constraints $\Pi_{ij}$ in $(G(\mathcal{H}))$
    a. Assign the unique state orthogonal to $\Pi_{ij}$ to qubits $i, j$.
Step 2 Propagate the previous assignments and if a contradiction is
       found return "Unsuccessful".
Step 3 For all rank-1 product constraints $\Pi_{ij} = |\alpha_i\beta_j\rangle\langle\alpha_i\beta_j|$ in $(G(\mathcal{H}))$
    a. Propagate in parallel the assignments $(i, |\alpha_i^\perp\rangle)$ and $(j, |\beta_j^\perp\rangle)$.
    b. If both are unsuccessful, return "Unsuccessful".
    c. Accept the assignments of the first successful propagation.
```



Part B: Rank-1 Entangled Constraints

Step 5 Pick an unassigned qubit i and propagate $(i, |0\rangle)$.

Step 6 If qubit j faces a contradiction, there are two paths p_1, p_2 from i to j.

Part B: Rank-1 Entangled Constraints

Step 5 Pick an unassigned qubit i and propagate $(i, |0\rangle)$.

Step 6 If qubit j faces a contradiction, there are two paths p_1, p_2 from i to j.

Step 7 Slide along p_1 and p_2 to obtain constraints $\Pi_{ij,1}$ and $\Pi_{ij,2}$.

Lemma (Sliding Lemma [JWZ11])

Consider a system on 3 qubits (i, j, k) with entangled constraints Π_{ij} and Π_{ik}. Then there is a constraint Π_{jk} such that the ground spaces of $\Pi_{ij} + \Pi_{ik}$ and $\Pi_{ij} + \Pi_{jk}$ are identical.

Part B: Rank-1 Entangled Constraints

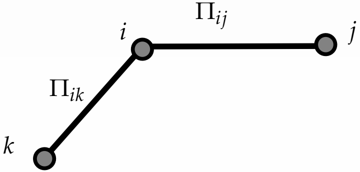
Step 5 Pick an unassigned qubit i and propagate $(i, |0\rangle)$.

Step 6 If qubit j faces a contradiction, there are two paths p_1, p_2 from i to j.

Step 7 Slide along p_1 and p_2 to obtain constraints $\Pi_{ij,1}$ and $\Pi_{ij,2}$.

Lemma (Sliding Lemma [JWZ11])

Consider a system on 3 *qubits* (i, j, k) *with entangled constraints* Π_{ij} *and* Π_{ik}. *Then there is a constraint* Π_{jk} *such that the ground spaces of* $\Pi_{ij} + \Pi_{ik}$ *and* $\Pi_{ij} + \Pi_{jk}$ *are identical.*



Part B: Rank-1 Entangled Constraints

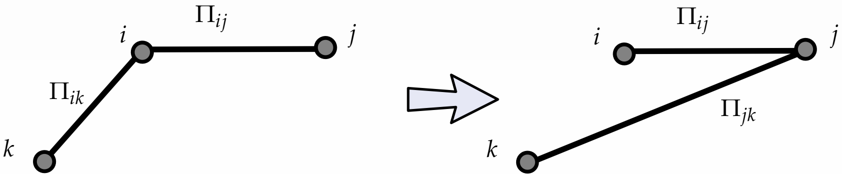
Step 5 Pick an unassigned qubit i and propagate $(i, |0\rangle)$.

Step 6 If qubit j faces a contradiction, there are two paths p_1, p_2 from i to j.

Step 7 Slide along p_1 and p_2 to obtain constraints $\Pi_{ij,1}$ and $\Pi_{ij,2}$.

Lemma (Sliding Lemma [JWZ11])

Consider a system on 3 *qubits* (i, j, k) *with entangled constraints* Π_{ij} *and* Π_{ik}*. Then there is a constraint* Π_{jk} *such that the ground spaces of* $\Pi_{ij} + \Pi_{ik}$ *and* $\Pi_{ij} + \Pi_{jk}$ *are identical.*

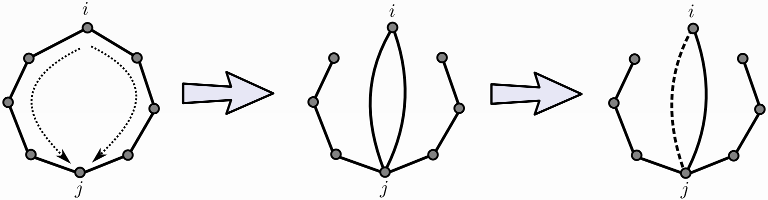


Dealing with Entangled Constraints

Step 5 Pick an unassigned qubit i and propagate $(i, |0\rangle)$.

Step 6 If qubit j faces a contradiction, there are two paths p_1, p_2 from i to j.

Step 7 Slide along p_1 and p_2 to obtain constraints $\Pi_{ij,1}$ and $\Pi_{ij,2}$.

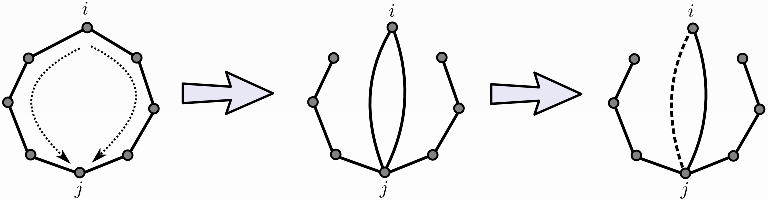


Dealing with Entangled Constraints

Step 5 Pick an unassigned qubit i and propagate $(i, |0\rangle)$.

Step 6 If qubit j faces a contradiction, there are two paths p_1, p_2 from i to j.

Step 7 Slide along p_1 and p_2 to obtain constraints $\Pi_{ij,1}$ and $\Pi_{ij,2}$.



Fact (Structure of 2-dim subspace)

Any 2-dimensional subspace V of the 2-qubit space $\mathbb{C}^2 \otimes \mathbb{C}^2$ contains at least one product state, which can be found in constant time.

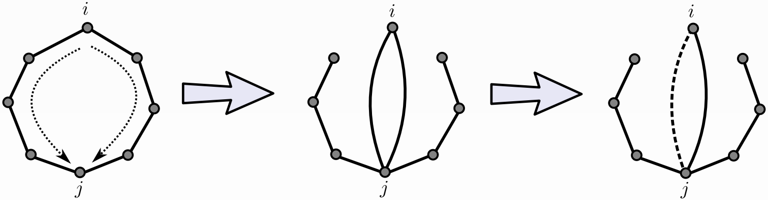
Dealing with Entangled Constraints

Step 5 Pick an unassigned qubit i and propagate $(i, |0\rangle)$.

Step 6 If qubit j faces a contradiction, there are two paths p_1, p_2 from i to j.

Step 7 Slide along p_1 and p_2 to obtain constraints $\Pi_{ij,1}$ and $\Pi_{ij,2}$.

Step 8 Use the product constraint in the space of $\Pi_{ij,1} + \Pi_{ij,2}$ to propagate $(i, |\psi_i\rangle)$ and $(j, |\psi_j\rangle)$ in parallel.



Fact (Structure of 2-dim subspace)

Any 2-dimensional subspace V of the 2-qubit space $\mathbb{C}^2 \otimes \mathbb{C}^2$ contains at least one product state, which can be found in constant time.

Why is it in Linear Time?

The cost of each component in the algorithm:

- Each call to Propagation is constrained by the number of edges accessed in it.

Why is it in Linear Time?

The cost of each component in the algorithm:

- Each call to Propagation is constrained by the number of edges accessed in it.
- An edge involved once in a successful propagation is never accessed again.

Why is it in Linear Time?

The cost of each component in the algorithm:

- Each call to Propagation is constrained by the number of edges accessed in it.
- An edge involved once in a successful propagation is never accessed again.
- For Parallel Propagation, the remaining graph can be copied in time $O(V+E)$ and the cost is constrained by the propagation that succeeds first.

Why is it in Linear Time?

The cost of each component in the algorithm:

- Each call to Propagation is constrained by the number of edges accessed in it.
- An edge involved once in a successful propagation is never accessed again.
- For Parallel Propagation, the remaining graph can be copied in time $O(V + E)$ and the cost is constrained by the propagation that succeeds first.
- The sliding is constrained by the length of the cycle as each slide can be done in constant time.

Why is it in Linear Time?

The cost of each component in the algorithm:

- Each call to Propagation is constrained by the number of edges accessed in it.
- An edge involved once in a successful propagation is never accessed again.
- For Parallel Propagation, the remaining graph can be copied in time $O(V + E)$ and the cost is constrained by the propagation that succeeds first.
- The sliding is constrained by the length of the cycle as each slide can be done in constant time.
- To deal with cycles of entangled constraints, each edge in the paths p_1, p_2 is considered at most 4 times – first propagation $+$ sliding $+$ parallel propagation.

Thanks for your attention!

Cost of performing operations on complex numbers

Bit Complexity

Bit complexity refers to the number of bit-wise operations performed in the course of an algorithm.

Frequently used to analyze algorithms that manipulate algebraic and complex numbers.

For classical 2SAT, optimal bit complexity is same as the number of complex number operations i.e. $O(n+m)$.

Bit Complexity

Bit complexity refers to the number of bit-wise operations performed in the course of an algorithm.

Frequently used to analyze algorithms that manipulate algebraic and complex numbers.

For classical 2SAT, optimal bit complexity is same as the number of complex number operations i.e. $O(n+m)$.

Main Result (from [dBG])

- Bit complexity of the algorithms is $O((m+n)M(n))$

  where $M(n) :=$ Cost of multiplying two n bit numbers.
- Explicit constructions show that $O(n+m)$ bit complexity is not possible for general 2-QSAT instances.
- When all constraints are product, the bit complexity matches the linear bit complexity of 2SAT