# Quantum Dynamic Programming Algorithm for DAGs. Applications for AND-OR DAG Evaluation, Diameter, Shortest and Longest Paths Search in DAG

Kamil Khadiev and Liliya Safina

Estonian Latvian Theory Days 2018

Kazan Federal UNIVERSITY

# Content

- Dinamic Programming for DAGs.

# Content

- Dinamic Programming for DAGs.
- AND-OR DAG Evaluation.

# Content

- Dinamic Programming for DAGs.
- AND-OR DAG Evaluation.
- Diameter Search for DAG.

# Content

- Dinamic Programming for DAGs.
- AND-OR DAG Evaluation.
- Diameter Search for DAG.
- Shortest Path Search for DAG.

# Content

- Dinamic Programming for DAGs.
- AND-OR DAG Evaluation.
- Diameter Search for DAG.
- Shortest Path Search for DAG.
- Longest Path Search for DAG.

# Dynamic Programming for DAGs

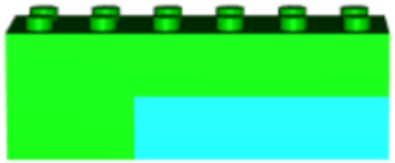
DP: From small parametrs to big ones.

# Dynamic Programming for DAGs

DP: From small parametrs to big ones.

# Dynamic Programming for DAGs

DP: From small parametrs to big ones.



# Dynamic Programming for DAGs

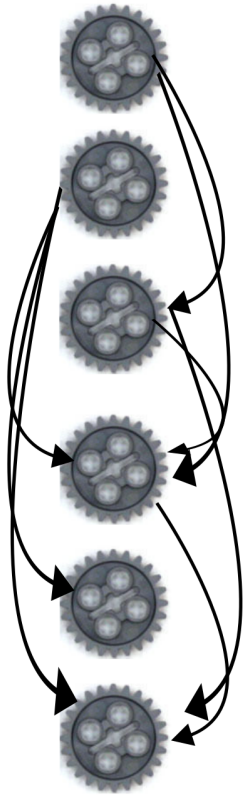
DP: From small parametrs to big ones.

# Dynamic Programming for DAGs

DP: From small parametrs to big ones.

## DP for Directed Acycling Graphs

- DAG $G = (V, E)$:

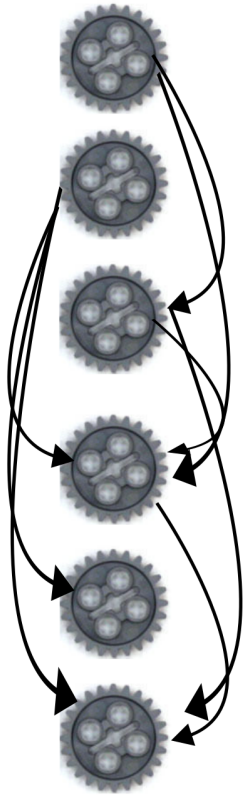


# Dynamic Programming for DAGs

DP: From small parametrs to big ones.

## DP for Directed Acycling Graphs

- DAG $G = (V, E)$:
  - $S_i = (v_{j_1}, \ldots, v_{j_w})$ are "succeeding" for $v_i$: $(v_i, v_{j_t}) \in E$;

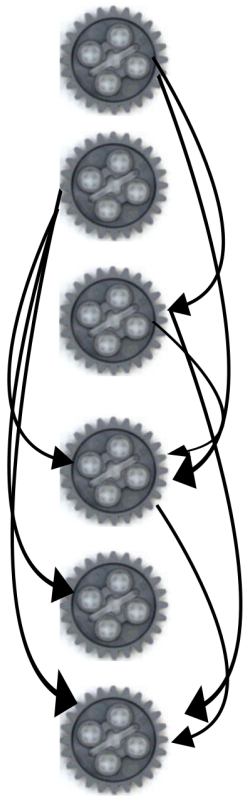


# Dynamic Programming for DAGs

DP: From small parametrs to big ones.

## DP for Directed Acycling Graphs

- DAG $G = (V, E)$:
  - $S_i = (v_{j_1}, \ldots, v_{j_w})$ are "succeeding" for $v_i$: $(v_i, v_{j_t}) \in E$;
  - $G$ is topologically sorted: $\forall (v_i, v_j) \in E : i < j$.

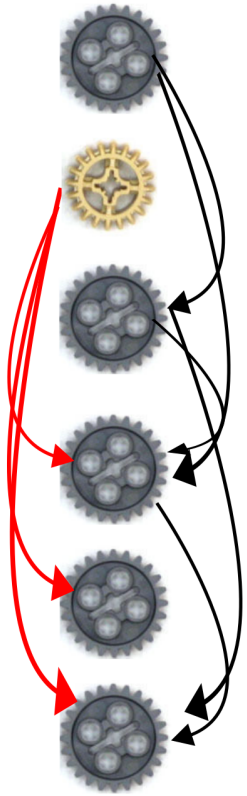


# Dynamic Programming for DAGs

DP: From small parametrs to big ones.

## DP for Directed Acycling Graphs

- DAG $G = (V, E)$:
  - $S_i = (v_{j_1}, \ldots, v_{j_w})$ are "succeeding" for $v_i$: $(v_i, v_{j_t}) \in E$;
  - $G$ is topologically sorted: $\forall (v_i, v_j) \in E : i < j$.
- Problem $Z$:
  - the solution can be presented as a result of a function $f(v_i)$.
  - $f(v_i) = h(f(v_{j_1}), \ldots, f(v_{j_w}))$, for $(v_{j_1}, \ldots, v_{j_w}) = S_i, w = |S_i|$.



# Dynamic Programming for DAGs

DP: From small parametrs to big ones.

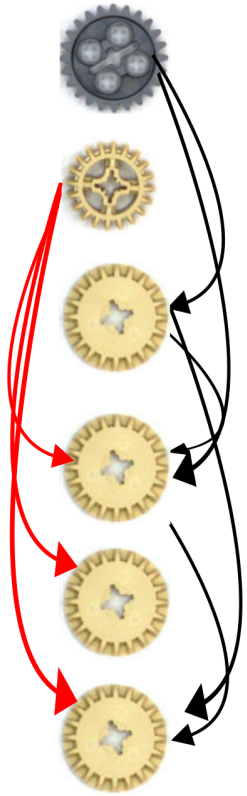
## DP for Directed Acycling Graphs

- DAG $G = (V, E)$:
  - $S_i = (v_{j_1}, \ldots, v_{j_w})$ are "succeeding" for $v_i$: $(v_i, v_{j_t}) \in E$;
  - $G$ is topologically sorted: $\forall (v_i, v_j) \in E : i < j$.
- Problem $Z$:
  - the solution can be presented as a result of a function $f(v_i)$.
  - $f(v_i) = h(f(v_{j_1}), \ldots, f(v_{j_w}))$, for $(v_{j_1}, \ldots, v_{j_w}) = S_i, w = |S_i|$.
- Let $t_i$ be an array with results of $f$.

  for $i$ in $(|V|, \ldots, 1)$:

  $\quad t_i = h(t_{j_1}, \ldots, t_{j_w})$

  , where $(v_{j_1}, \ldots, v_{j_w}) = S_i, w = |S_i|$



# Dynamic Programming for DAGs

## Classical DP for DAG

for $i$ in $(|V|, \ldots, 1)$:

$t_i = h(t_{j_1}, \ldots, t_{j_w})$

- Time complexity of one step is $|S_i|$
- Time complexity of whole algorithm is $O(|E| + |V|)$.

# Dynamic Programming for DAGs

## Classical DP for DAG

for $i$ in $(|V|, \ldots, 1)$:
$\quad t_i = h(t_{j_1}, \ldots, t_{j_w})$

- Time complexity of one step is $|S_i|$
- Time complexity of whole algorithm is $O(|E| + |V|)$.

## Quantum DP for DAG

for $i$ in $(|V|, \ldots, 1)$:
$\quad t_i = Q(t_{j_1}, \ldots, t_{j_w})$

If $h$ is based on MAX, MIN, AND, OR, NAND then

- Time complexity of one step is $\sqrt{|S_i|}$
- Time complexity of whole algorithm is $O(\sum_{i=1}^{|V|} \sqrt{|S_i|}) = O(\sqrt{|V||E|})$.

# The Improvement of the Quantum Algorithm

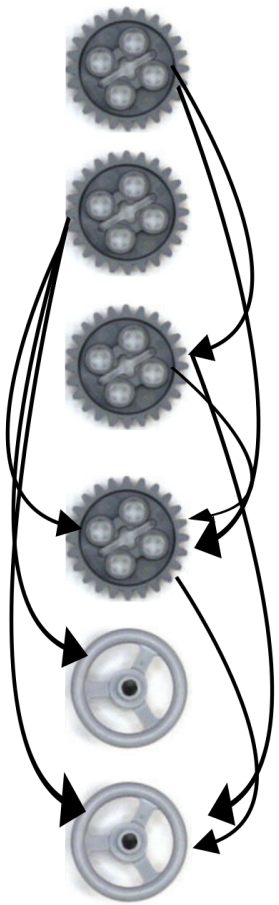
## Inner vertexes of DAG

for $i$ in $(|V|-|L|,\ldots,1)$:

$$t_i = h(t_{j_1},\ldots,t_{j_w})$$

Here $L$ is the set of "leaf" vertexes.

$L = \{v_i : |S_i| = 0\}$



# The Improvement of the Quantum Algorithm

## Inner vertexes of DAG

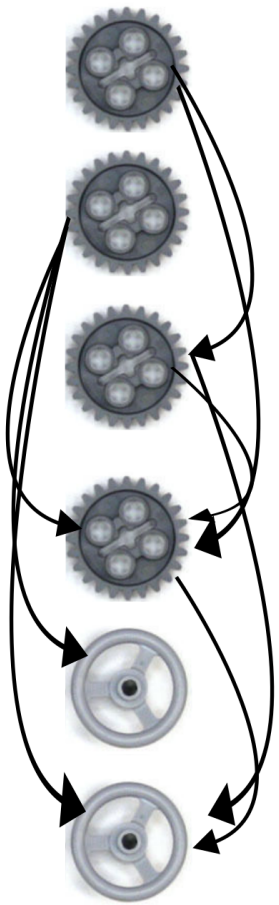
for $i$ in $(|V|-|L|,\ldots,1)$:

$$t_i = h(t_{j_1},\ldots,t_{j_w})$$

Here $L$ is the set of "leaf" vertexes.

$L = \{v_i : |S_i| = 0\}$

- Time complexity is $O(\sqrt{(|V|-|L|)|E|})$.



# Improvement of Quantum algorithm

## Small error probability

# Improvement of Quantum algorithm

## Small error probability

- A success probability of the main algorithm is $O\left(\frac{1}{2^{|V|}}\right)$.

# Improvement of Quantum algorithm

## Small error probability

- A success probability of the main algorithm is $O\left(\frac{1}{2^{|V|}}\right)$.
- We repeat the algorithm $Q$ on each vertex $2\log_2|V|$ times.

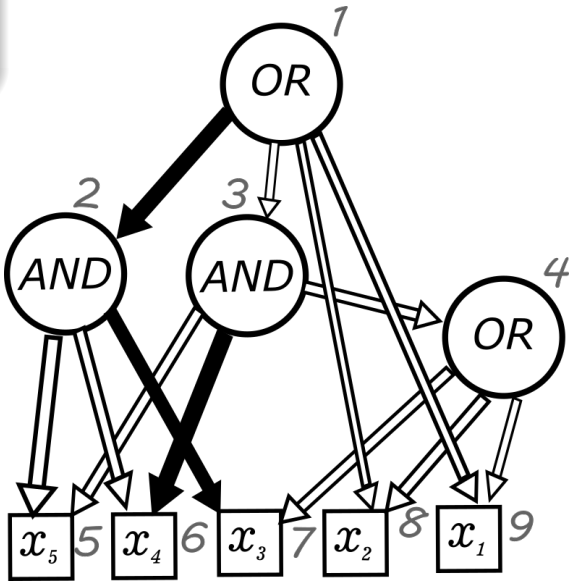
# Improvement of Quantum algorithm

## Small error probability

- A success probability of the main algorithm is $O\left(\frac{1}{2^{|V|}}\right)$.
- We repeat the algorithm $Q$ on each vertex $2\log_2|V|$ times.
- Time complexity is $O(\sqrt{(|V|-|L|)|E|}\log|V|)$,
  Error probability is $O\left(\frac{1}{|V|}\right)$.

# AND-OR DAG Evaluation

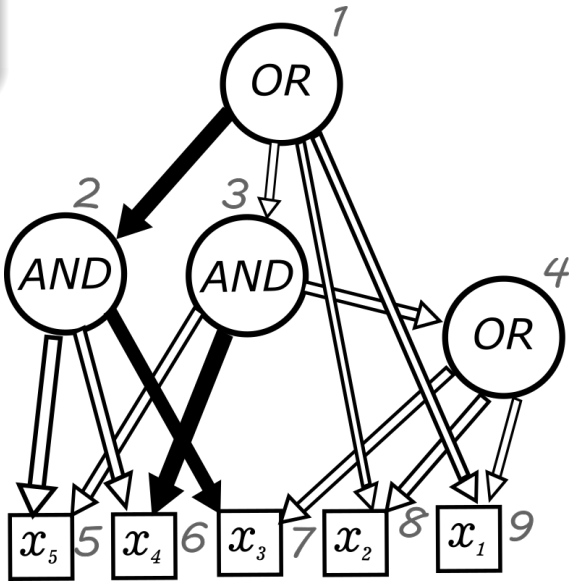
## AND-OR DAG



# AND-OR DAG Evaluation

## AND-OR DAG

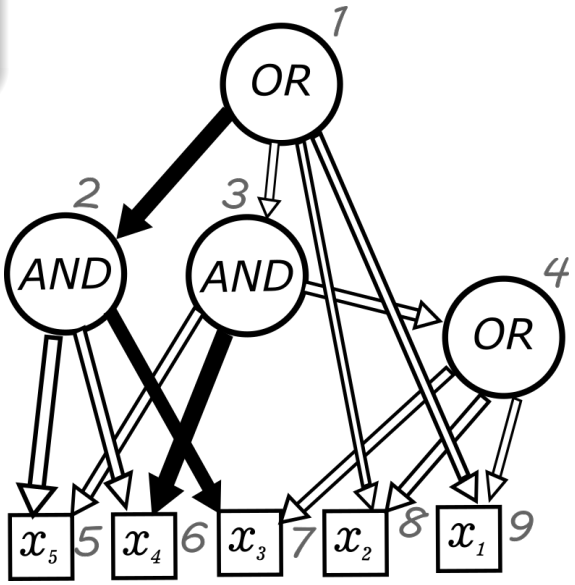
- AND, OR, NAND-vertexes



# AND-OR DAG Evaluation

## AND-OR DAG

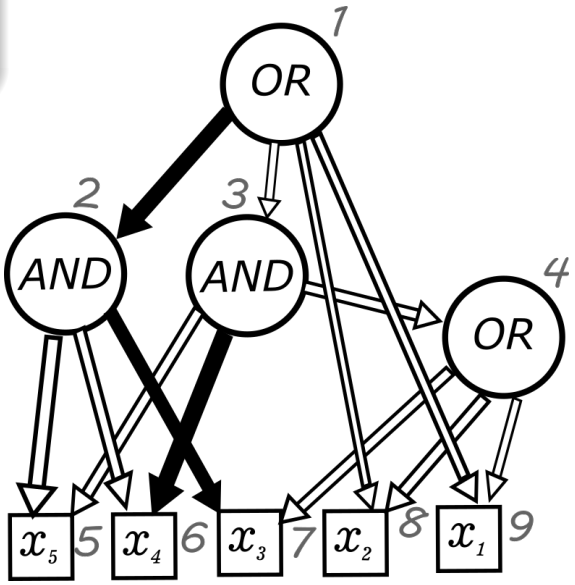
- AND, OR, NAND-vertexes
- can be NOT-edges



# AND-OR DAG Evaluation

## AND-OR DAG

- AND, OR, NAND-vertexes
- can be NOT-edges
- variable-“leafs”



# AND-OR DAG Evaluation

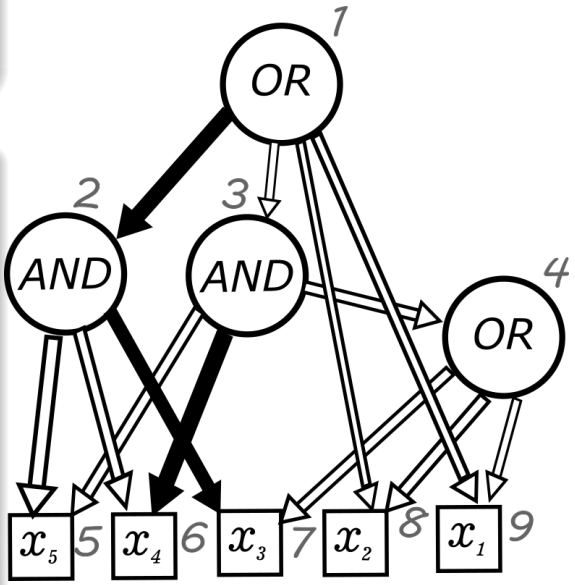
## AND-OR DAG

- AND, OR, NAND-vertexes
- can be NOT-edges
- variable-"leafs"

## Quantum DP Algorithm

for $i$ in $(|V| - |L|, \ldots, 1)$:

$$t_i = Q(2\log_2|V|, \{t_{j_1}^{l(j_1)}, \ldots, t_{j_w}^{l(j_w)}\})$$



# AND-OR DAG Evaluation

## AND-OR DAG

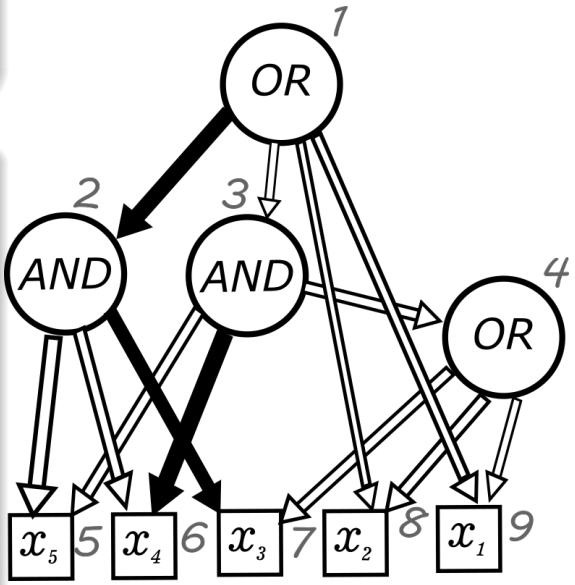
- AND, OR, NAND-vertexes
- can be NOT-edges
- variable-"leafs"

## Quantum DP Algorithm

for $i$ in $(|V| - |L|, \ldots, 1)$:

$$t_i = Q(2\log_2|V|, \{t_{j_1}^{l(j_1)}, \ldots, t_{j_w}^{l(j_w)}\})$$

- $l(j) \in \{0,1\}$, $t_j^1 = t_j^1$, $t_j^0 = NOT\ t_j$



# AND-OR DAG Evaluation

## AND-OR DAG

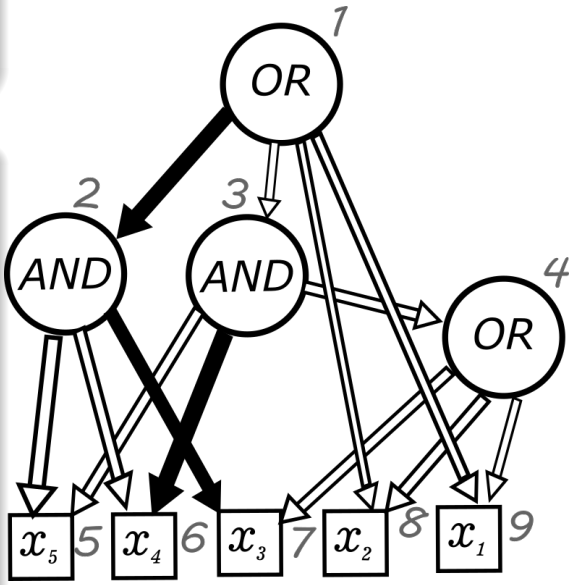
- AND, OR, NAND-vertexes
- can be NOT-edges
- variable-"leafs"

## Quantum DP Algorithm

for $i$ in $(|V| - |L|, \ldots, 1)$:

$$t_i = Q(2\log_2|V|, \{t_{j_1}^{l(j_1)}, \ldots, t_{j_w}^{l(j_w)}\})$$

- $l(j) \in \{0,1\}$, $t_j^1 = t_j^1$, $t_j^0 = NOT\ t_j$
- $Q(K,S)$ for OR is $K$ times Grover search of 1s in $S$.



# AND-OR DAG Evaluation

## AND-OR DAG

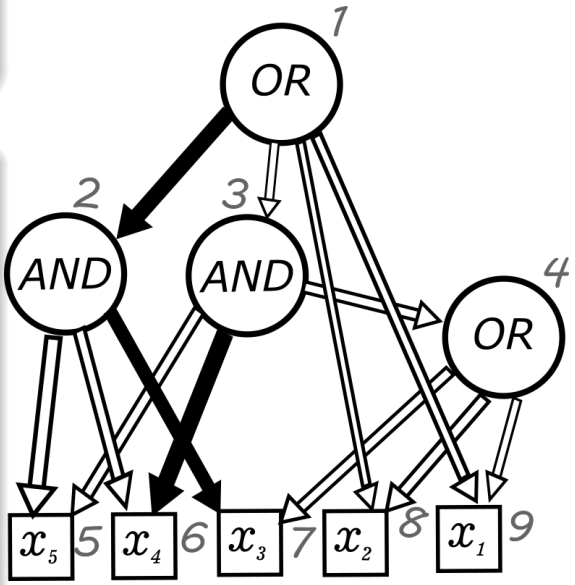
- AND, OR, NAND-vertexes
- can be NOT-edges
- variable-"leafs"

## Quantum DP Algorithm

for $i$ in $(|V| - |L|, \ldots, 1)$:

$$t_i = Q(2\log_2|V|, \{t_{j_1}^{l(j_1)}, \ldots, t_{j_w}^{l(j_w)}\})$$

- $l(j) \in \{0,1\}$, $t_j^1 = t_j^1$, $t_j^0 = NOT\ t_j$
- $Q(K,S)$ for OR is $K$ times Grover search of 1s in $S$.
- $Q(K,S)$ for AND, NAND is $K$ times Grover search of 0s in $S$.



# AND-OR DAG Evaluation

## Existing Algorithms

- Deterministic: $T = O(|E|) = O(|V|^2)$;

# AND-OR DAG Evaluation

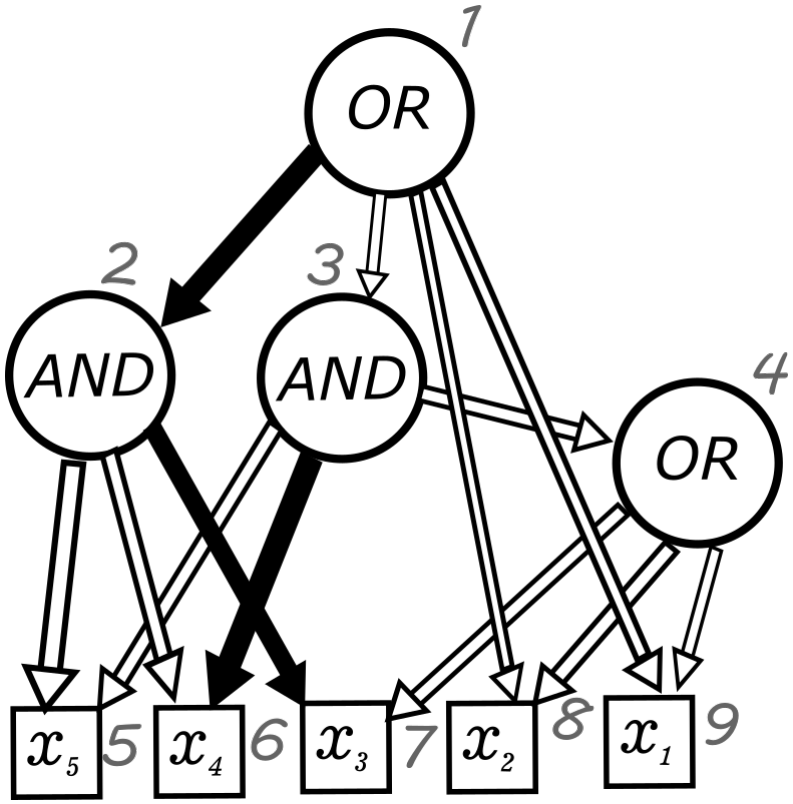
## Existing Algorithms

- Deterministic: $T = O(|E|) = O(|V|^2)$;
- Quantum (Ambainis, 2007, 2010) for a tree: $T = O(|V|^{0.5})$.

# AND-OR DAG Evaluation

## Existing Algorithms

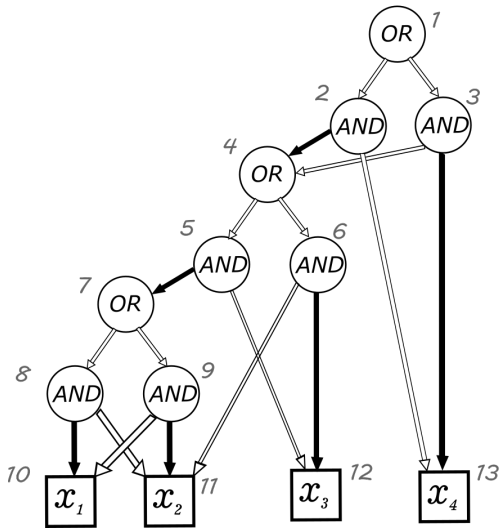
- Deterministic: $T = O(|E|) = O(|V|^2)$;
- Quantum (Ambainis, 2007, 2010) for a tree: $T = O(|V|^{0.5})$.
- DAG $\rightarrow$ tree: can be $T = 2^{O(|V|)}$.

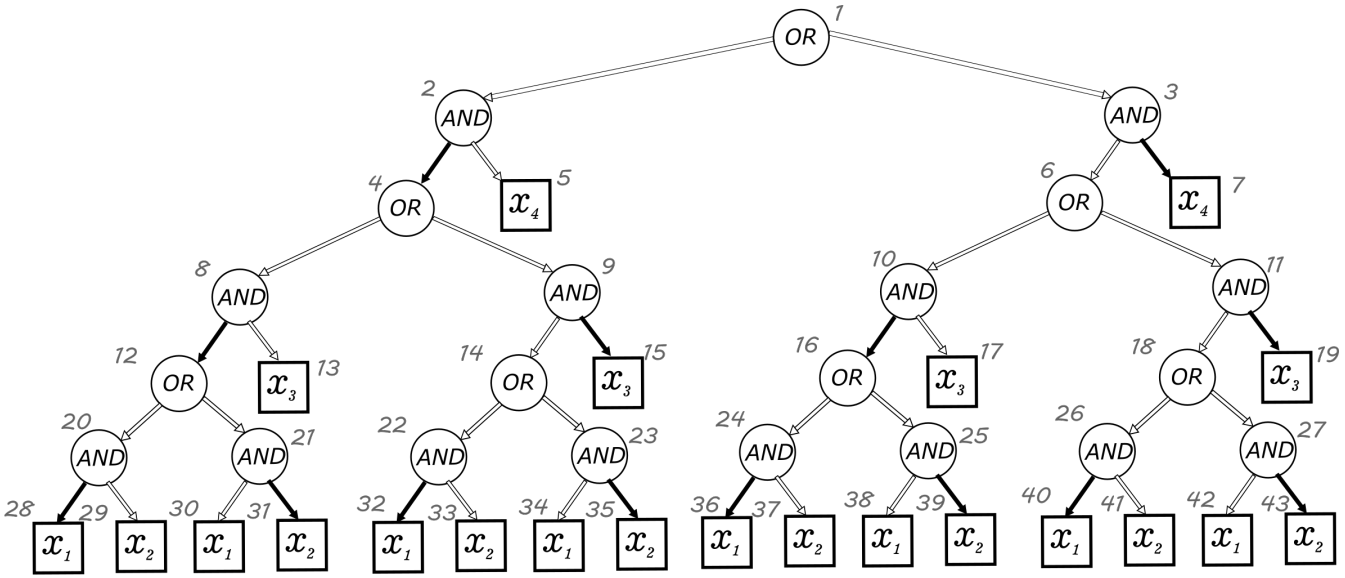


# AND-OR DAG Evaluation

## Existing Algorithms

- Deterministic: $T = O(|E|) = O(|V|^2)$;
- Quantum (Ambainis, 2007, 2010) for a tree: $T = O(|V|^{0.5})$.
- DAG $\rightarrow$ tree: can be $T = 2^{O(|V|)}$.



# AND-OR DAG Evaluation

## Existing Algorithms

- Deterministic: $T = O(|E|) = O(|V|^2)$;
- Quantum (Ambainis, 2007, 2010) for a tree: $T = O(|V|^{0.5})$.
- DAG $\rightarrow$ tree: can be $T = 2^{O(|V|)}$.

## Quantum Algorithm

# AND-OR DAG Evaluation

## Existing Algorithms

- Deterministic: $T = O(|E|) = O(|V|^2)$;
- Quantum (Ambainis, 2007, 2010) for a tree: $T = O(|V|^{0.5})$.
- DAG $\rightarrow$ tree: can be $T = 2^{O(|V|)}$.

## Quantum Algorithm

- Grover Search algorithm's $T = O(\sqrt{N})$ and $Pr_{error} \leq 0.5$.

# AND-OR DAG Evaluation

## Existing Algorithms

- Deterministic: $T = O(|E|) = O(|V|^2)$;
- Quantum (Ambainis, 2007, 2010) for a tree: $T = O(|V|^{0.5})$.
- DAG $\rightarrow$ tree: can be $T = 2^{O(|V|)}$.

## Quantum Algorithm

- Grover Search algorithm's $T = O(\sqrt{N})$ and $Pr_{error} \leq 0.5$.
- $Q$ algorithm's $T = O(\sqrt{|S_i|} \log |V|)$ and $Pr_{error} = O\left(\frac{1}{|V|^2}\right)$

# AND-OR DAG Evaluation

## Existing Algorithms

- Deterministic: $T = O(|E|) = O(|V|^2)$;
- Quantum (Ambainis, 2007, 2010) for a tree: $T = O(|V|^{0.5})$.
- DAG $\rightarrow$ tree: can be $T = 2^{O(|V|)}$.

## Quantum Algorithm

- Grover Search algorithm's $T = O(\sqrt{N})$ and $Pr_{error} \leq 0.5$.
- $Q$ algorithm's $T = O(\sqrt{|S_i|} \log |V|)$ and $Pr_{error} = O\left(\frac{1}{|V|^2}\right)$
- Time complexity of whole algorithm is $T = O(\sqrt{(|V| - |L|)|E|} \log |V|) = O(|V|^{1.5} \log |V|)$.

# AND-OR DAG Evaluation. Example

$$F^{k,l}(X) = \bigoplus_{i=1}^{l} \bigwedge_{j=1}^{k} x_{i,j}$$

- Deterministic algorithm: $T = O(kl)$;
- Quantum algorithm for a tree: $T = O(2^{l/2})$.
- Our quantum algorithm for a DAG: $T = O(l\sqrt{k}\log l)$

# AND-OR DAG Evaluation. Example

$$F^{k,l}(X) = \bigoplus_{i=1}^{l} \bigwedge_{j=1}^{k} x_{i,j}$$

- Deterministic algorithm: $T = O(kl)$;
- Quantum algorithm for a tree: $T = O(2^{l/2})$.
- Our quantum algorithm for a DAG: $T = O(l\sqrt{k}\log l)$

## Zhegalkin polynomial

$$F^m(X) = \bigoplus_{i=1}^{l} x_{j_1} \wedge \cdots \wedge x_{j_{k_i}}$$

$$m = \sum_{i=1}^{l} k_i.$$

- Deterministic: $T = O(m)$.
- Quantum algorithm for a tree: $T = O(2^{l/2})$.
- Our quantum algorithm for a DAG: $T = O(\sqrt{lm}(\log l + \log m))$.

# Diameter Search Problem

## Diameter Search Problem

$d$ is distance between most far vertexes.

# Diameter Search Problem

## Diameter Search Problem

$d$ is distance between most far vertexes.

## Quantum Algorithm

for $i$ in $(|V| - |L|, \ldots, 1)$:

$u = QMAX\_IND\_K\_TIMES(2\log_2 h, \{t_{j_1}, \ldots, t_{j_w}\})$

$z = QMAX\_IND\_K\_TIMES(2\log_2 h, \{t_{j_1}, \ldots, t_{j_w}\} \setminus \{u\})$

$ans = max(ans, t_u + t_z + 2)$

$t_i = t_u + 1$

Here $QMAX\_IND\_K\_TIMES(K, S)$ is the quantum algorithm that searches index of the maximal element in $S$ (Dürr, Høyer, 1996). We invoke it $K$ times.

# Diameter Search Problem

## Deterministic Algorithm

$T = O(|E|) = O(|V|^2)$.

# Diameter Search Problem

## Deterministic Algorithm

$T = O(|E|) = O(|V|^2)$.

## Quantum Algorithm

# Diameter Search Problem

## Deterministic Algorithm

$T = O(|E|) = O(|V|^2)$.

## Quantum Algorithm

- Dürr, Høyer algorithm's $T = O(\sqrt{N})$ and $Pr_{error} \leq 0.5$.

# Diameter Search Problem

## Deterministic Algorithm

$T = O(|E|) = O(|V|^2)$.

## Quantum Algorithm

- Dürr, Høyer algorithm's $T = O(\sqrt{N})$ and $Pr_{error} \leq 0.5$.
- $QMAX\_IND\_K\_TIMES$ algorithm's $T = O(\sqrt{|S_i|} \log |V|)$ and $Pr_{error} = O\left(\frac{1}{|V|^2}\right)$

# Diameter Search Problem

## Deterministic Algorithm

$T = O(|E|) = O(|V|^2)$.

## Quantum Algorithm

- Dürr, Høyer algorithm's $T = O(\sqrt{N})$ and $Pr_{error} \leq 0.5$.
- $QMAX\_IND\_K\_TIMES$ algorithm's $T = O(\sqrt{|S_i|}\log|V|)$ and $Pr_{error} = O\left(\frac{1}{|V|^2}\right)$
- Time complexity of whole algorithm is $T = O(\sqrt{(|V| - |L|)|E|}\log|V|) = O(\sqrt{|V||E|}\log|V|) = O(|V|^{1.5}\log|V|)$.

# Shortest Path Search Problem

## Shortest Path Search Problem

Shortest paths from $v_s$-vertex to others in weighed DAG

# Shortest Path Search Problem

## Shortest Path Search Problem

Shortest paths from $v_s$-vertex to others in weighed DAG

## Quantum Algorithm

for $i$ in $(s, \ldots, |V|)$:

$$t_i = QMIN\_K\_TIMES(2\log_2|V|, \{t_{j_1}, \ldots, t_{j_w}\})$$

where $P_i = \{t_{j_1}, \ldots, t_{j_w}\}$ are "preceding" for $v_i$: $(v_{j_t}, v_i) \in E$

# Shortest Path Search Problem

## Deterministic Algorithms

- DP for DAGs: $T = O(|E|) = O(|V|^2)$.
- Dijkstra's algorithm with Fibonacci heap for graphs: $T = O(|E| + |V| \log |V|)$.
- Dörn'2009 quantum algorithm for graphs: $T = O(\sqrt{|V||E|}(\log |V|)^2)$.

# Shortest Path Search Problem

## Deterministic Algorithms

- DP for DAGs: $T = O(|E|) = O(|V|^2)$.
- Dijkstra's algorithm with Fibonacci heap for graphs: $T = O(|E| + |V| \log |V|)$.
- Dörn'2009 quantum algorithm for graphs: $T = O(\sqrt{|V||E|}(\log |V|)^2)$.

## Quantum Algorithm

$T = O(\sqrt{(|V| - s)|E|} \log |V|) = O(\sqrt{|V||E|} \log |V|) = O(|V|^{1.5} \log |V|)$.

# Longest Path Search Problem

## Longest Path Search Problem

Longest paths from $v_s$-vertex to others in weighed DAG

# Longest Path Search Problem

## Longest Path Search Problem

Longest paths from $v_s$-vertex to others in weighed DAG

## Quantum Algorithm

for $i$ in $(s, \ldots, |V|)$:

$$t_i = QMAX\_K\_TIMES(2\log_2|V|, \{t_{j_1}, \ldots, t_{j_w}\})$$

where $P_i = \{t_{j_1}, \ldots, t_{j_w}\}$ are "preceding" for $v_i$: $(v_{j_t}, v_i) \in E$

# Shortest Path Search Problem

## Deterministic Algorithms

- DP for DAGs: $T = O(|E|) = O(|V|^2)$.
- a NP-complete problem for graphs.

# Shortest Path Search Problem

## Deterministic Algorithms

- DP for DAGs: $T = O(|E|) = O(|V|^2)$.
- a NP-complete problem for graphs.

## Quantum Algorithm

$T = O(\sqrt{(|V| - s)|E|} \log |V|) = O(\sqrt{|V||E|} \log |V|) = O(|V|^{1.5} \log |V|)$.

Thank you for your attention!

Paldies! Aitäh!