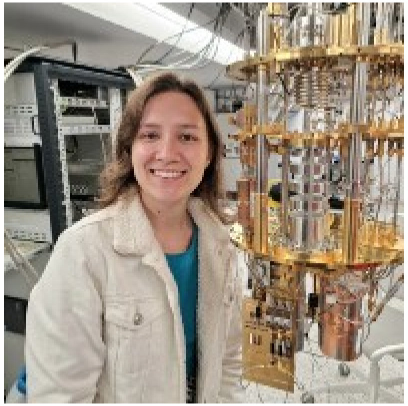
Quantum Speedups for Bayesian Network Structure Learning



Juha Harviainen
University of Helsinki

Kseniya Rychkova
University of Queensland

Mikko Koivisto
University of Helsinki

The BNSL problem

Input

Families $\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_n$ of subsets of $[n] := \{1, 2, \ldots, n\}$ and weights $w_i(S)$ for each set $S \in \mathbf{F}_i$.

Output

A DAG $([n], A)$, with $A_i \in \mathbf{F}_i$, maximizing

$$w(A) := w_1(A_1) + w_2(A_2) + \ldots + w_n(A_n) .$$

Here A_i is the set of parents of node i in A.

The BNSL problem

Input

Families $\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_n$ of subsets of $[n] := \{1, 2, \ldots, n\}$ and weights $w_i(S)$ for each set $S \in \mathbf{F}_i$.

Output

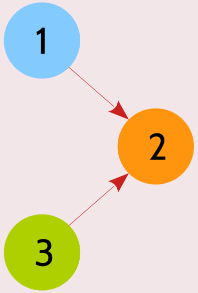
A DAG $([n], A)$, with $A_i \in \mathbf{F}_i$, maximizing

$$w(A) := w_1(A_1) + w_2(A_2) + \ldots + w_n(A_n) .$$

Here A_i is the set of parents of node i in A.

S	$w_1(S)$
{ }	3.0
{2}	3.5
{3}	2.0
{2, 3}	4.0

S	$w_3(S)$
{ }	2.0
{1}	1.5
{2}	2.5
{1, 2}	3.0



S	$w_2(S)$
{ }	1.5
{1}	3.0
{3}	2.0
{1, 3}	5.0

The BNSL problem

Input

Families $F_1, F_2, \ldots, F_n$ of subsets of $[n] := \{1, 2, \ldots, n\}$ and weights $w_i(S)$ for each set $S \in F_i$.

Output

A DAG $([n], A)$, with $A_i \in F_i$, maximizing

$$w(A) := w_1(A_1) + w_2(A_2) + \ldots + w_n(A_n) .$$

Here A_i is the set of parents of node i in A.

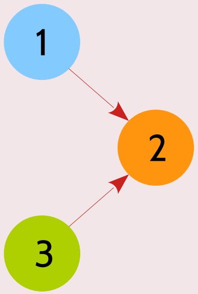
Similar to TSP and the Feedback Arc Set problem

NP-hard Chickering 1996

Can be solved in time $O(2^n n^2)$ Ott & Miyano 2003, Koivisto & Sood 2004, Singh & Moore 2005, Silander & Myllymäki 2006

S	$w_1(S)$
{ }	3.0
{2}	3.5
{3}	2.0
{2, 3}	4.0

S	$w_3(S)$
{ }	2.0
{1}	1.5
{2}	2.5
{1, 2}	3.0



S	$w_2(S)$
{ }	1.5
{1}	3.0
{3}	2.0
{1, 3}	5.0

Our results

Theorem 1

BNSL, with subexponentially many potential parent sets, cannot be solved classically in time $O(c^n)$ for any $c < 2$ under SETH.

Our results

Theorem 1
BNSL, with subexponentially many potential parent sets, cannot be solved classically in time $O(c^n)$ for any $c < 2$ under SETH.

Theorem 2
BNSL, with subexponentially many potential parent sets, admits a bounded-error quantum algorithm that runs in time $O(1.817^n)$.

Our results

Theorem 1
BNSL, with subexponentially many potential parent sets, cannot be solved classically in time $O(c^n)$ for any $c < 2$ under SETH.

Theorem 2
BNSL, with subexponentially many potential parent sets, admits a bounded-error quantum algorithm that runs in time $O(1.817^n)$.

Theorem 3
BNSL, with $O(1.453^n)$ potential parent sets, admits a bounded-error quantum algorithm that runs in time $O(1.982^n)$.

Hardness – proof idea

Reduce from the k-Hitting Set problem:

> **Input:** A family **T** of subsets of $[n]$, each of size at most k, and a number t.
> **Question:** Is there a subset of $[n]$ of size t intersecting all members of **T**?

Hardness – proof idea

Reduce from the k-Hitting Set problem:

Input: A family **T** of subsets of $[n]$, each of size at most k, and a number t.
Question: Is there a subset of $[n]$ of size t intersecting all members of **T**?

Theorem Cygan, Dell, Lokshtanov, Marx, Nederlof, Okamoto, Paturi, Saurabh, Wahlström 2016
Under SETH, for any $c < 2$ there exists a k such that the k-Hitting Set problem cannot be solved in time $O(c^n)$ by a classical algorithm.

Hardness – proof idea

Reduce from the k-Hitting Set problem:

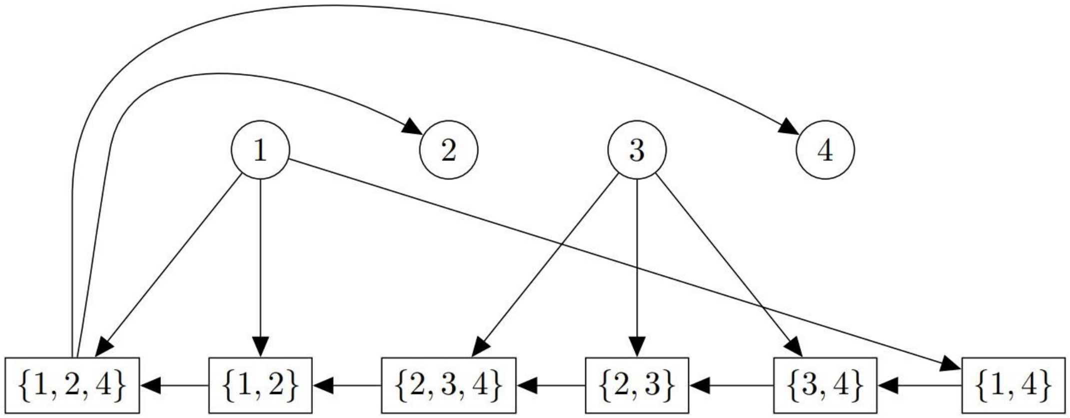
Input: A family **T** of subsets of $[n]$, each of size at most k, and a number t.
Question: Is there a subset of $[n]$ of size t intersecting all members of **T**?

Theorem Cygan, Dell, Lokshtanov, Marx, Nederlof, Okamoto, Paturi, Saurabh, Wahlström 2016
Under SETH, for any $c < 2$ there exists a k such that the k-Hitting Set problem cannot be solved in time $O(c^n)$ by a classical algorithm.

Reduction

1. A simple reduction to a BNSL instance with $n + |T|$ nodes.

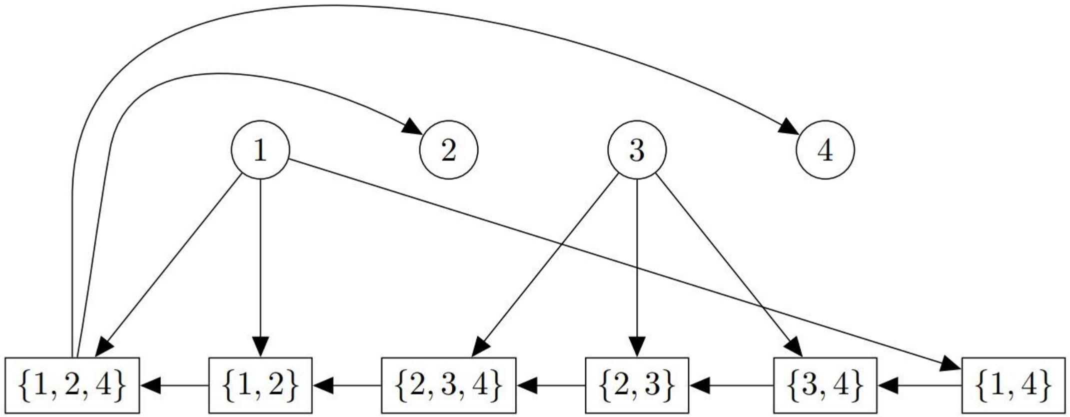
Hardness – proof idea



Reduction

1. A simple reduction to a BNSL instance with n + |T| nodes.

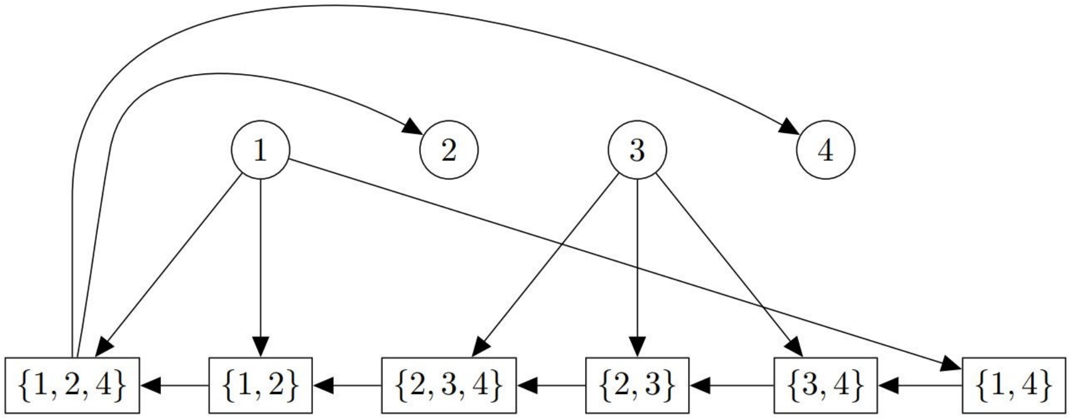
Hardness – proof idea



Reduction

1. A simple reduction to a BNSL instance with $n + |T|$ nodes.
2. Sparsify the instance, yielding just $n' := n + O(n^{k/(k+1)}) = n + o(n)$ nodes.

Hardness – proof idea



Reduction

1. A simple reduction to a BNSL instance with $n + |T|$ nodes.
2. Sparsify the instance, yielding just $n' := n + O(n^{k/(k+1)}) = n + o(n)$ nodes.
3. If the BNSL instance could be solved in time $O(b^{n'})$ with $b < 2$, then the k-Hitting Set problem could be solved in time $O(c^n)$ with $c = (2b)^{1/2} < 2$.

Quantum algorithms

Quantum computation refers to a theoretical model inspired by quantum physics.

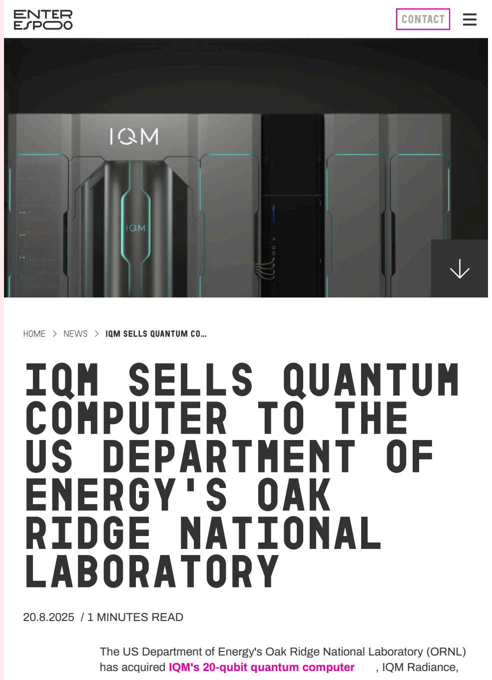
Enables solving **some problems faster** than by classical computation:

- Shor's algorithm for the Factoring problem
- Grover's algorithm for unstructured search

No practical value in the foreseeable future:

- The largest integer factored using Shor's algorithm: 21 = 3 x 7

  Martín-López, Laing, Lawson, Alvarez, Zhou, O'Brien 2012


ENTER ESPOO
CONTACT
HOME > NEWS > IQM SELLS QUANTUM CO…

IQM SELLS QUANTUM COMPUTER TO THE US DEPARTMENT OF ENERGY'S OAK RIDGE NATIONAL LABORATORY

20.8.2025 / 1 MINUTES READ

The US Department of Energy's Oak Ridge National Laboratory (ORNL) has acquired IQM's 20-qubit quantum computer , IQM Radiance,

Building blocks

Recursive quantum search Dürr & Høyer 1996; Ambainis, Balodis, Iraids, Kokainis, Prusis, Vihrovs 2019

Suppose $f(x)$ is an integer computable for any given $x \in \{1, 2, \ldots, m\}$ by a bounded-error quantum algorithm in time T. Then there is a bounded-error quantum algorithm that computes $\max f(x)$ in time $O(T\, m^{1/2} \log m)$.

Building blocks

Recursive quantum search Dürr & Høyer 1996; Ambainis, Balodis, Iraids, Kokainis, Prusis, Vihrovs 2019

Suppose $f(x)$ is an integer computable for any given $x \in \{1, 2, \ldots, m\}$ by a bounded-error quantum algorithm in time T. Then there is a bounded-error quantum algorithm that computes $\max f(x)$ in time $O(T\, m^{1/2} \log m)$.

Quantum RAM Giovannetti, Lloyd, Maccone 2008

Any time-T classical algorithm that uses random access memory can be invoked as a subroutine for a quantum algorithm in time $O(T)$.

Building blocks

Recursive quantum search Dürr & Høyer 1996; Ambainis, Balodis, Iraids, Kokainis, Prusis, Vihrovs 2019

Suppose $f(x)$ is an integer computable for any given $x \in \{1, 2, \ldots, m\}$ by a bounded-error quantum algorithm in time T. Then there is a bounded-error quantum algorithm that computes $\max f(x)$ in time $O(T\, m^{1/2} \log m)$.

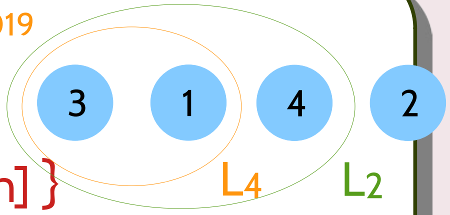
Quantum RAM Giovannetti, Lloyd, Maccone 2008

Any time-T classical algorithm that uses random access memory can be invoked as a subroutine for a quantum algorithm in time $O(T)$.

Vertex ordering problems Ambainis, Balodis, Iraids, Kokainis, Prusis, Vihrovs 2019

There is a bounded-error quantum algorithm that computes

$$\max \{ f(L_1, 1) + f(L_2, 2) + \ldots + f(L_n, n) : L \text{ is a linear order on } [n] \}$$



in time $O(1.817^n\, T)$, supposing f can be evaluated in time T.

A speedup for BNSL – proof of Theorem 2

Idea: Formulate BNSL as a vertex ordering problem with an appropriate f.

A speedup for BNSL – proof of Theorem 2

Idea: Formulate BNSL as a vertex ordering problem with an appropriate f.

Observations

1. Every DAG has a topological ordering of the nodes – a linear order L.

A speedup for BNSL – proof of Theorem 2

Idea: Formulate BNSL as a vertex ordering problem with an appropriate f.

Observations

1. Every DAG has a topological ordering of the nodes – a linear order L.
2. Maximize the weight among DAGs A whose topological ordering is L:

   for each node i: $\max \{ w_i(A_i) : A_i \text{ is a subset of } L_i \} =: f(L_i, i)$.

A speedup for BNSL – proof of Theorem 2

Idea: Formulate BNSL as a vertex ordering problem with an appropriate f.

Observations

1. Every DAG has a topological ordering of the nodes – a linear order L.
2. Maximize the weight among DAGs A whose topological ordering is L:

   for each node i: $\max \{ w_i(A_i) : A_i \text{ is a subset of } L_i \} =: f(L_i, i)$.

3. Evaluating f takes time linear in the number of potential parents sets.

   Or, about a square root of that using quantum computing.

   => a subexponential factor => time $O(1.817^n)$.

A speedup for BNSL – proof of Theorem 2

Idea: Formulate BNSL as a vertex ordering problem with an appropriate f.

Observations

1. Every DAG has a topological ordering of the nodes – a linear order L.
2. Maximize the weight among DAGs A whose topological ordering is L:

   for each node i: $\max \{ w_i(A_i) : A_i \text{ is a subset of } L_i \} =: f(L_i, i)$.
3. Evaluating f takes time linear in the number of potential parents sets.

   Or, about a square root of that using quantum computing.

   => a subexponential factor => time $O(1.817^n)$.

What if we have exponentially many potential parent sets?

Another speedup – proof of Theorem 3

Idea: Connect with a known space–time tradeoff for vertex ordering problems.

Another speedup – proof of Theorem 3

Idea: Connect with a known space–time tradeoff for vertex ordering problems.

Partial order cover Koivisto & Parviainen 2010

A family **P** of partial orders on [n] is a cover if every linear order L on [n] is a linear extension of some member P of **P**, i.e., P ⊆ L.

Another speedup – proof of Theorem 3

Idea: Connect with a known space–time tradeoff for vertex ordering problems.

Partial order cover Koivisto & Parviainen 2010
A family **P** of partial orders on [n] is a cover if every linear order L on [n] is a linear extension of some member P of **P**, i.e., $P \subseteq L$.

Write $\max \{ f(L) : L \text{ is a linear order on } [n] \}$ as

$\max \{ g(P) : P \in \mathbf{P} \}$, with $g(P) := \max \{ f(L) : L \text{ is a linear extension of } P \}$.

Another speedup – proof of Theorem 3

Idea: Connect with a known space–time tradeoff for vertex ordering problems.

Partial order cover Koivisto & Parviainen 2010
A family **P** of partial orders on [n] is a cover if every linear order L on [n] is a linear extension of some member P of **P**, i.e., $P \subseteq L$.

Write max { f(L) : L is a linear order on [n] } as

max { g(P) : P ∈ **P** } , with g(P) := max { f(L) : L is a linear extension of P } .

Space x time Koivisto & Parviainen 2010
Any vertex ordering problem can be solved in space $O^*(S)$ and time $O^*(T)$ with

$$S := \max \{ |\mathrm{Downsets}(P)| : P \in \mathbf{P} \} \quad \text{and} \quad T := \sum \{ |\mathrm{Downsets}(P)| : P \in \mathbf{P} \} .$$

There exists a cover such that $S\,T = S^2\,|\mathbf{P}| = O(3.93^n)$ and $S = O(1.453^n)$.

Another speedup – proof of Theorem 3

Idea:

Space x time Koivisto & Parviainen 2010

Any vertex ordering problem can be solved in space $O^*(S)$ and time $O^*(T)$ with

$$S := \max \{ |\text{Downsets}(P)| : P \in \mathbf{P} \} \text{ and } T := \sum \{ |\text{Downsets}(P)| : P \in \mathbf{P} \} .$$

There exists a cover such that $S\,T = S^2\,|\mathbf{P}| = O(3.93^n)$ and $S = O(1.453^n)$.

Write $\max \{ f(L) : L \text{ is a linear order on } [n] \}$ as

$\max \{ g(P) : P \in \mathbf{P} \}$, with $g(P) := \max \{ f(L) : L \text{ is a linear extension of } P \}$.

=> ?

Another speedup – proof of Theorem 3

Idea:

Space x time Koivisto & Parviainen 2010

Any vertex ordering problem can be solved in space $O^*(S)$ and time $O^*(T)$ with

$$S := \max \{ |\text{Downsets}(P)| : P \in \mathbf{P} \} \text{ and } T := \sum \{ |\text{Downsets}(P)| : P \in \mathbf{P} \} .$$

There exists a cover such that $S\,T = S^2\,|\mathbf{P}| = O(3.93^n)$ and $S = O(1.453^n)$.

Write $\max \{ f(L) : L \text{ is a linear order on } [n] \}$ as

$\max \{ g(P) : P \in \mathbf{P} \}$, with $g(P) := \max \{ f(L) : L \text{ is a linear extension of } P \}$.

=> BNSL using quantum search in time

$O(S\,|\mathbf{P}|^{1/2} \log |\mathbf{P}|) = O(1.982^n)$,

supposing the number of potential parent sets is $O(S)$.

Summary

Theorem 1
BNSL, with subexponentially many potential parent sets, cannot be solved classically in time $O(c^n)$ for any $c < 2$ under SETH.

Theorem 2
BNSL, with subexponentially many potential parent sets, admits a bounded-error quantum algorithm that runs in time $O(1.817^n)$.

Theorem 3
BNSL, with $O(1.453^n)$ potential parent sets, admits a bounded-error quantum algorithm that runs in time $O(1.982^n)$.

Su

Open problem
Can BNSL, with polynomially many potential parent sets, be solved classically in time $O(c^n)$ for some $c < 2$?

Theorem 1
BNSL, with subexponentially many potential parent sets, cannot be solved classically in time $O(c^n)$ for any $c < 2$ under SETH.

Theorem 2
BNSL, with subexponentially many potential parent sets, admits a bounded-error quantum algorithm that runs in time $O(1.817^n)$.

Theorem 3
BNSL, with $O(1.453^n)$ potential parent sets, admits a bounded-error quantum algorithm that runs in time $O(1.982^n)$.

Su

Open problem
Can BNSL, with polynomially many potential parent sets, be solved classically in time $O(c^n)$ for some $c < 2$?

Theorem 1
BNSL, with subexponentially many potential parent sets, cannot be solved classically in time $O(c^n)$ for any $c < 2$ under SETH.

Open problem
Does BNSL, with $O(2^n)$ potential parent sets, admit a bounded-error quantum algorithm that runs in time $O(c^n)$ for some $c < 2$?

Theorem 3
BNSL, with $O(1.453^n)$ potential parent sets, admits a bounded-error quantum algorithm that runs in time $O(1.982^n)$.