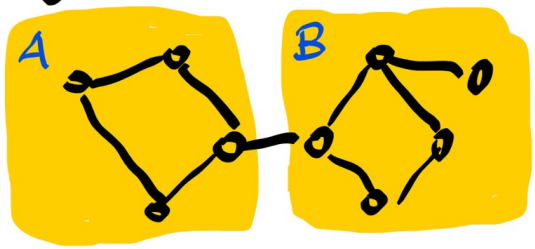
A literature review of quantum parallelisation

C. Dürr
LIP6

2 Oct 2024

Divide and Conquer....
Saleem, Tomesh, +4
2022

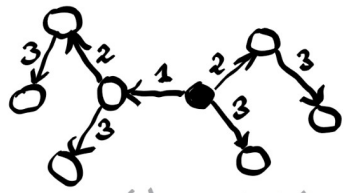
Figure I don't understand



It says: vertex = qubit
edge = entanglement
A, B = parallel qu. computers

In my understanding we capture only part of the picture. Ideally we need knot theory to describe entanglement.

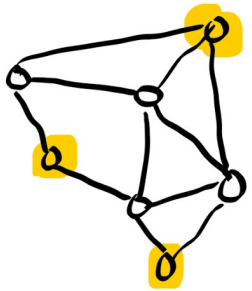
Example ○ = initial $|0\rangle$
● = initial $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$
Ⓤ → Ⓥ controled NOT
t: $V := V \oplus u$
t = time of operation



now qubits are all $|0\rangle$ w.ampl. $\frac{1}{\sqrt{2}}$
and all $|1\rangle$ w.ampl. $\frac{1}{\sqrt{2}}$
entanglement graph = clique ≠ interaction graph

Maximum Independent Set

input $G = (V, E)$
output S s.t. $\forall uv \in E$
$u \notin S$ or
$v \notin S$
objective $\max |S|$



State of the art for classical computers

easy for bipartite graphs
interval graphs

hard to approximate (cannot decide if $\alpha \leq n^{\varepsilon}$ or $\alpha \geq n^{1-\varepsilon}$)
is also hard for quantum, so don't expect much

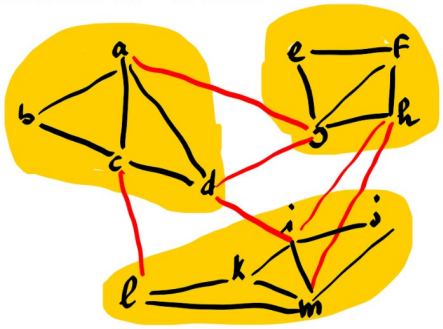
exact algorithm in $1.1996^n \cdot n^{O(1)}$ [Xiao, Nagamochi 2017]
My suggestion: combine with quantum computing

Quantum Algorithm

qubit = vertex
$|1\rangle$ = belong to solution set
constraint
$|0\rangle \to |1\rangle$ only possible if all neighbors are $|0\rangle$

1) partition graph, each part → dedicated quantum comp.
2) run quantum optimisation algorithm on each qu. comp.

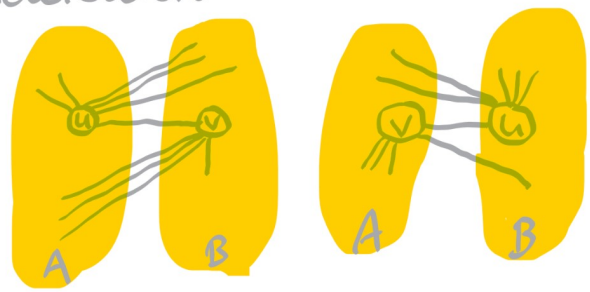
Partition the Graph



Size of each part ≤ # qbits per machine
Minimize # crossing edges

Their choice

Kernighan-Lin 1970
partition $V = A \cup B$, $|A|, |B| = \frac{|V|}{2}$
local search



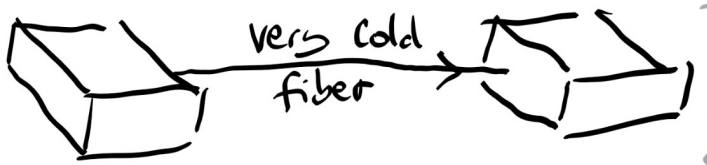
My suggestion

Use integer linear programming formulation
Appropriate for small instances.
Easier than actual maximum independent set problem?

Communication between Parts

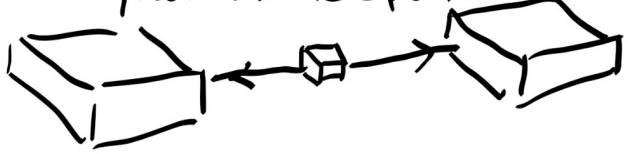
technical aspect

either



I see problem of sharing a clock (sync.)

or quantum teleportation

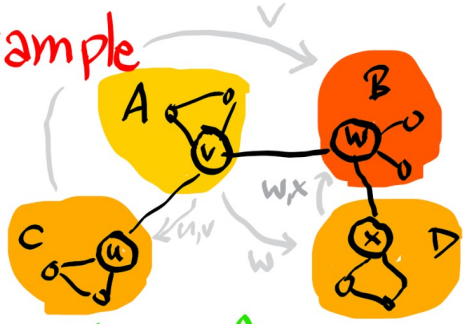


Need one additional qudit on each side.
I think we need **sequential** computation

1) run QC1 on vertices abcdgil
2) teleport ad from QC1 to QC2
3) run QC2 on adefghim
4) teleport e from QC1 to QC3
5) teleport dhim from QC2 to QC3
6) run QC3 on cdhijklm

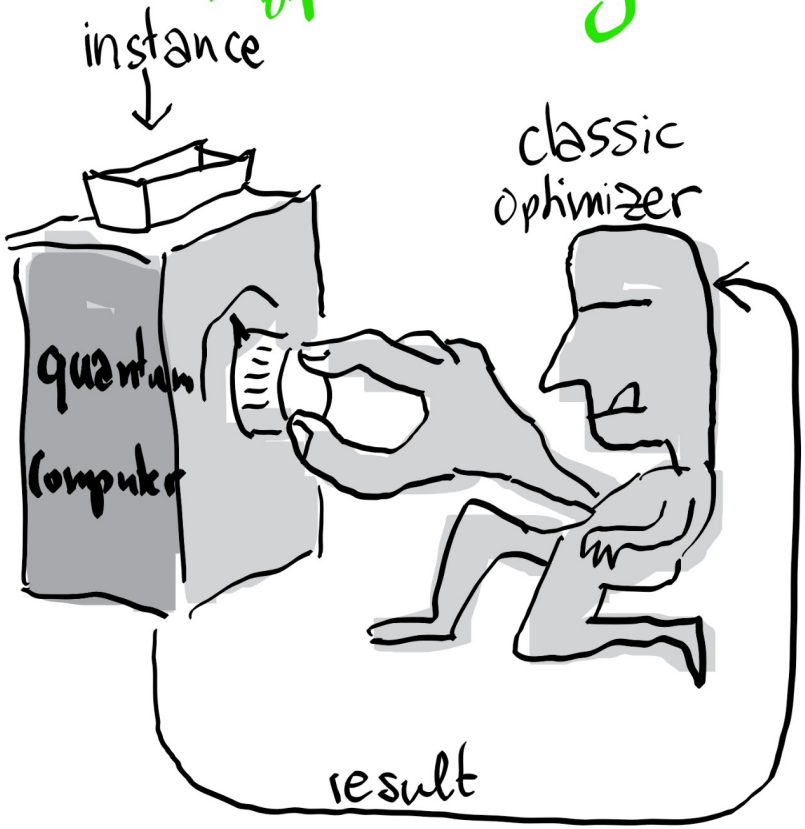
I suspect, rather minimizing # cross edges,
we would like to minimize a different objective
- d is teleported twice (↔ degree)
- take cross edges into account for QC capacity

example



execution order
1) A∪{u,w}
2) C∪{v} || D∪{w}
3) B∪{v,x}

Quantum Approximate Algorithm



alternate between
- phase separation step (constraint in obj)
- mixing step (pb indep.)

classic Optimize time spend on each step

Quantum Alternating operator Ansatz (Approach)

invariant: all states with nonzero amplitude are feasible
objective = Hamming Weight.
mixer operator on specific qubit $|0\rangle \to |1\rangle$ only if all neighbors in state $|0\rangle$.
Apply sequentially to all vertices in some arbitrary order

[...]

Their Experiments

random graphs
- 3-regular
- 2-community

high low high edge probability

16 and 26 nodes

Simulated quantum computer
- Qiskit

Compared different methods for partitioning the graph

Compare with classical alg. for max. indep. set but not with state of the art