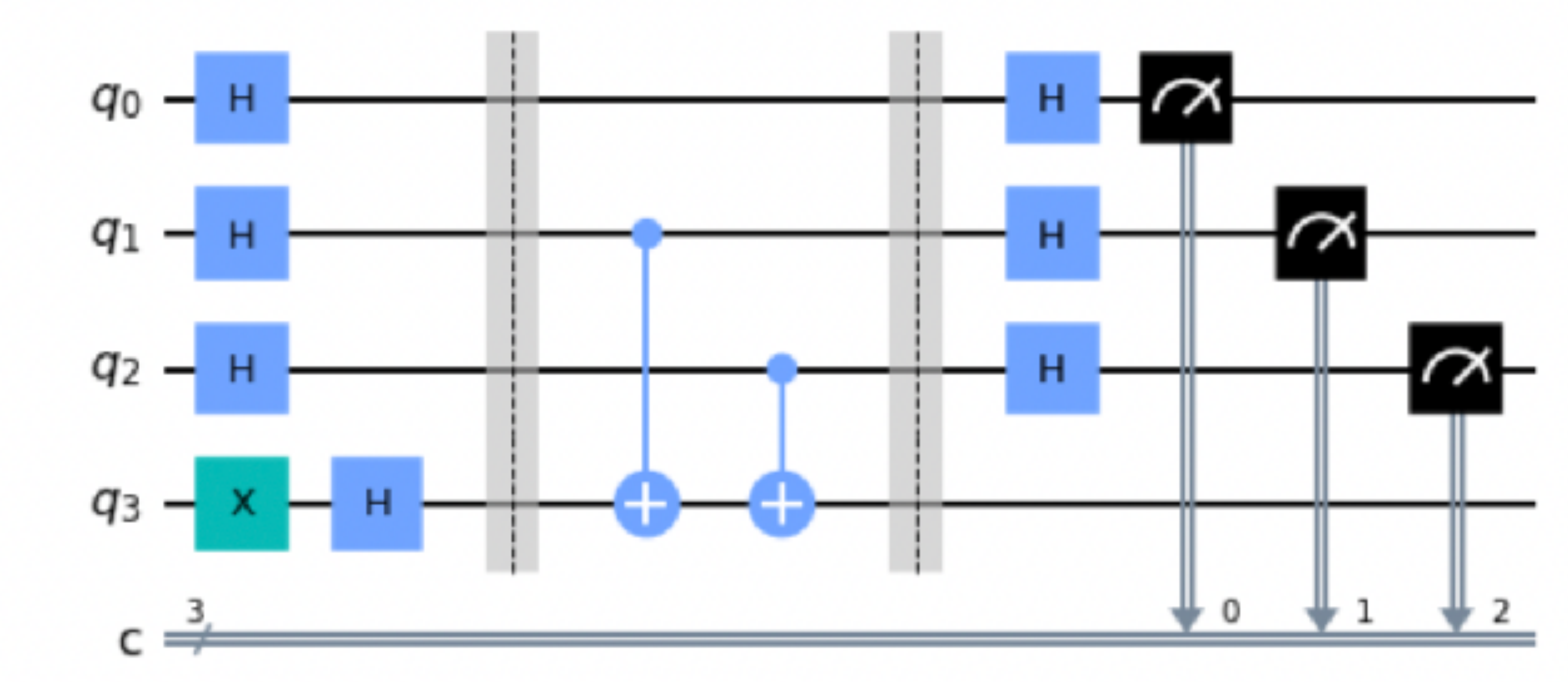
Guaranteed approximation algorithms through quantum annealing

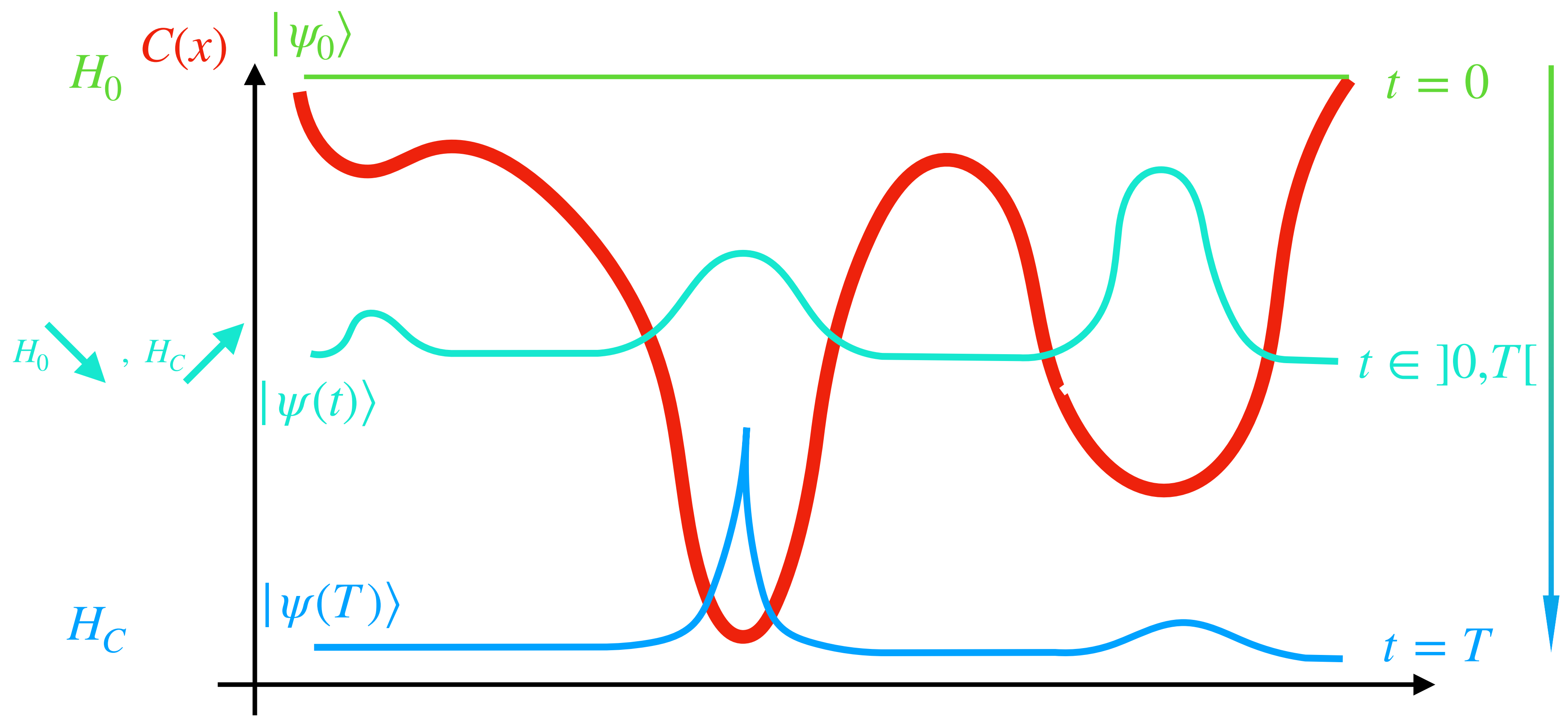
Tours, July 2023, Chiral Matter colloquium

Arthur Braida, Simon Martiel, Ioan Todinca

“Typical” results in quantum algorithms:

- Based on **quantum circuits** (quantum gates)
- Improve upon the time complexity of classic algorithms





This work : **quantum annealing**.

- Universal model for quantum computation
- Adiabatic theorem: if the process is **slow** enough, an optimal solution is obtained with high probability
- What can we do in **short** running time?
- Seeking for theoretical guarantees, not for quantum advantage

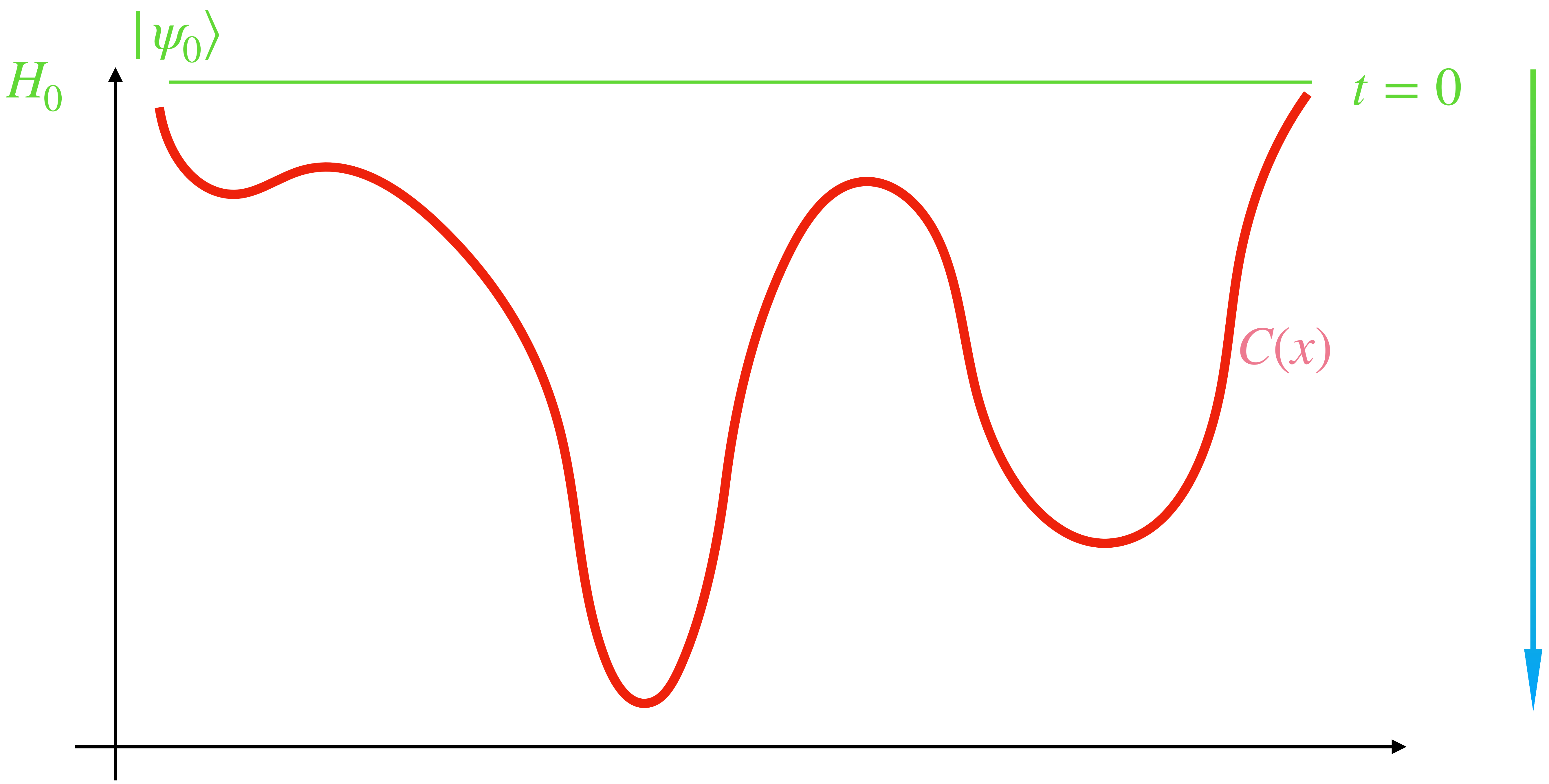
Computing with quantum annealing

Minimum of a function $C(x)$

1. Start with a simple Hamiltonian H_0, and a simple ground state $|\psi_0\rangle$
2. Target Hamiltonian H_C: its ground state corresponds to $\min_x C(x)$
3. Evolution from H_0 to H_C in time T

$$i\hbar\frac{\partial}{\partial t}|\psi^G(t)\rangle = H(t,G)|\psi^G(t)\rangle, \quad s(t) = \frac{t}{T}$$

$$H(t,G) = (1 - s(t))H_0 + s(t)H_C(G)$$



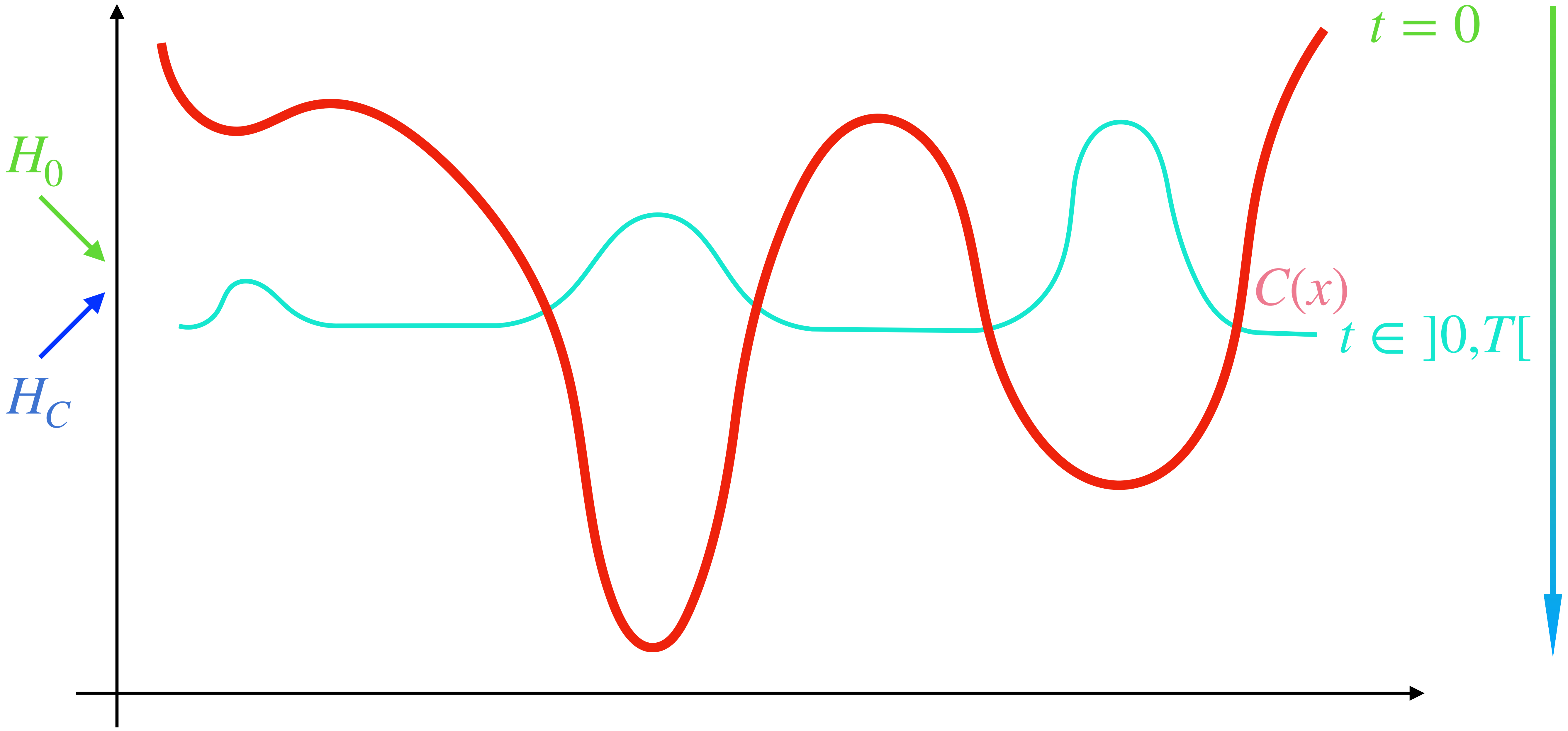
Computing with quantum annealing

Minimum of a function $C(x)$

1. Start with a simple Hamiltonian H_0, and a simple ground state $|\psi_0\rangle$
2. Target Hamiltonian H_C: its ground state corresponds to $\min_x C(x)$
3. Evolution from H_0 to H_C in time T

$$i\hbar \frac{\partial}{\partial t} |\psi^G(t)\rangle = H(t,G)|\psi^G(t)\rangle, \quad s(t) = \frac{t}{T}$$

$$H(t,G) = (1 - s(t))H_0 + s(t)H_C(G)$$



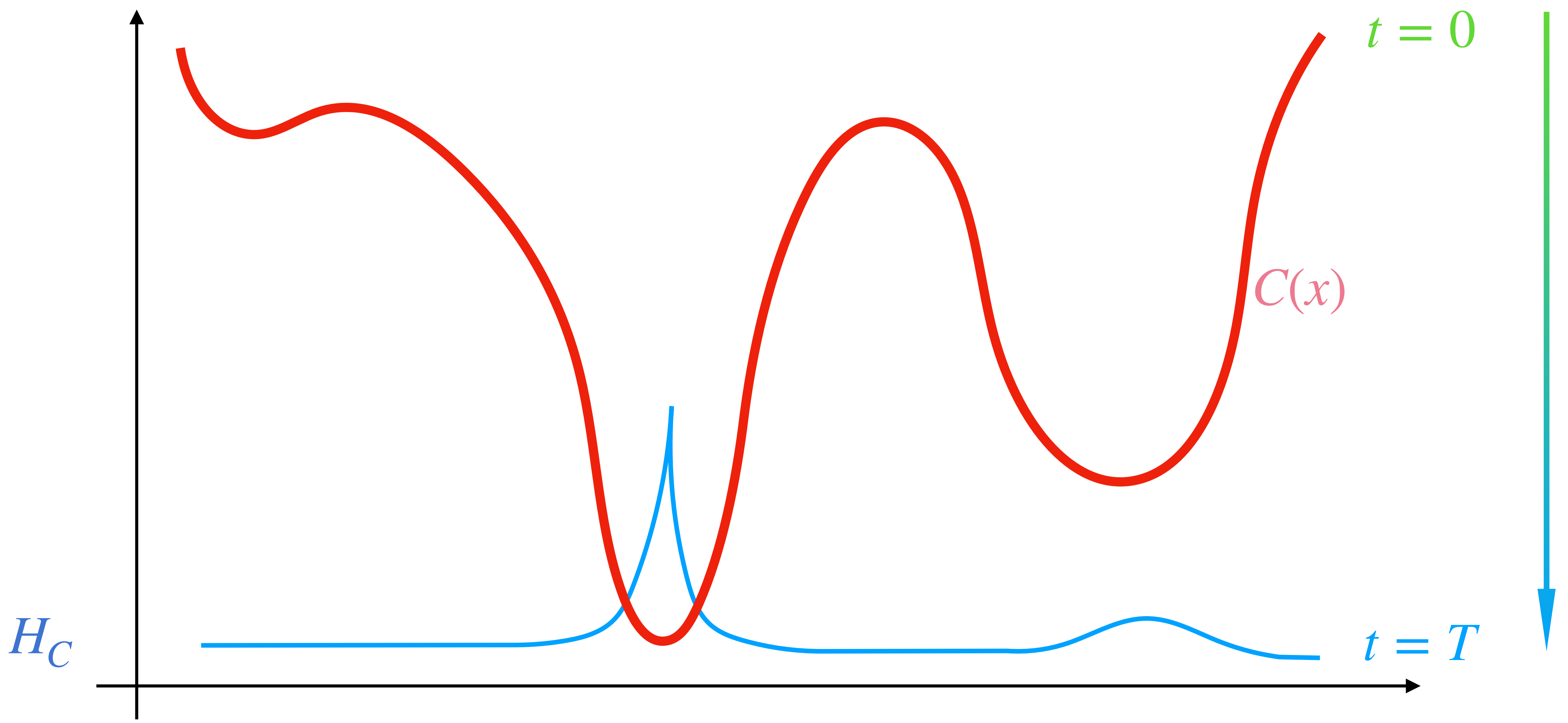
Computing with quantum annealing

Minimum of a function $C(x)$

1. Start with a simple Hamiltonian H_0, and a simple ground state $|\psi_0\rangle$
2. Target Hamiltonian H_C: its ground state corresponds to $\min_x C(x)$
3. Evolution from H_0 to H_C in time T

$$i\hbar\frac{\partial}{\partial t}|\psi^G(t)\rangle = H(t,G)|\psi^G(t)\rangle, \quad s(t) = \frac{t}{T}$$

$$H(t,G) = (1 - s(t))H_0 + s(t)H_C(G)$$



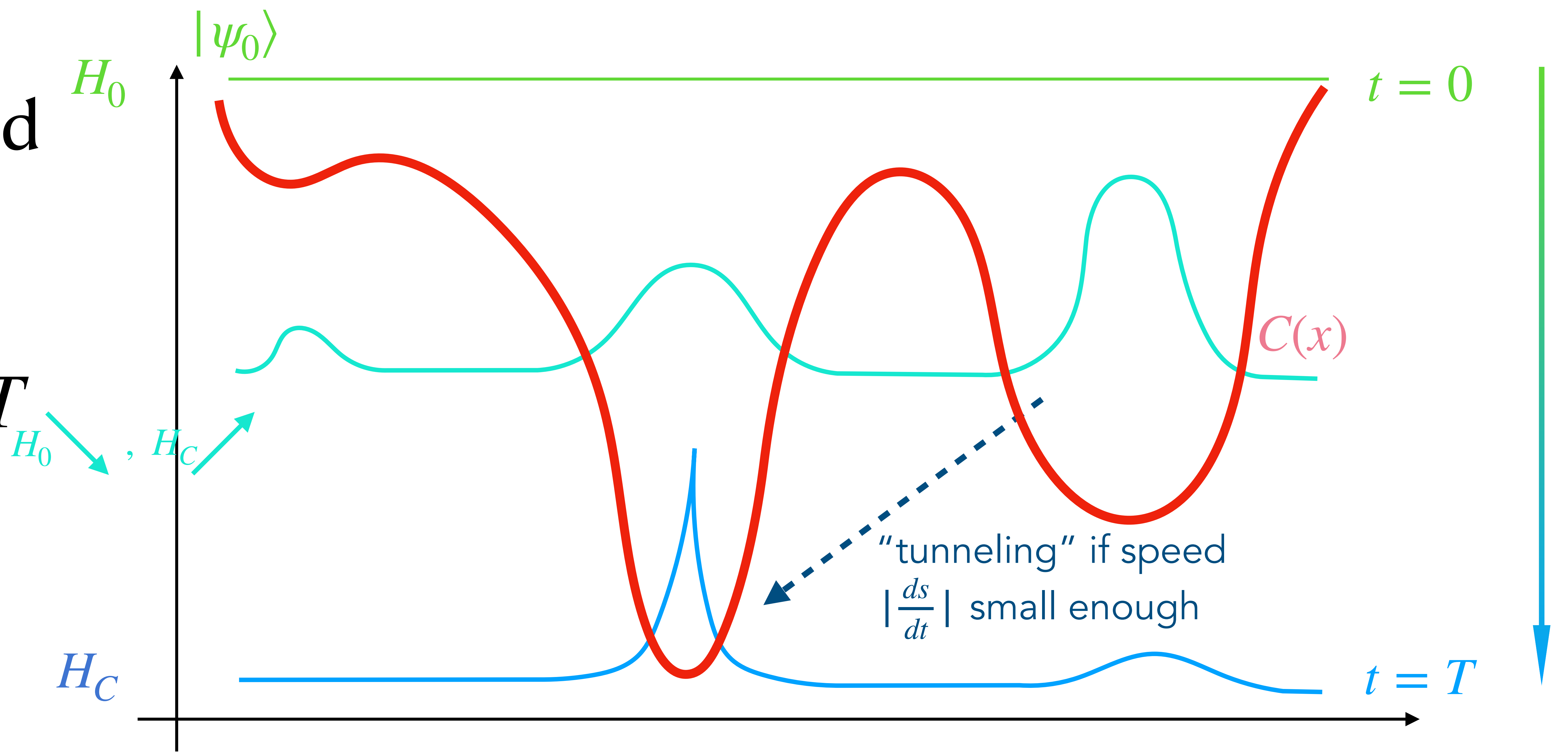
Computing with quantum annealing

$$i\hbar \frac{\partial}{\partial t} |\psi^G(t)\rangle = H(t, G) |\psi^G(t)\rangle, \quad s(t) = \frac{t}{T}$$

$$H(t, G) = (1 - s(t))H_0 + s(t)H_C(G)$$

Minimum of a function $C(x)$

1. Start with a simple Hamiltonian H_0, and a simple ground state $|\psi_0\rangle$
2. Target Hamiltonian H_C: its ground state corresponds to $\min_x C(x)$
3. Evolution from H_0 to H_C in time T

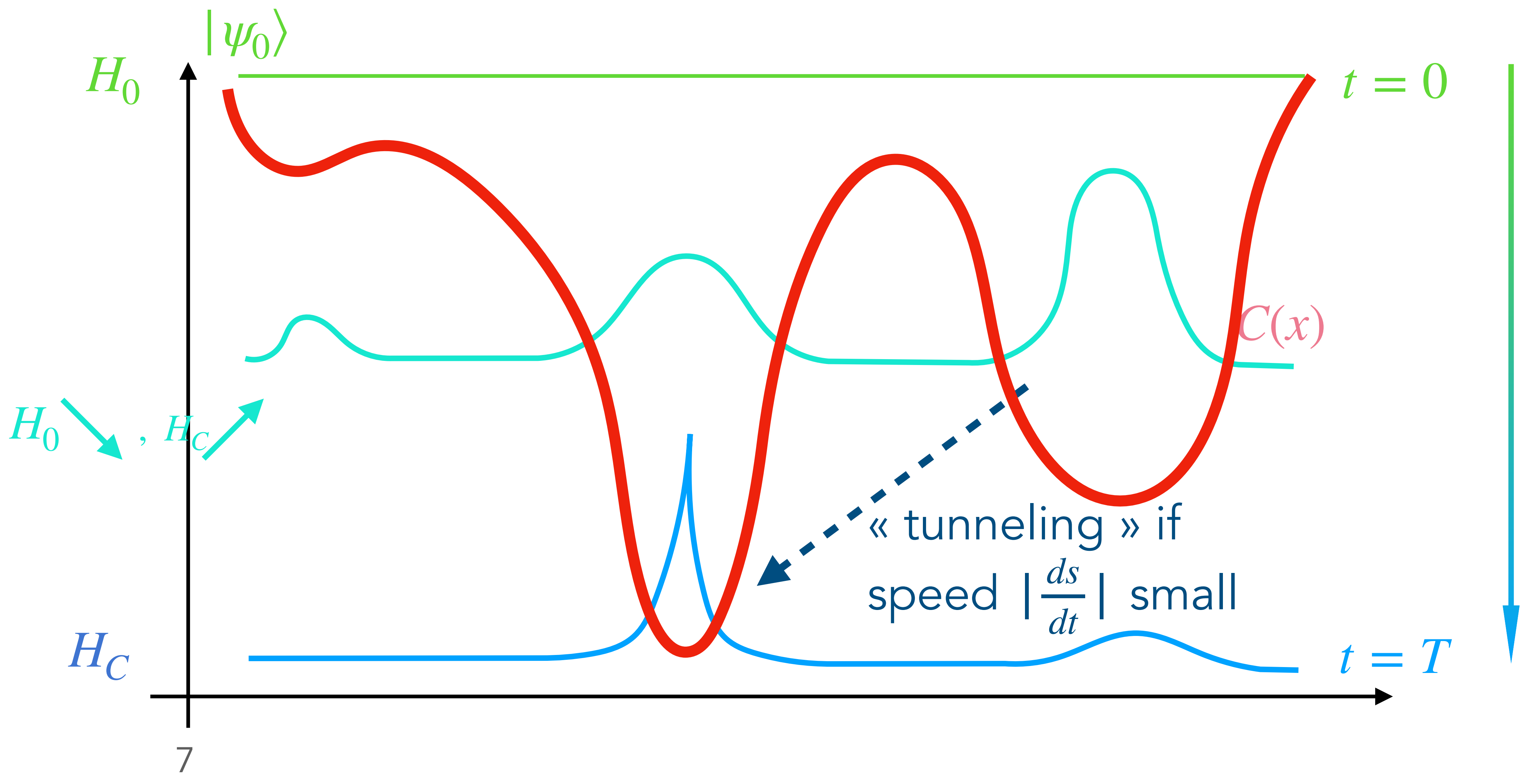


Convergence of QA

Adiabatic Theorem **:** if T is large enough, then the system stays in the eigen subspace from which it started the evolution.

$H(t) = (1 - s(t))H_0 + s(t)H_C$ with $s(0) = 0$ and $s(T) = 1$, e.g., $s(t) = \frac{t}{T}$

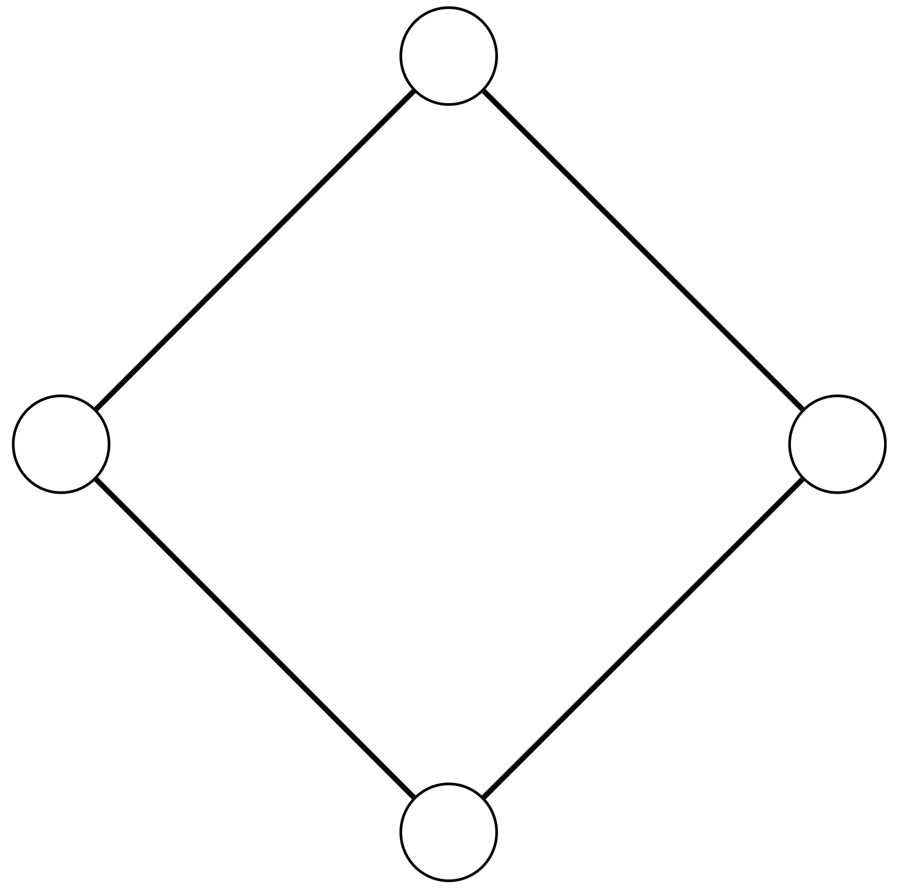
In practice **:** if $T \sim \mathcal{O}(\frac{1}{\Delta_{min}^2})$, where Δ_{min} is the minimum spectral gap of $H(t, G)$, then measuring $|\psi(T)\rangle$ gives, w.h.p., the state x which minimizes $C(x)$.



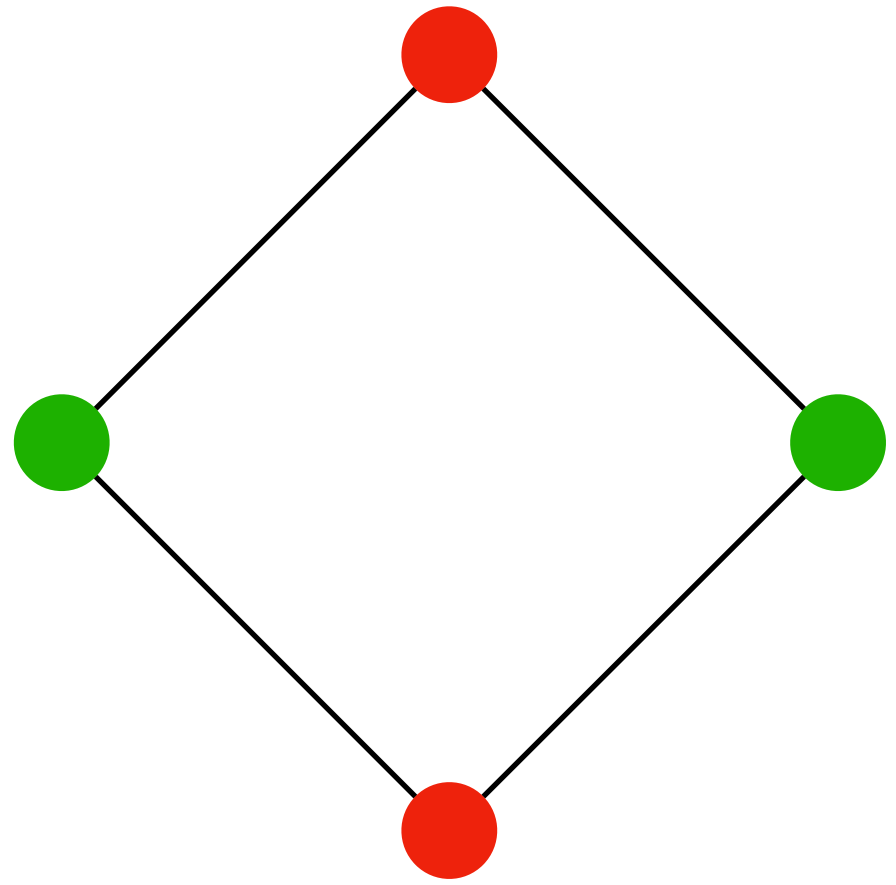
Typically, T should be exponential w.r.t. input size

Concrete application: MaxCut

1. Input: graph $G = (V, E)$.
2. Output: a bipartition of V
3. Objective: maximize the number of crossing edges



Alternatively: color the vertices in red/green, maximize the number of bicolored edges



MaxCut is NP-hard. There exists a polynomial approximation algorithm with approximation factor $\rho_{GW} = 0.878$ [Goemans, Williamson 1995].

I.e., for any input, the output cuts at least $\rho_{GW} \times OPT$ edges

MaxCut through QA

$C(x) = -\sum_{\{i,j\}\in E} x_i \oplus x_j$ where x is a bicoloration of the nodes V

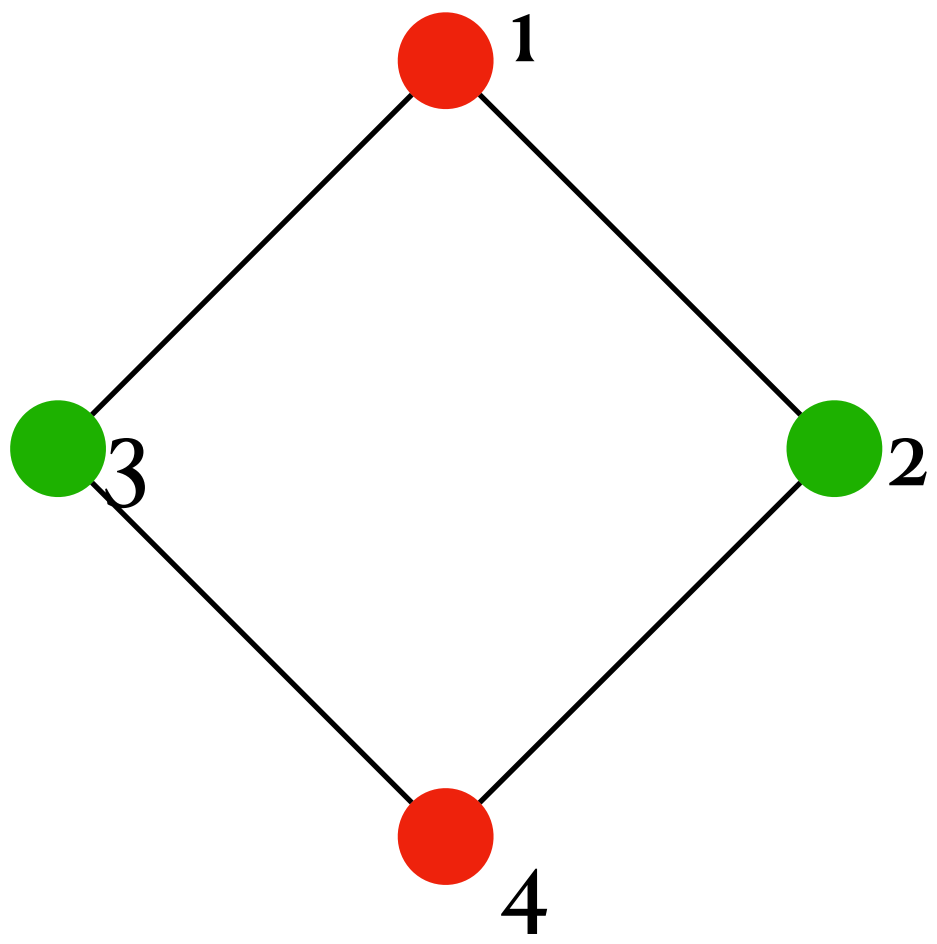
(0 and 1 in the computational basis)

$C(x) = -\sum_{\{i,j\}\in E} \frac{1 - z_i z_j}{2}$ for $z_i = (-1)^{x_i}$ (so z_i equals +1 or -1).

C counts the number of edges bicolored edges.

$H_0 = -\sum_i \sigma_x^{(i)}$ = -adjacency matrix of the hypercube

$$H_C(G) = -\sum_{\{i,j\}\in E} \frac{1 - \sigma_z^{(i)}\sigma_z^{(j)}}{2} = \sum_e O_e$$, counts the bicoloured edges



$x = 0101$

$C(x) = -4$

MaxCut through QA: approximation ratio

At the end of the process $t = T$, the measure gives one sample x of the final superposition $|\psi(T)\rangle$ with probability $|\langle x|\psi(T)\rangle|^2$.

For any probabilistic algorithm $\mathscr{A}$ that solves MaxCut C, we define the approximation ratio as the ratio of the expected output of $\mathscr{A}$ and the optimal value:

$$\rho_{MC}(\mathscr{A}) = \min_G \frac{\mathbb{E}_{\mathscr{A}}(C)}{C_{opt}}.$$

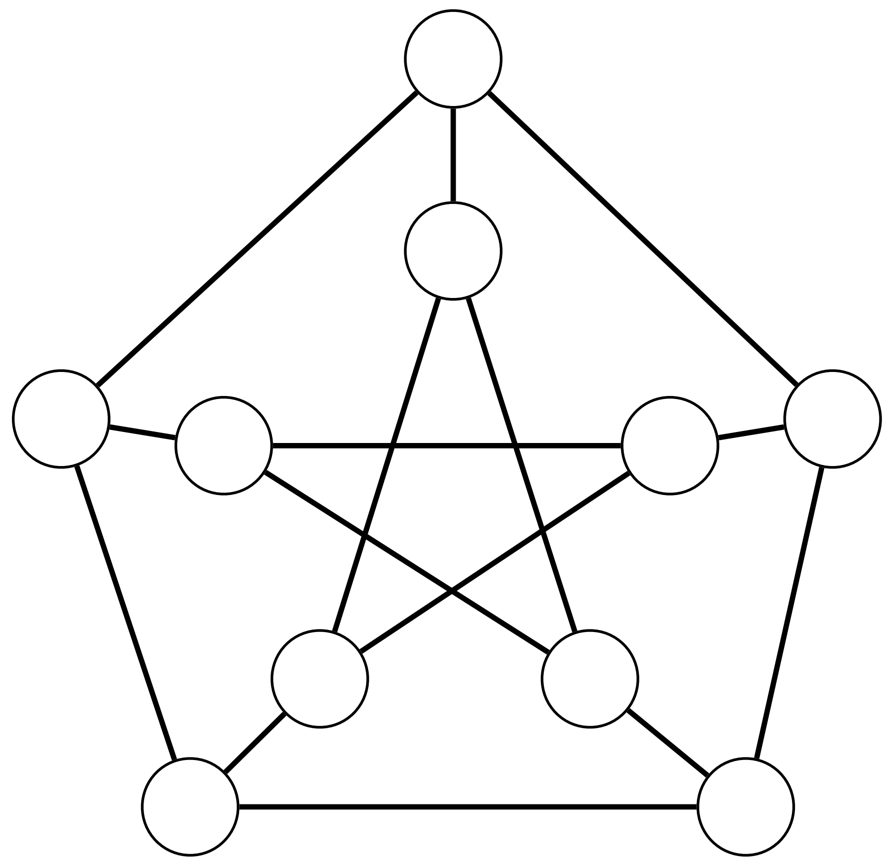
The goal is to find the worst possible ratio or an input independent lower bound.

For $\mathscr{A} = QA$, $\mathbb{E}_{QA}(C) = \langle\psi^G(T)|H_C(G)|\psi^G(T)\rangle = \langle H_C(G)\rangle_G = \sum_e \langle O_e\rangle_G$.

Our result

Theorem: QA at $T = 1.63$ achieves a 0.5933 approximation of MaxCut on *cubic graphs*, i.e., where all vertices have degree 3.

- better than the naive algorithm, which achieves 0.5…
- but weaker than other classic algorithms



Tools

- Short-time QA is "almost local"
- Cubic graphs: constant number of local configurations
- Lieb-Robinson type bound on the "difference" to locality

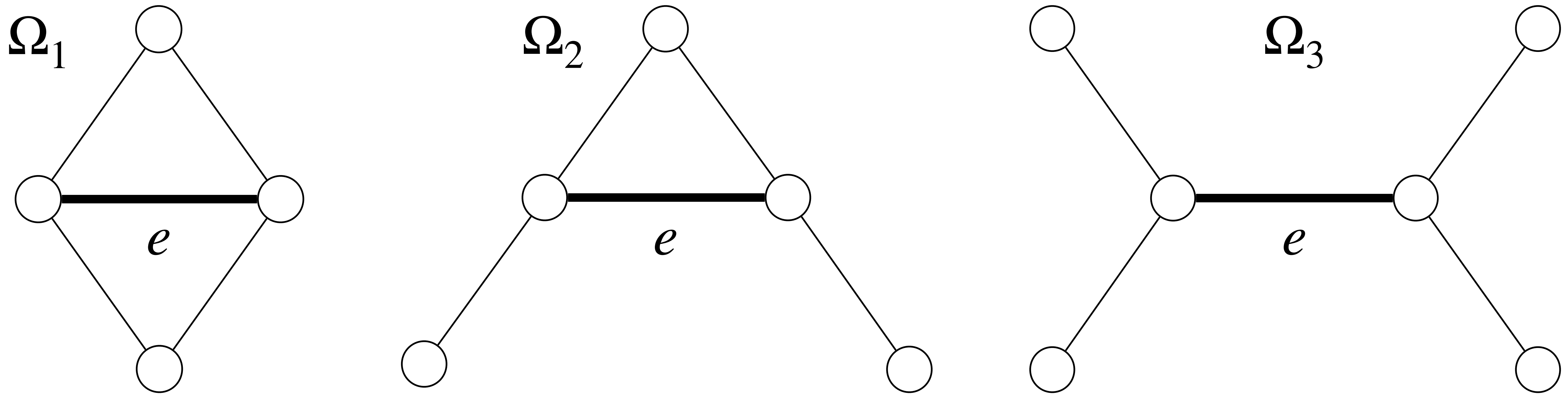
Take-away: LR-type bounds are "good enough" to guarantee approximation ratios for bounded degree graphs

Some details

Let $\mathscr{A}$ be a quantum algorithm, by linearity $\mathbb{E}_{\mathscr{A}}(C) = \langle H_C(G) \rangle_G = \sum_{e \in E} \langle O_e \rangle_G$. Assume :

(H1) Local Algorithm: $\langle O_e \rangle_G = \langle O_e \rangle_{B_e}$ where B_e is the ball of radius 1 around the edge e.

(H2) Focus on cubic graphs: any edge e has one of the following possible B_e:



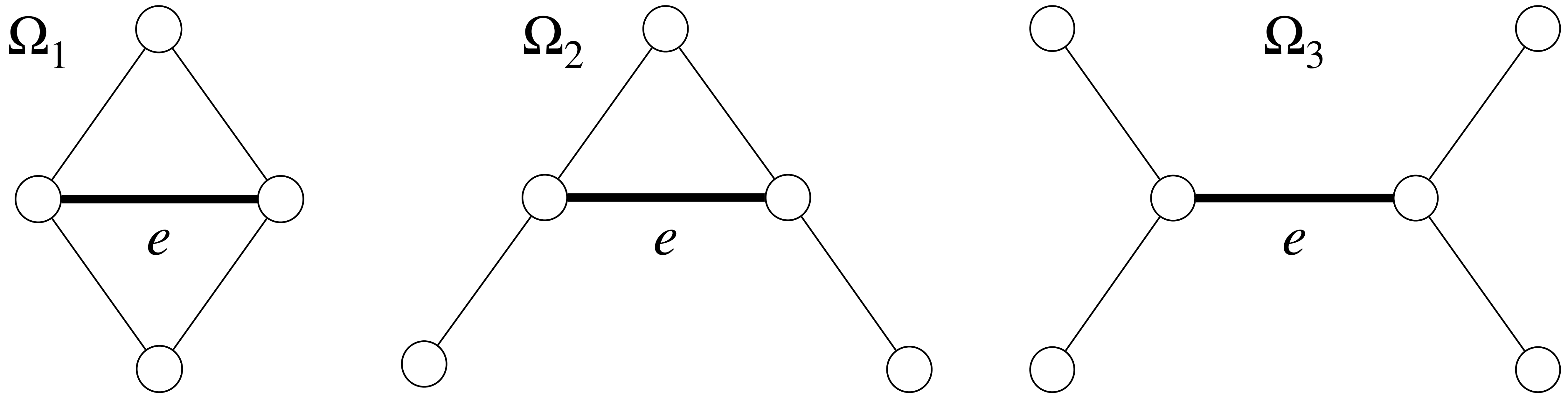
$$\mathbb{E}_{\mathscr{A}}(C) = \sum_{e \in E} \langle O_e \rangle_G \underset{\text{(H1)}}{=} \sum_e \langle O_e \rangle_{B_e}$$

$$\underset{\text{(H2)}}{=} S \langle O_e \rangle_{\Omega_1} + (4S + 3F) \langle O_e \rangle_{\Omega_2} + \left(\frac{3n}{2} - 5S - 3F\right) \langle O_e \rangle_{\Omega_3}$$

where S is the number of squares and F is the number of isolated triangles .

Some details

After optimization over possible values S and F, the worst ratio is achieved for $S = F = 0$, and the approximation ratio itself corresponds to $\langle O_e \rangle_{\Omega_3}$.



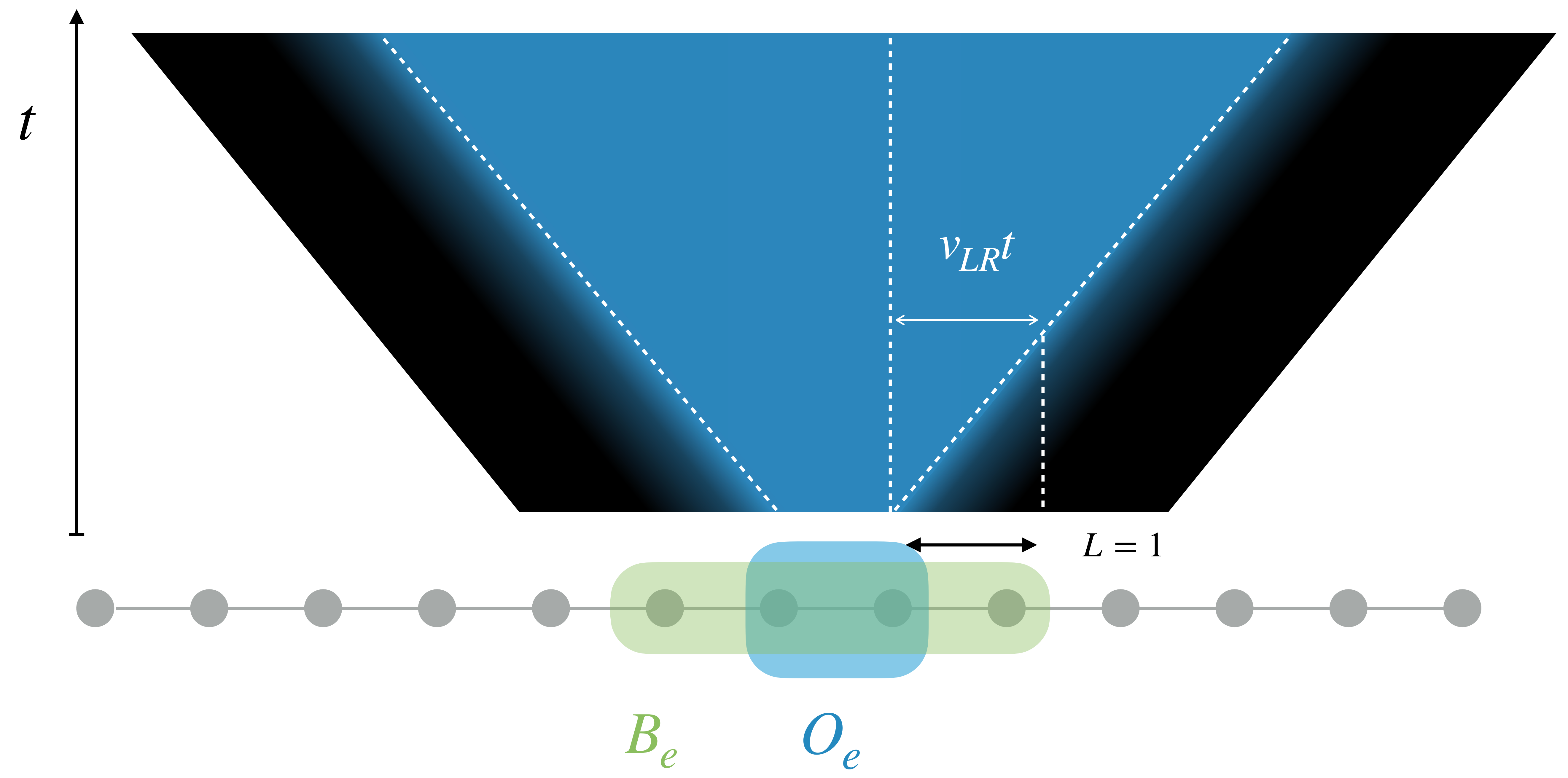
This only works under the assumption that the algorithm is local! Not quite true… but true up to 10^{-3} for our short (constant) running time.

Lieb-Robinson "like" bounds

QA is a priori non local, the unitary is mixing all qubits.

Lieb-Robinson bound (1972): bound on the speed of information flow.

Evolution of the support of $\langle O_e \rangle_G$



If t is small enough $\langle O_e \rangle_G \simeq \langle O_e \rangle_{B_e}$

⇒ **Corollary** (*almost* local) :

$$\langle O_e \rangle_G \geq \underbrace{\langle O_e \rangle_{B_e} - LR^{B_e}_{O_e}(t)}_{\langle O_e \rangle^*_{B_e}}$$

See article for explicit computation of LR.

Theorem: QA at $T = 1.63$ achieves a 0.5933 approximation of MaxCut on cubic graphs.

Conclusion and discussion

Quantum annealing in short (constant) time: guaranteed approximation for several optimization problems, for graphs of bounded degree.

Tools: pseudo-locality through Lieb-Robinson type bounds, limitation: bounded degree

See article for comparisons with QAOA (quantum approximate approximation algorithms), which **is** local

Personal frustration: MaxCut on cubic graphs has (exponentially) many solutions close to OPT. No tools for studying the probability that QA remains in the “lowest” states, for reasonable T.

Ongoing work: better bounds, study of “anti-crossings”, inputs on which QA can be efficient

Thank you!

Main contributor: arthur.braida@atos.net

arXiv:2202.01636

On constant-time quantum annealing and guaranteed approximations for graph optimization problems,

Published in Quantum Science and Technology