# Unstructured search problem

- $N$ items $\{1, 2, ..., N\}$
- One "marked item" $w$ represents the solution
- Query: "is $w = x$?" is a black box function

$$f(x) = \begin{cases} 0 & x \neq w \\ 1 & x = w \end{cases}$$

- **Classical**: $O(N)$

# Grover's quantum search

- Takes advantage of the **superposition** of states
- Algorithm consists in repeated application of the **Grover's operator** $U_G = U_\psi U_w$
- **Oracle** $U_w$: a black box that **marks** the solution with a phase

- **Quantum Grover**: $O(\sqrt{N})$

## Search procedure

- Preparation of the superposition state

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle$$

- Oracle application

$$|x\rangle \xrightarrow{U_w} (-1)^{f(x)} |x\rangle$$

- Application of $U_\psi$

$$2 |\psi\rangle \langle\psi| - I$$

- Readout

# Spatial search by quantum walks

- $N$ elements distributed in space, described by **$N$-vertex graph**
- Quantum walk dynamics determined by the **Laplacian** $L$ of the graph
- Marked state is **identified** by an **oracle** $H_w = -|w\rangle\langle w|$

$$H = -\gamma L - |w\rangle\langle w|$$

- **Complete graph**: $O(\sqrt{N})$

**Search procedure**

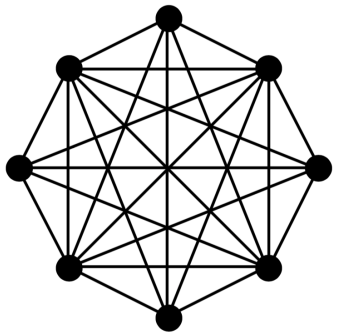
- Start in state $|s\rangle = \frac{1}{\sqrt{N}} \sum_j |j\rangle$
- Schroedinger evolve for time $T$ using the Hamiltonian $H$
- Measure the state
- Goal: choose $\gamma$, $T$ so that the probability $|\langle w|e^{-iHT}|s\rangle|^2$ is as close to 1 for the smallest $T$.

A. Child and J. Goldstone, Spatial search by quantum walk. *Physical Review A* (2004)

# Search by quantum walks

- $N$ elements distributed in space (e.g., a physical database)
- Model: $N$-vertex graph $G$
- Dynamics of the quantum walk on the graph is determined by the Laplacian $L$ of the graph

$$H = -\gamma L$$



**Adjacency matrix**:

$$A_{ij} = \begin{cases} 1 & \text{if } (i,j) \in G \\ 0 & \text{otherwise} \end{cases}$$

**Diagonal degree matrix**:

$$D_{jj} = deg(j)$$

**Laplacian matrix**:

$$L = A - D$$

# Search by adiabatic evolution

- The **adiabatic theorem** ensures that under certain conditions if a system evolves slow enough, it remains in its ground state
- It is used to solve computational problem via a **time-dependent evolution**:

$$H(s) = (1-s)H_i + sH_f$$

### Global adiabatic search

- Adiabatic theorem is applied globally
- Linear $s(t)$
- Time scaling: $O(N)$

### Local adiabatic search

- Adiabatic theorem is applied locally
- Non-linear $s(t)$
- Time scaling: $O(\sqrt{N})$

J. Roland and N. Cerf, Quantum search by local adiabatic evolution. *Physical Review A*

# Impossibility of adiabatic quantum walk search

- Both approches individually perform well, what happens if we combine them?

- Wong *et. al* (2016) showed that is is **impossible** to construct an **adiabatic-quantum walk** search algorithm (for the complete graph)
- It would need a stronger structure than the Grover's oracle

It leaves space for a time-dependent approach **inspired** by the adiabatic evolution, but **free** from the **constraints** of the adiabatic theorem

T. Wong and D. Meyer, Irreconcilable difference between quantum walks and adiabatic quantum computing. *Physical Review A* (2016)

# Time-dependent Hamiltonian

- We consider a time-dependent Hamiltonian **inspired** by the adiabatic evolution

$$H(s) = (1-s)L - s\gamma \left|w\right\rangle \left\langle w\right|$$

where $L$ is the **Laplacian** of the graph, $s$ is the **interpolating schedule** and $\left|w\right\rangle \left\langle w\right|$ is the **oracle** Hamiltonian

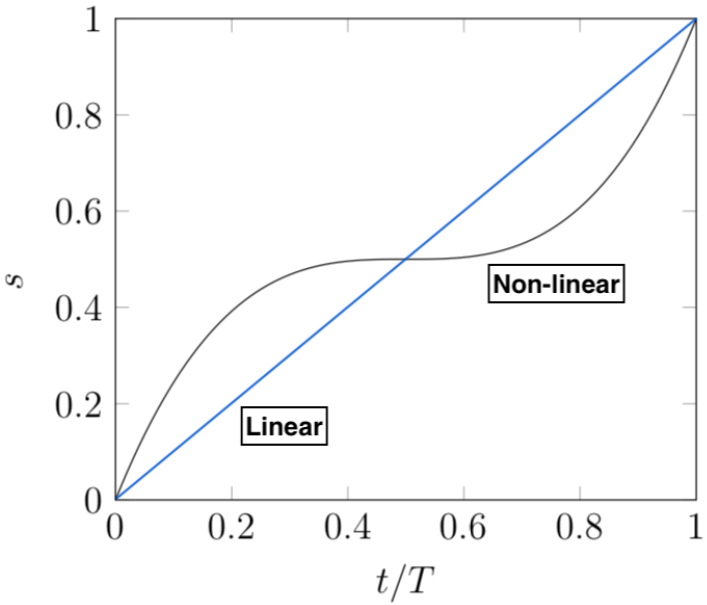
- The evolution of the state is determined by solving the schroedinger equation

$$i\frac{d}{dt}\left|\psi(t)\right\rangle = H\left|\psi(t)\right\rangle$$

with the necessary boundary conditions.

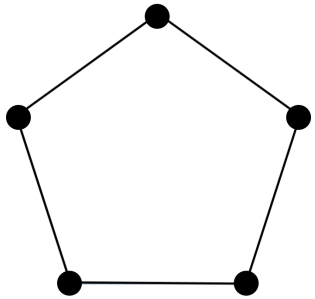
# Interpolating schedule and multiple runs for one search

- We considered **linear** $s_L(t)$ and **non-linear** $s_{NL}(t)$ interpolating schedules




- If the search is **imperfect** (solution found with $p < 1$) we consider the possibility of repeating the search **multiple times**

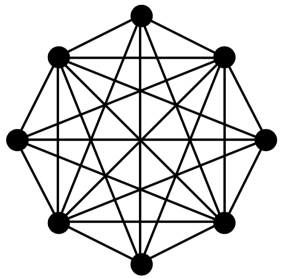
# Selected graphs: cycle and complete



**Cycle graph** $Cy(N)$

Worst case scenario

- Search problem is **not solved** with the standard quantum walks approach



**Complete graph** $C(N)$

Best case scenario

- Search problem **solved** with standard quantum walks approach
- The equivalent *unstructured search* is **solved** with the local adiabatic evolution

# Results characterization: search, localization and robustness

**Search**
Finding of the solution with high probability for the **smallest** $T$ possible

**Localization**
Finding of the solution with high probability **without** the need to **minimize** the time

A description of the **localization** is necessary given the **adiabatic nature** of the algorithm

**Robustness $R_\gamma$ and $R_T$**
Quantifies the variation of the probability due to variation of $\gamma$ and $T$

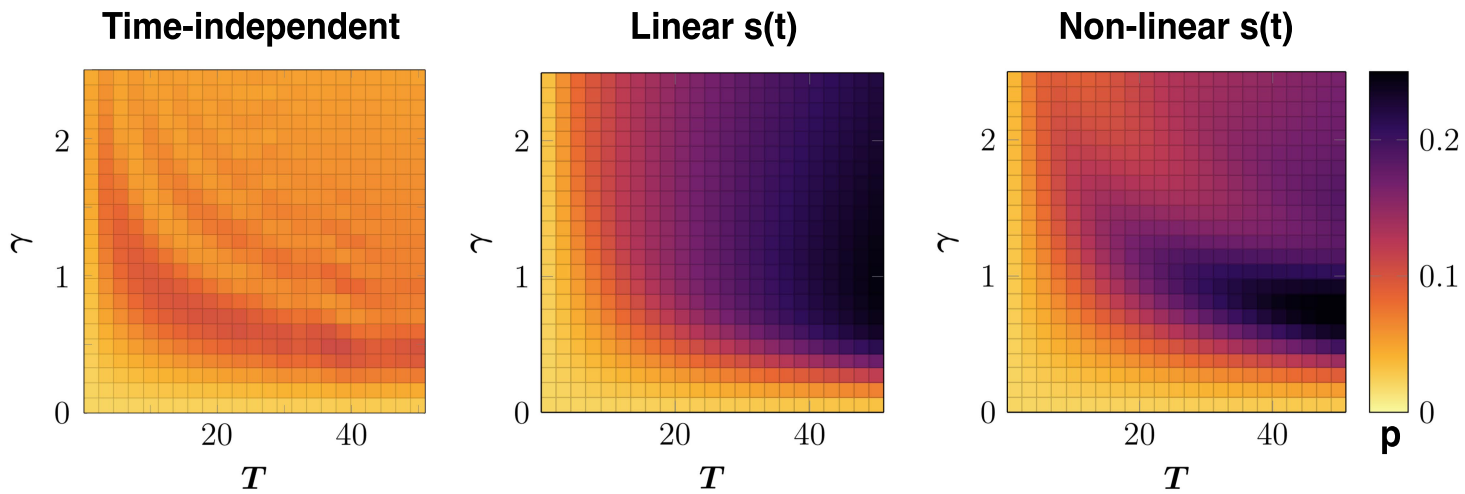
$$R^{\pm} = p(T, \gamma) - p(T, \gamma \pm \delta)$$

$$R_\gamma = \left[\frac{R^+ + R^-}{2}\right]$$

# Cycle graph: probability grid evaluation

We compared the **time-independent** approach and the **time-dependent** one with **linear** and **non-linear** $s(t)$

- Computed the probability over a **grid** for the different combination of $T$ and $\gamma$, with graphs up to $N = 71$
- Intuitive way to visualize the results: **probability heatmaps** (in figure $N = 51$)

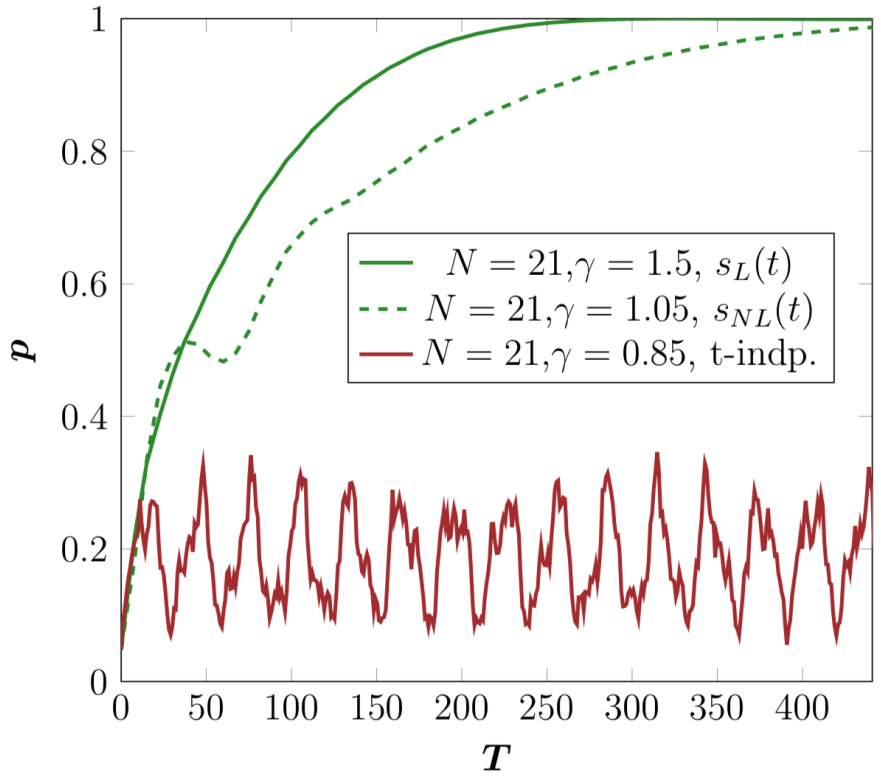


# Cycle graph: localization

## Time-dependent approach shows localization properties

- The time-independent approach (red) does **not** have **localization** properties

- The time-dependent approach (green) is able to achieve $p = 1$, although for **large** $T$ ($\approx N^2$)




- Time-dependent approach produces high probability ($p < 1$) in much less time than for $p = 1$
  (e.g. for $s_L(t)$ $p = 1$ at $T = 300$, $p = 0.9$ at $T = 150$)

# Cycle graph: search

## Multiple runs search with constraint on $T$

- We consider the multiple runs for one search approach
  (*) Time-dependent is not optimized on $T$
  (*) Time-independent is an imperfect search
- We introduce a **new quantity** $\tau$ in order to compare the two approaches

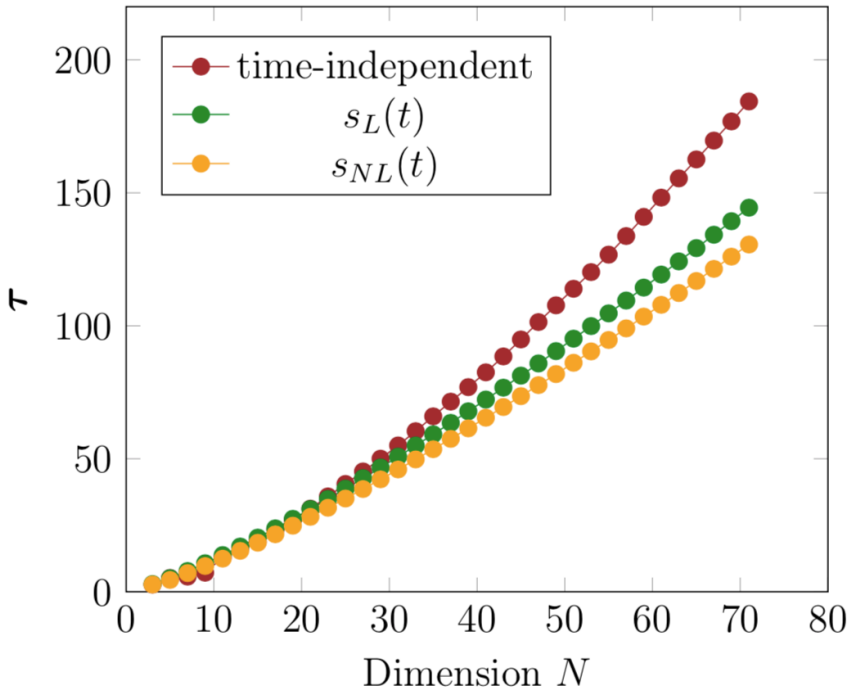
$$\tau = \min \left( \frac{T}{p} \right)_{T > T_{\min}, \gamma}$$

- $T_{\min}$ is a constraint on the minimum time
- $T_{\min} = \frac{\pi}{2}\sqrt{N}$

# Cycle graph: search (2)

## Time-dependent performs better than time-independent

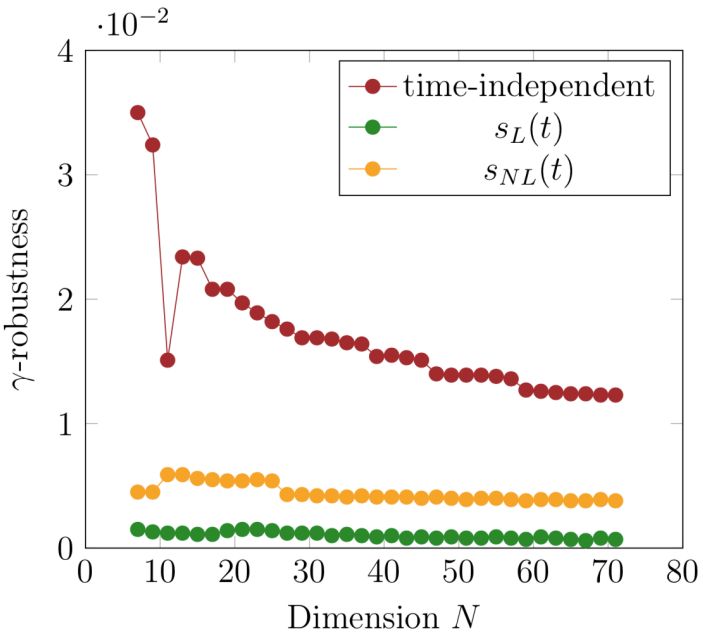
- Similar performance for small $N$ ($< 35$)
- Time-dependent approach wins for large $N$
- Non-linear $s(t)$ performs slightly better than the linear one

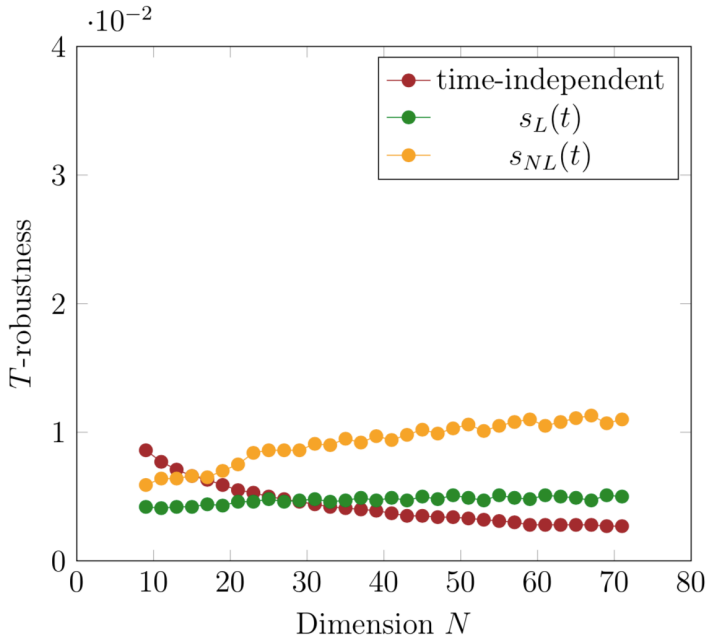


# Cycle graph: robustness

## Time-dependent approach with linear $s(t)$ is the most robust

- Robustness is evaluated for a $T$, $\gamma$ variation of 2 **grid blocks**
- **$\gamma$-Robustness**: time-dependent is best, with linear $s(t)$ the most robust
- **$T$-Robustness**: surprisingly the time-independent is more robust for large $N$, although the values are very similar

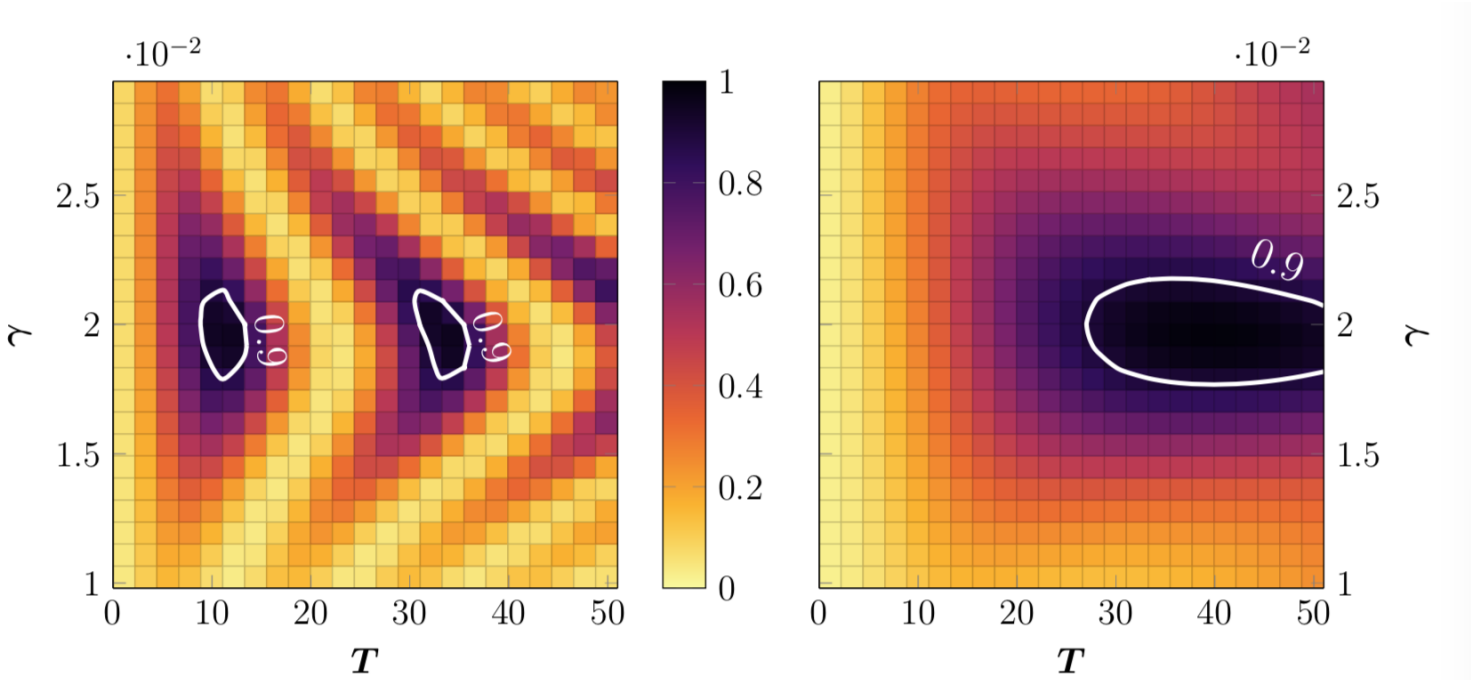


# Complete graph: probability and qualitative robustness

- Time-dependent is more robust but with much worse time scaling

- For completeness we study the complete graph
- We do not expect to be able to improve an already optimal approach
- We might get some insights on the performance of our algoritihm
- We study the **qualitative robustness** (in figure $N = 51$)

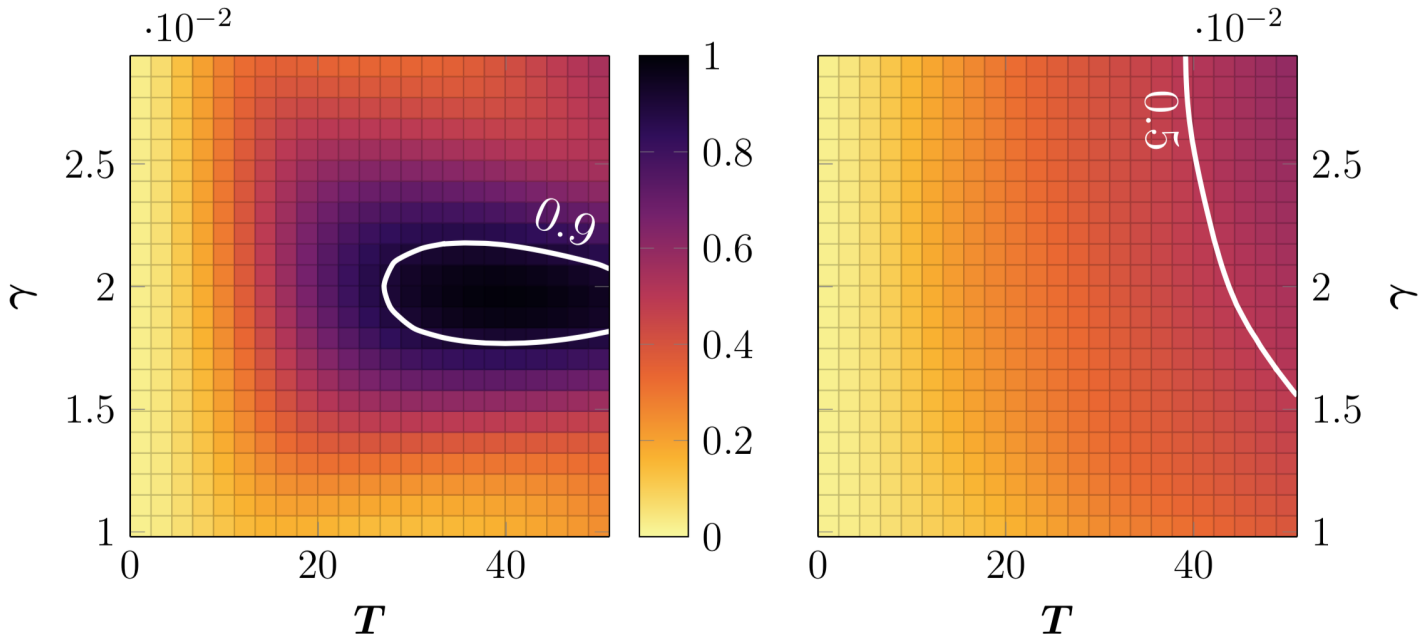



(a) time-independent

(b) $s_{NL}(t)$

# The choice of $s(t)$ is crucial

- As already mentioned the choice of $s(t)$ is crucial
- Improvements on the performance come from the choice of the optimal $s(t)$
- Linear $s(t)$ performs **worse** than **classical** algorithms (in figure $N = 51$)




(a) $s_{NL}(t)$ (b) $s_L(t)$

# What have we learned?

Results from applying the time-dependent Hamiltonian algorithm

**Cycle graph**

- Shows localization properties
- Is more robust
- Better search performance overall
- Non-linear $s(t)$ performs better but is less robust

**Complete graph**

- Is (qualitatively) more robust
- Performs much worse than Grover
- Performs better than classical
- Shows that $s(t)$ has great impact on the performance

# What is next?

- The application of the algorithm to the complete graph suggests that **improvements** on the performance come from the choice of the **optimal** $s(t)$

- Application to **different graph topologies**

**Thanks for your attention**