## Brief Announcement:

# Time-Travel Planning with Tenet Turnstiles

Thibaut Blanc    Jean-Romain Luttringer    Quentin Bramas    Sébastien Tixeuil

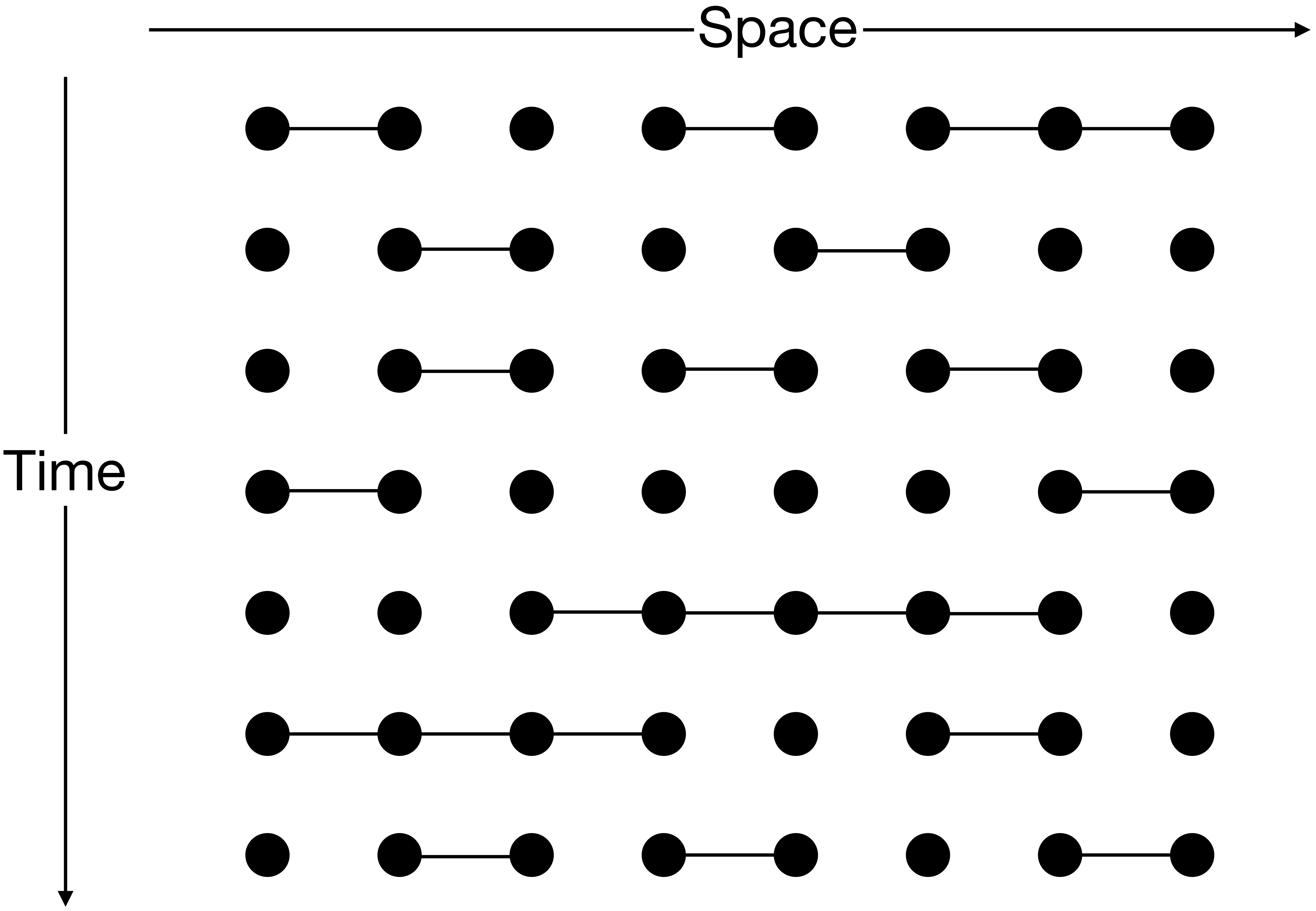
ENS DE LYON
Université de Strasbourg
SORBONNE UNIVERSITÉ

SAND 2026

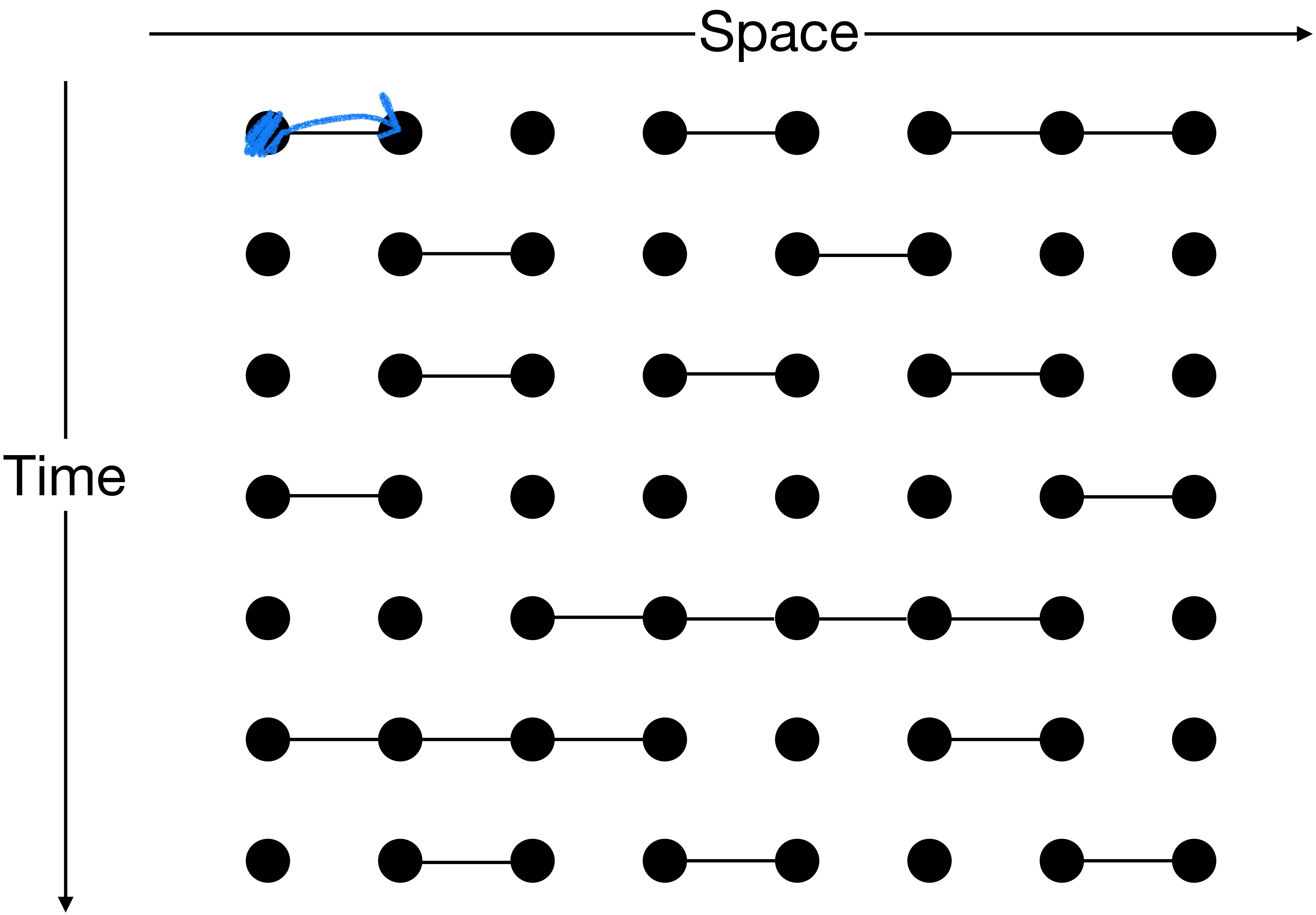
# Conclusion

Interesting gap under the Space-online setting

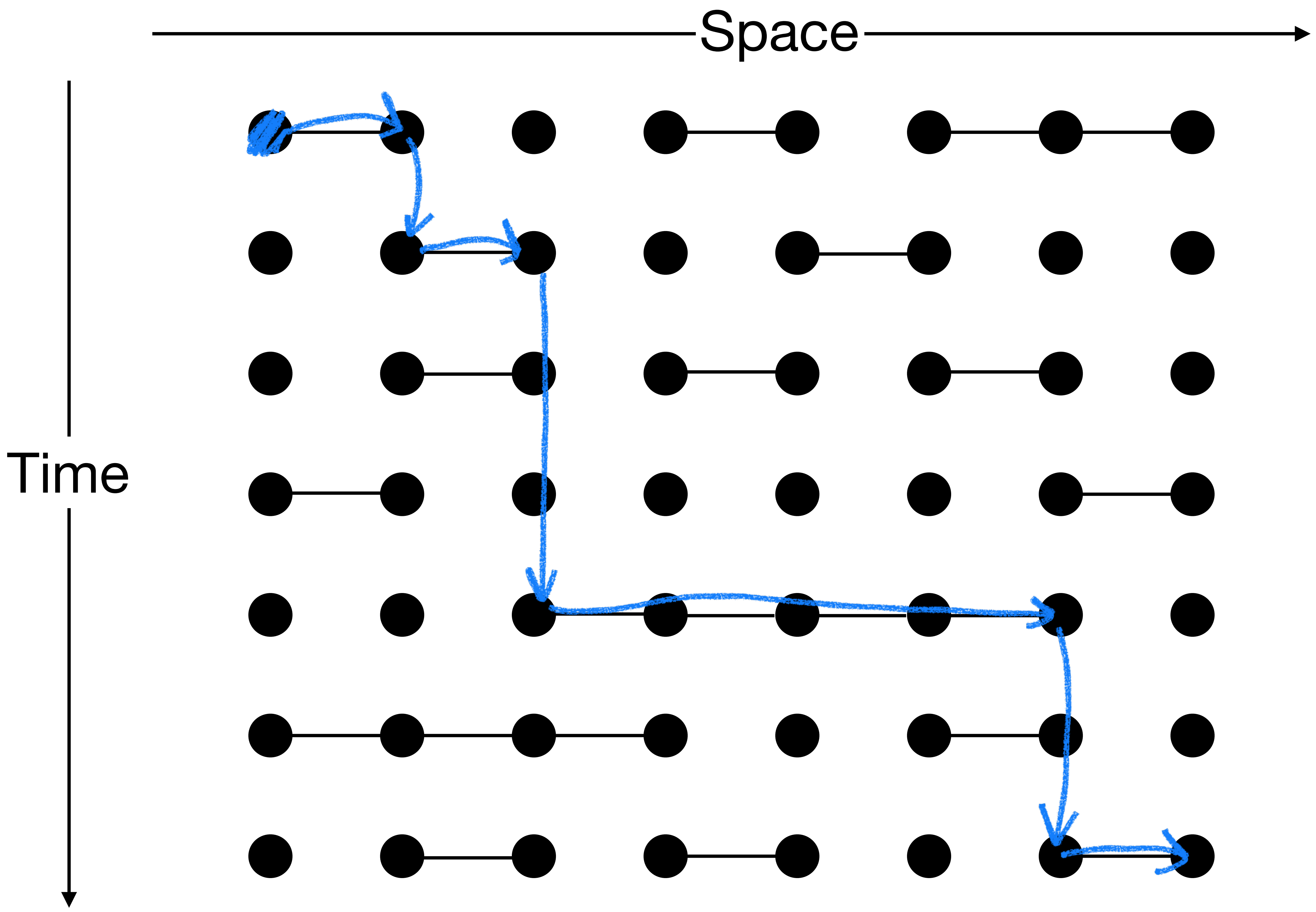
# Evolving Graphs



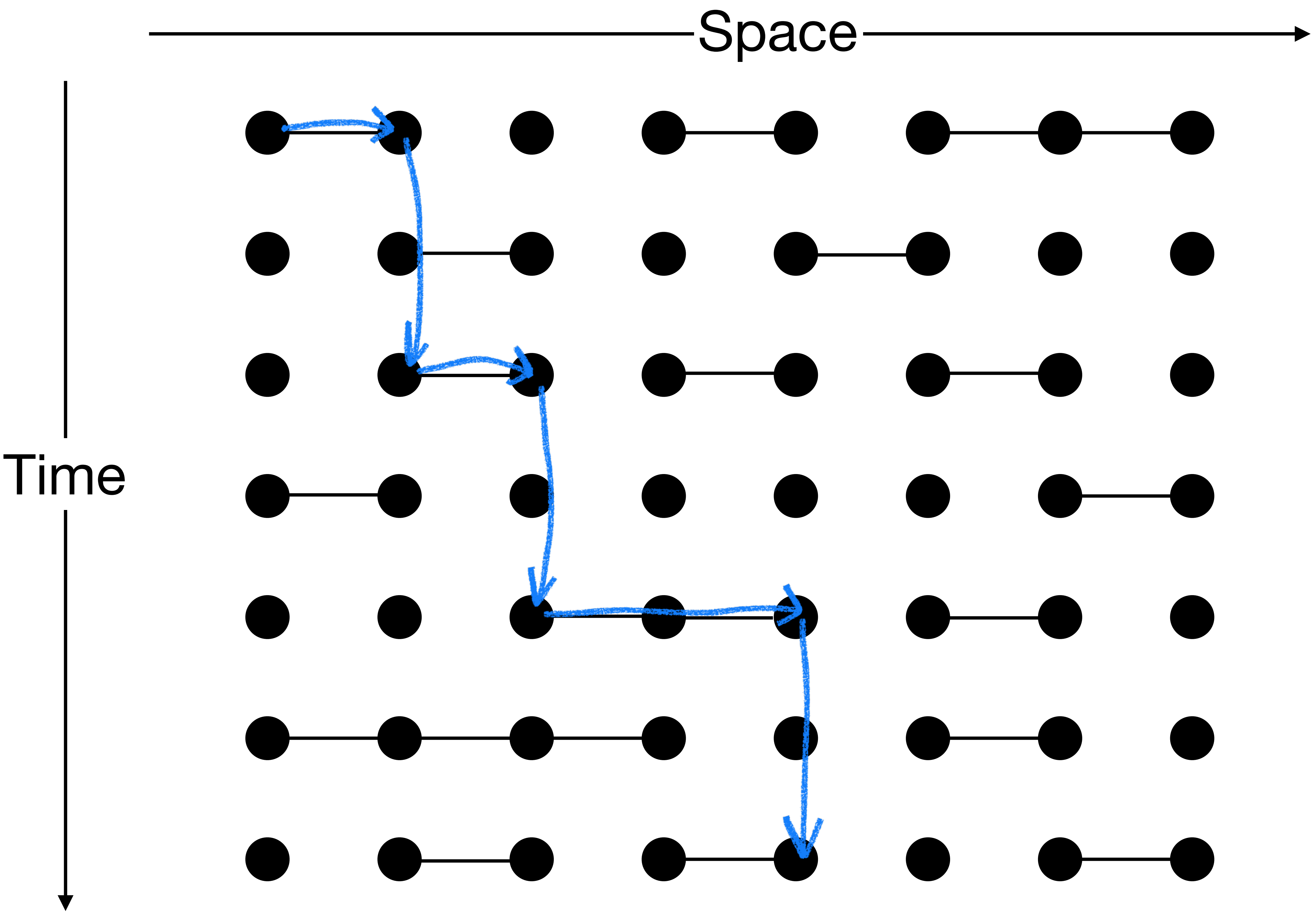
# Journey

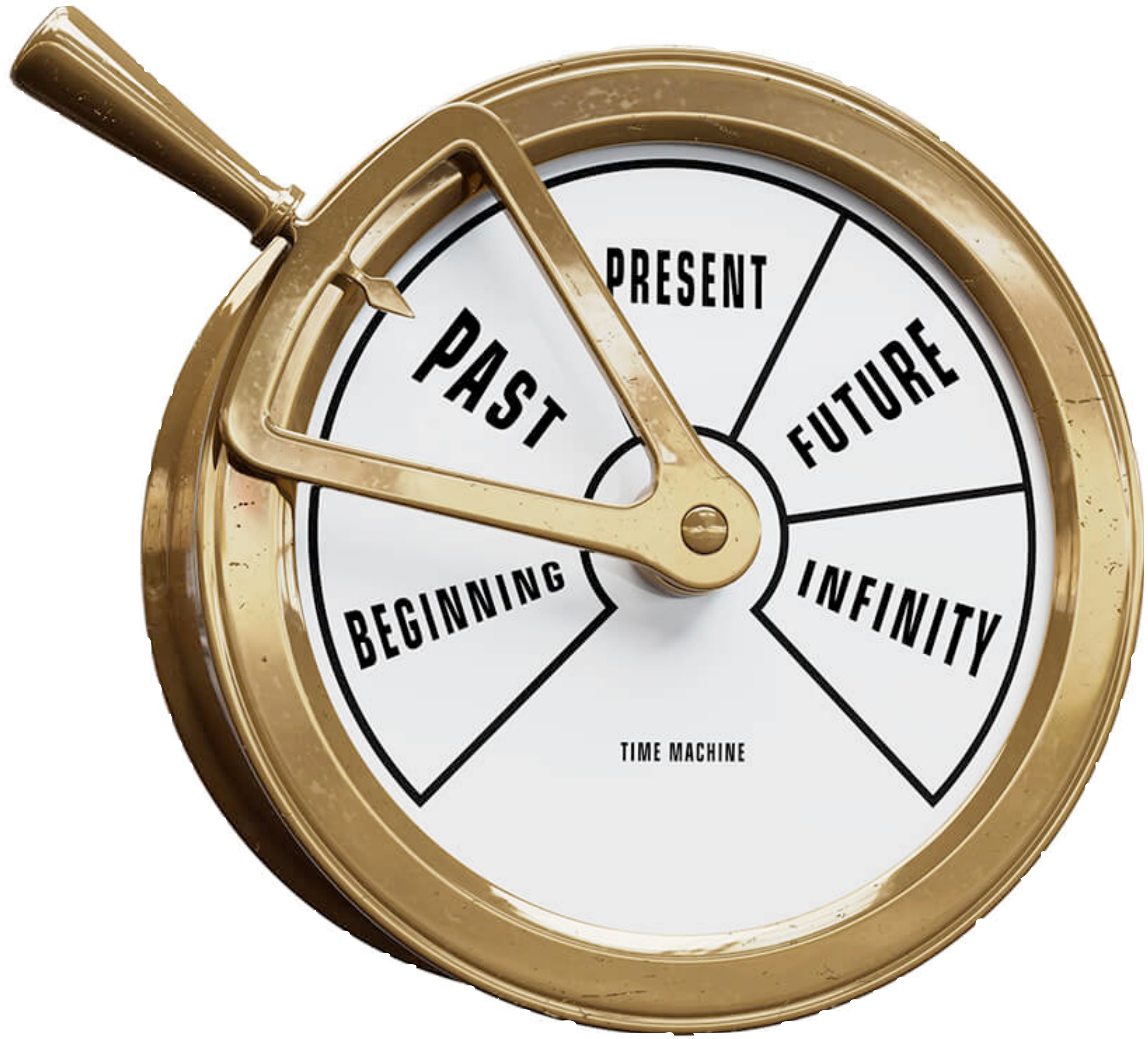


# Journey

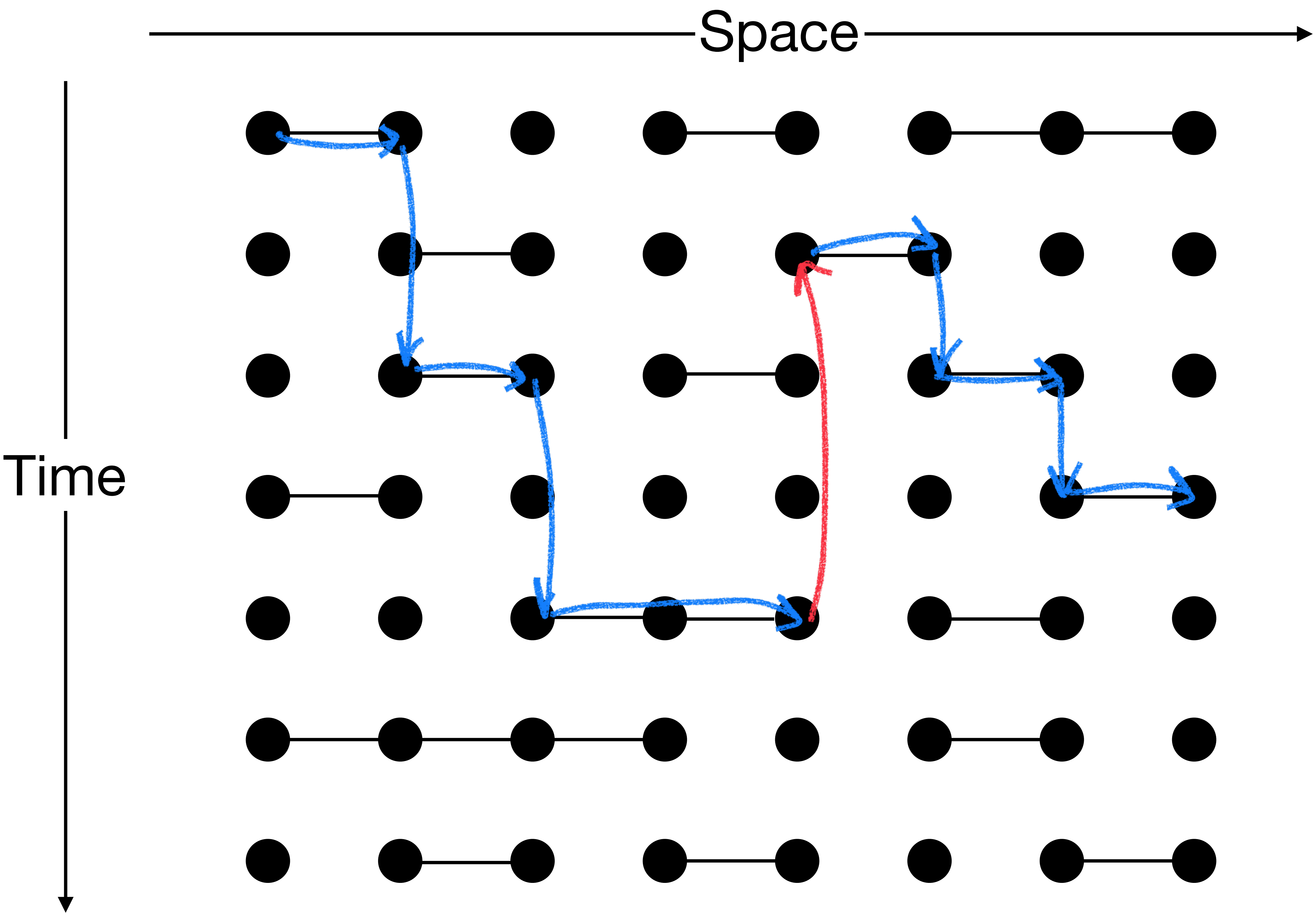


# Journey

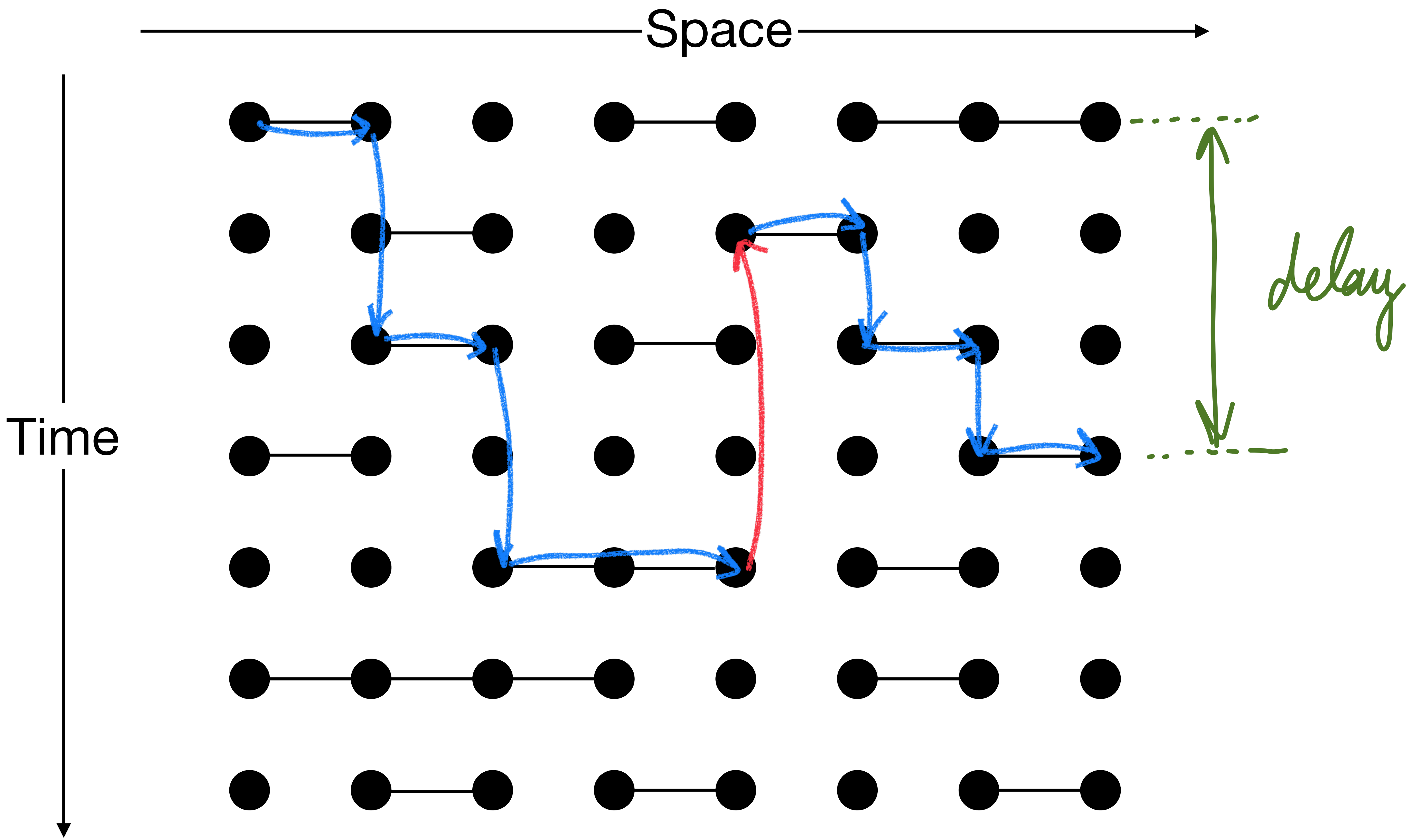



PRESENT
PAST
FUTURE
BEGINNING
INFINITY
TIME MACHINE

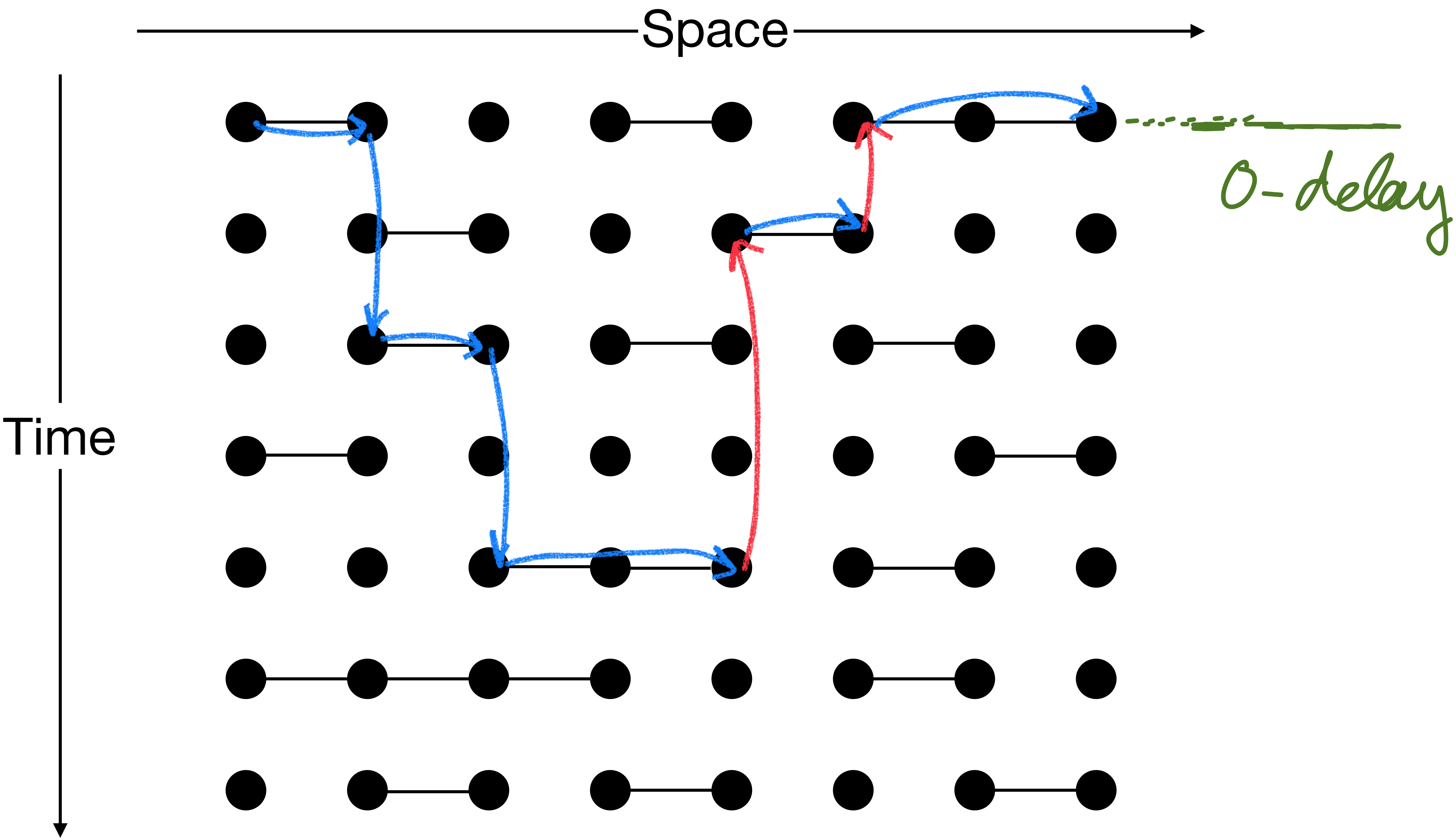
# Space-Time Travel



# Travel Delay



# Travel Delay



# Travel Cost

Travelling back in time requires **Chronotonium.**

The price recently increased, because the **Markarian 771 Wormgate** has been blocked for three galactic quarters.

How cost is computed ?

# Travel Cost

**SSS2023, SAND2024:** Cost depends on how far back you travel:

- Back to the Future (1985): fixed cost (1.21 GWatts)
- Primer (2004): linear cost

**SAND2026:** You pay when you reverse direction of time

- Tenet (2020)

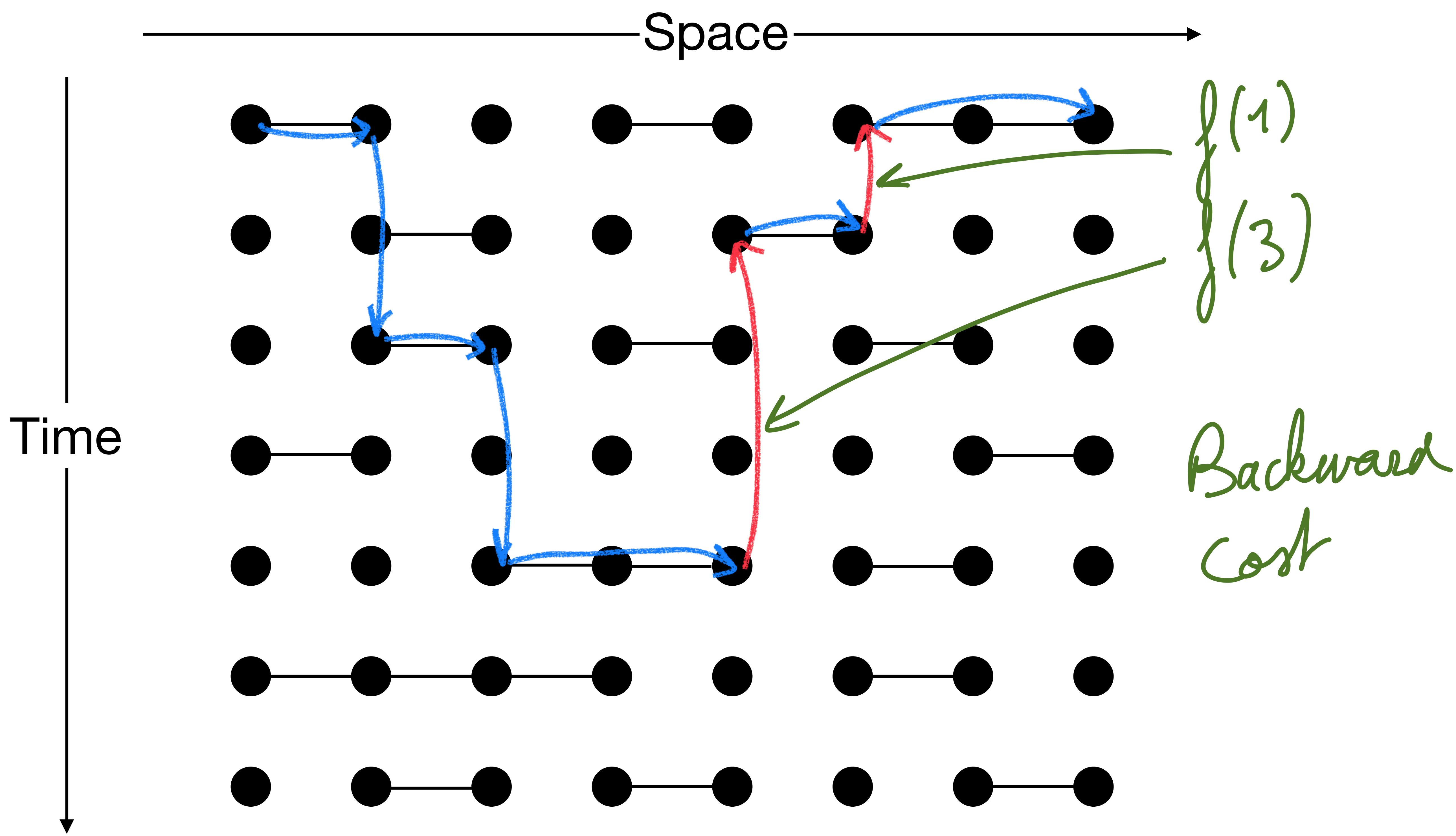
**SAND2028:** Space-time travel device **(FIFA Best paper award)**

- Dr Who
- Wormhole

Great survey in TCS MacDonald Edition 2031

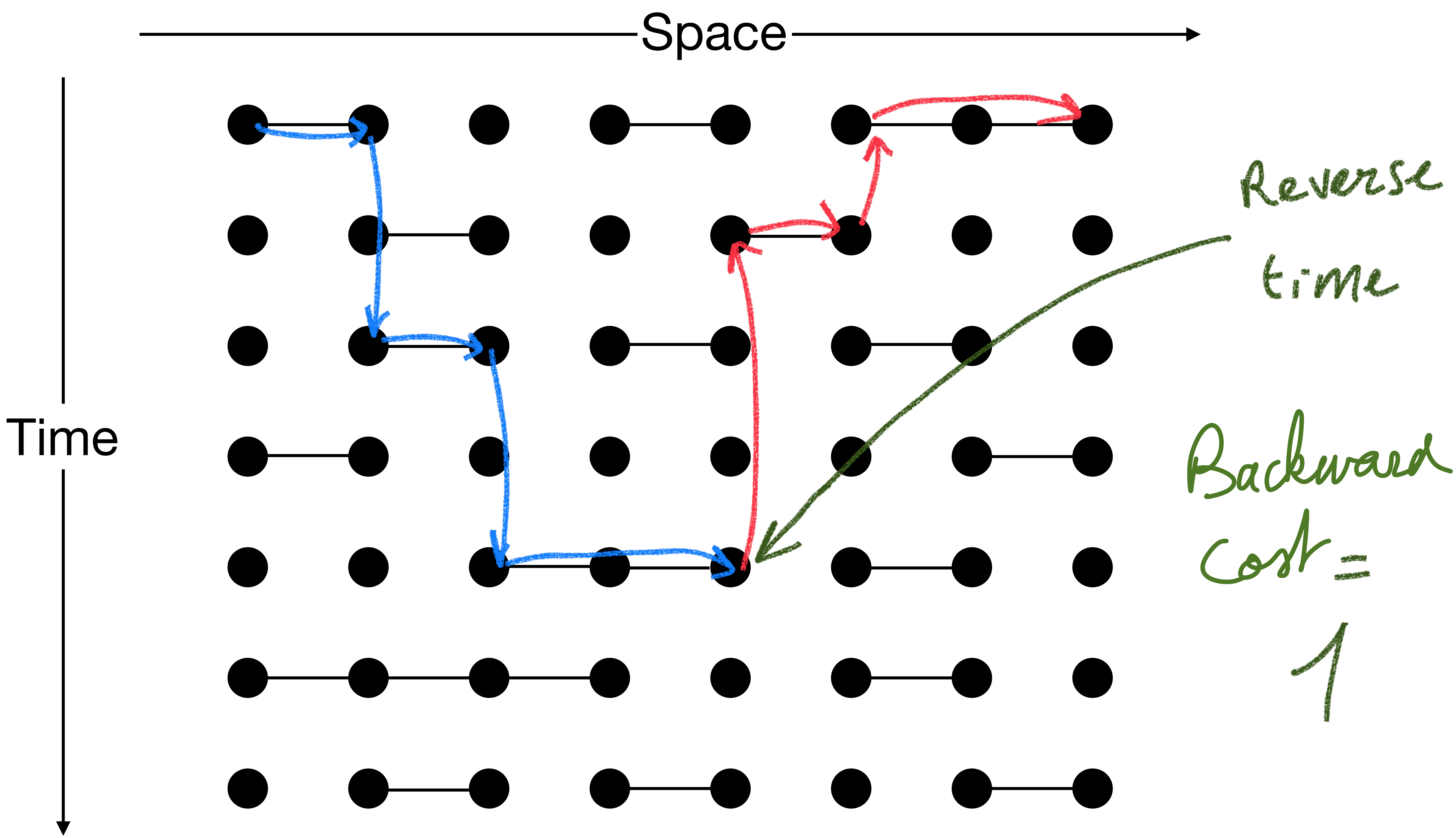
# Travel Cost

as a function of backward time



# Tenet Turnstiles

# Travel Cost as in TeneT



# The offline ODOC Problem

$$\text{cost}(T) = \#\ \text{time inversion}$$

Given an evolving graph G, a source, and a destination

Find a 0-delay space-time travel with lowest cost.

# The offline ODOC Problem

$$O(m \log(n))$$

$m$ : # edges in underlying graph

(if we can request next / previous edge in $O(1)$)

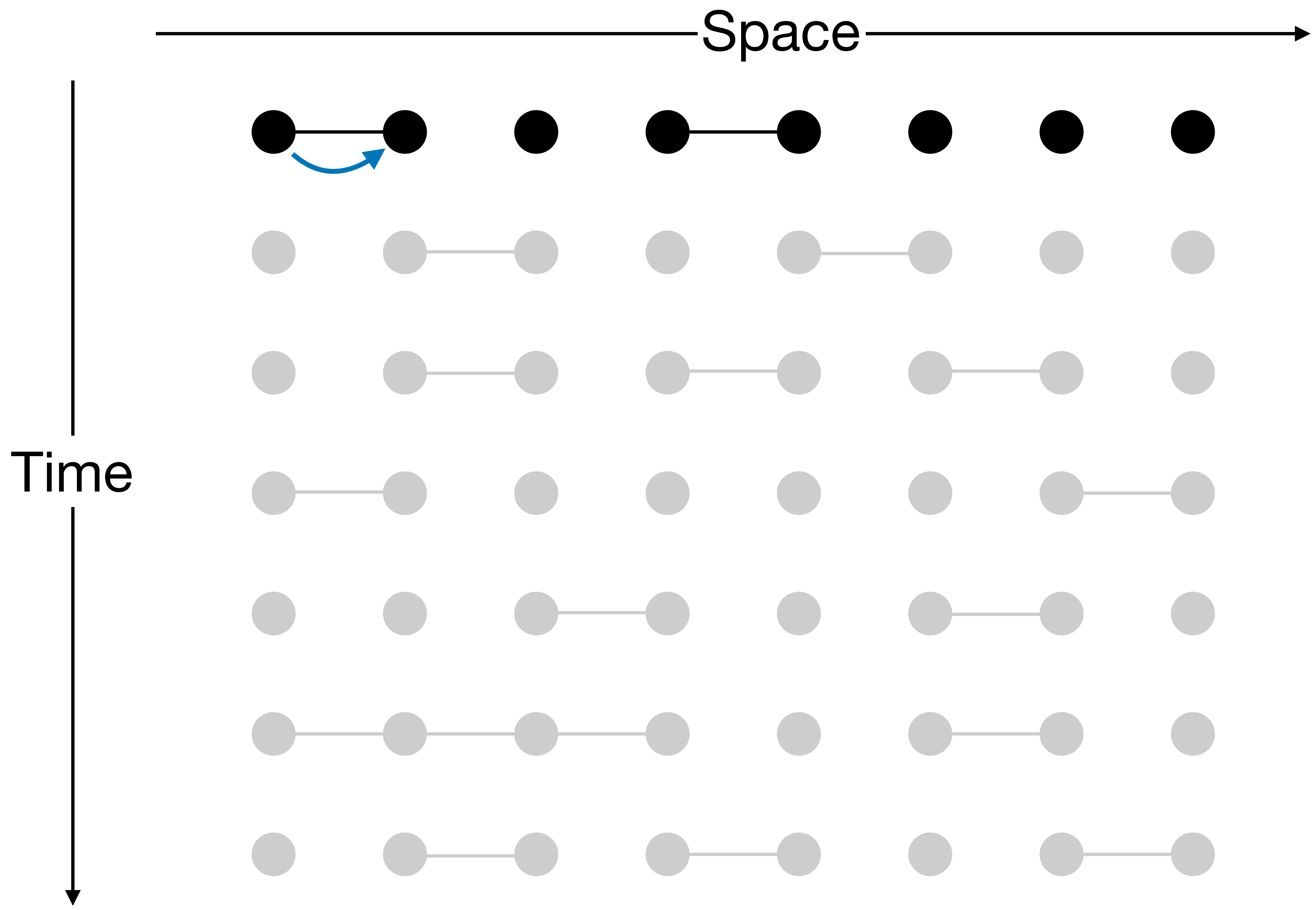
# The offline ODOC Problem

cost(T) = # time inversion

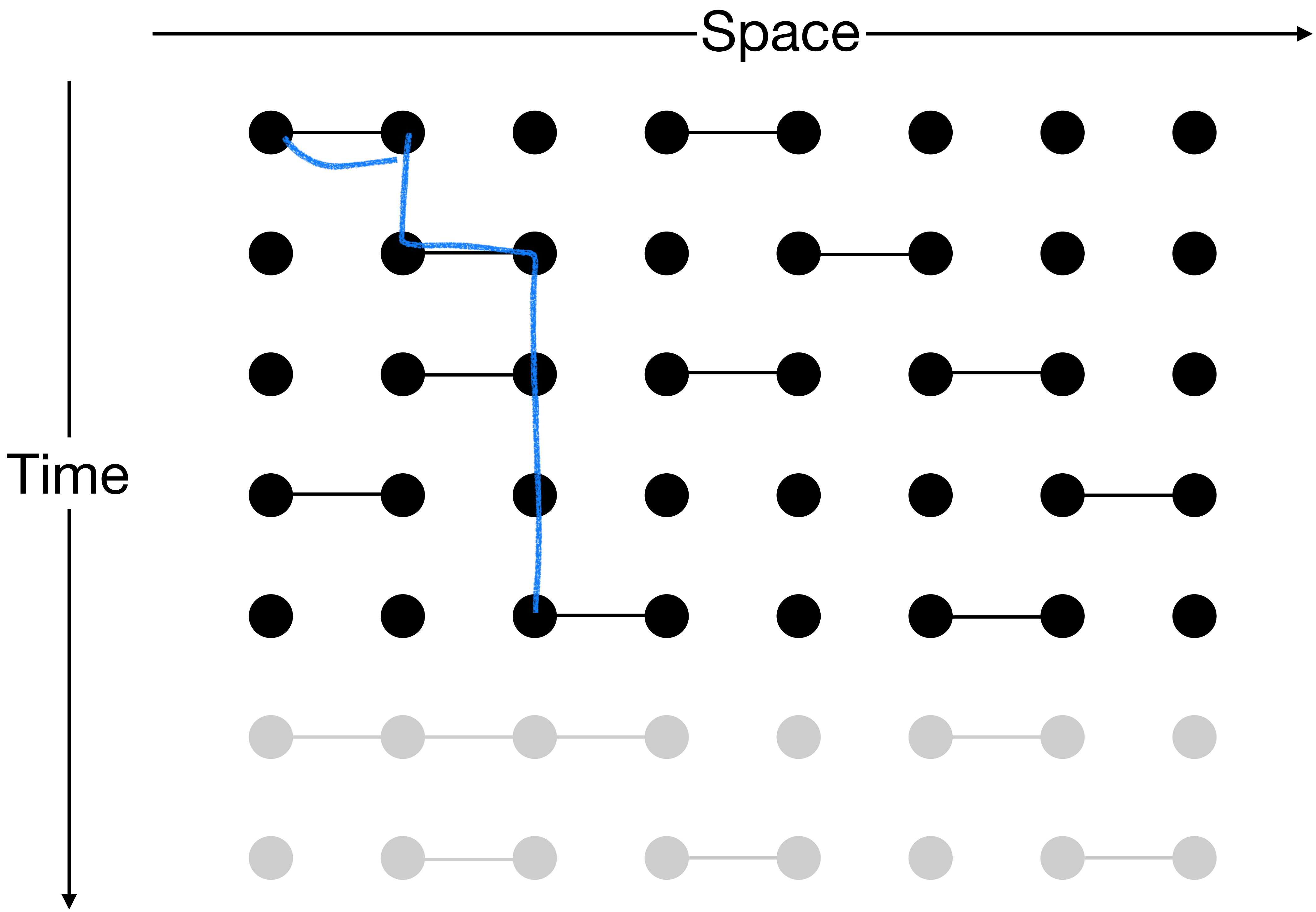
Given an evolving graph G, a source, and a destination

Find a 0-delay space-time travel with lowest cost.

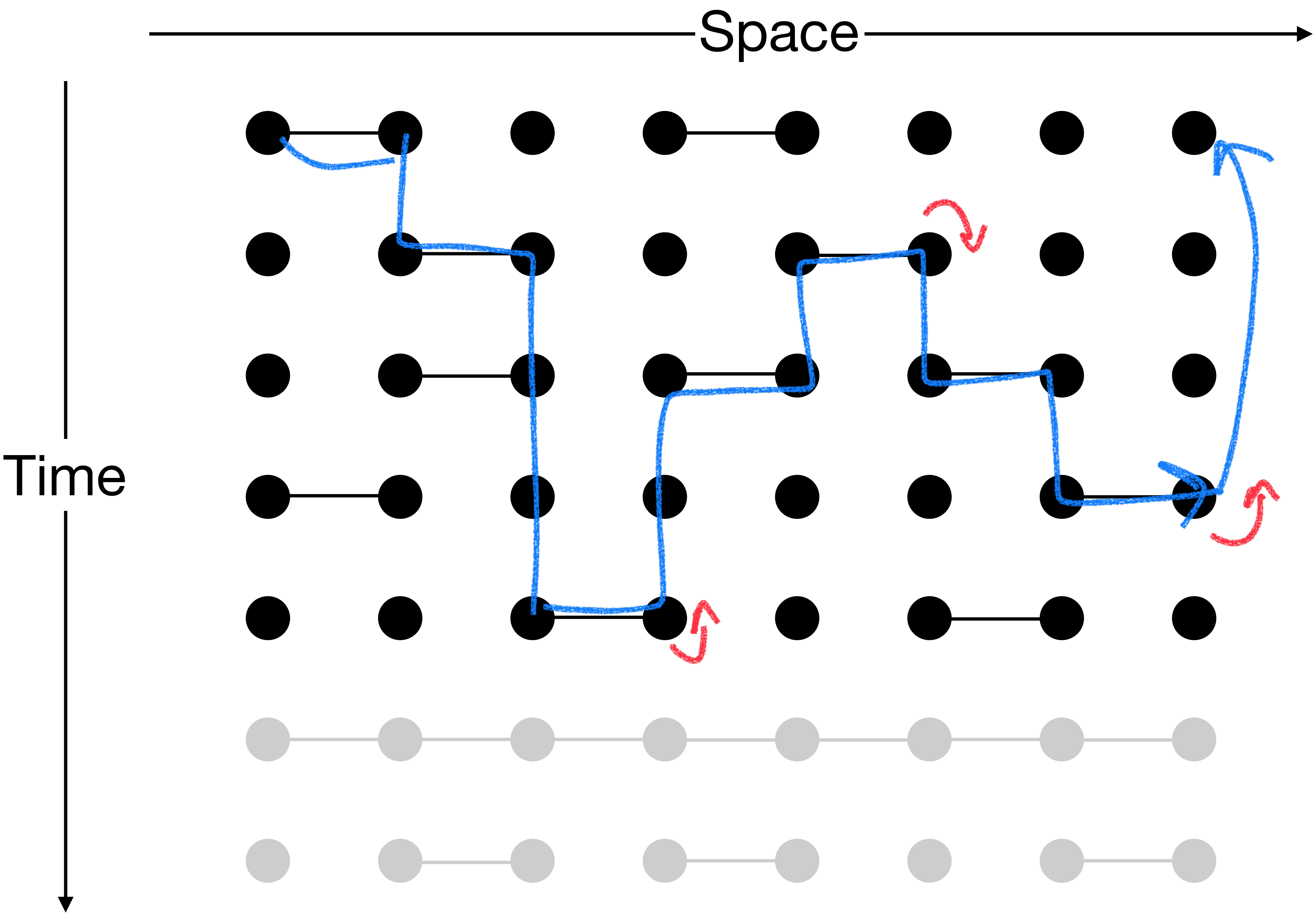
# Time-Online Algorithm



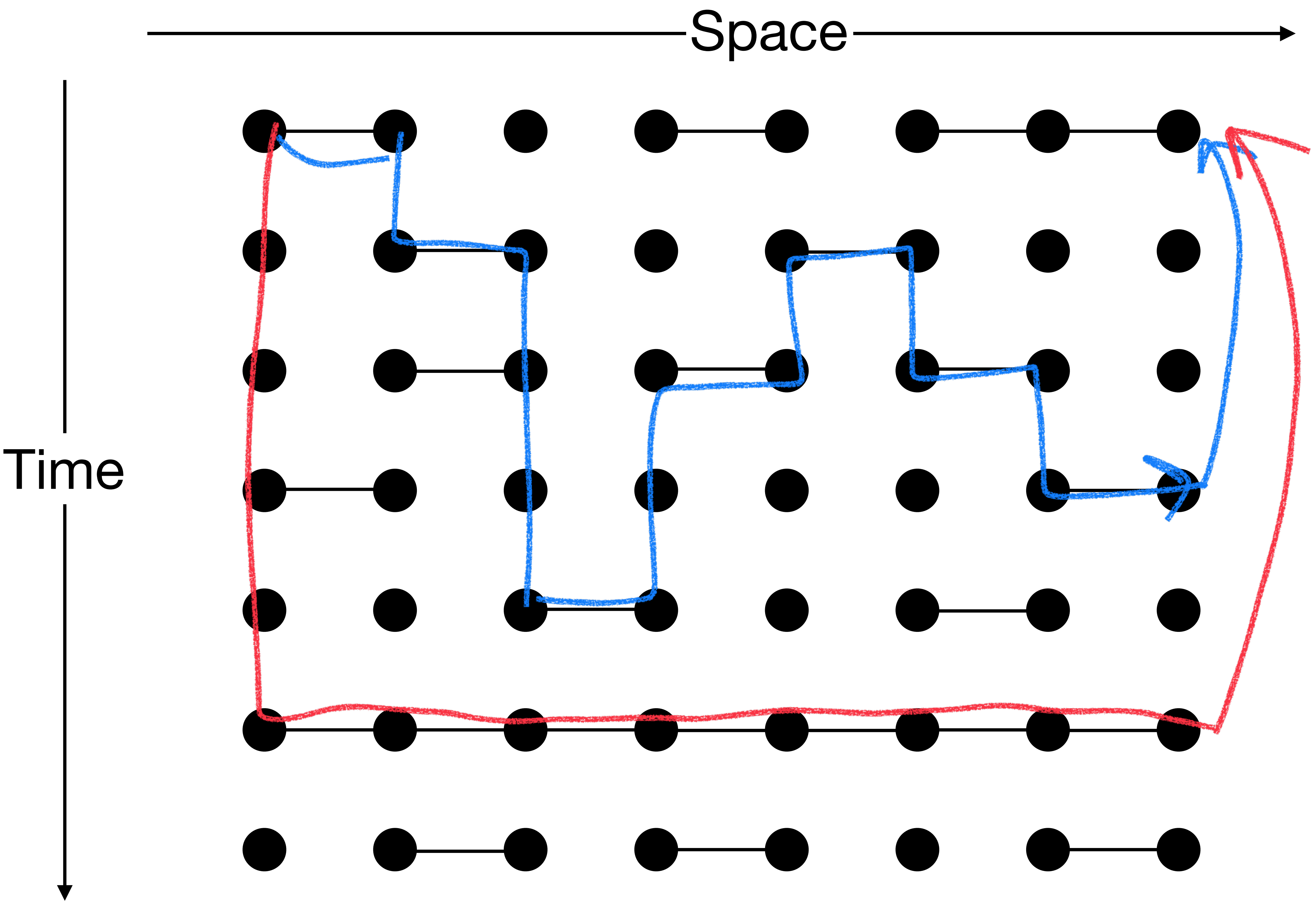
# Time-Online Algorithm



# Time-Online Algorithm



# Competitive Ratio

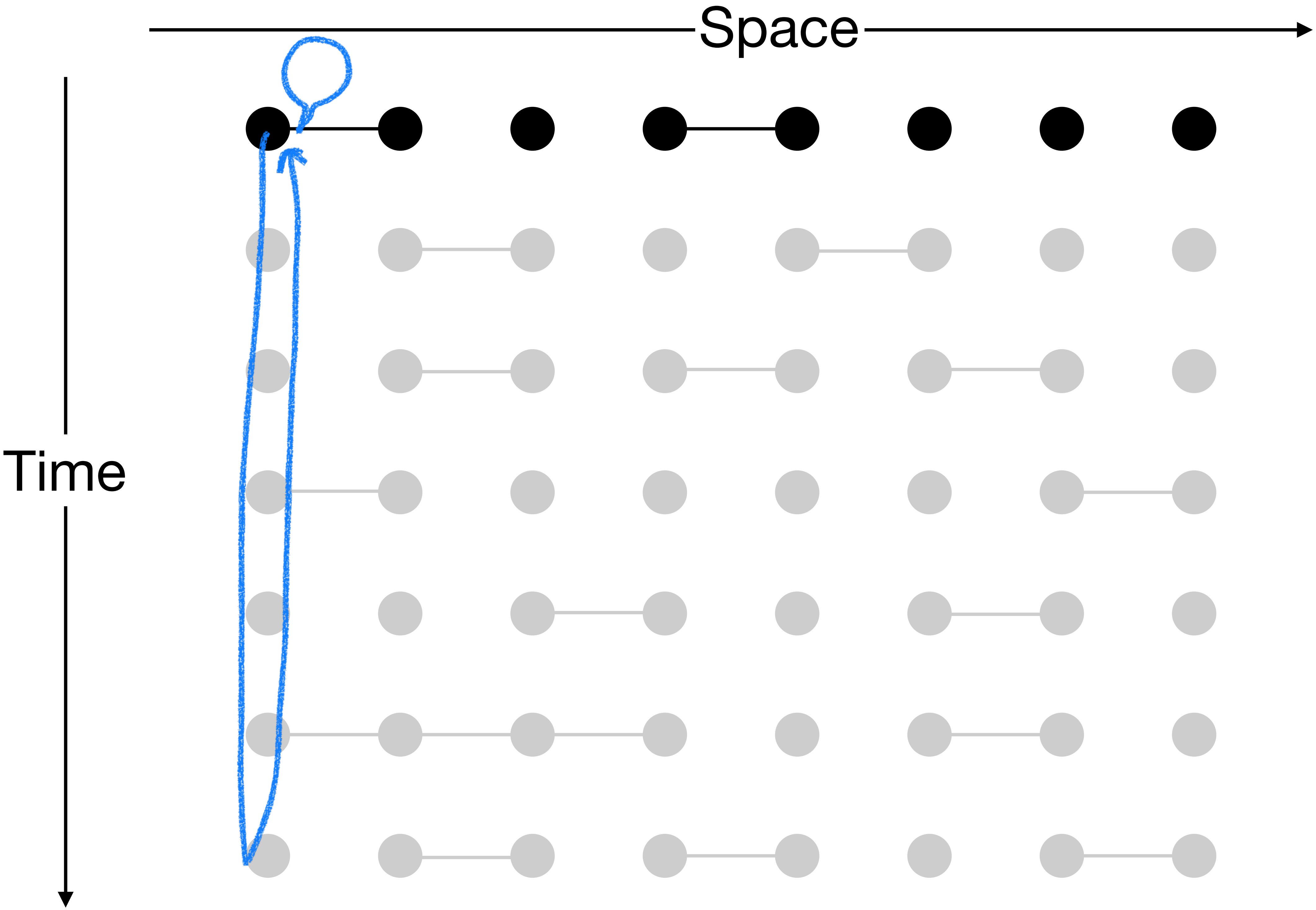


# Competitive Ratio

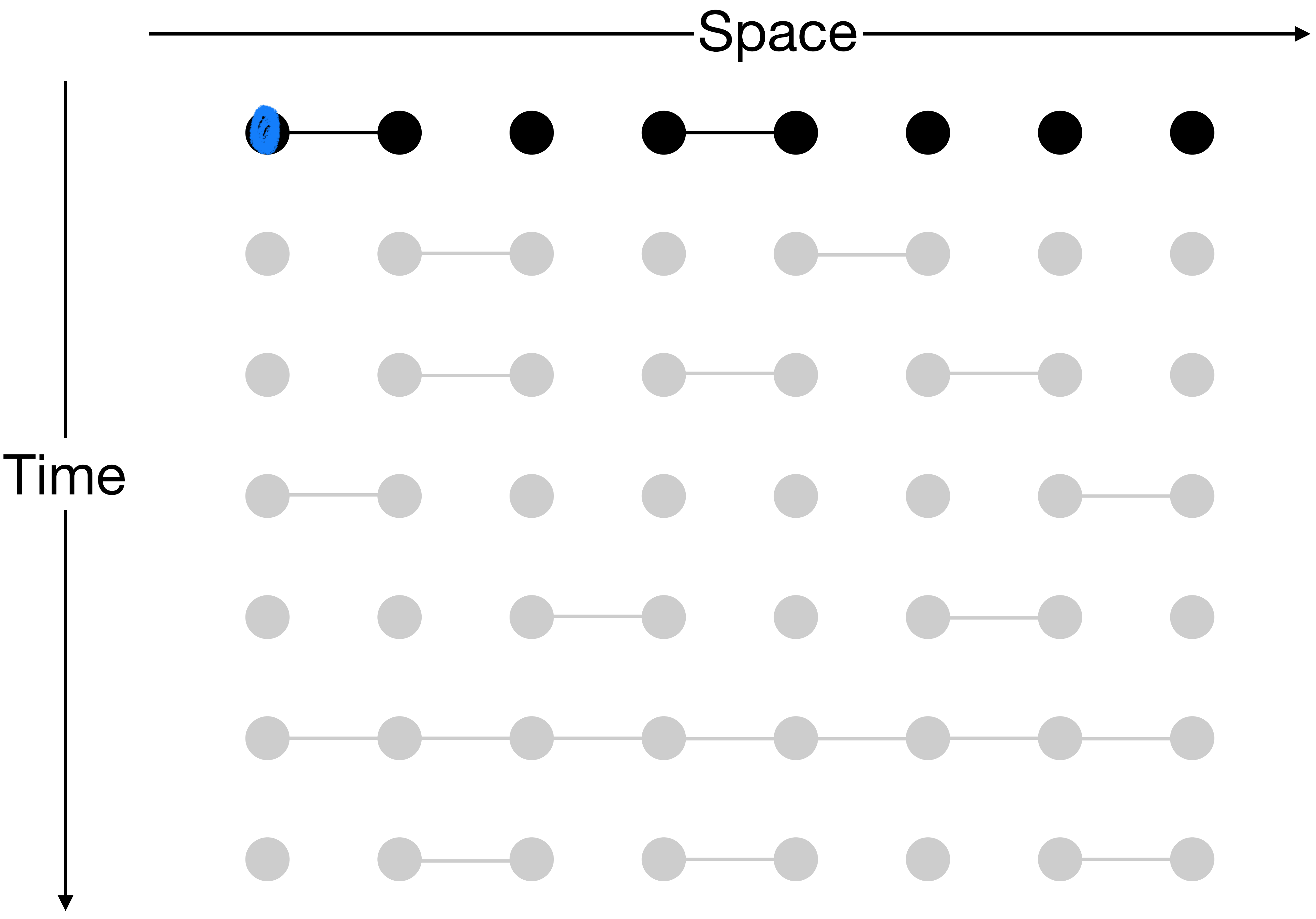
$A$ is $\rho$-competitive if

$$\forall G, \quad \frac{Cost(A, G)}{Cost(Opt, G)} \leq \rho$$

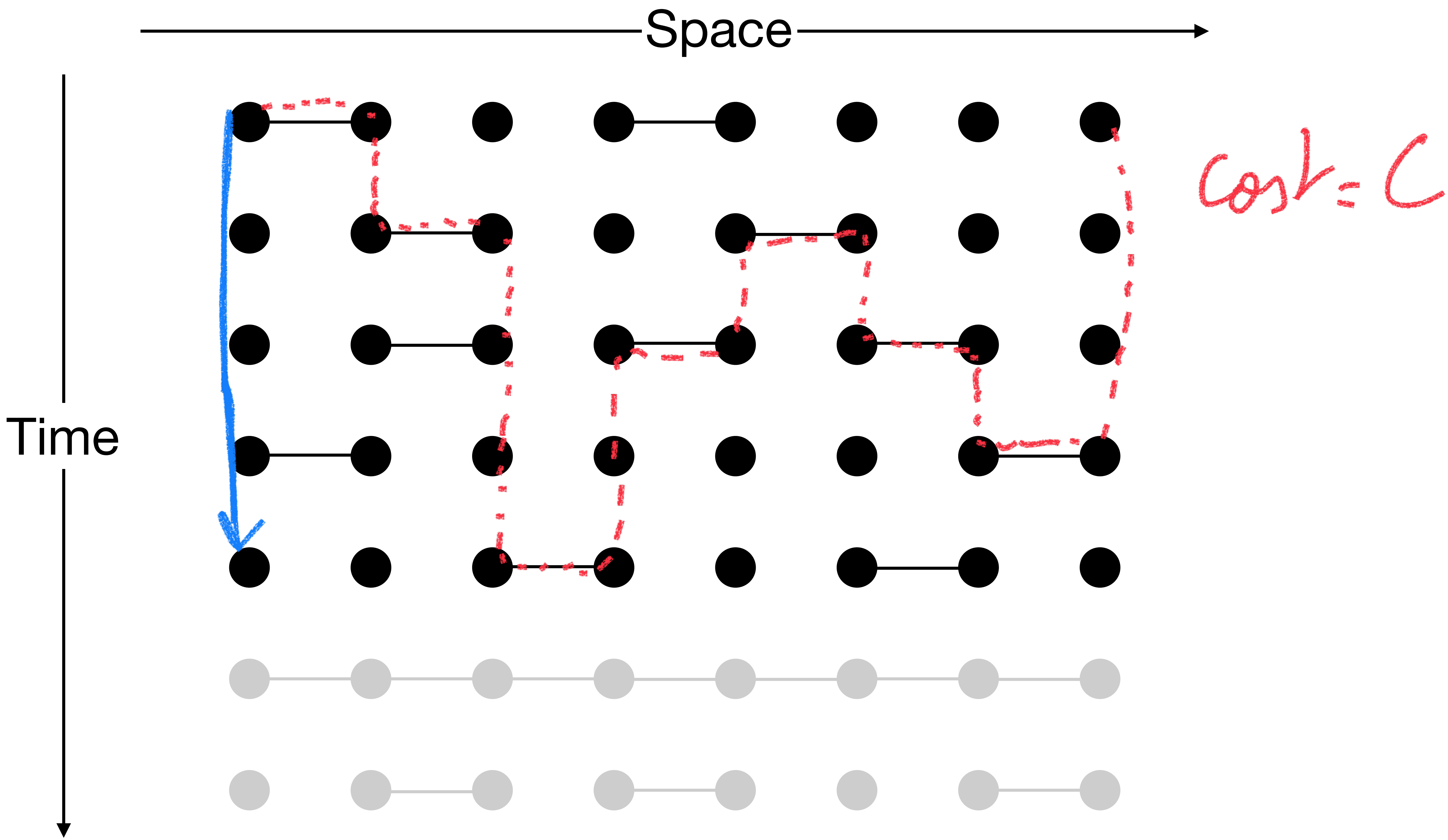
# Learning paradoxe



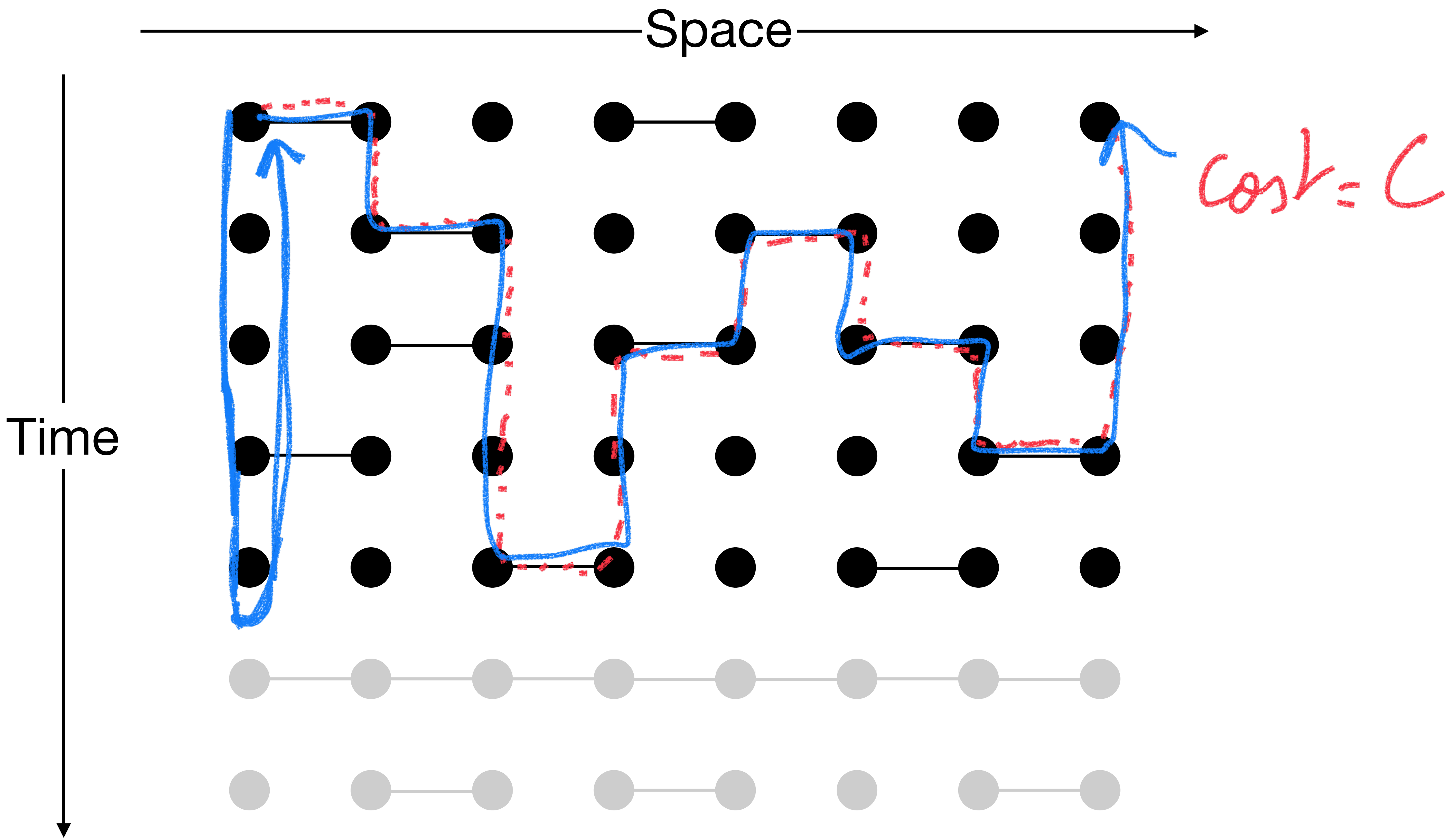
# Time-Online Algorithm



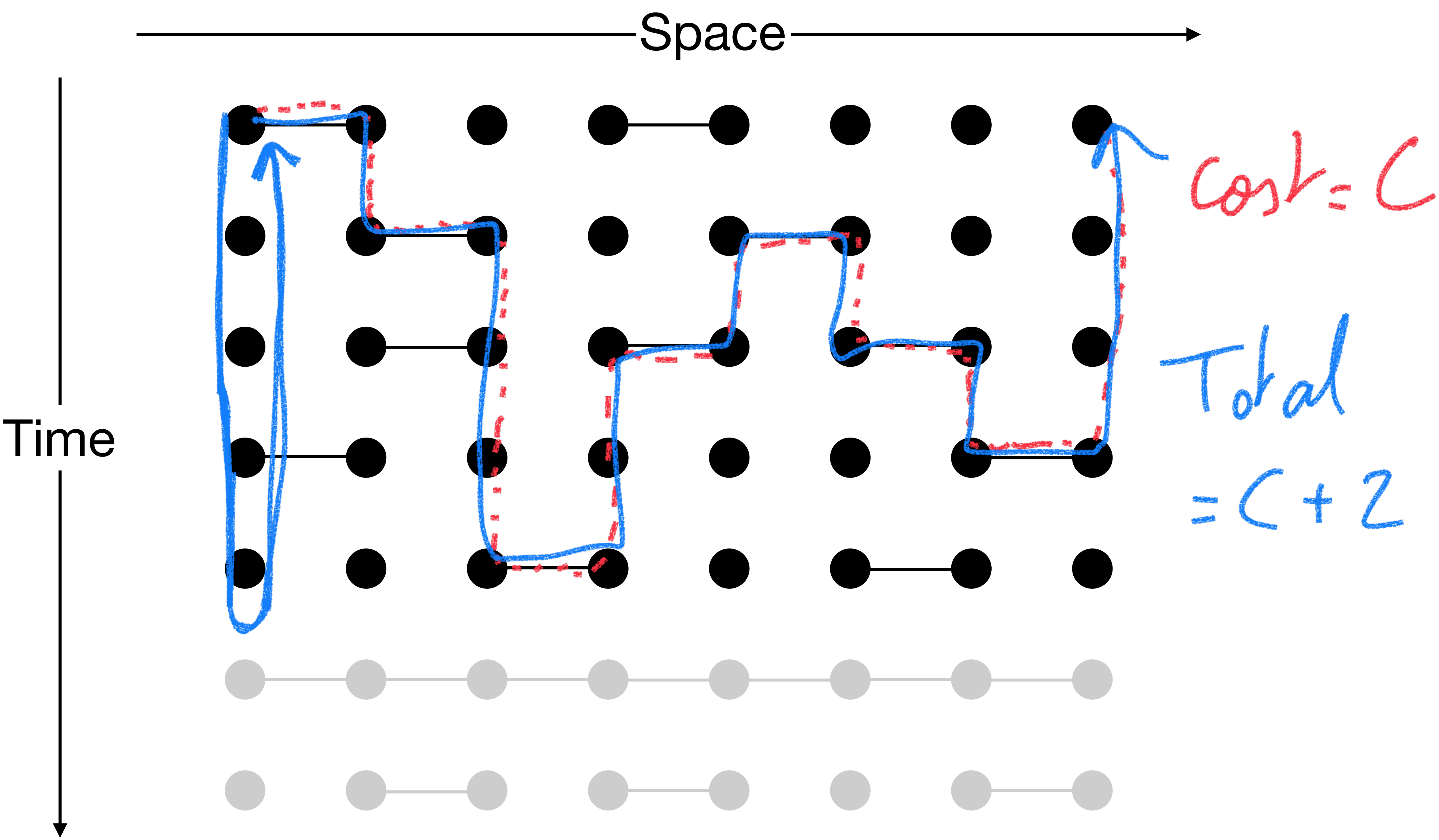
# Time-Online Algorithm



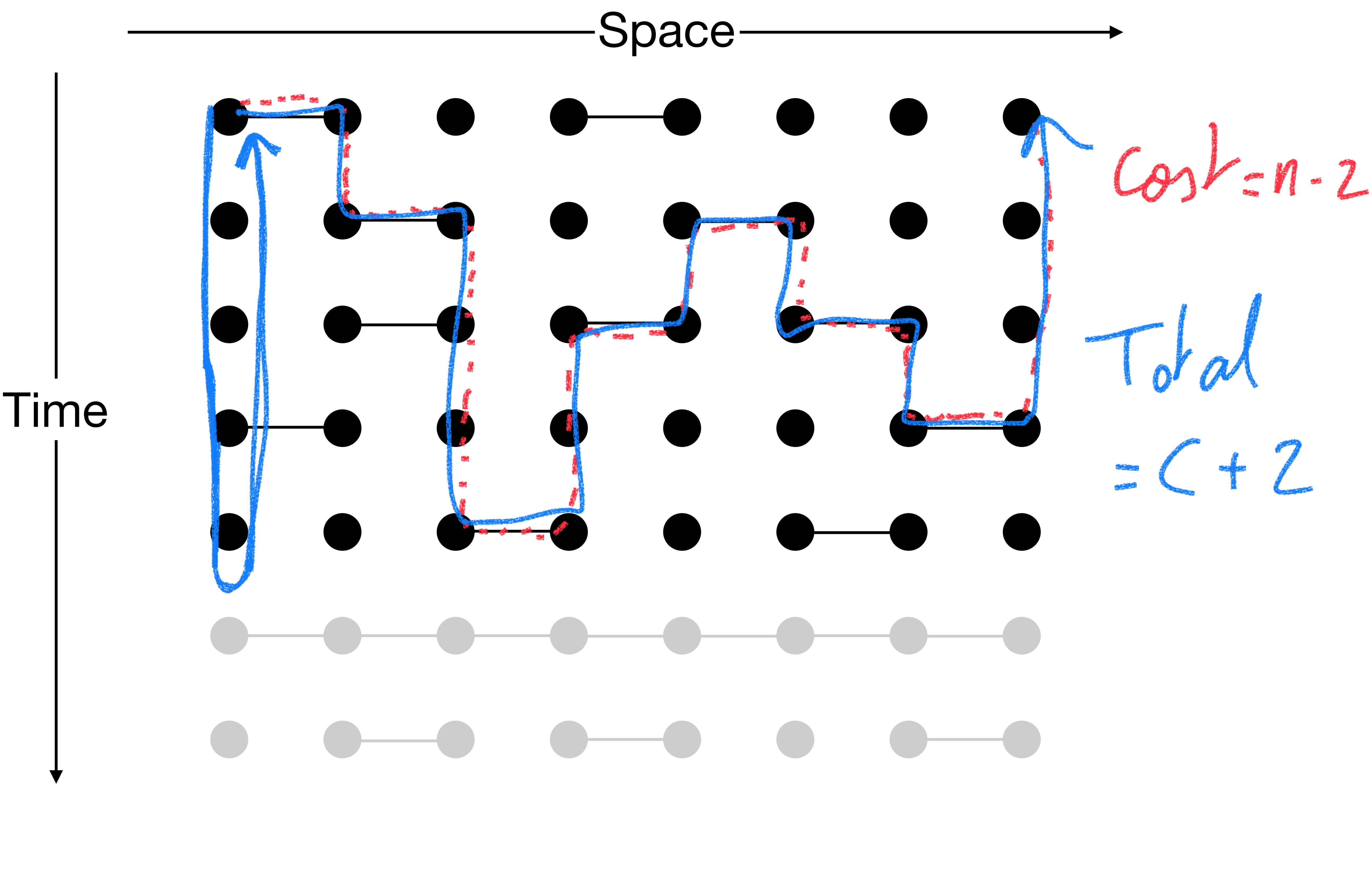
# Time-Online Algorithm



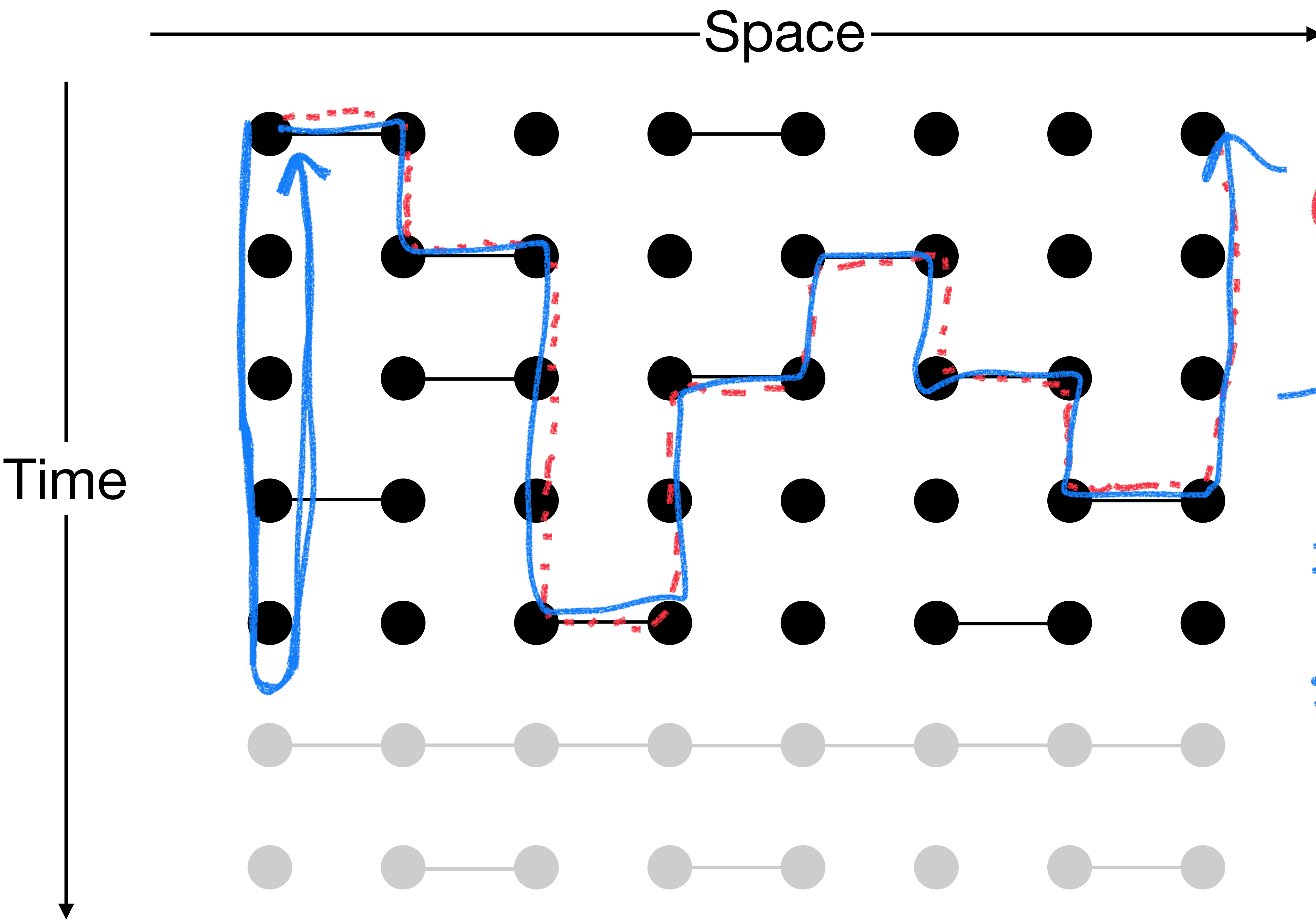
# Time-Online Algorithm



# Time-Online Algorithm



# Time-Online Algorithm




Cost = n - 2

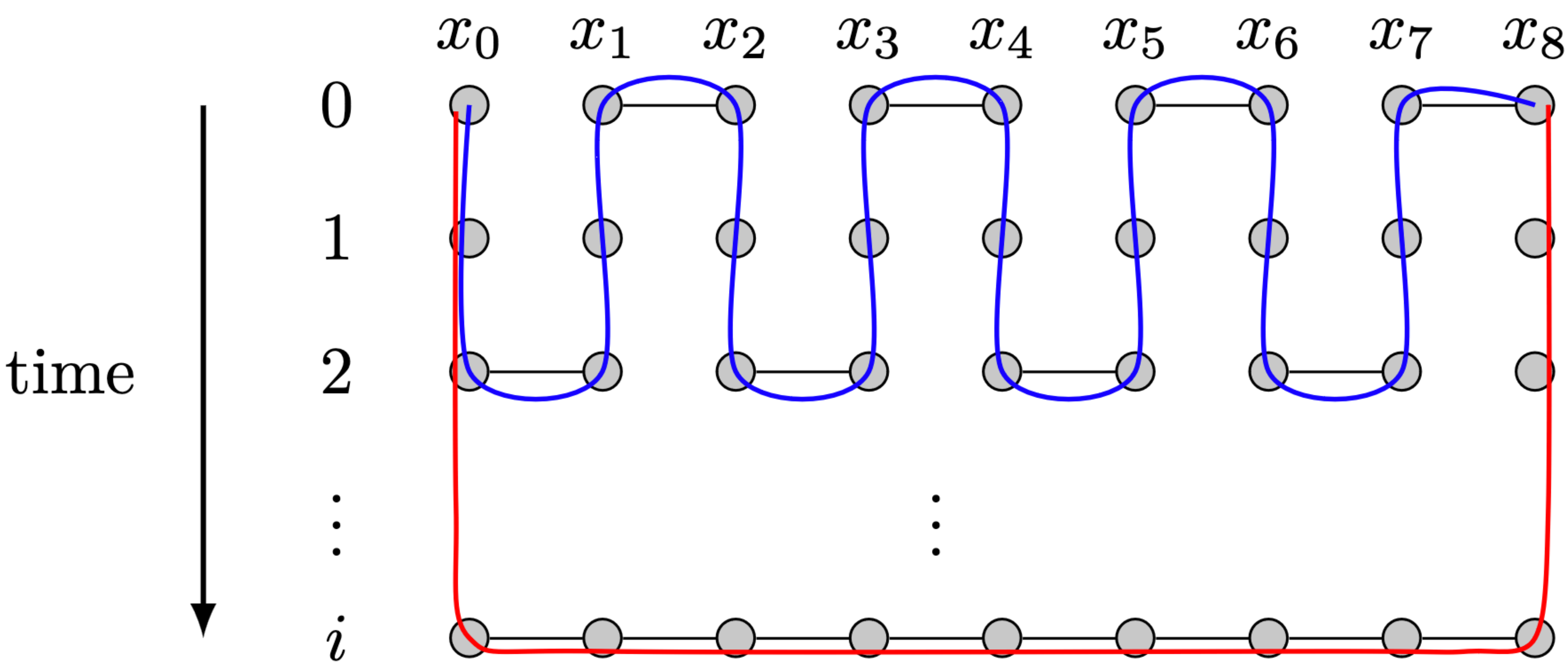
Total

= C + 2

= n

# Time-Online Algorithm

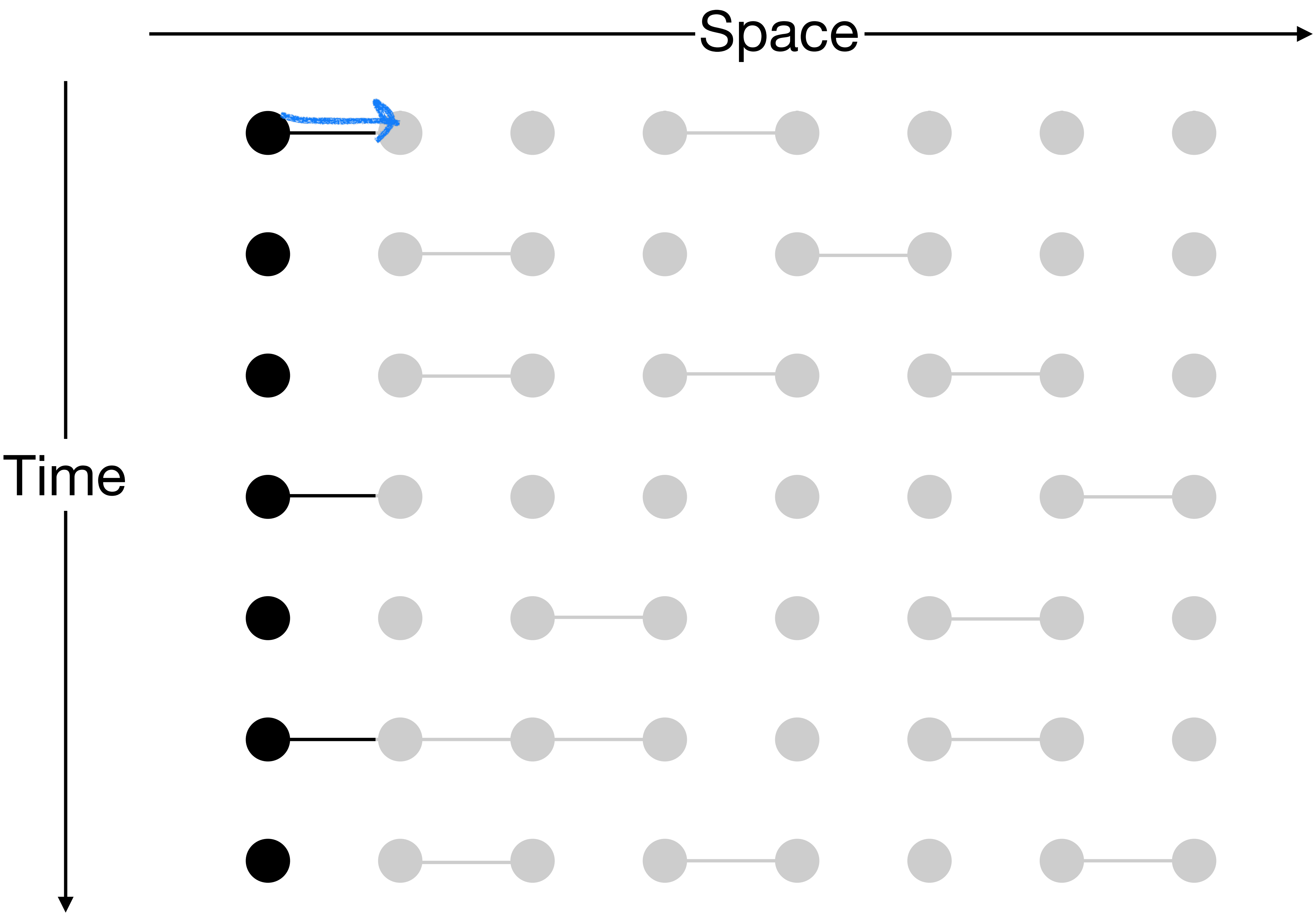
Let $G_i$ be



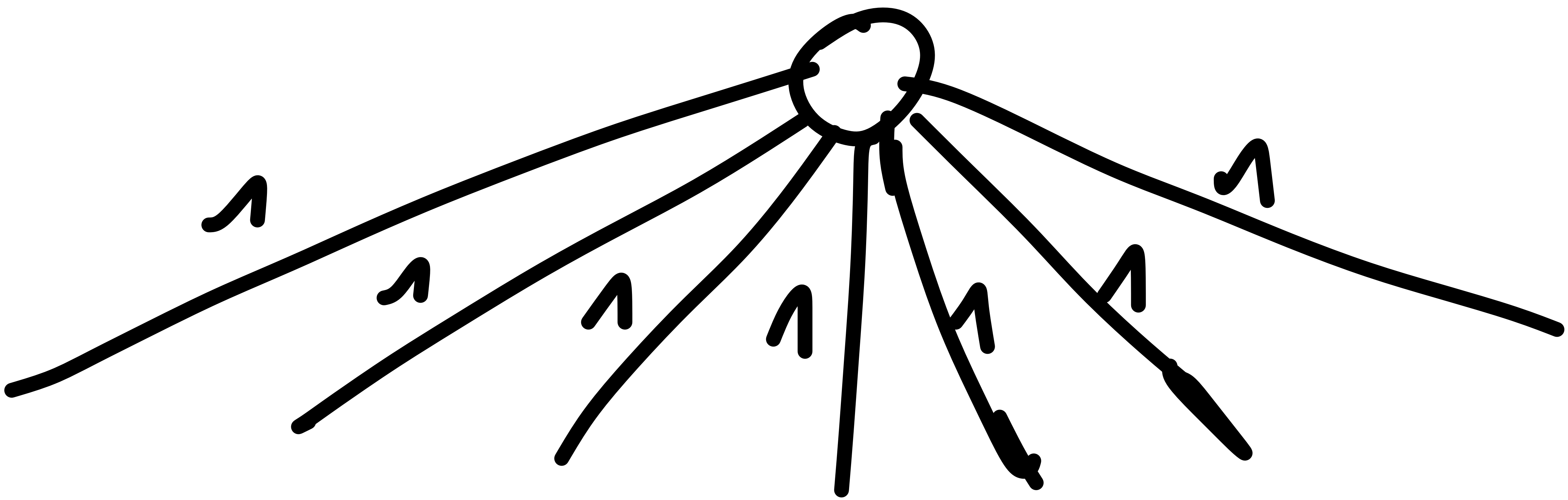
# Time-Online Algorithm

Optimal competitive ratio : $n$

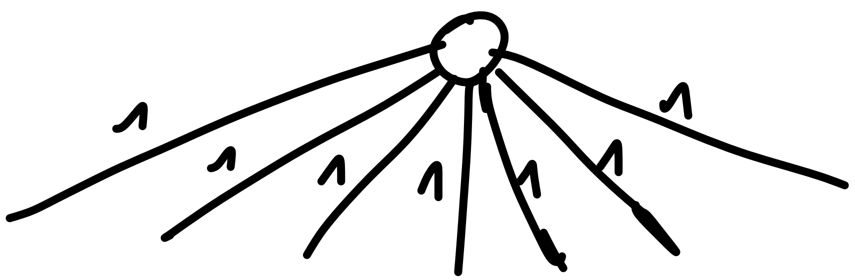
# Space-Online Algorithm



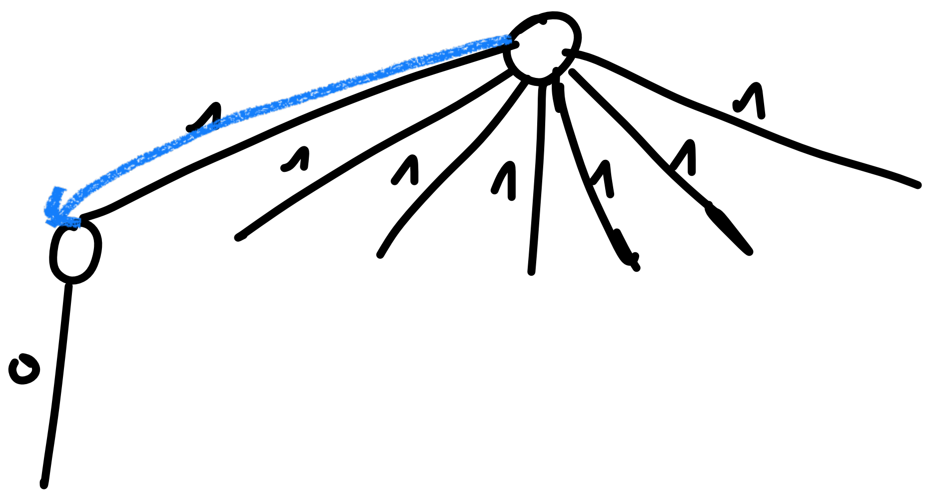
# Space-Online Algorithm Lower Bound



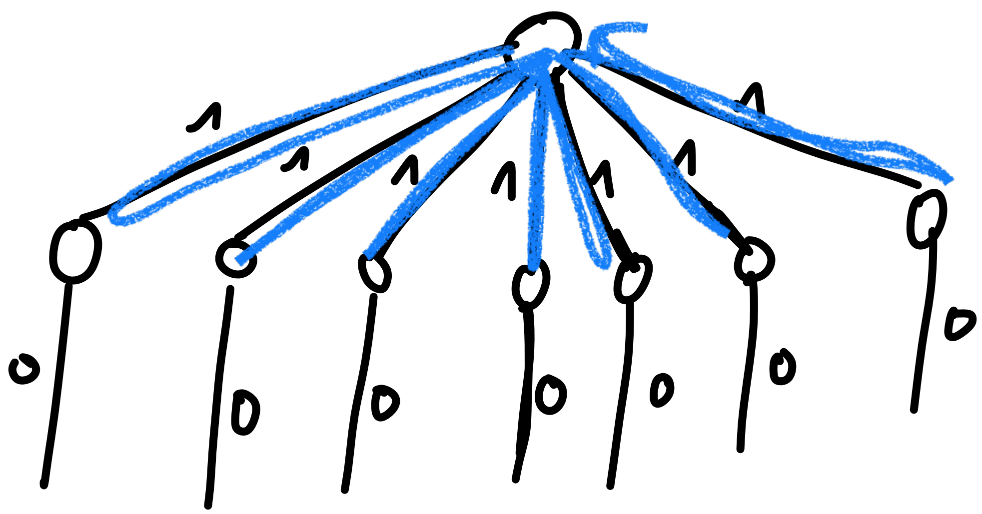
# Space-Online Algorithm Lower Bound



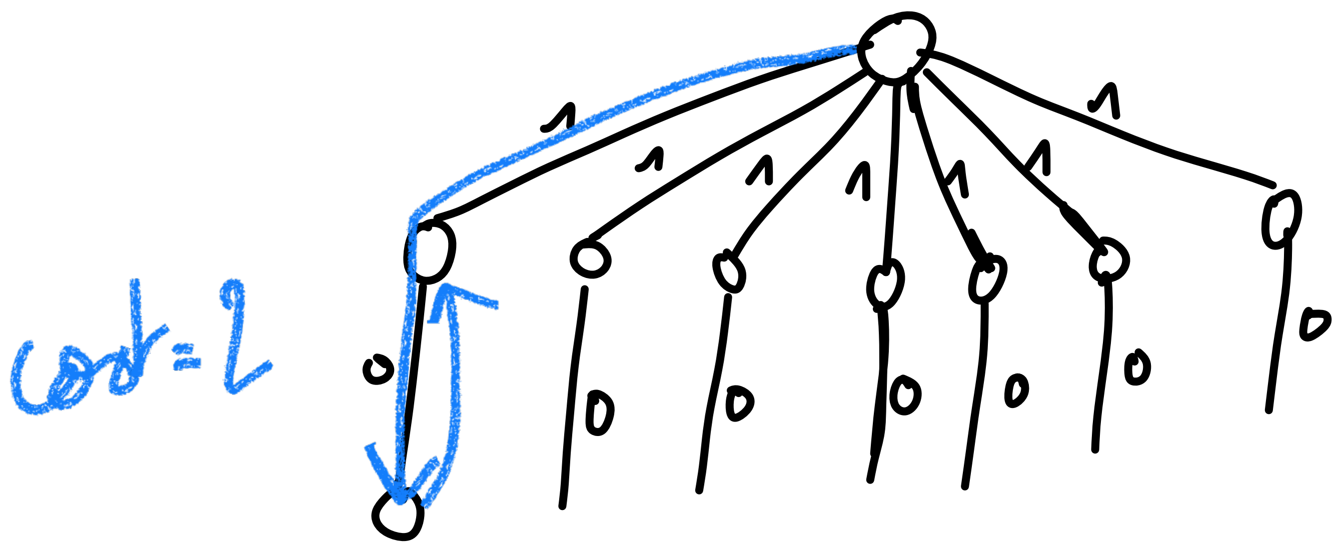
# Space-Online Algorithm Lower Bound



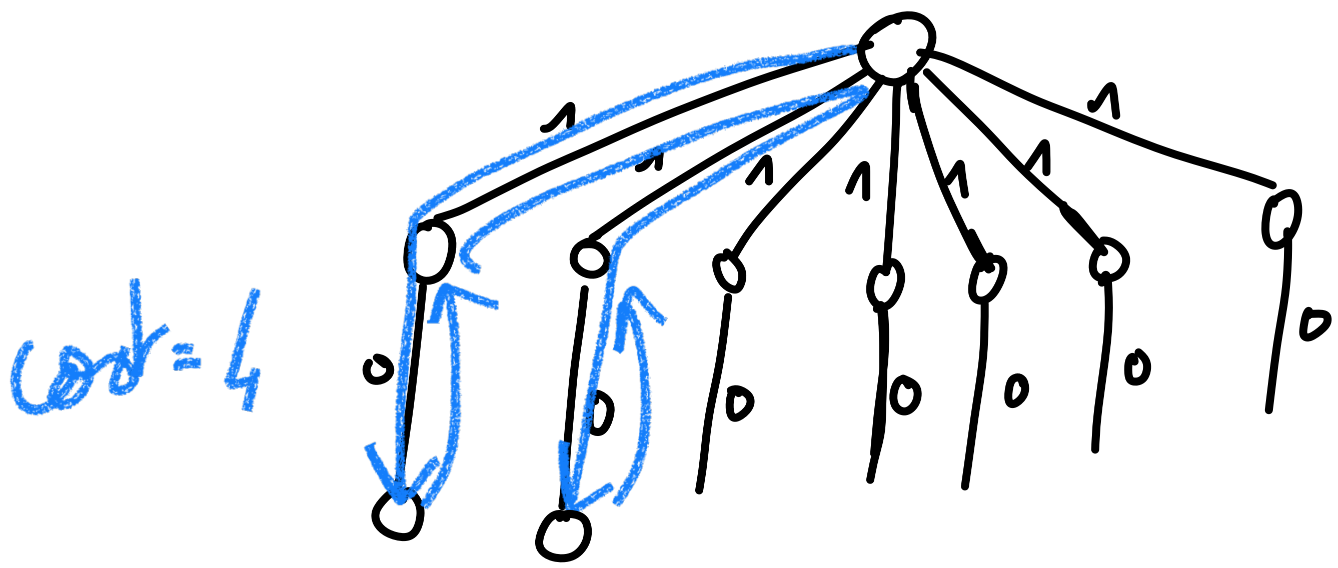
# Space-Online Algorithm Lower Bound



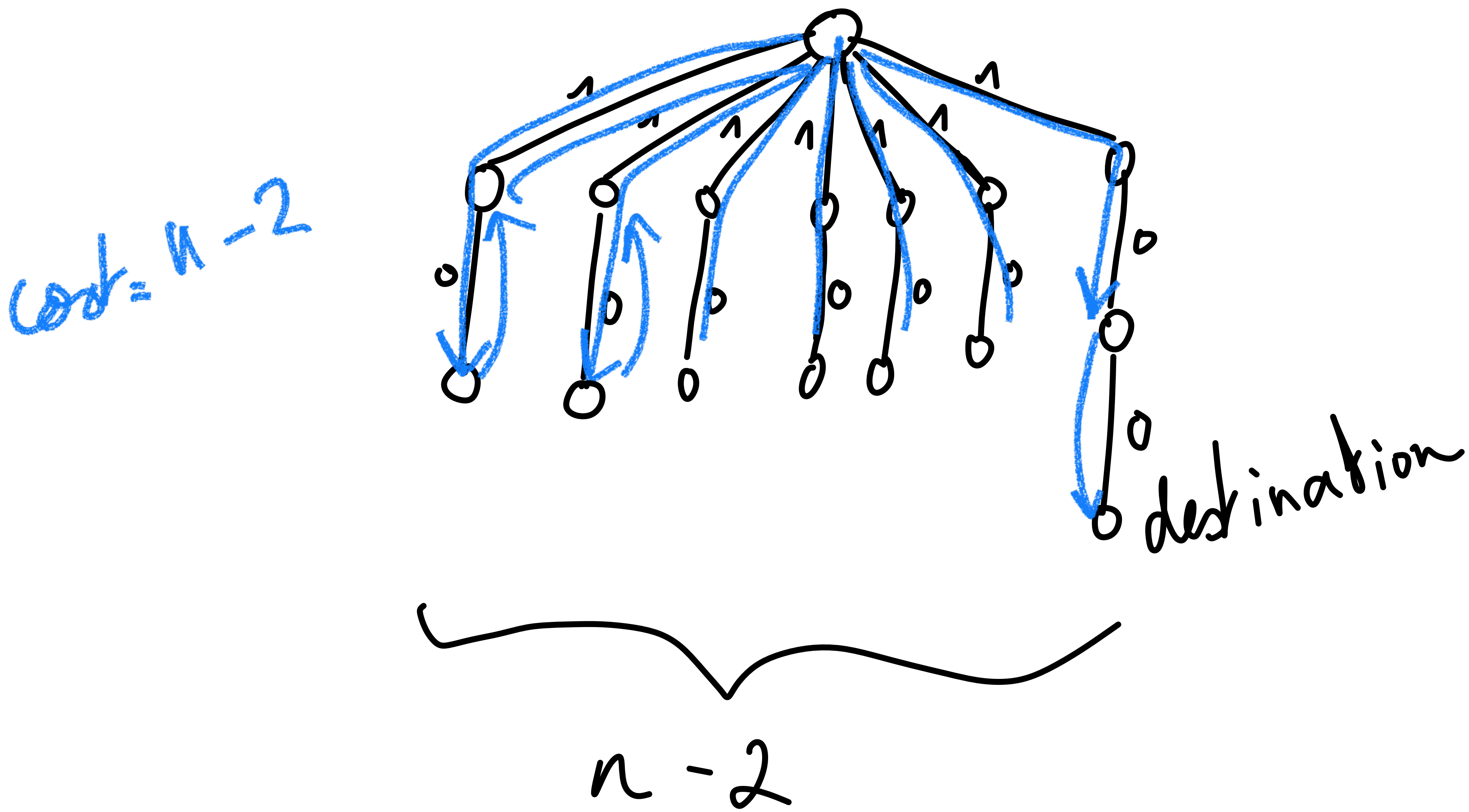
# Space-Online Algorithm Lower Bound



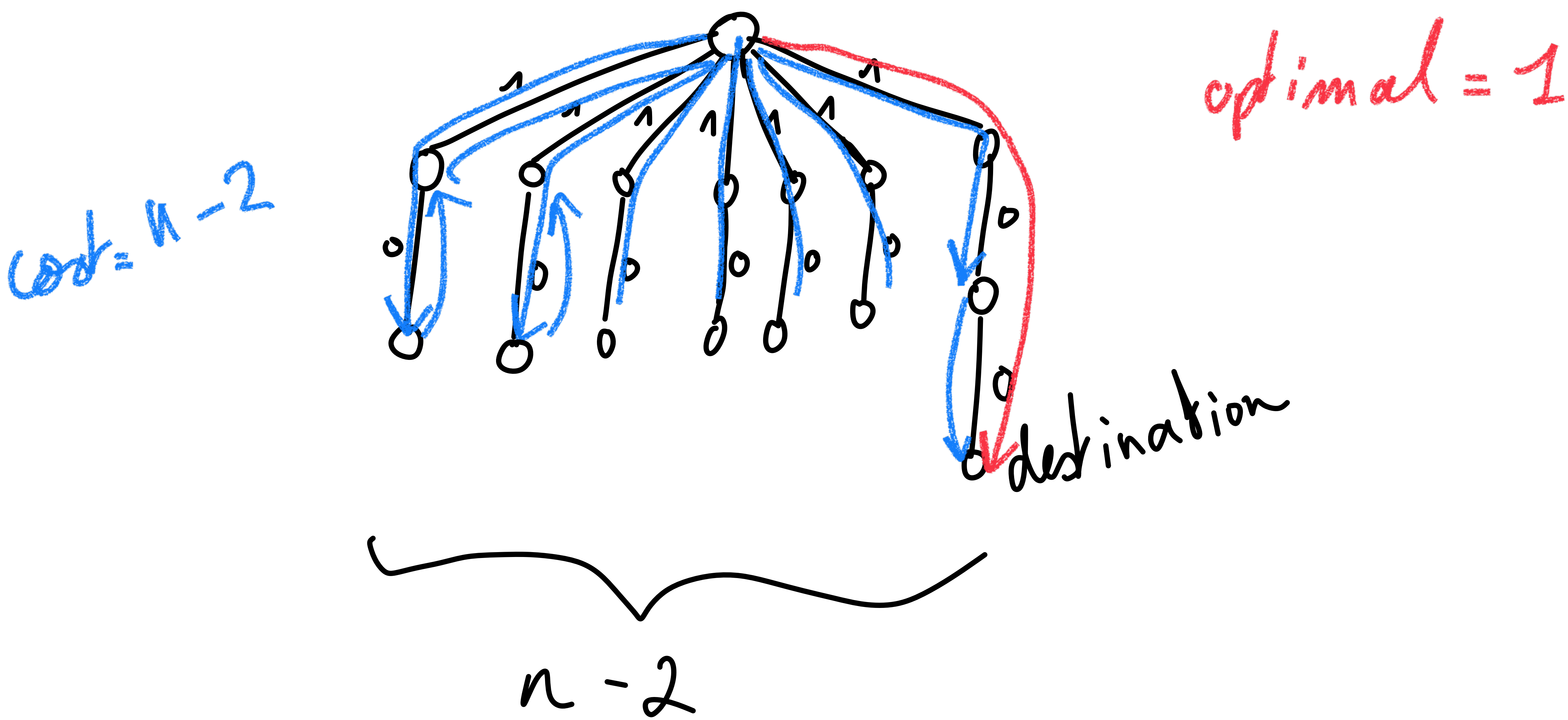
# Space-Online Algorithm Lower Bound



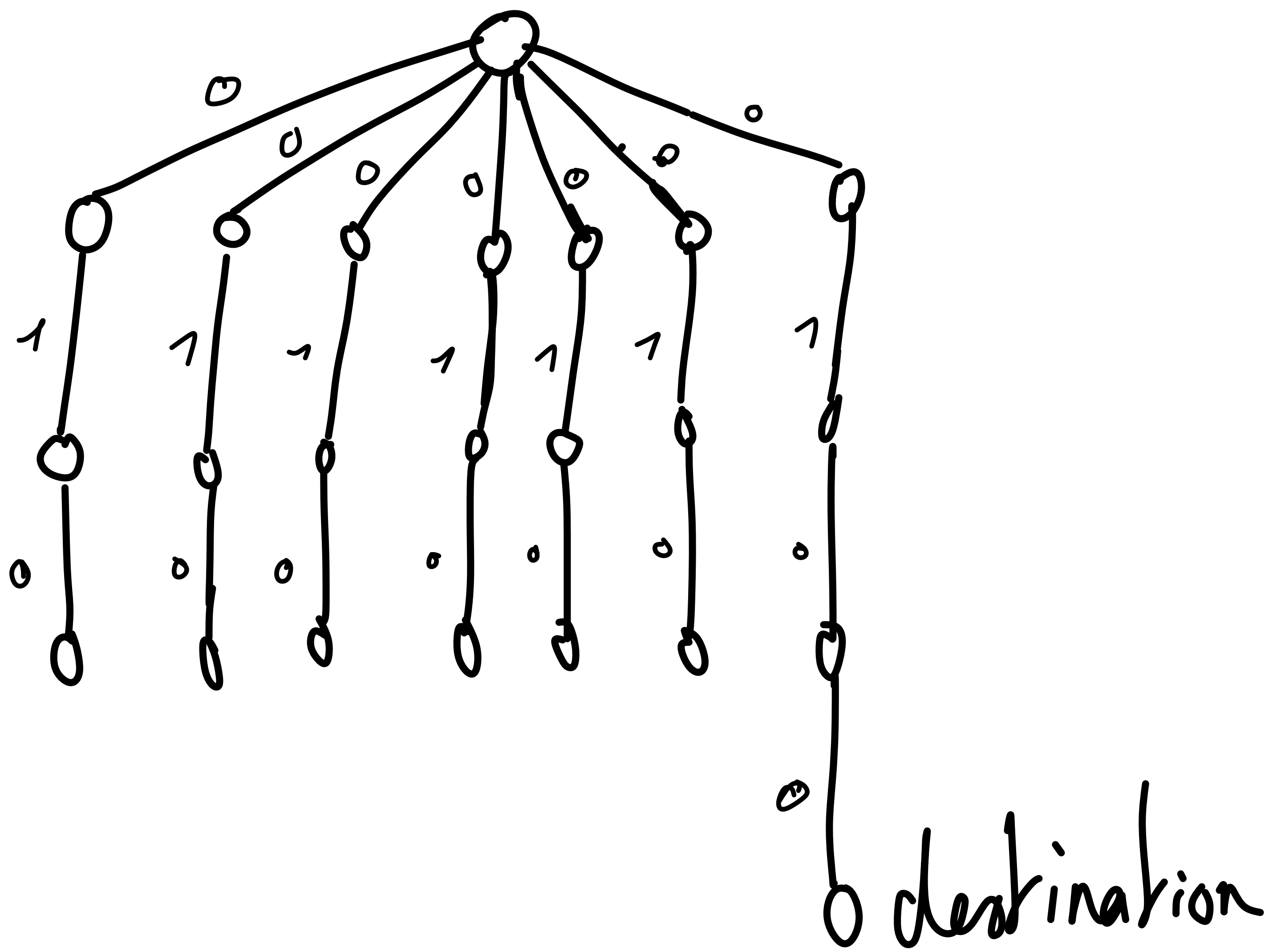
# Space-Online Algorithm Lower Bound



# Space-Online Algorithm Lower Bound



# Space-Online Algorithm Lower Bound

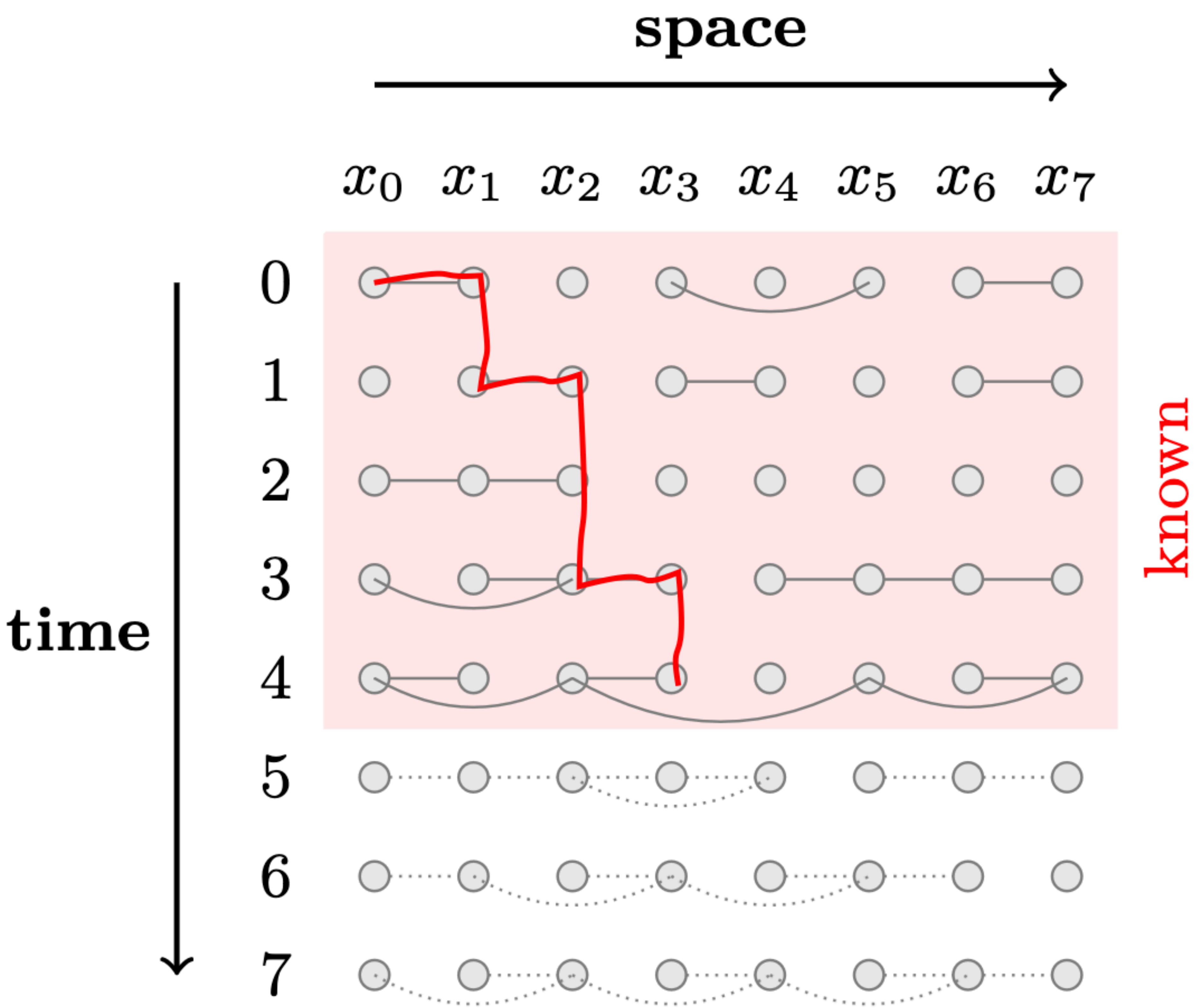


# Space-Online Algorithm Lower Bound

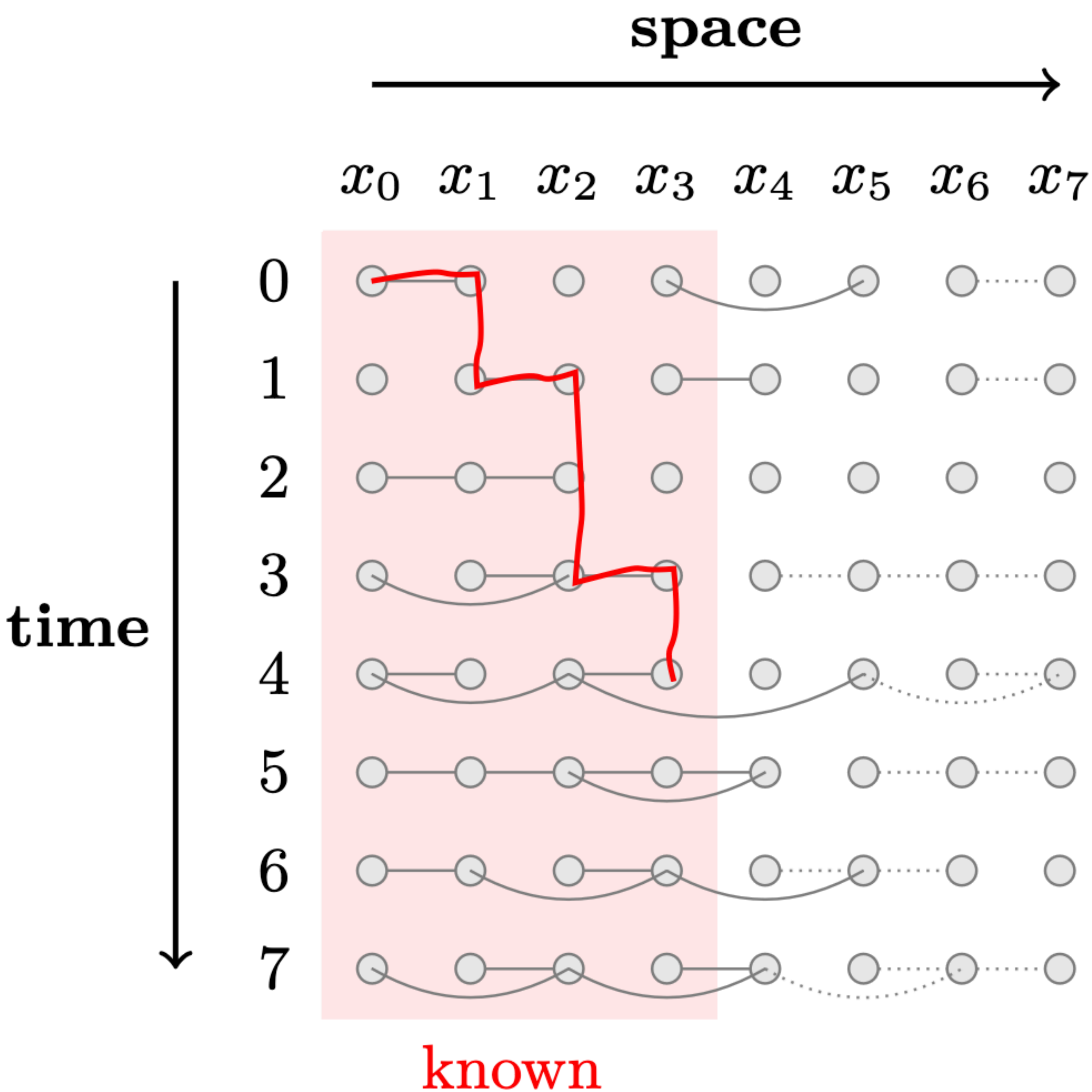
Lower bound : $\frac{4n}{3} - \frac{17}{3}$

best algo : $2n - 4$

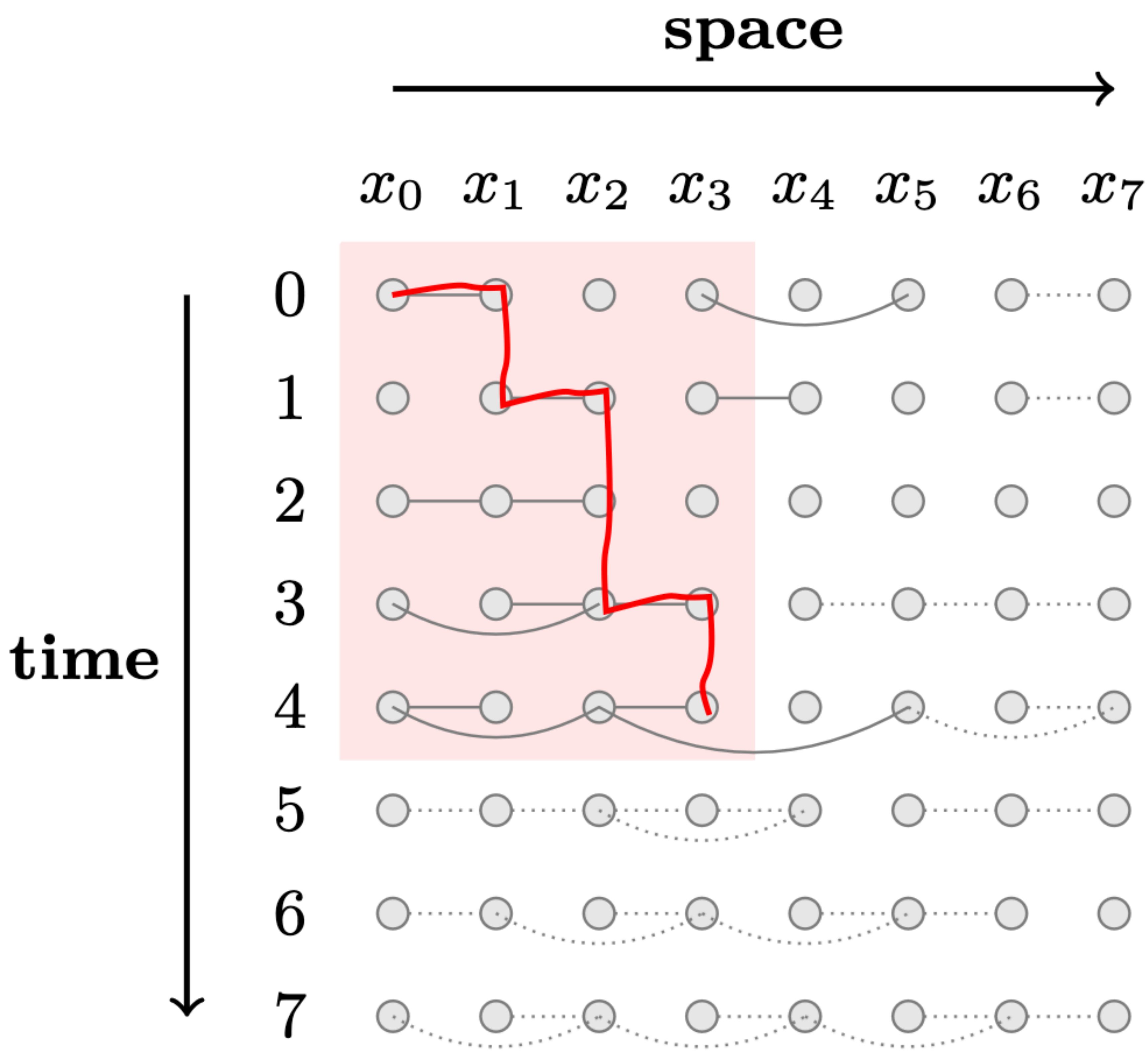
# Time-Online Model



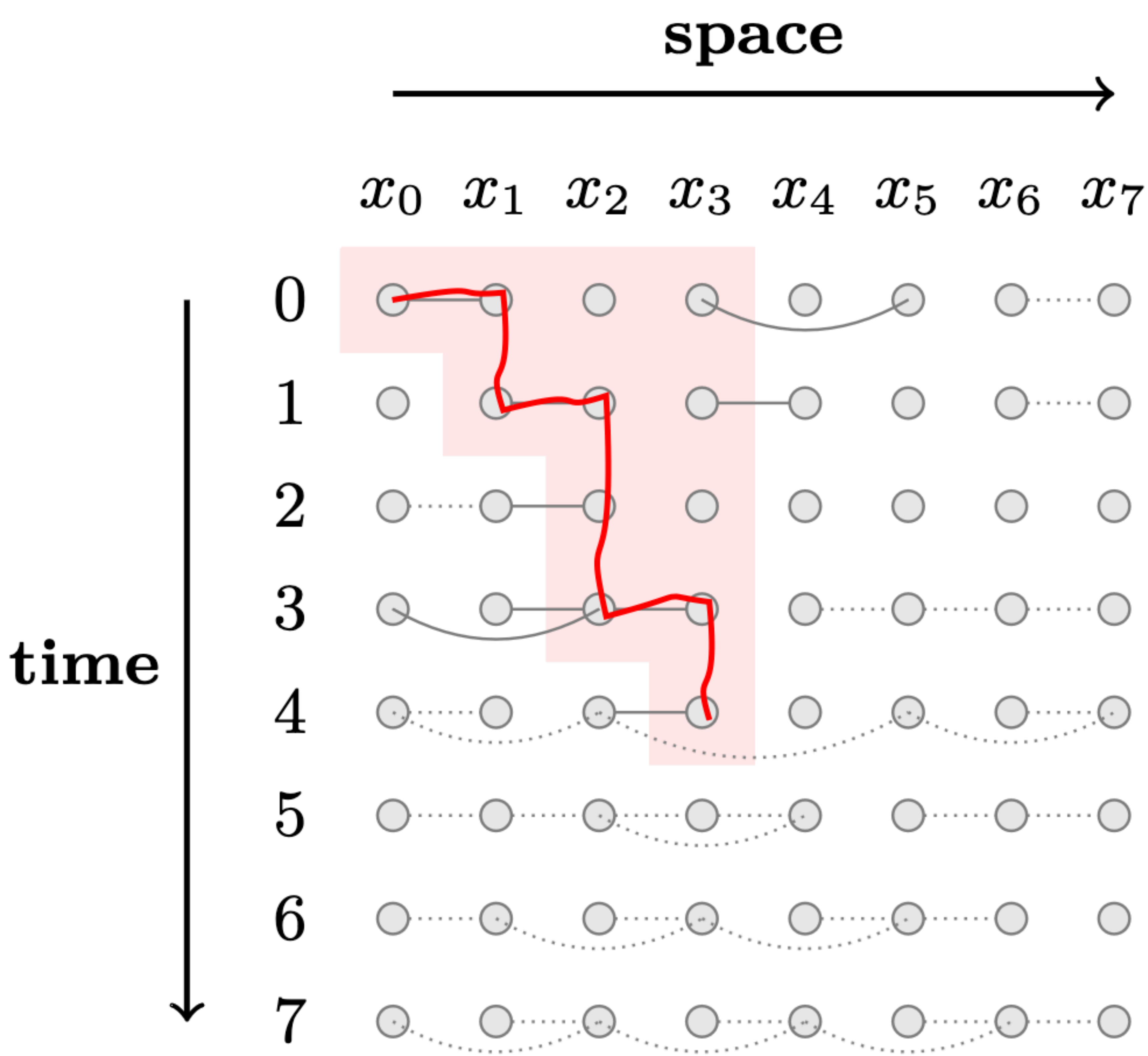
# Space-Online Model



# ST-easy-Online Model



# ST-hard-Online Model



# ST-hard-Online Algorithm

No finite competitive ratio (if $n \geqslant 3$)

# ST-easy-Online Algorithm

Optimal competitive ratio: $\frac{(2n-1)^2}{12}+1$

# Outline

- Conclusion
- Polynomial offline algorithm
- Optimal Time-Online algorithm
- Upper and lower bounds for the Space-Online problem
- Optimal ST-easy-Online algorithm
- Introduction

## Brief Announcement:

# Time-Travel Planning with Tenet Turnstiles


**Thibaut Blanc** **Jean-Romain Luttringer** **Quentin Bramas** **Sébastien Tixeuil**

ENS DE LYON

Université de Strasbourg

SORBONNE UNIVERSITÉ


SAND 2026

NO