Online Space-time Travel Planning in Dynamic Graphs

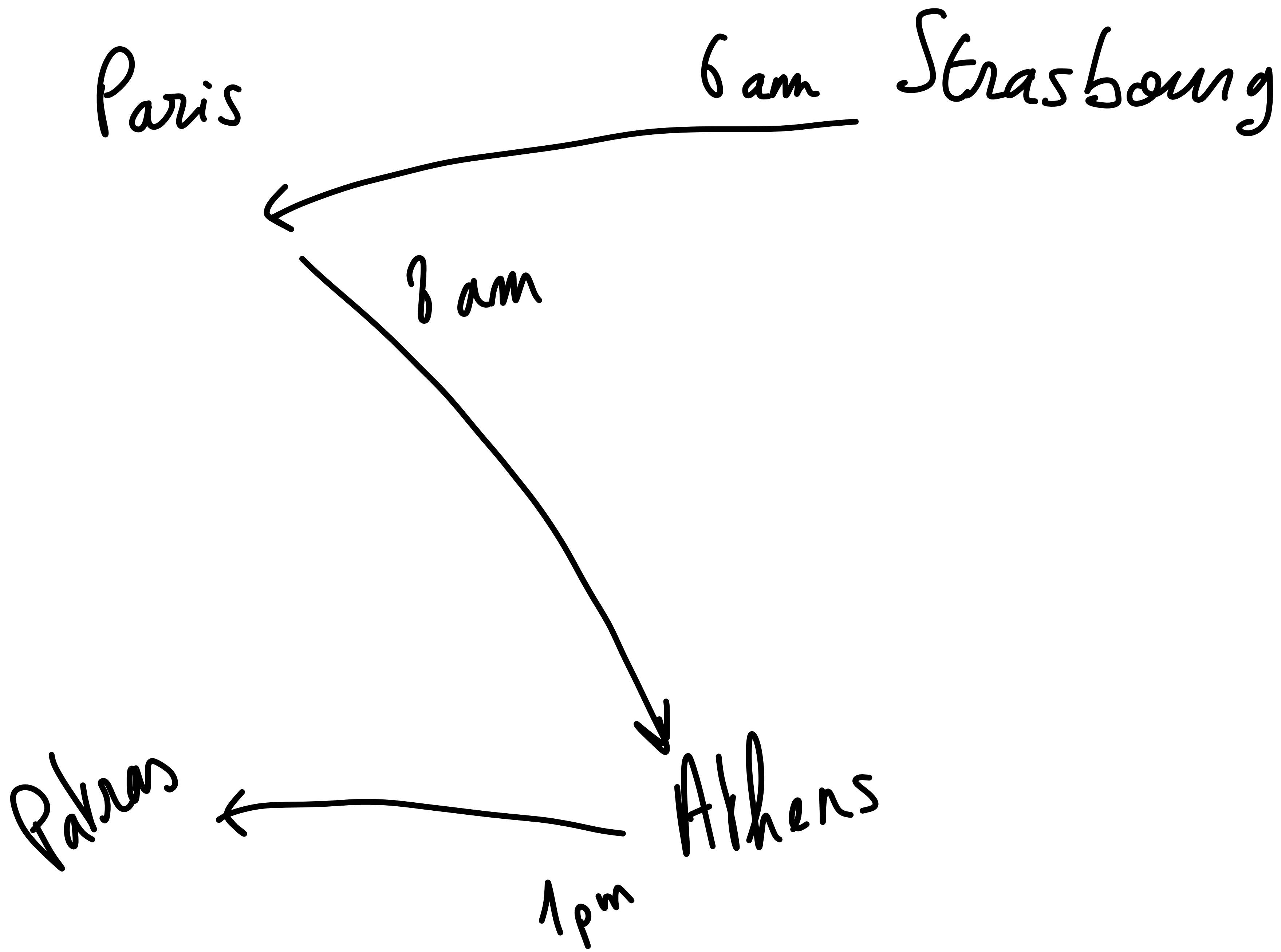
Jean-Romain Luttringer   Quentin Bramas   Sébastien Tixeuil

Université de Strasbourg

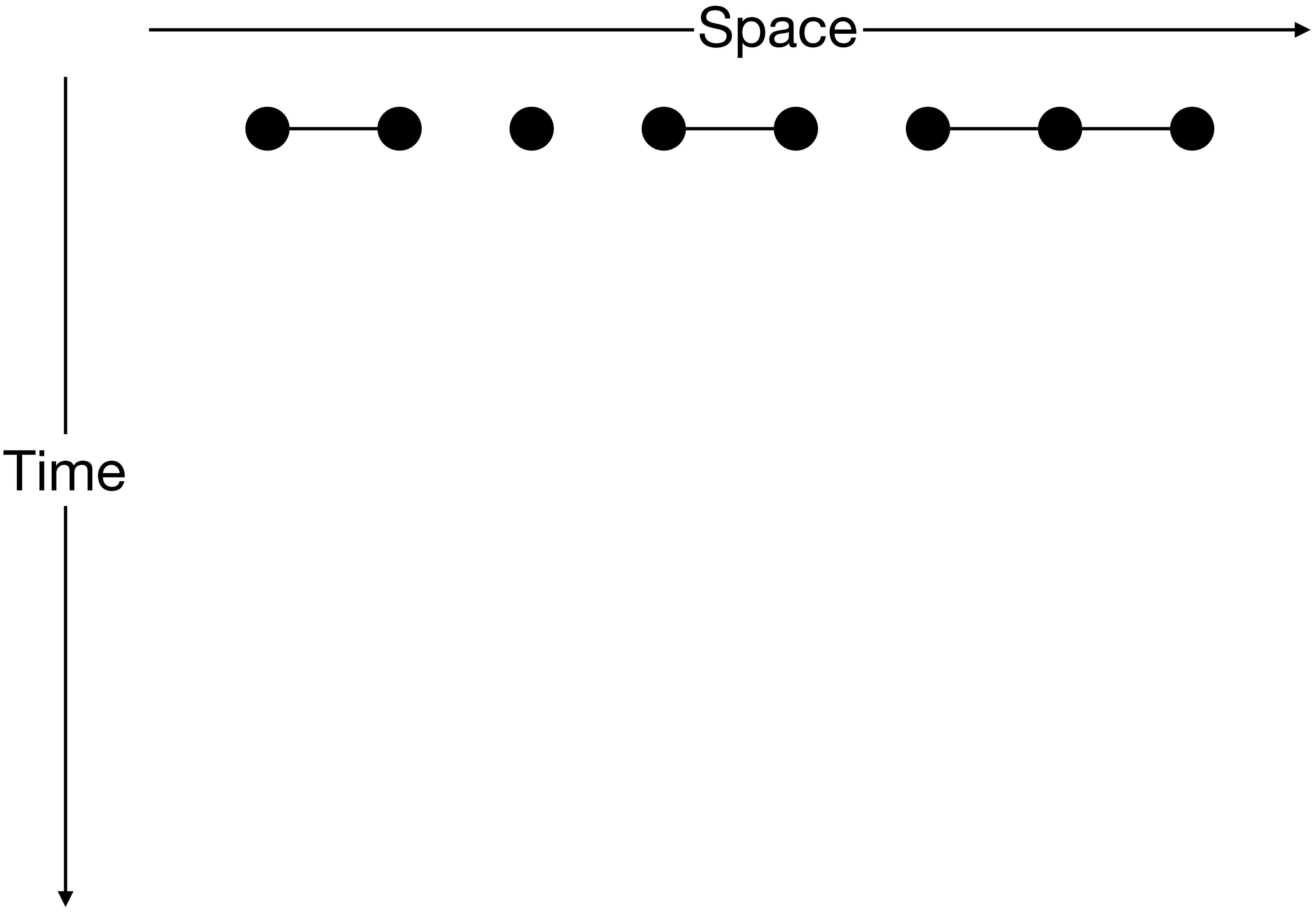
SORBONNE UNIVERSITÉ

SAND 2024

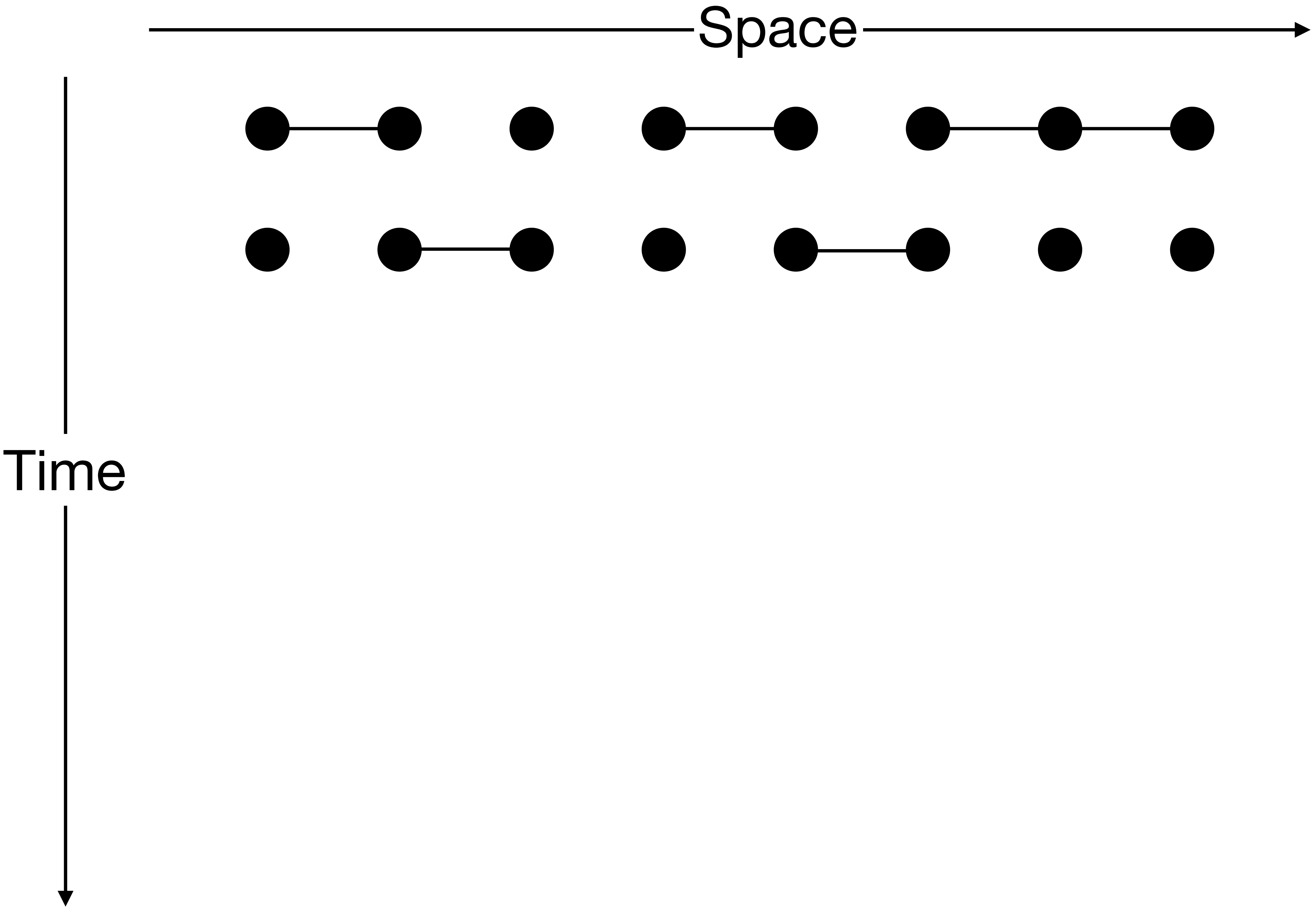
Dynamic Graphs



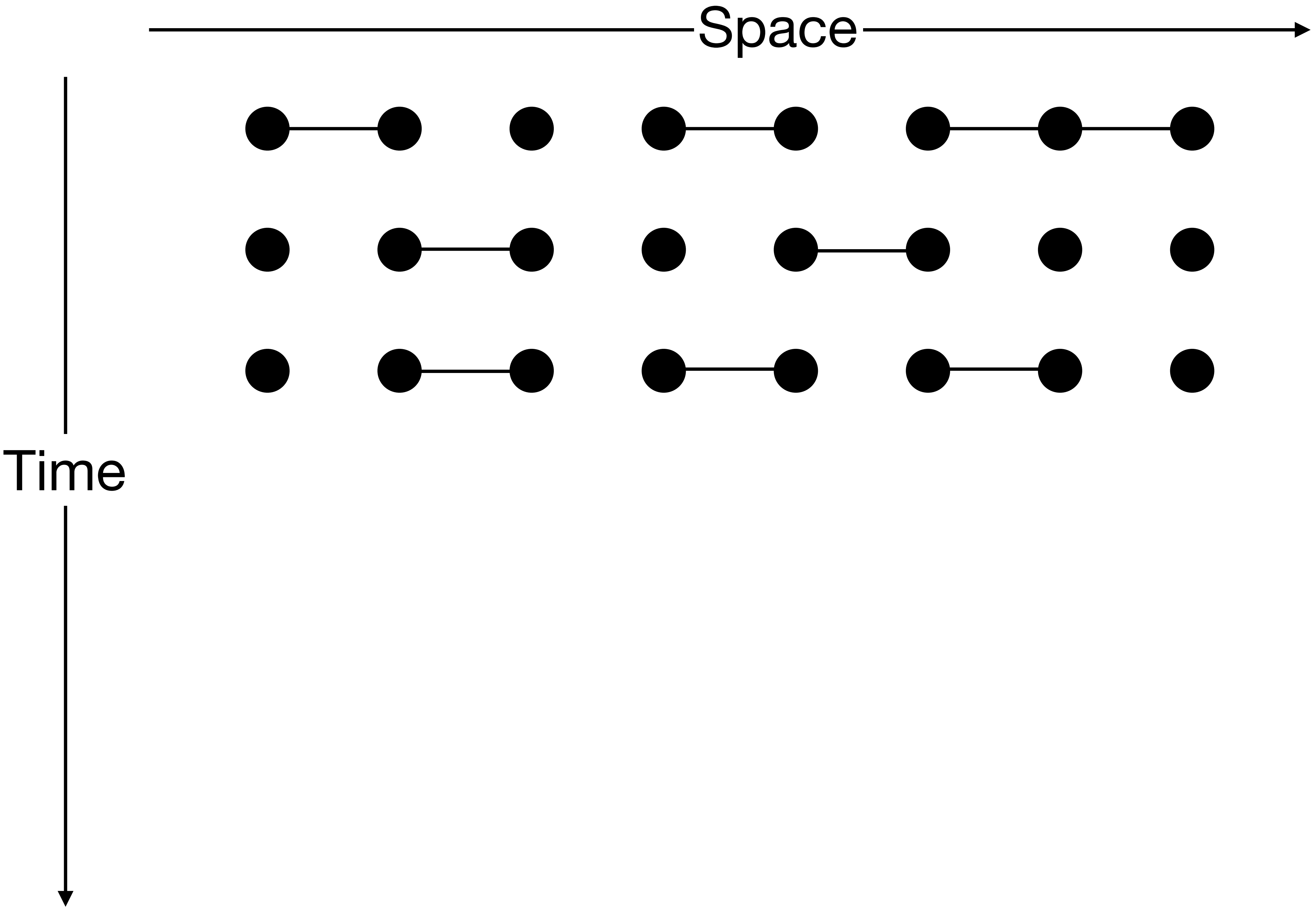
Evolving Graphs



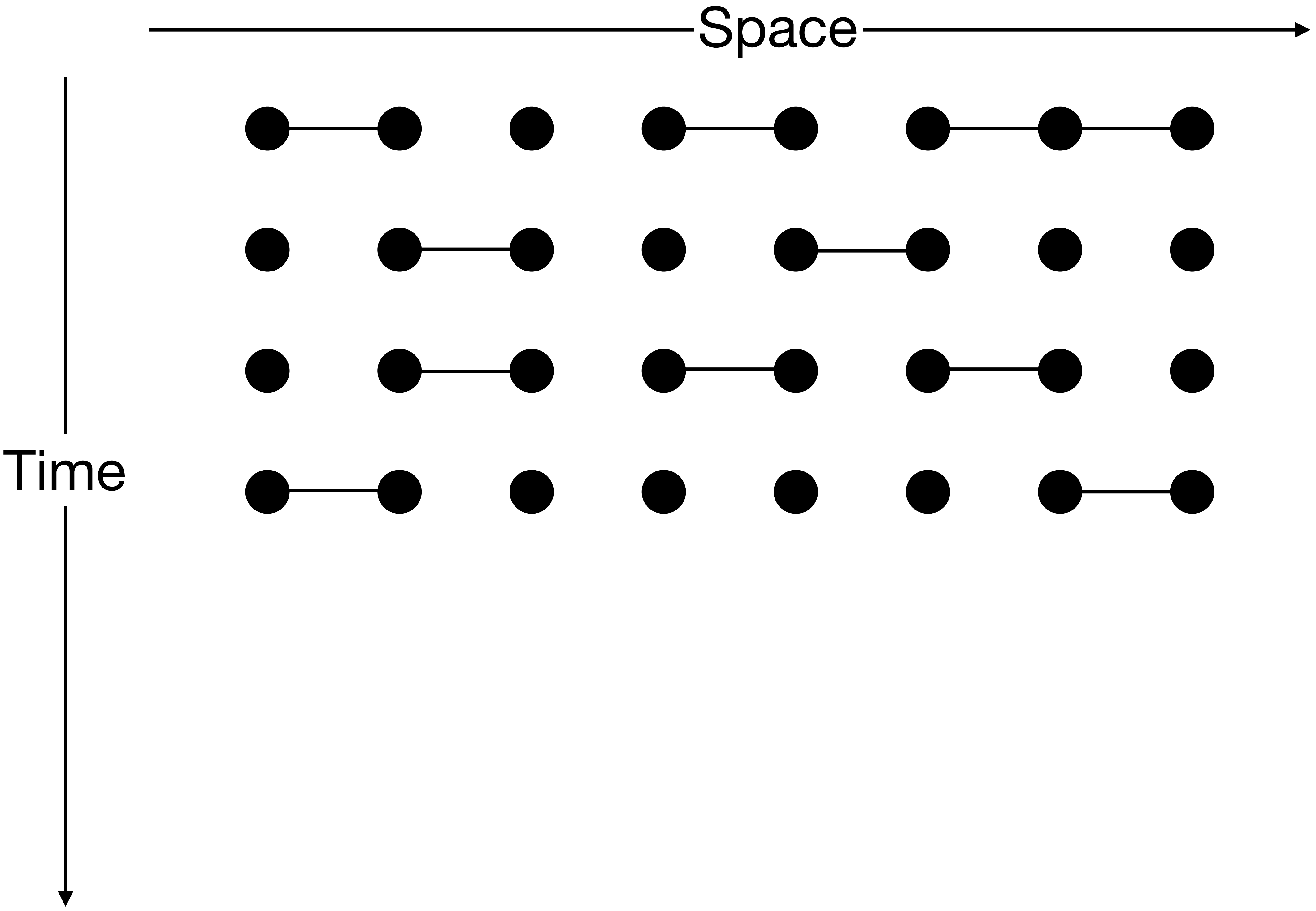
Evolving Graphs



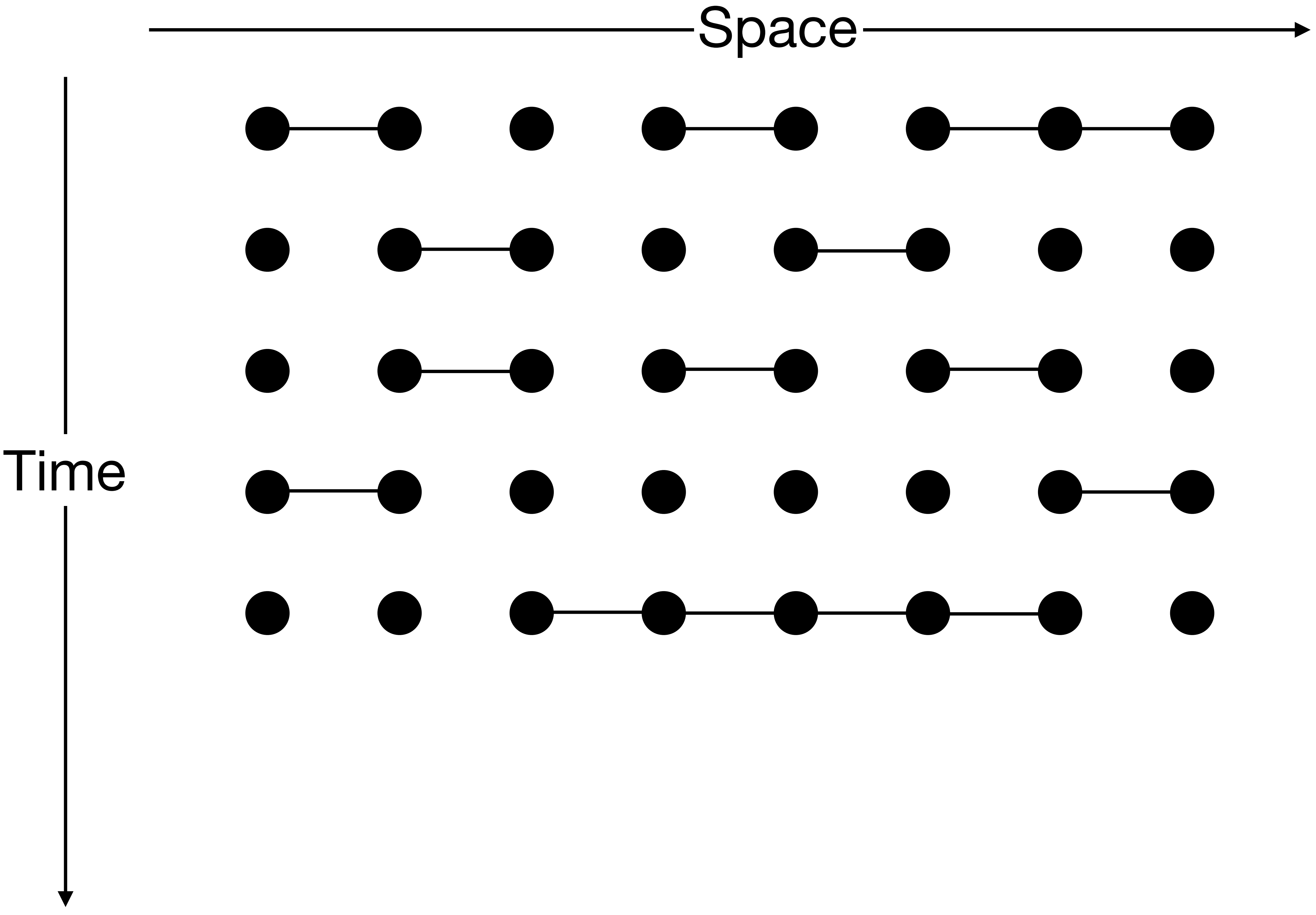
Evolving Graphs



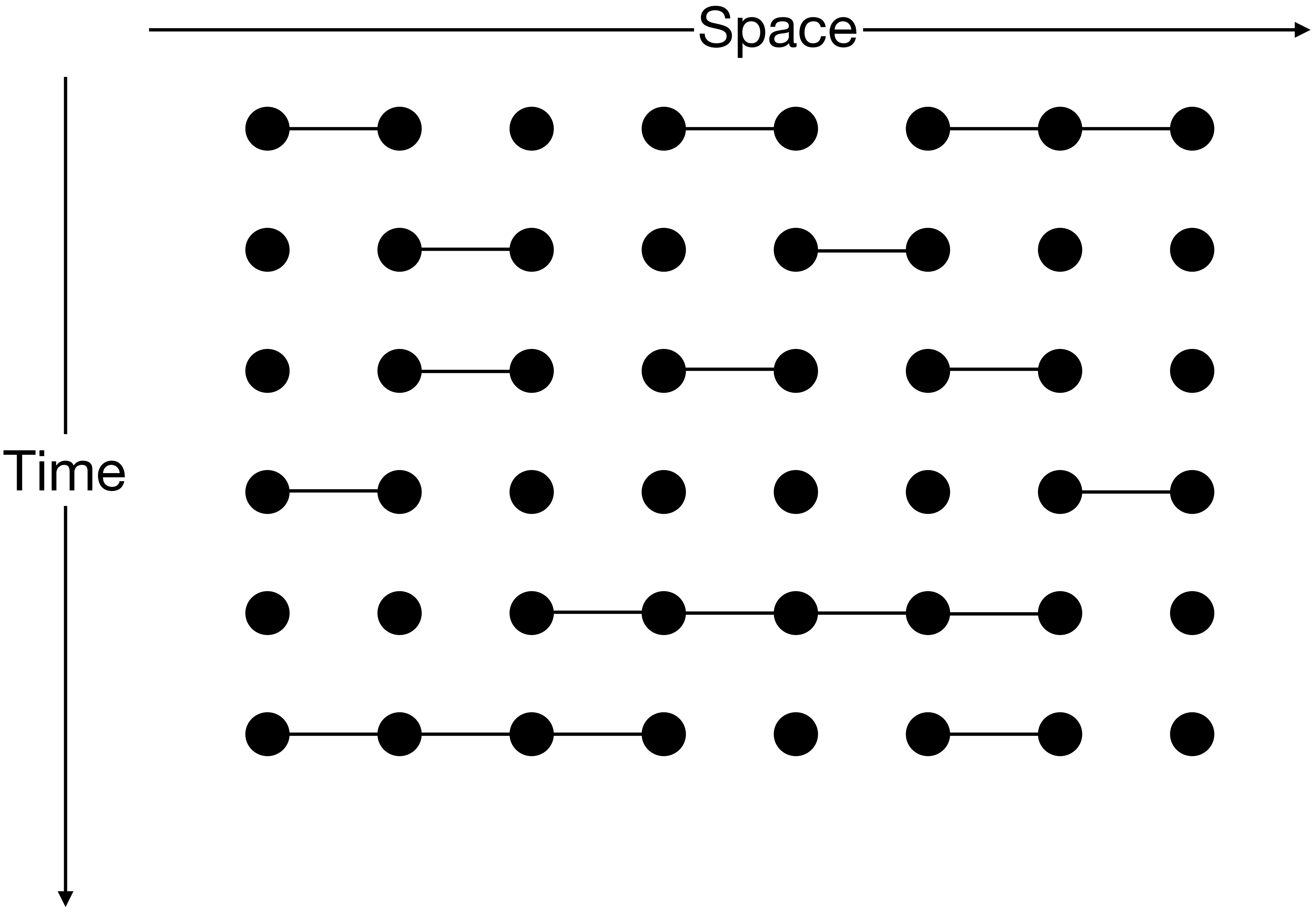
Evolving Graphs



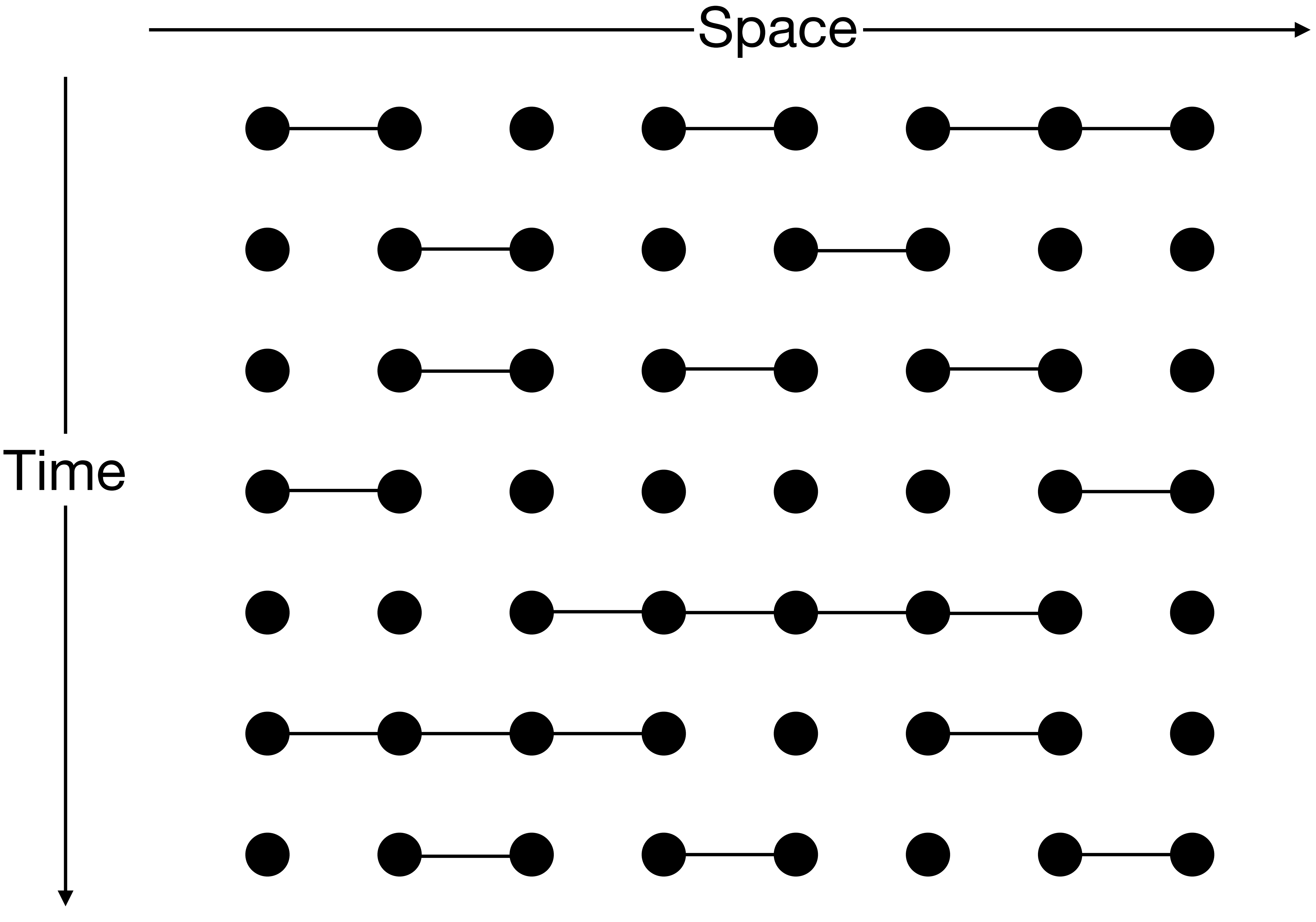
Evolving Graphs



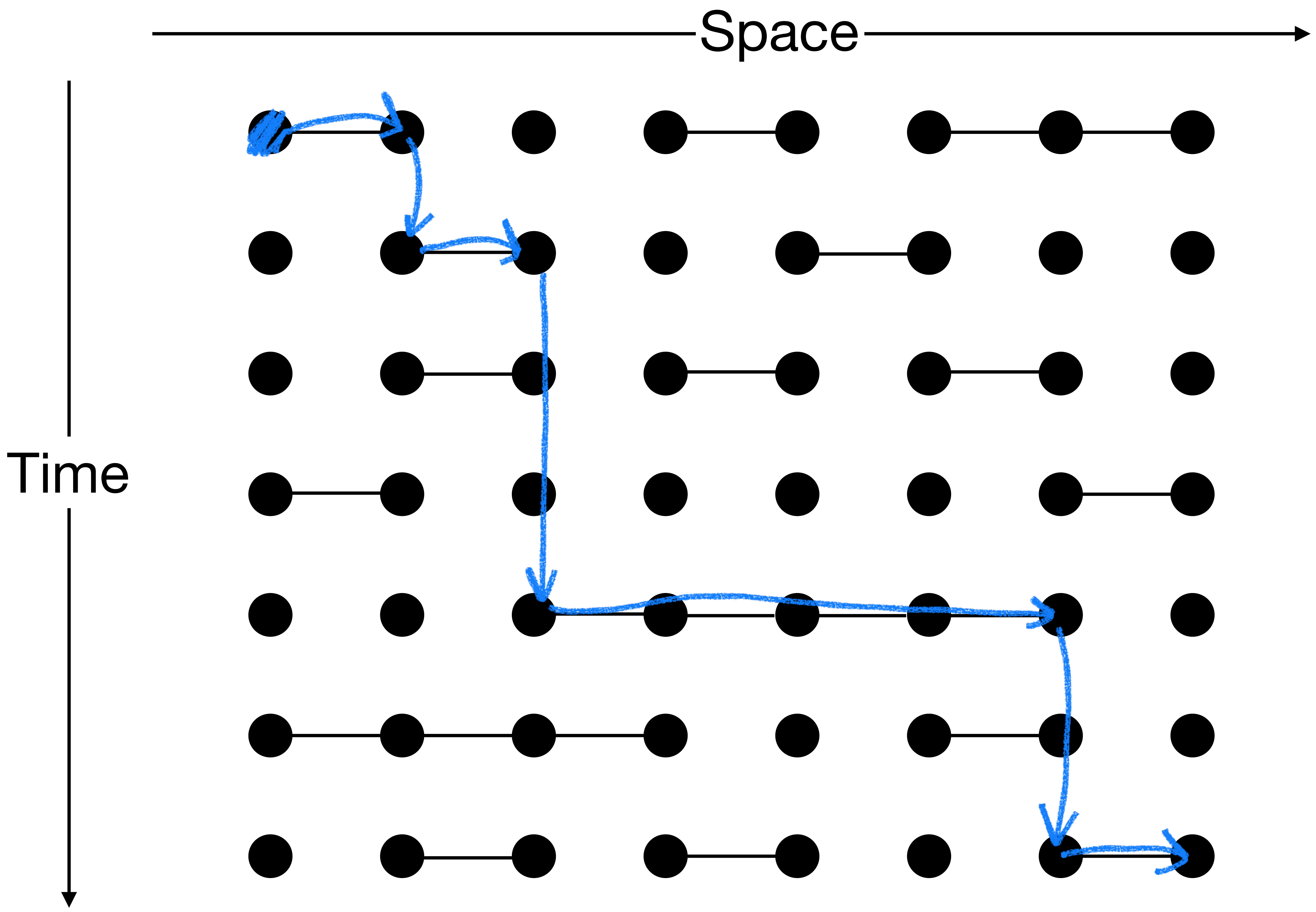
Evolving Graphs



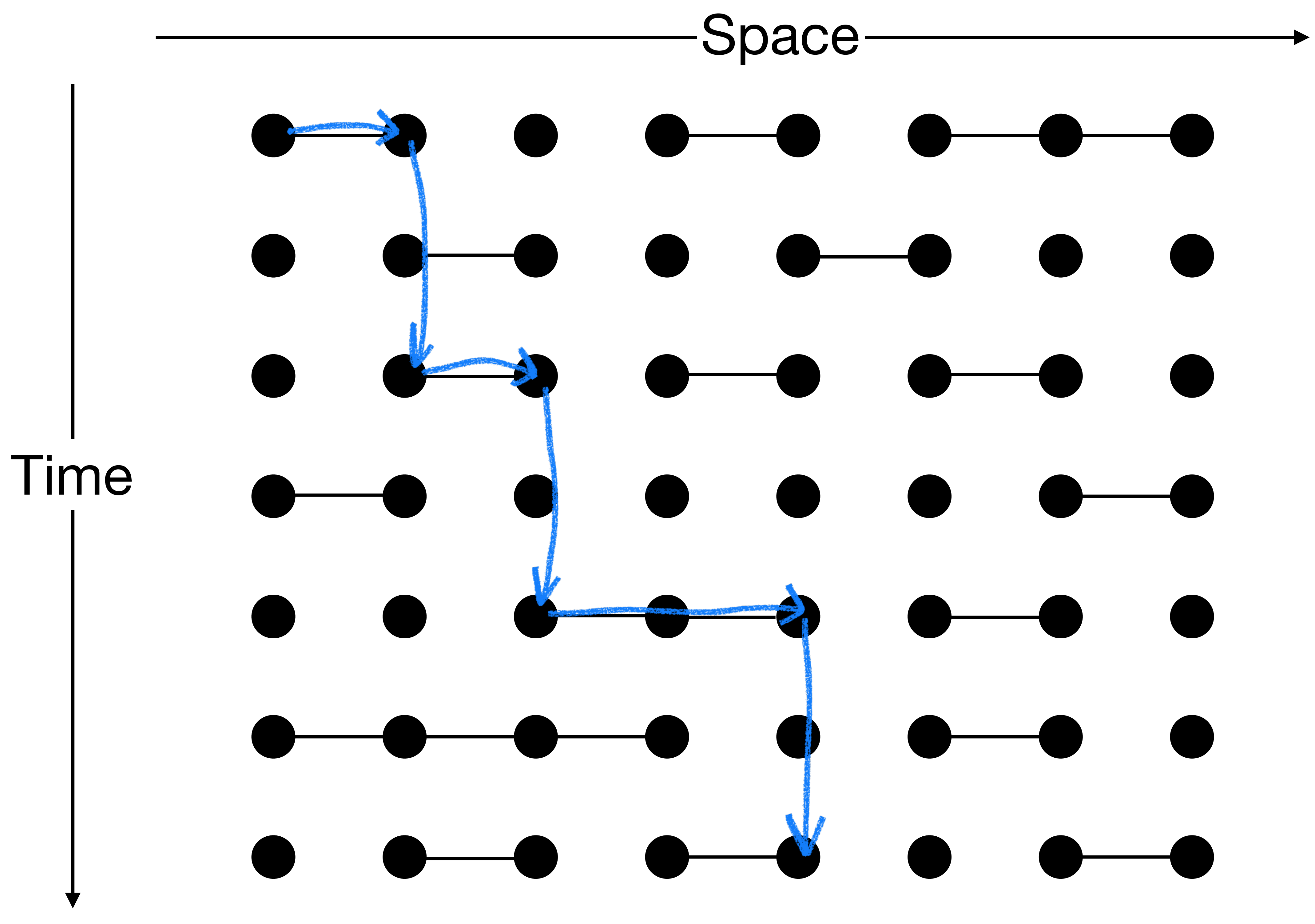
Evolving Graphs

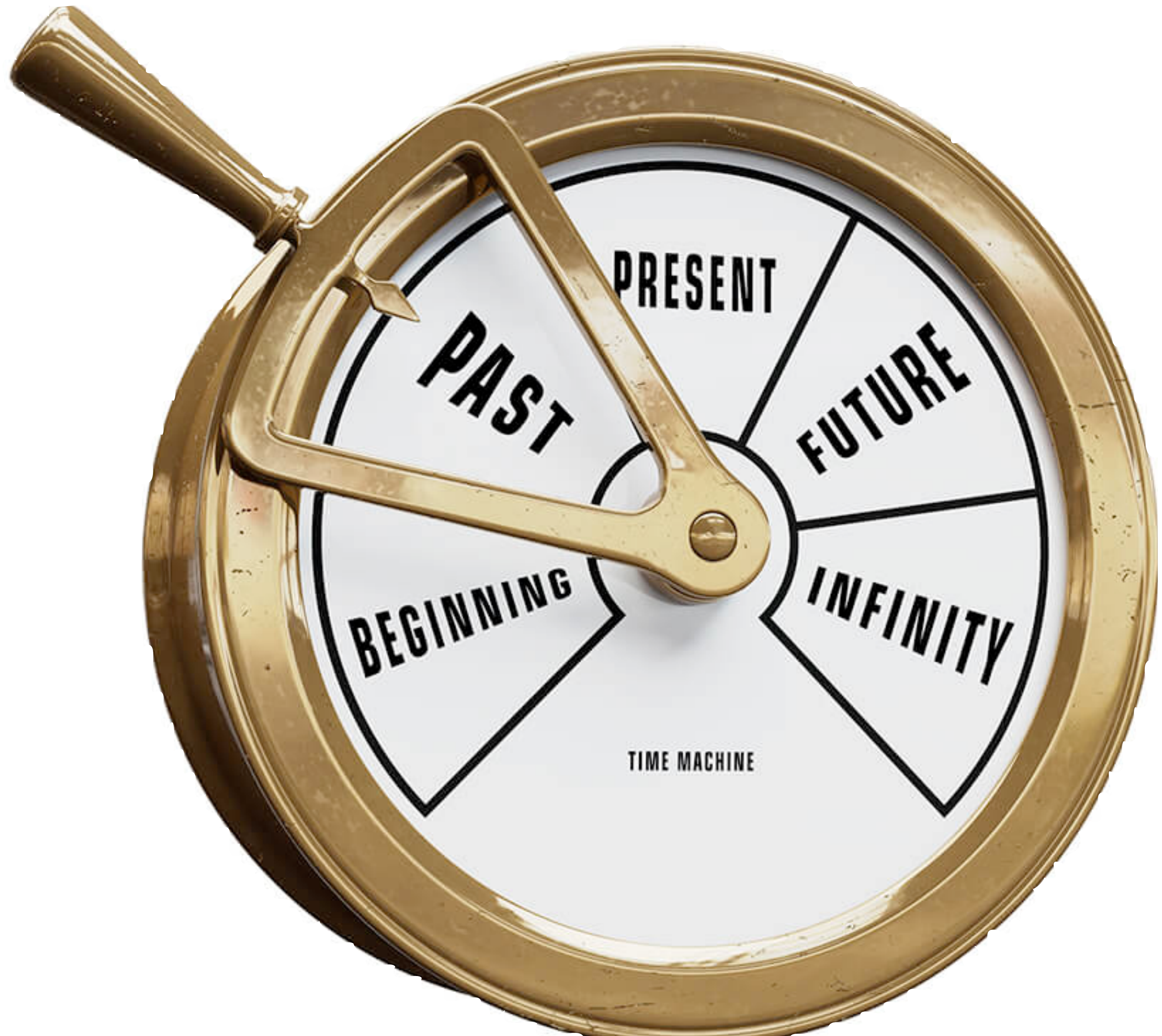


Journey

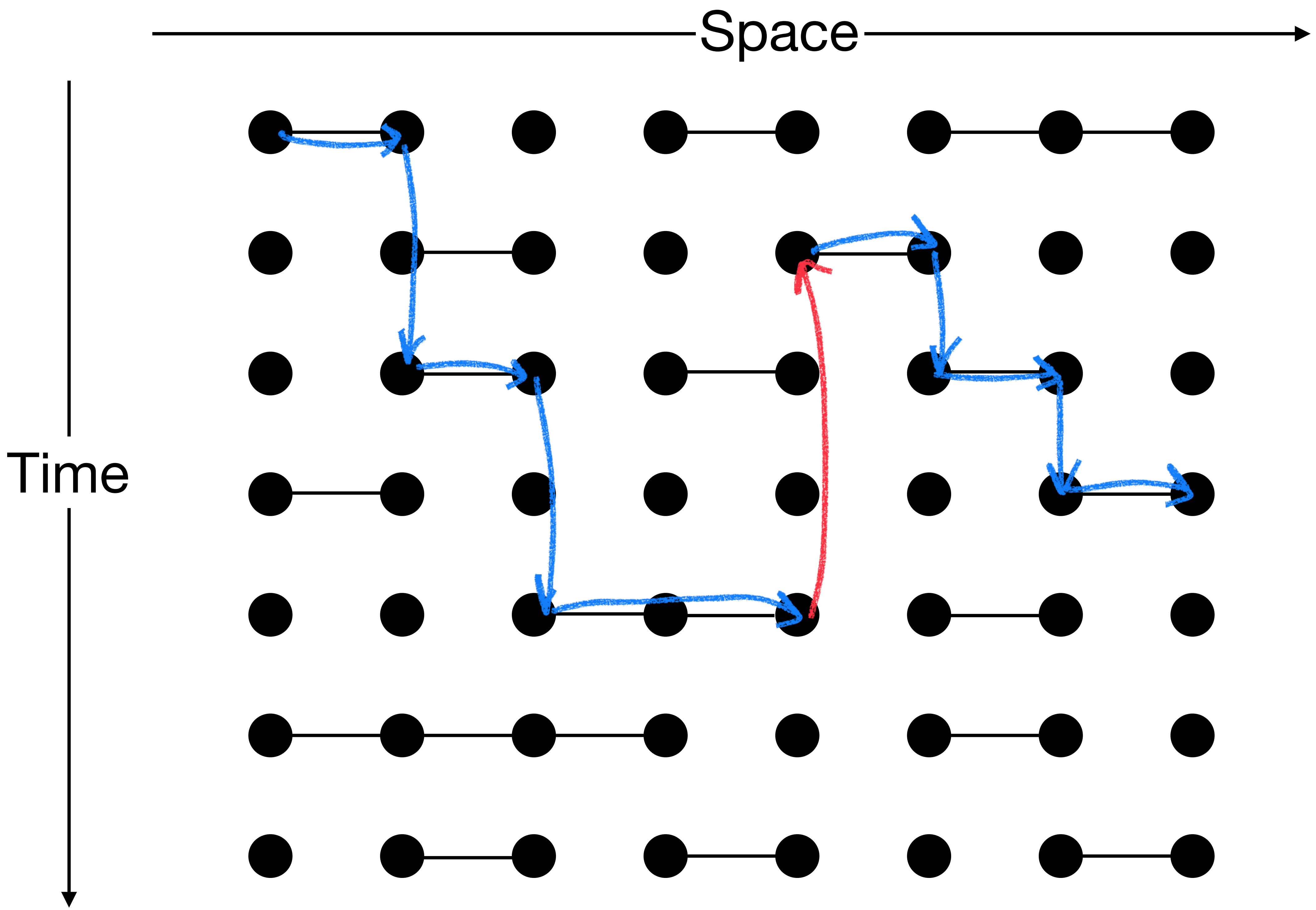


Journey

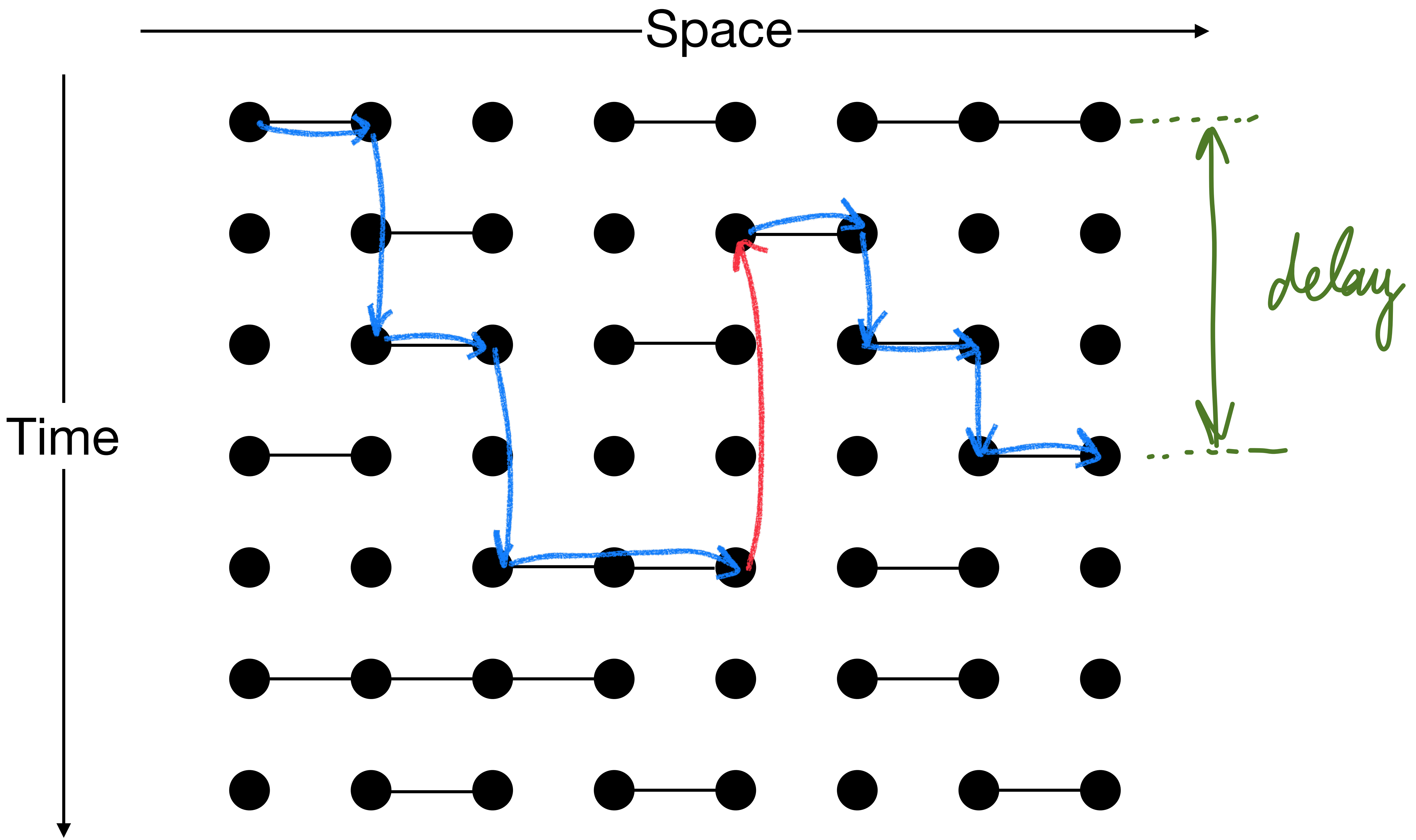



PRESENT
PAST
FUTURE
BEGINNING
INFINITY
TIME MACHINE

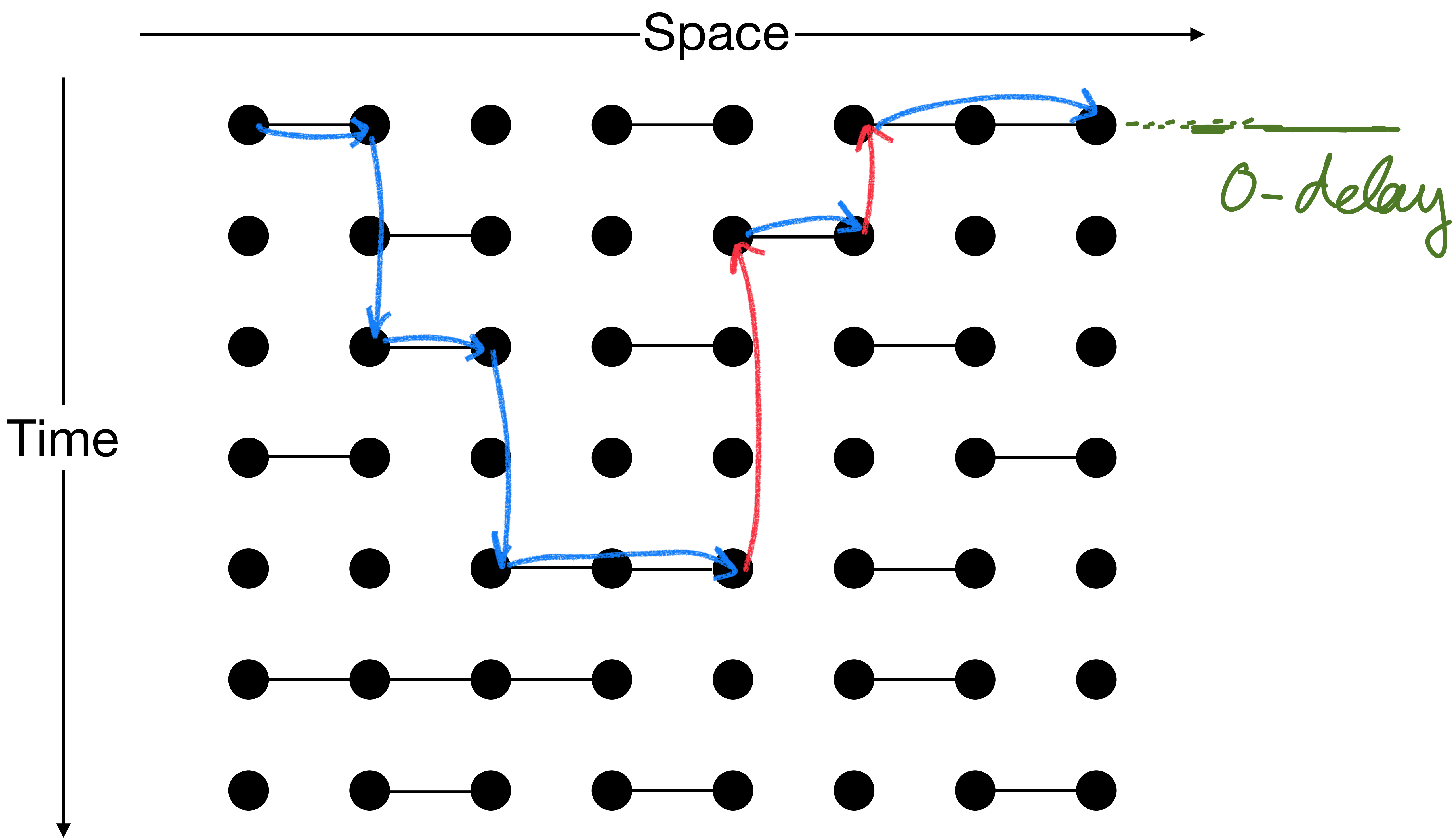
Space-Time Travel



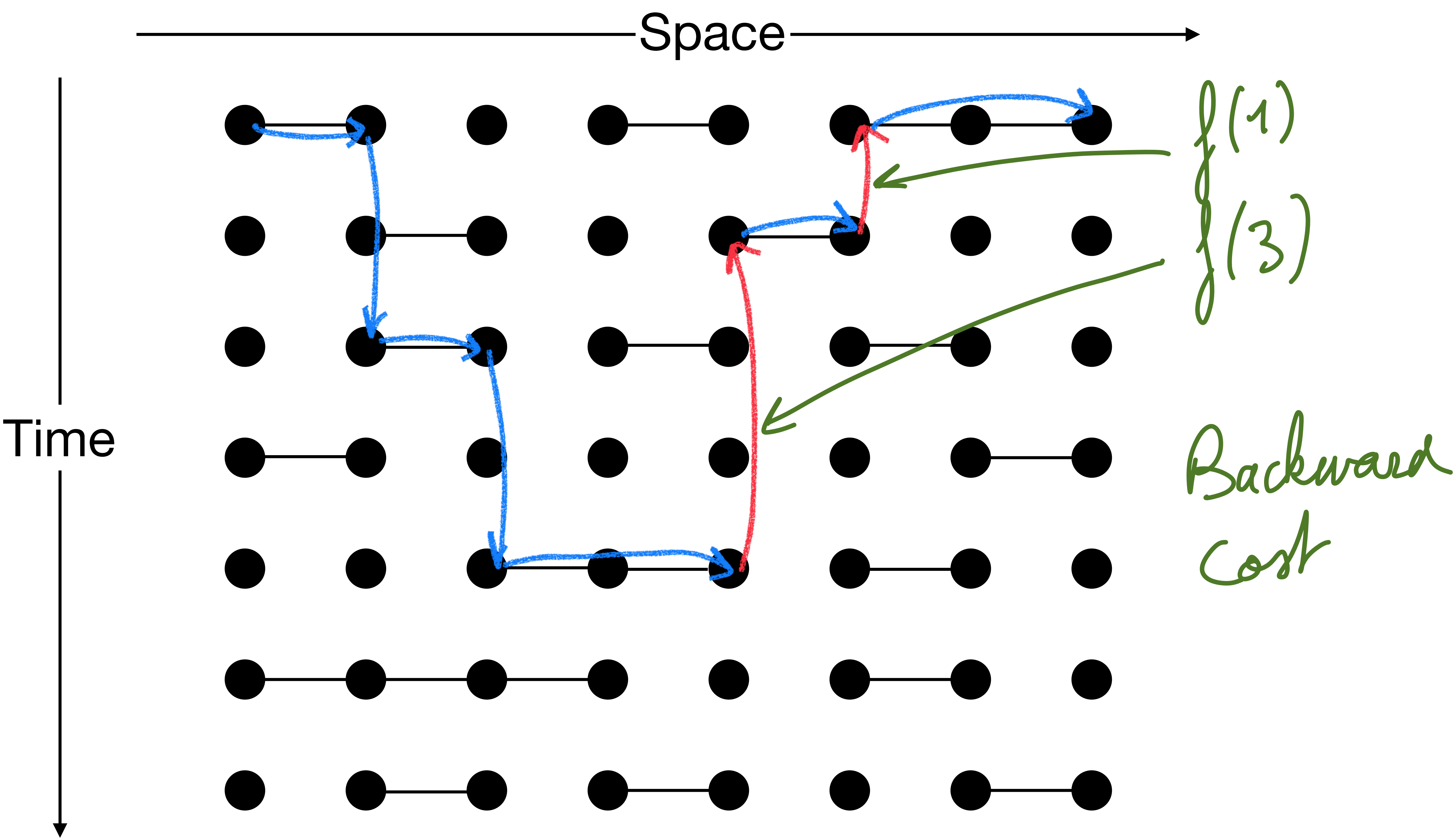
Travel Delay



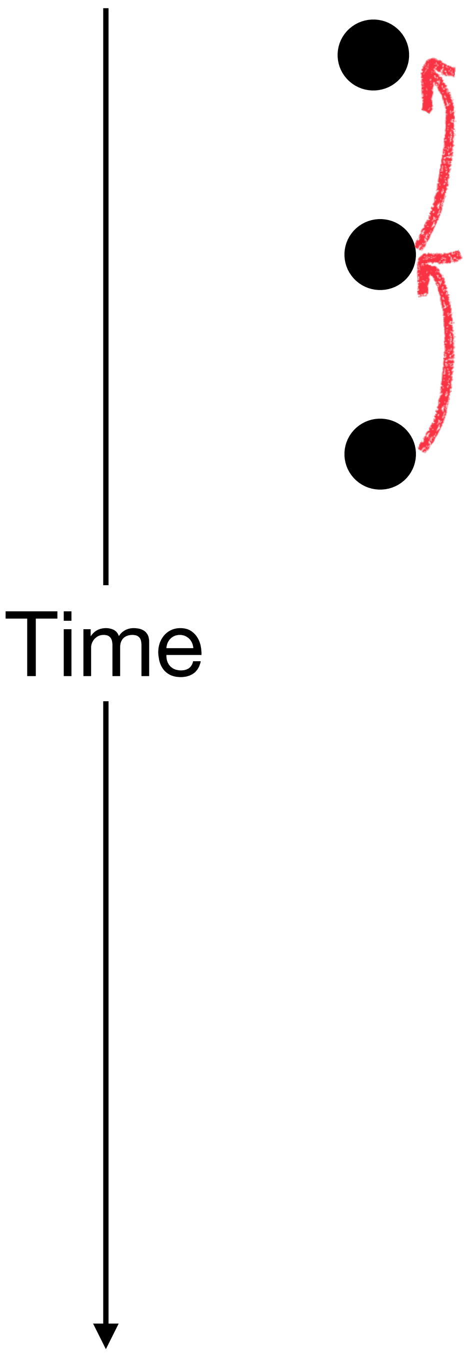
Travel Delay



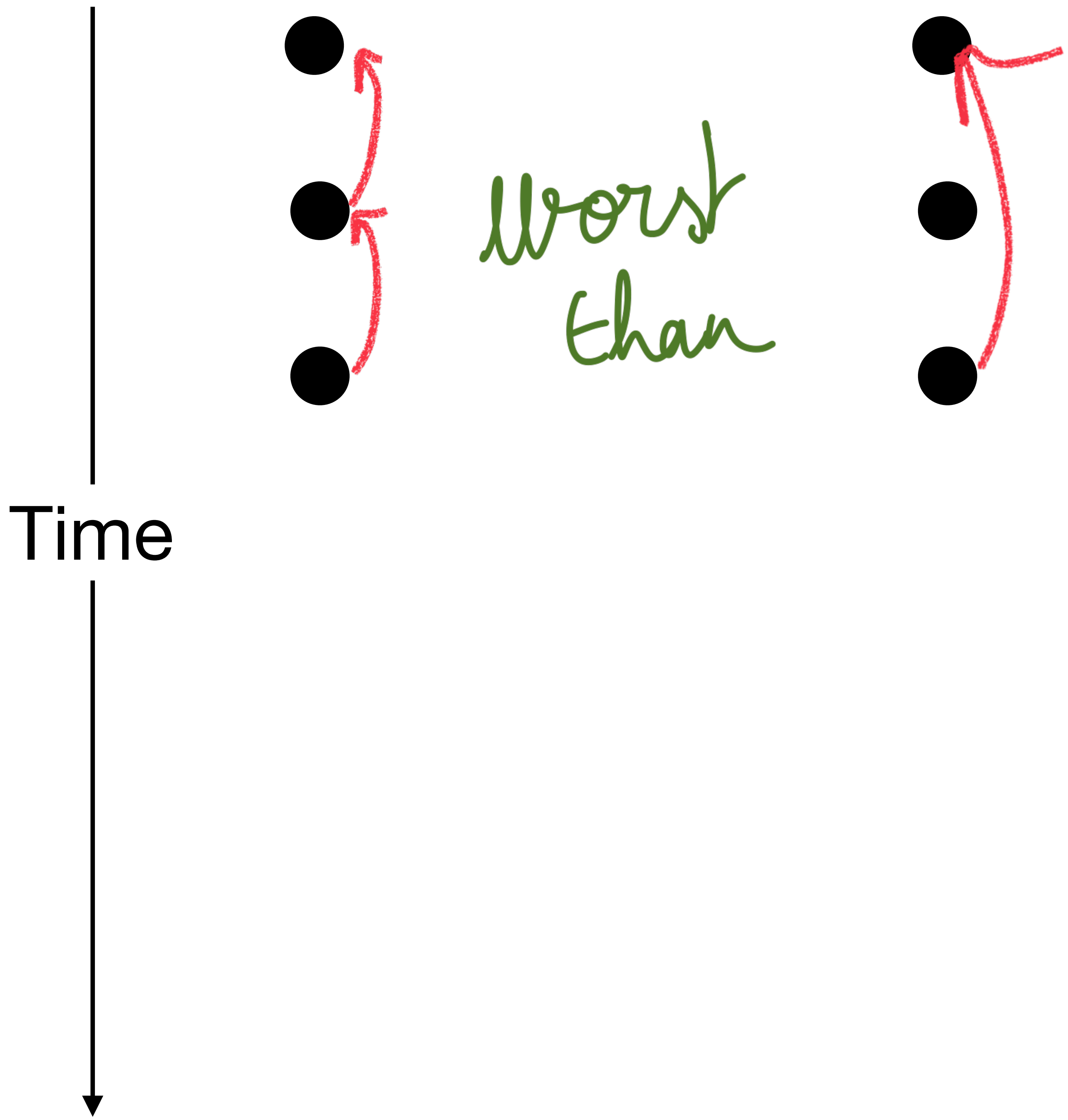
Travel Cost



Cost function nice properties



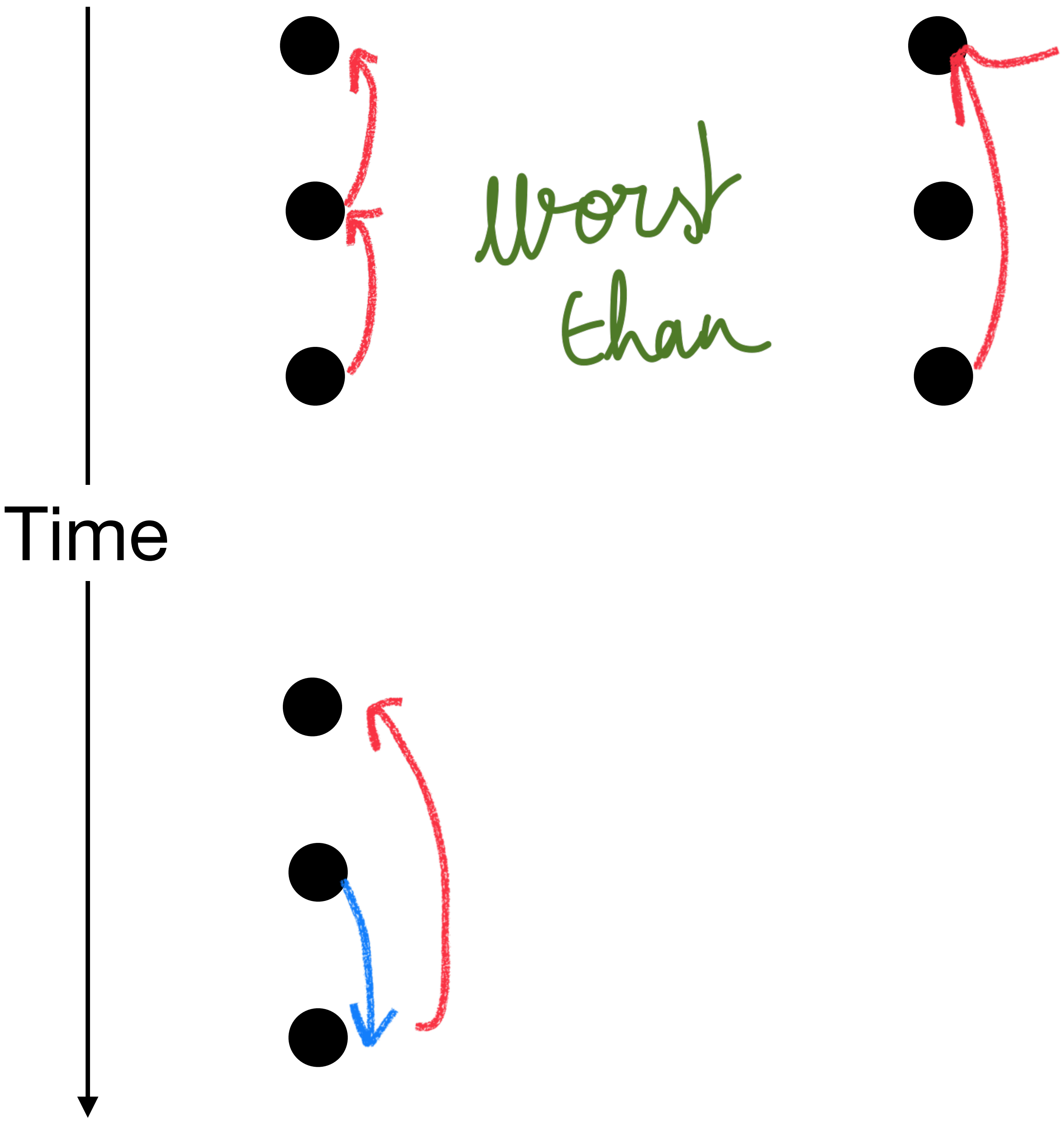
Cost function nice properties



$$f(2) \leq f(1) + f(1)$$

$\Rightarrow$ Sub-additive

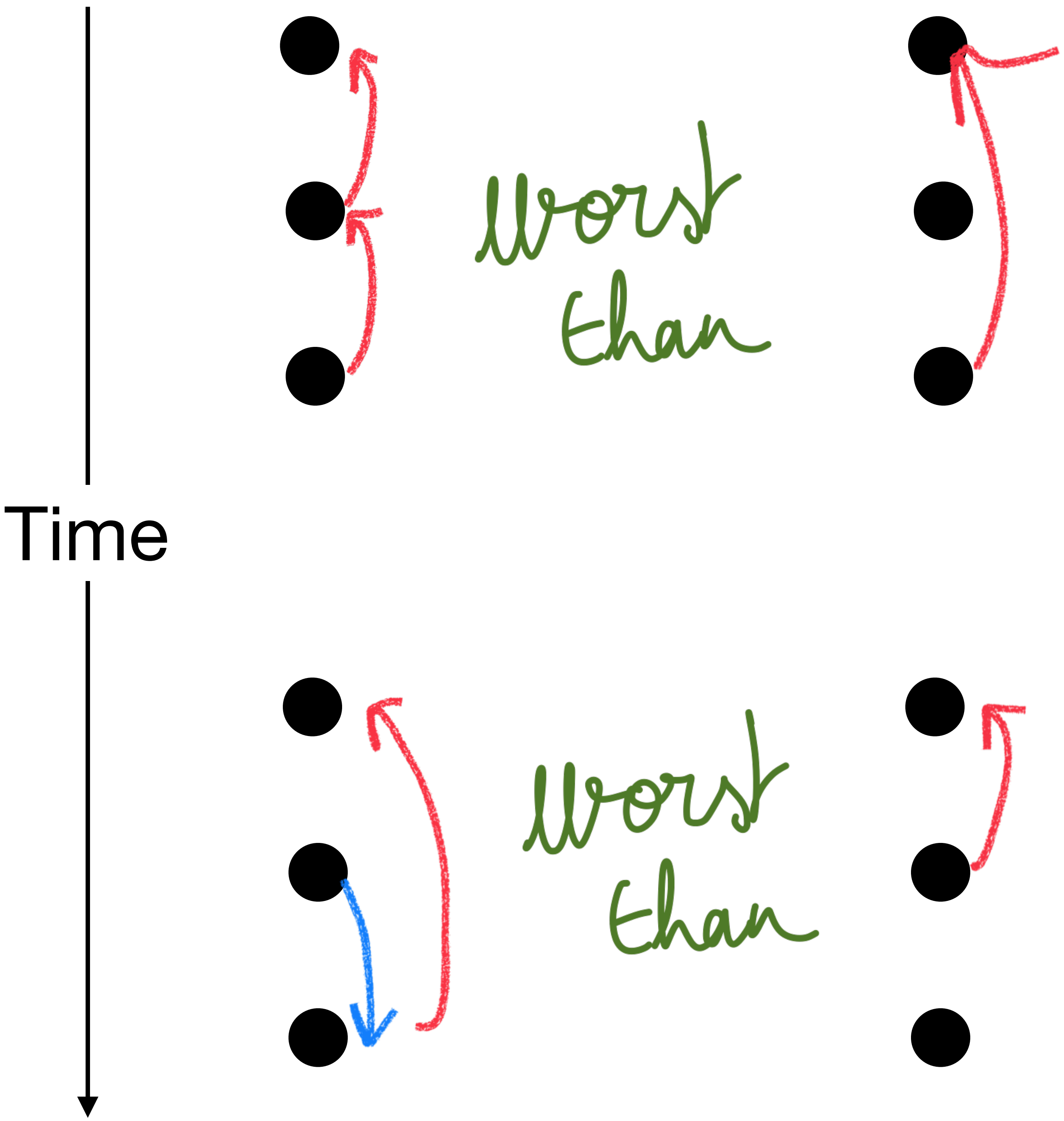
Cost function nice properties



$f(2) \leq f(1) + f(1)$

$\Rightarrow$ Sub-additive

Cost function nice properties



$f(2) \leq f(1) + f(1)$

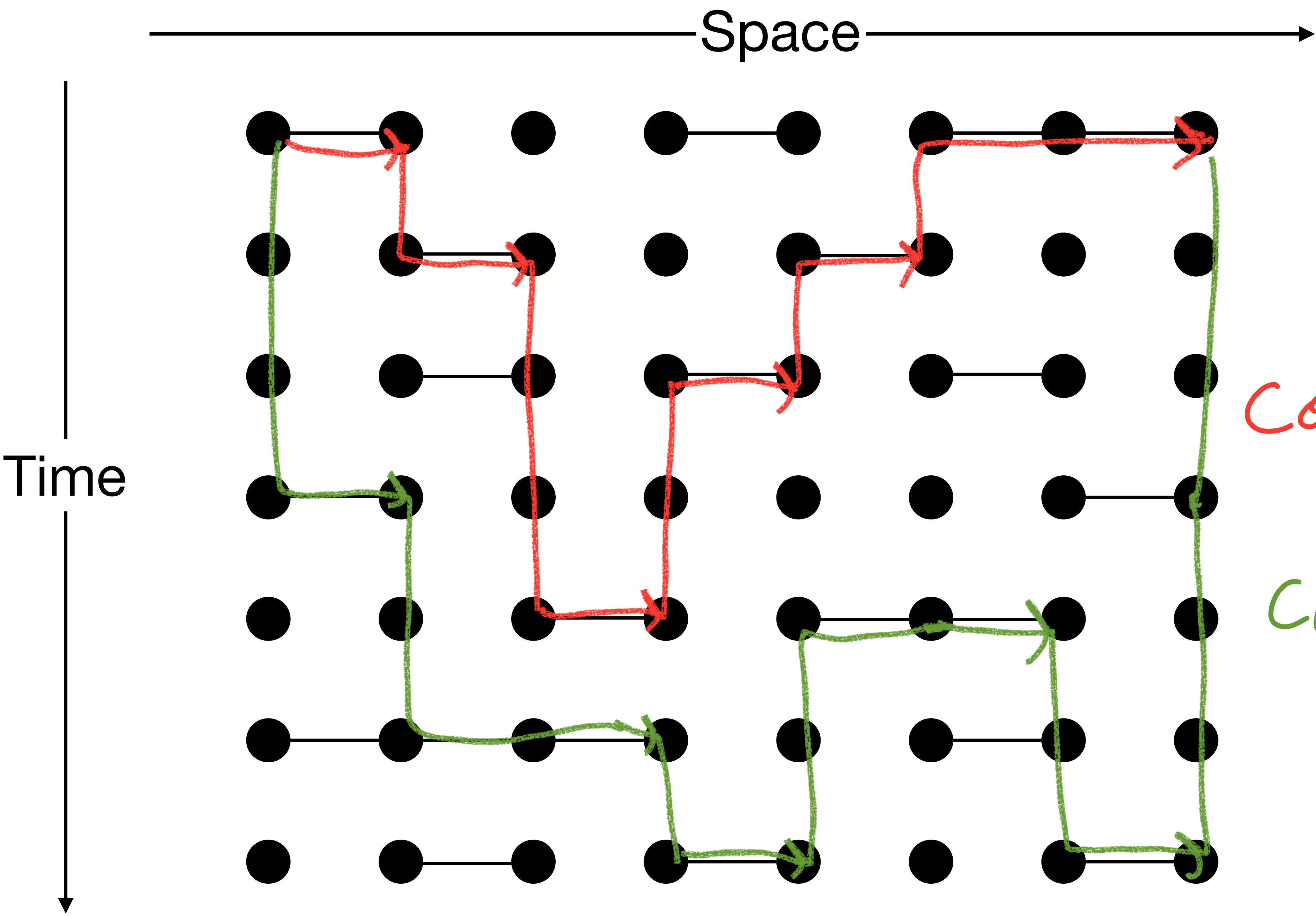
$\Rightarrow$ Sub-additive

$\Rightarrow$ non-decreasing

(and positive)

The ODOC Problem

$f : x \mapsto x$

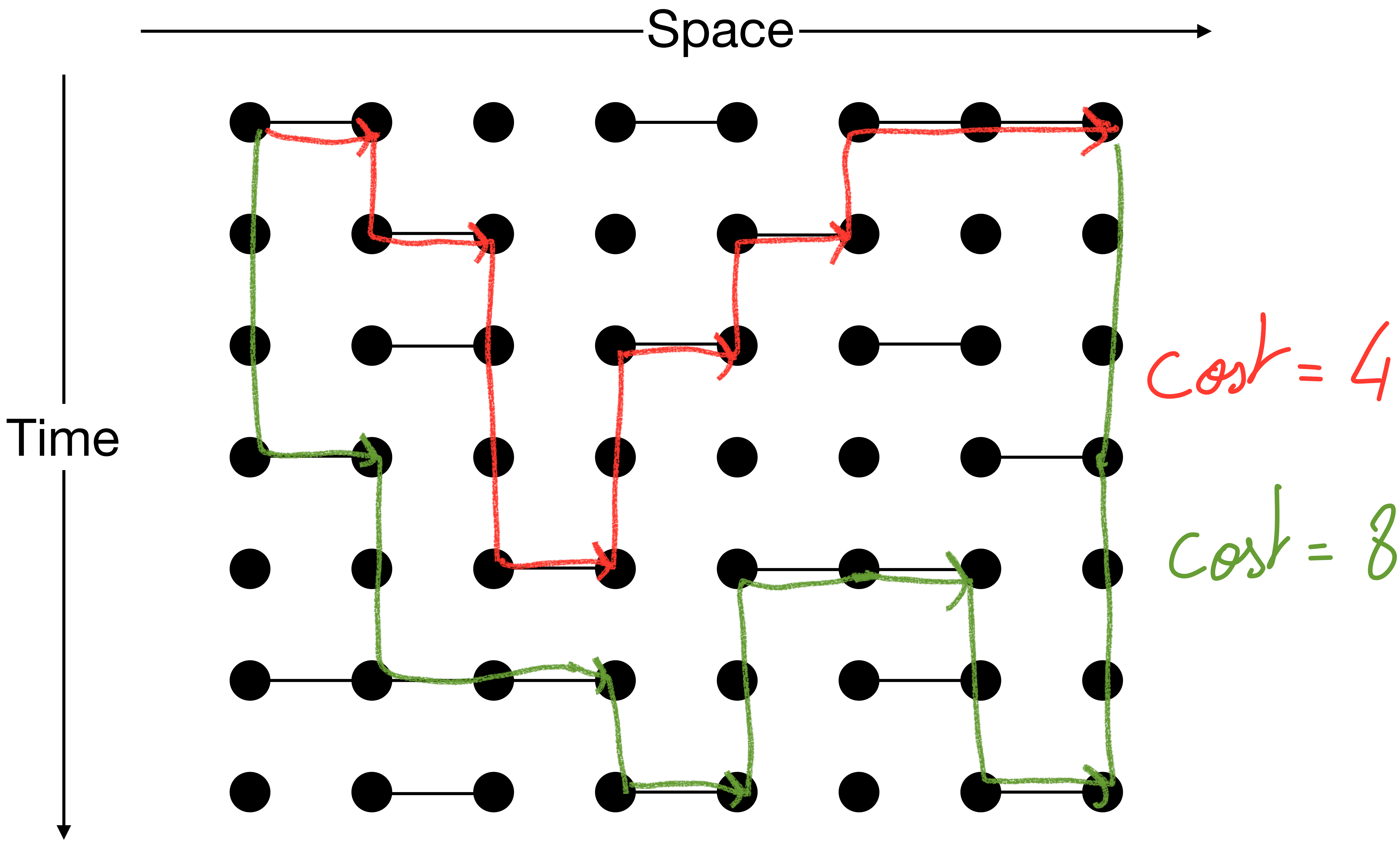


cost =

cost =

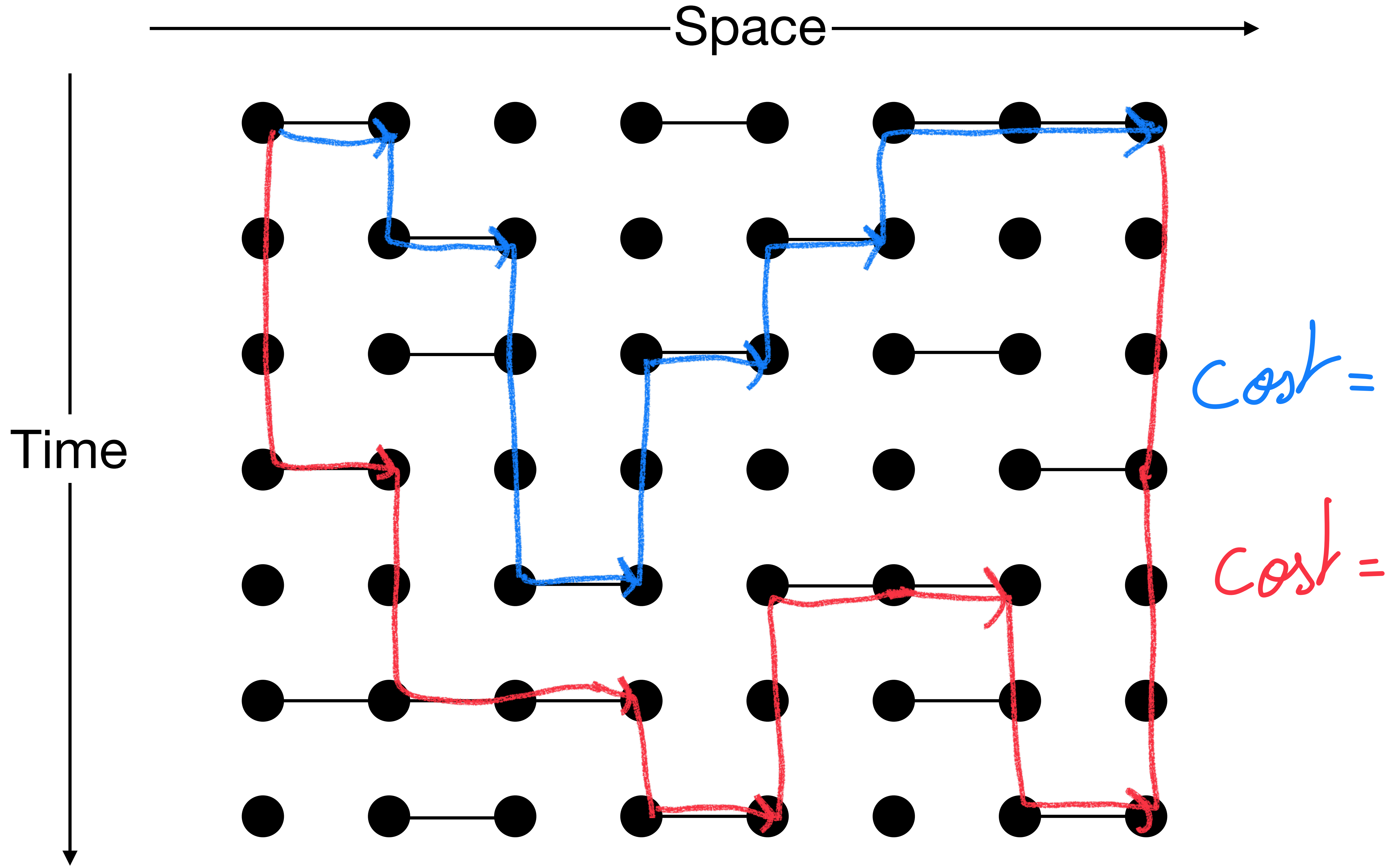
The ODOC Problem

$f : x \mapsto x$



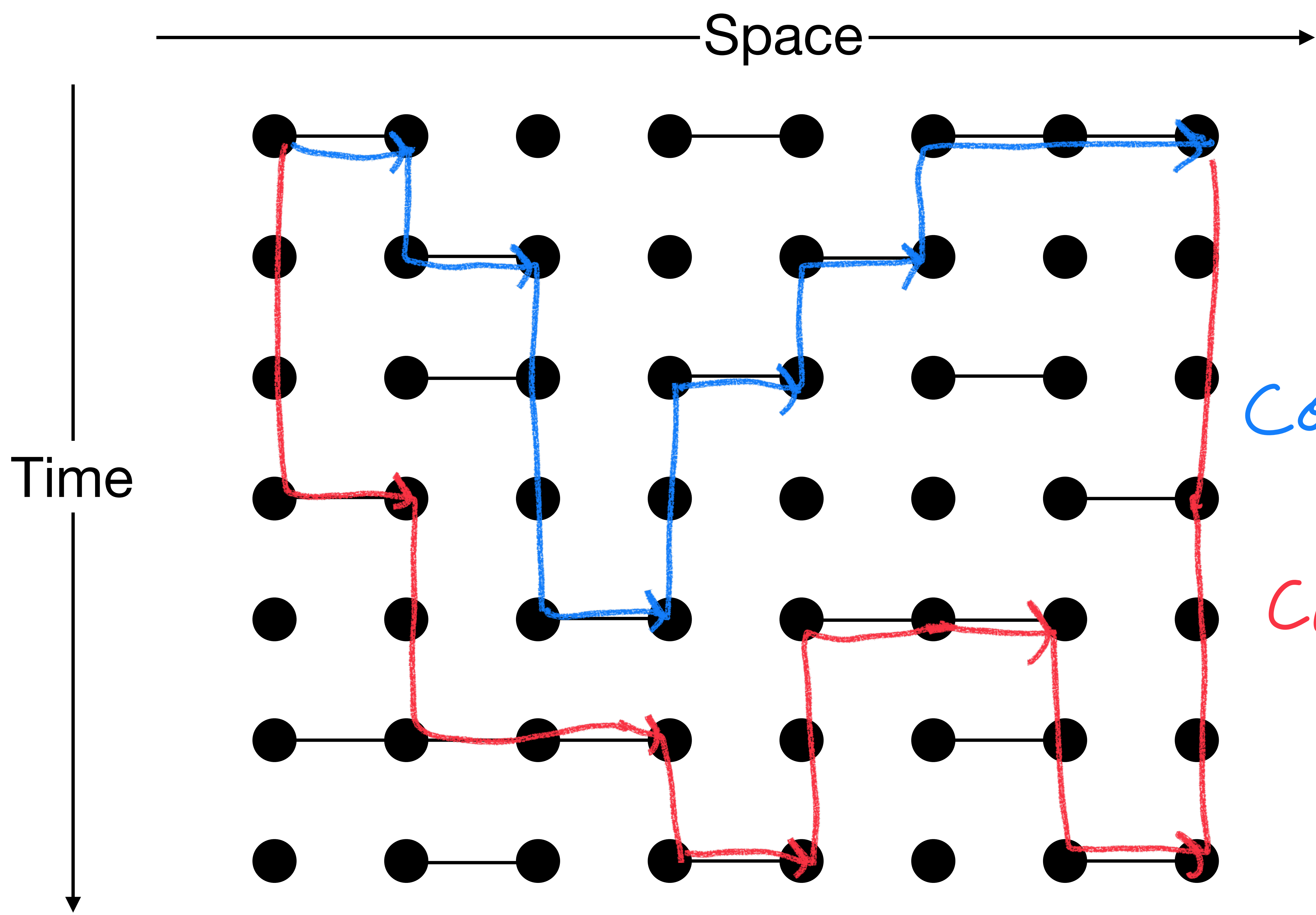
The ODOC Problem

$f: x \mapsto 1$



The ODOC Problem

$f: x \mapsto 1$



cost = 3

cost = 2

Related Work

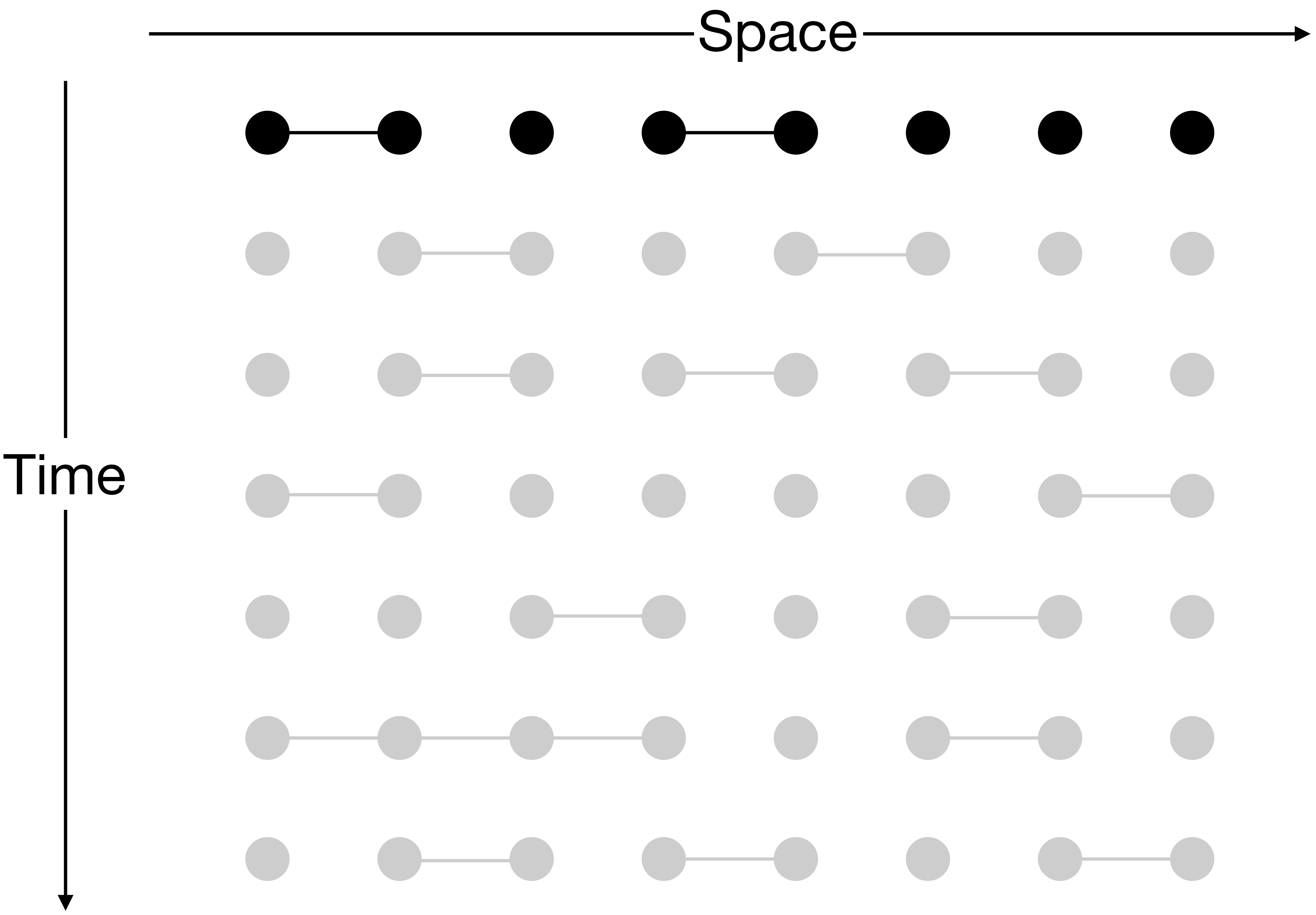
SSS 2023

Necessary condition on the cost function

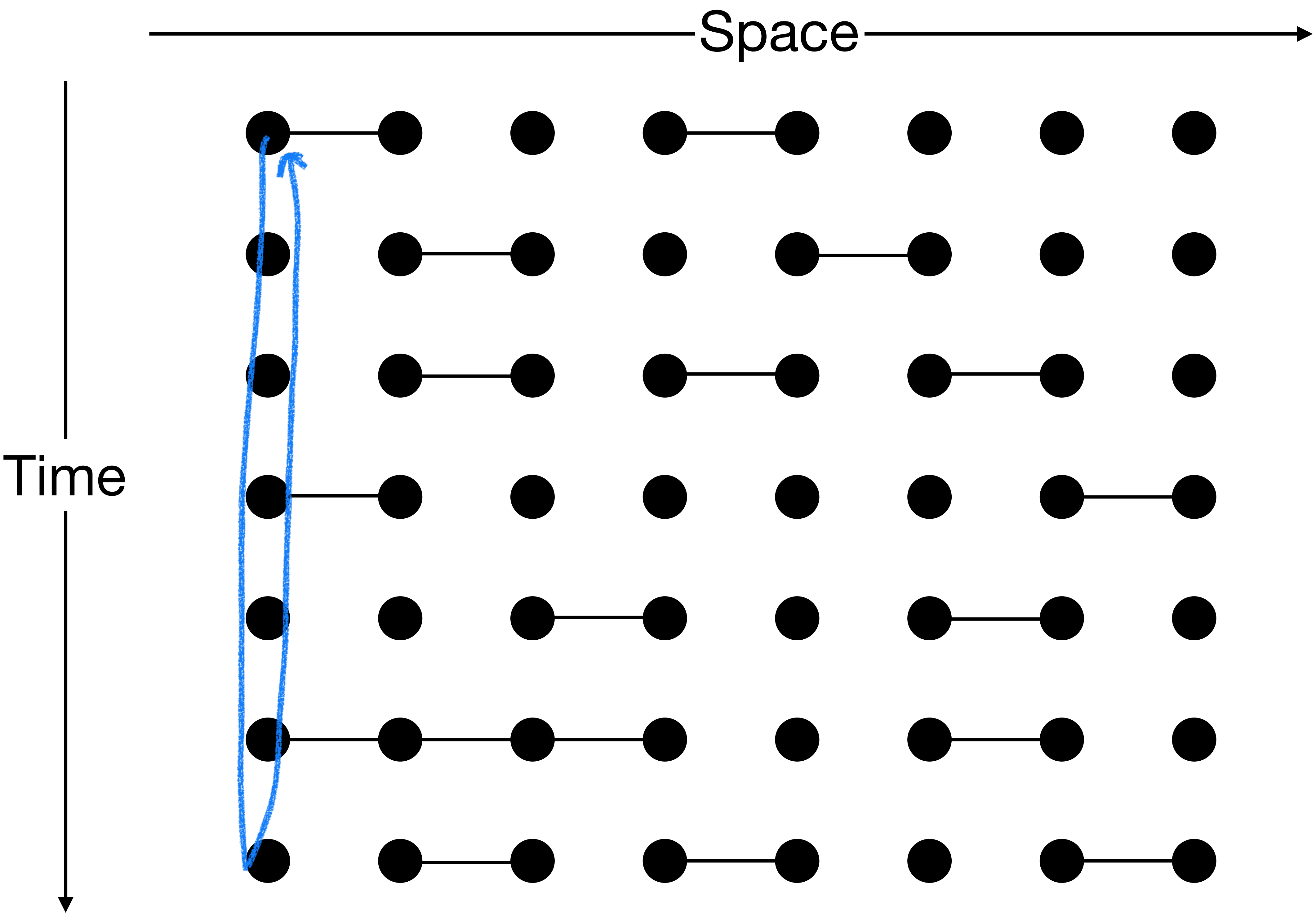
Offline ODOC problem is in P, even with a constraint on the cost, even in graphs evolving for infinity.

But, may create paradoxes!

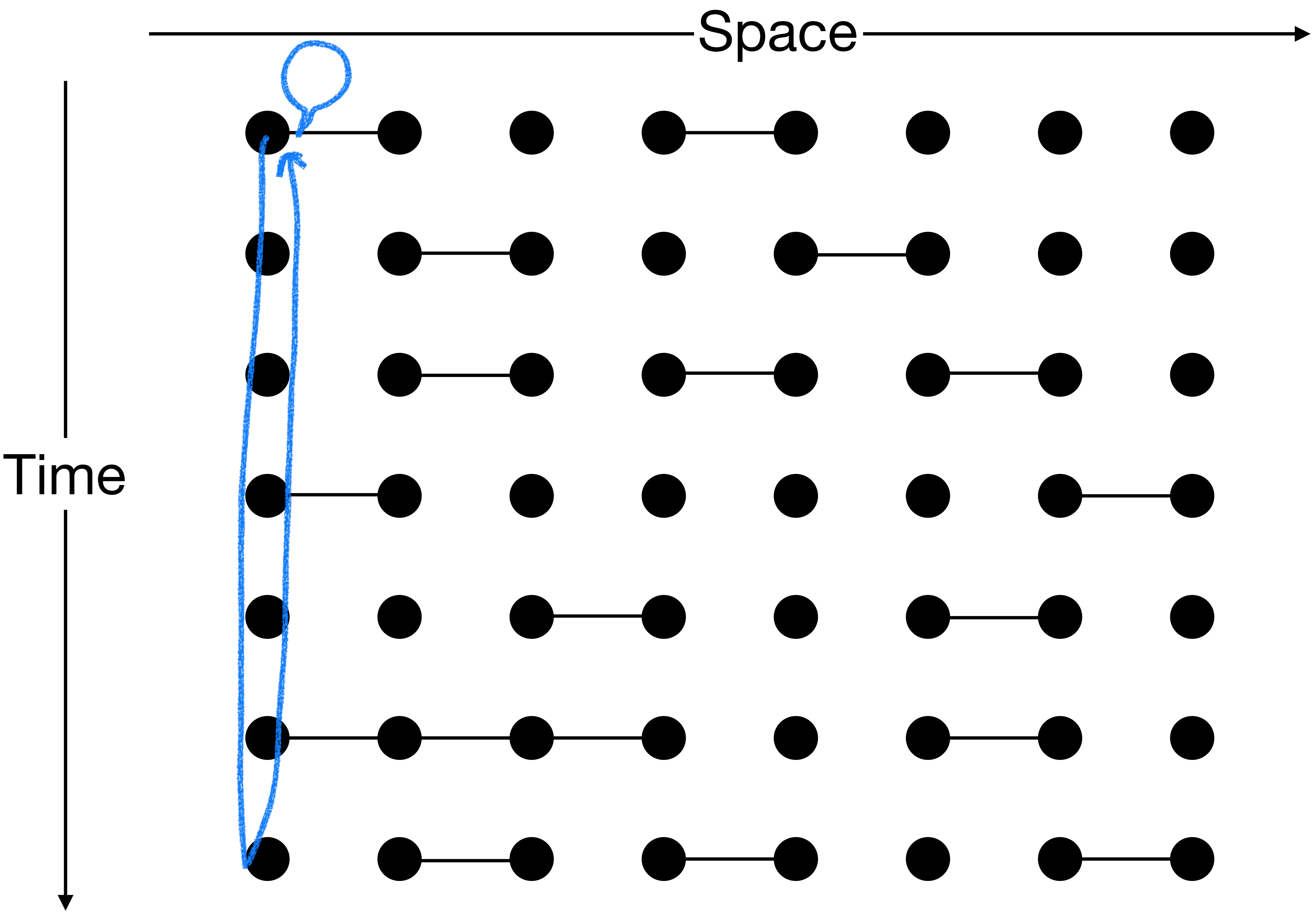
Learning paradoxe



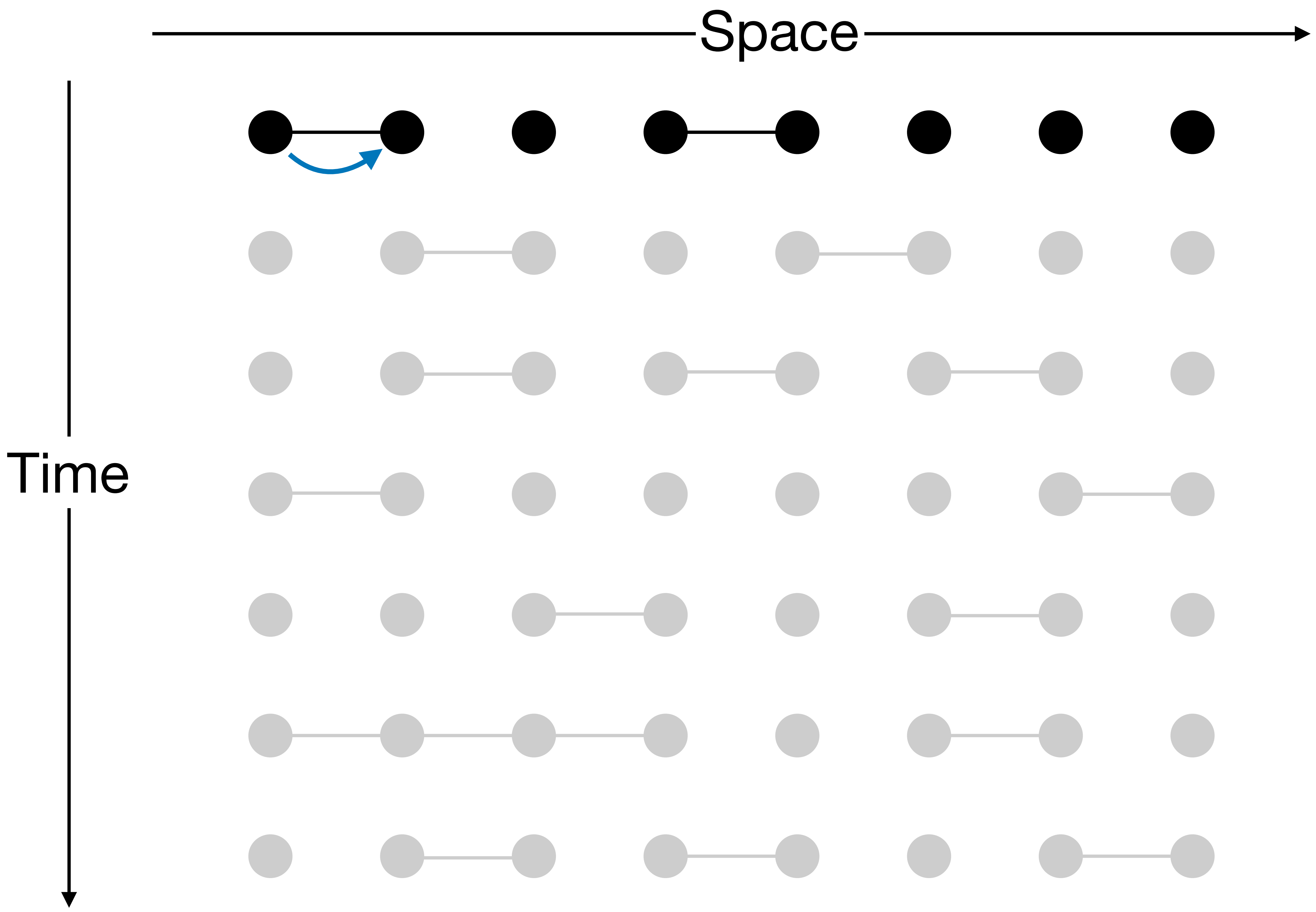
Learning paradoxe



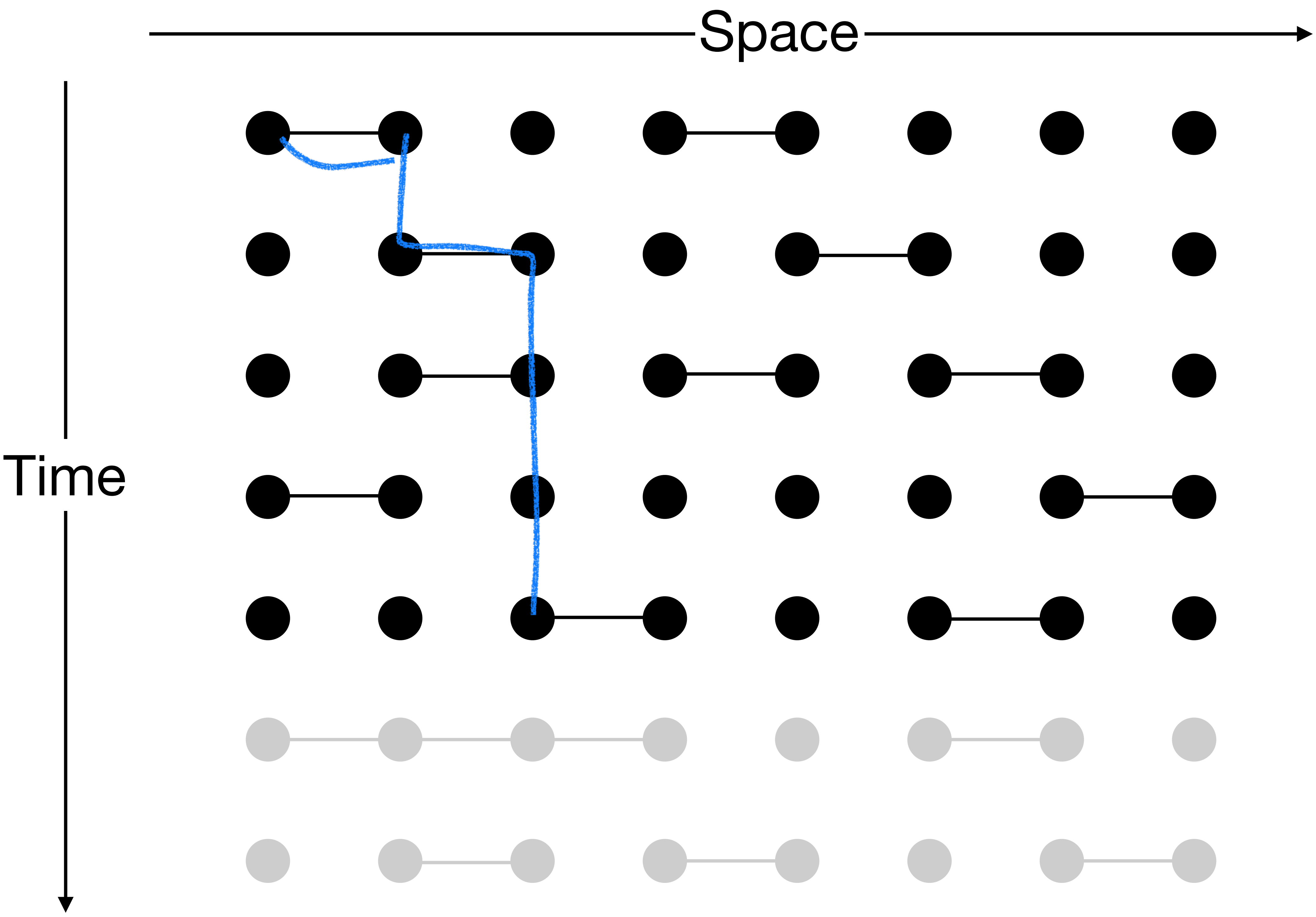
Learning paradoxe



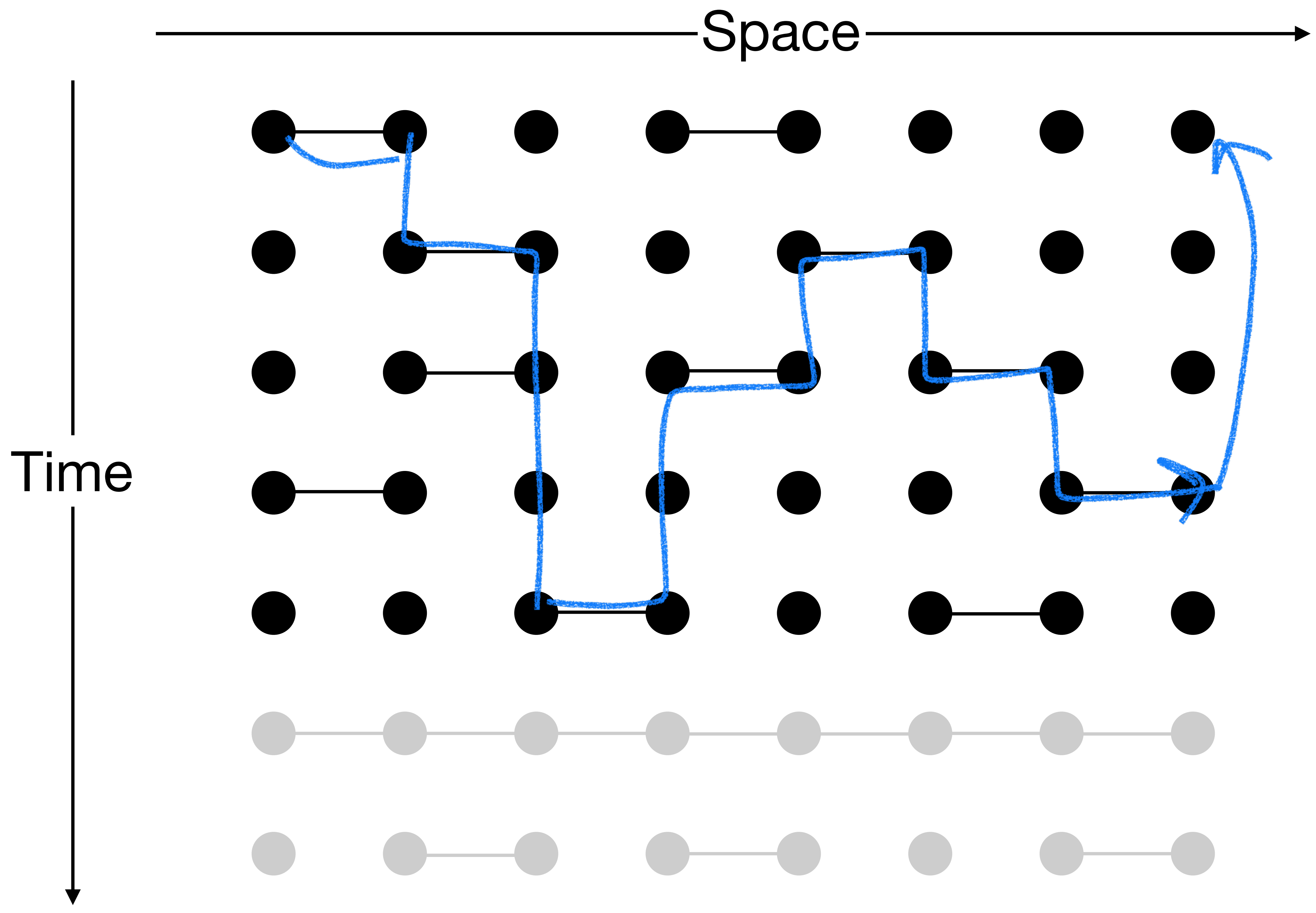
Time-Online Algorithm



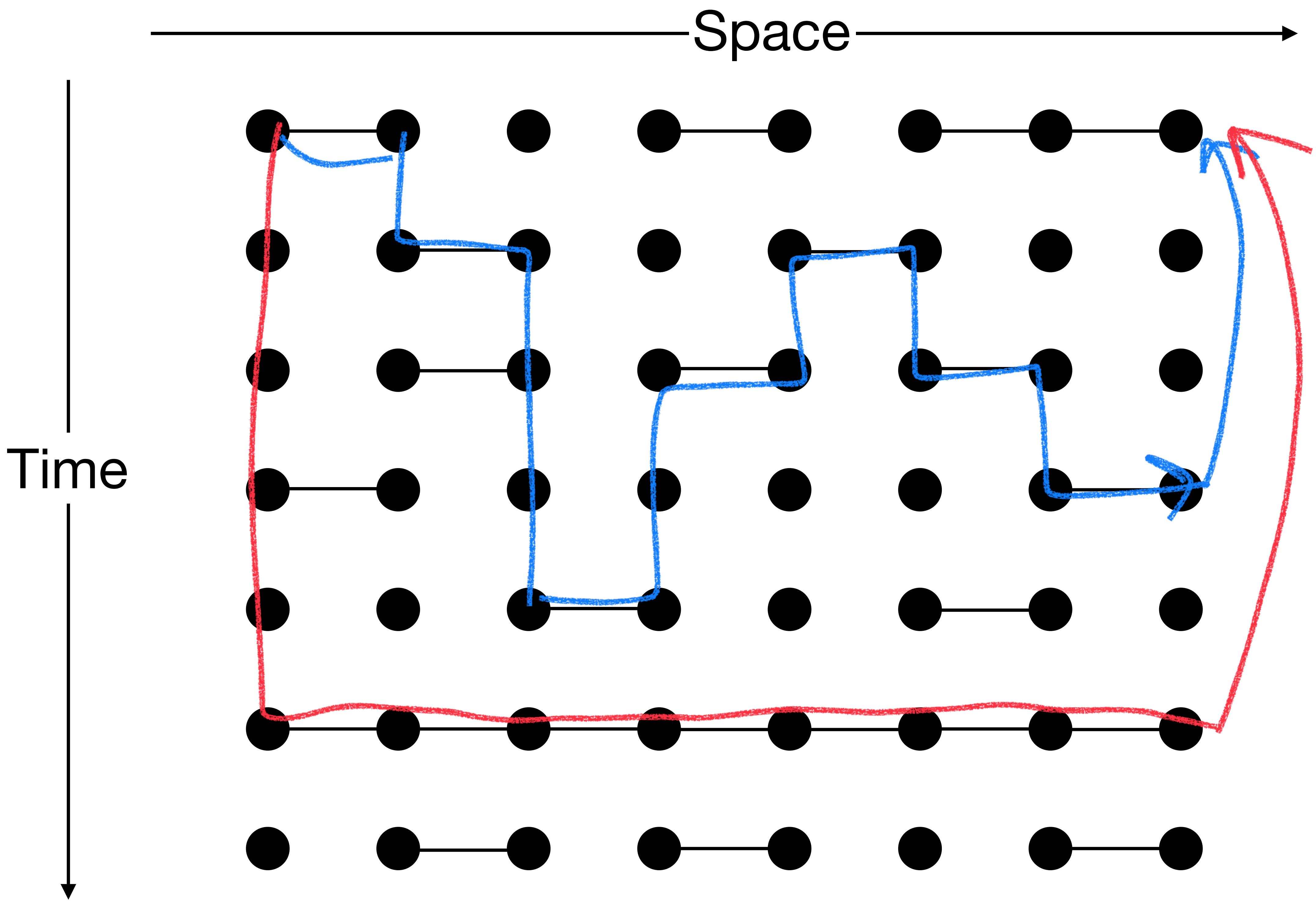
Time-Online Algorithm



Time-Online Algorithm



Competitive Ratio



Competitive Ratio

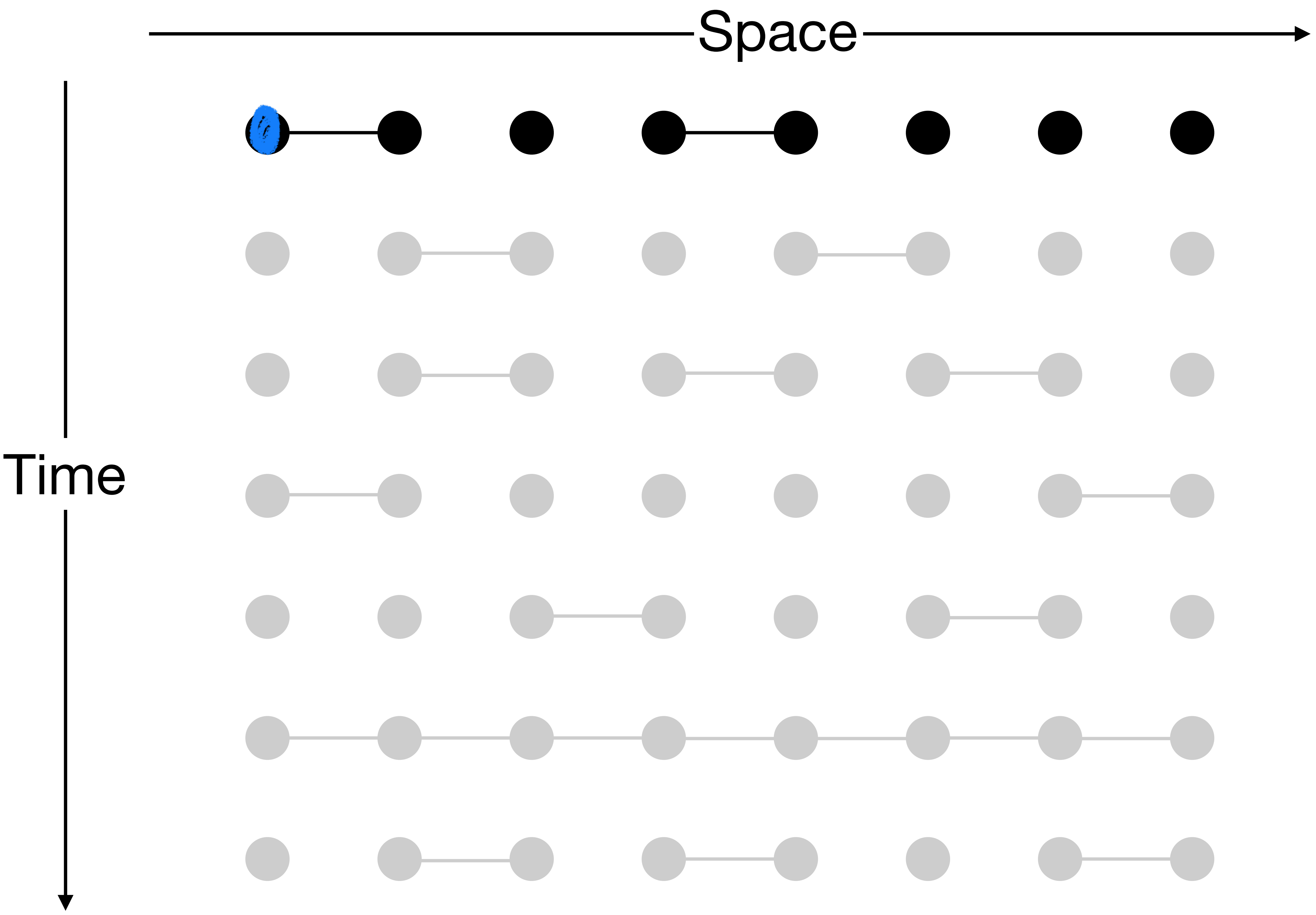
A is か-competitive if

Competitive Ratio

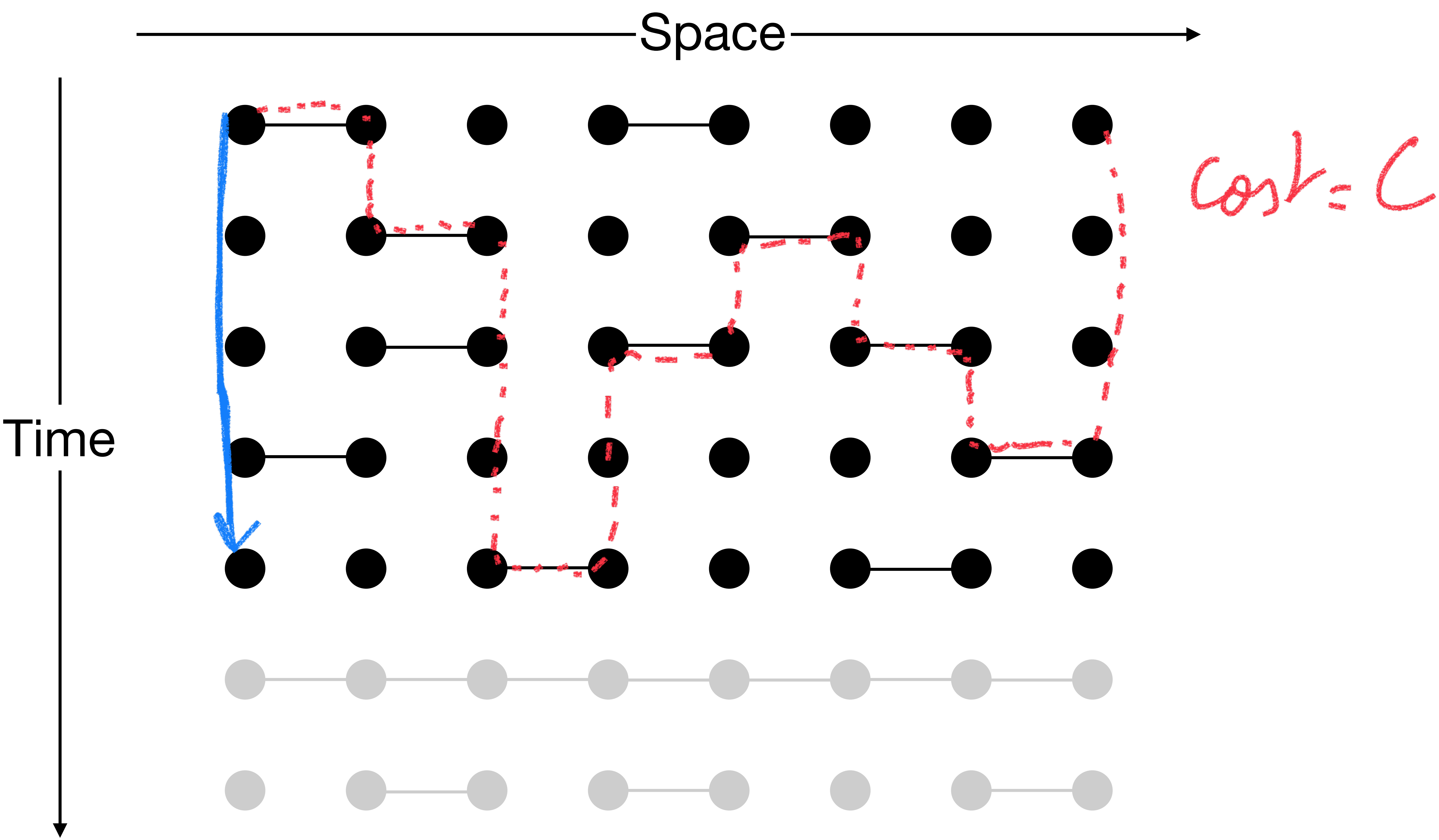
A is か-competitive if

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

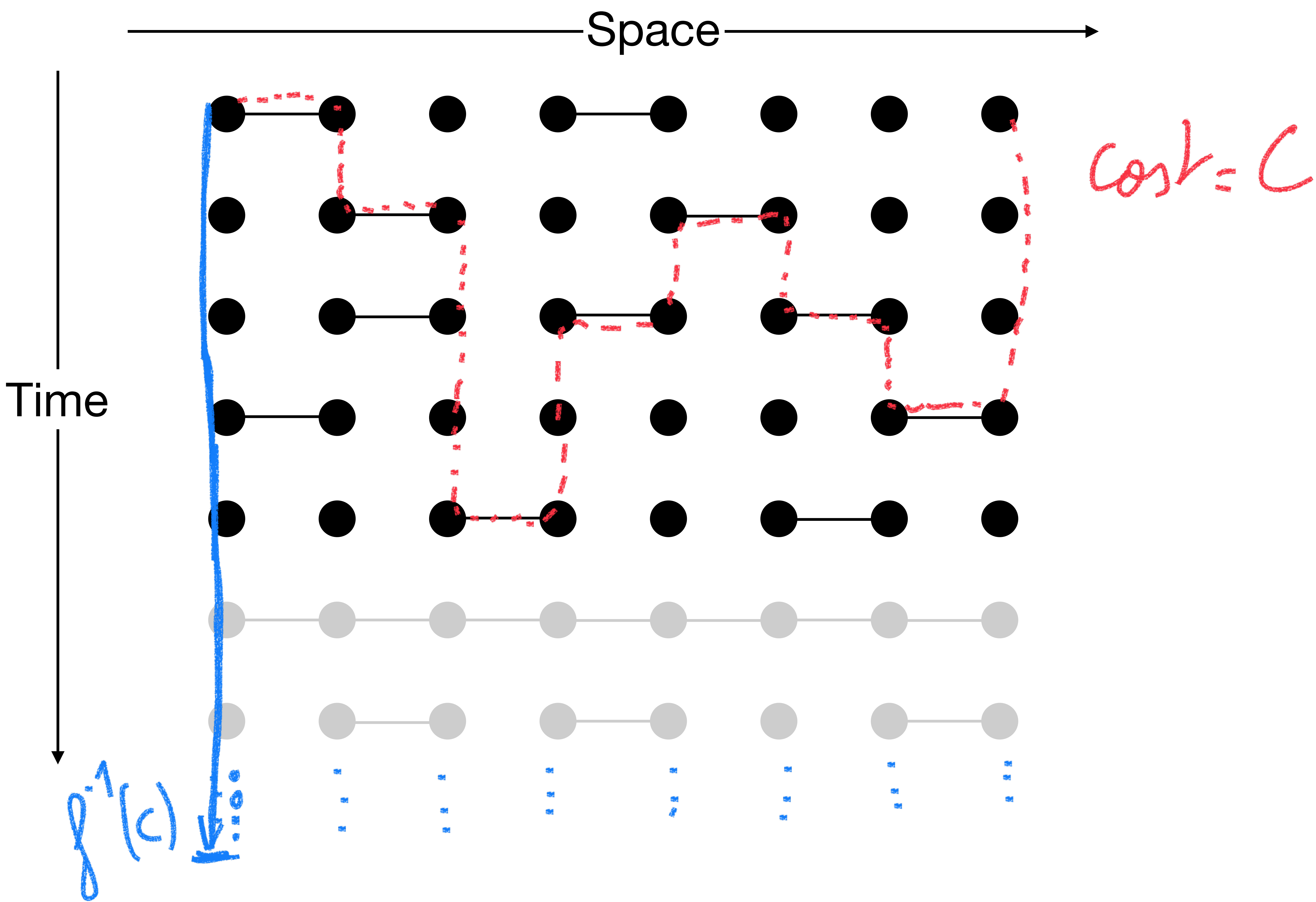
Time-Online Algorithm



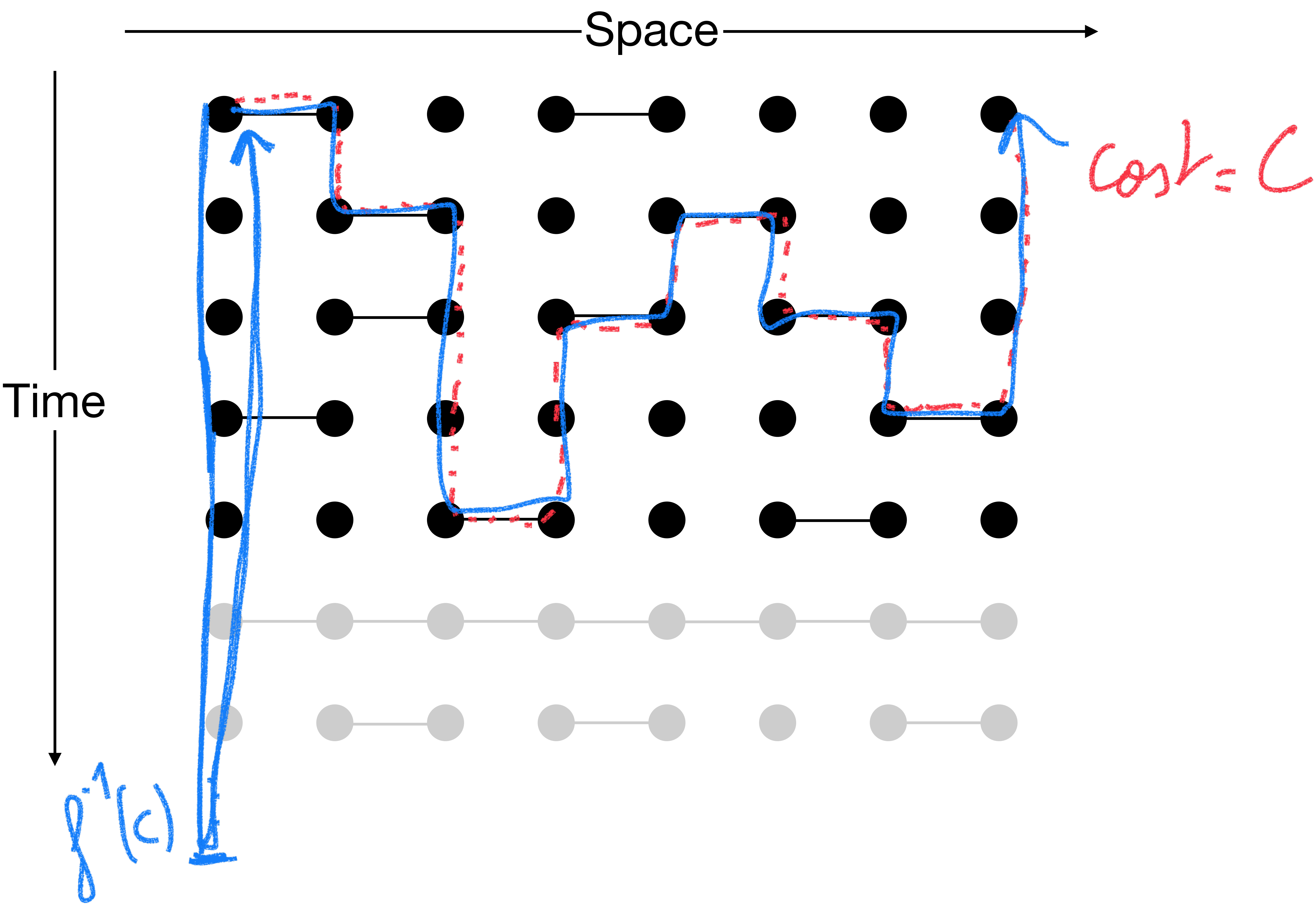
Time-Online Algorithm



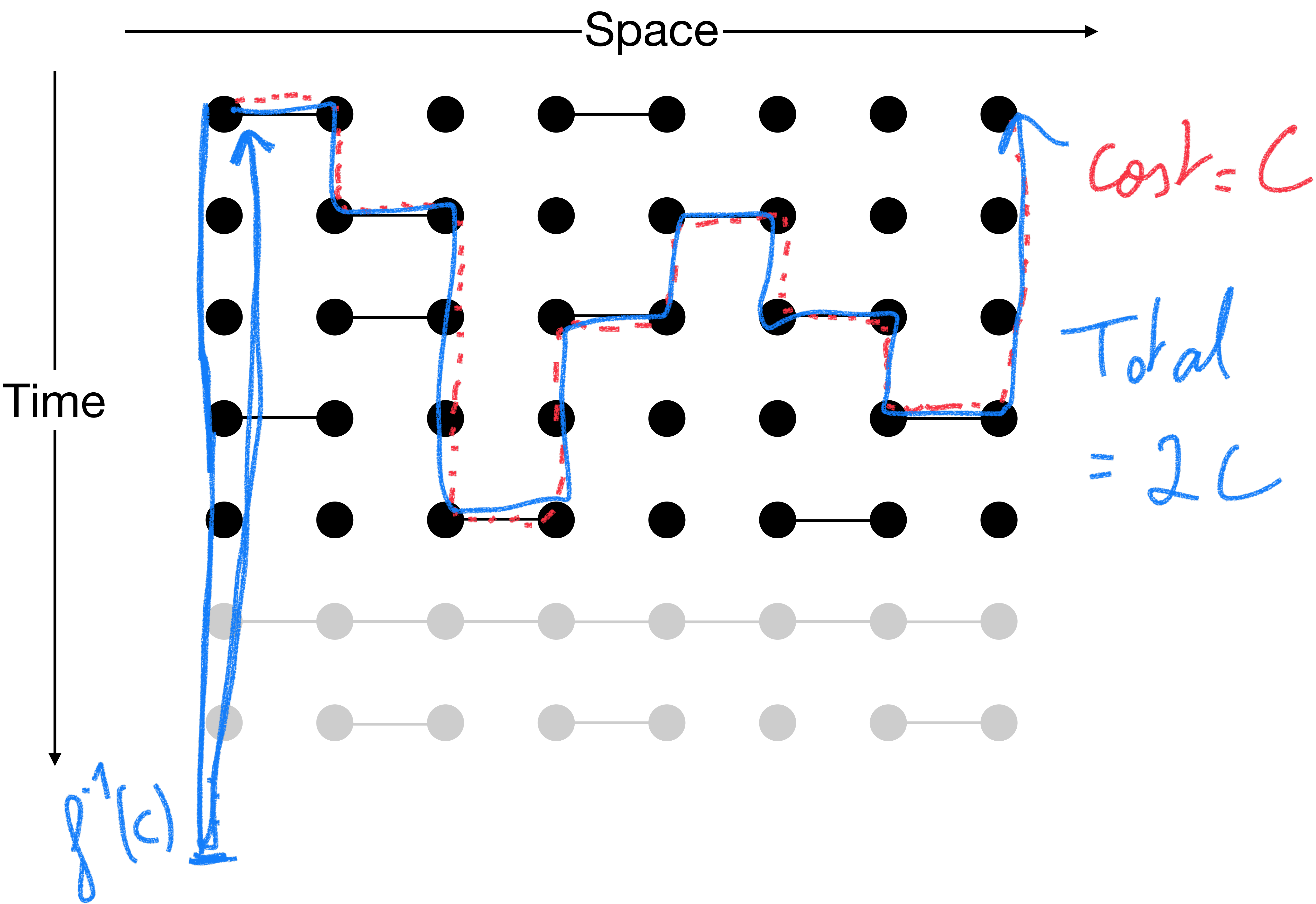
Time-Online Algorithm



Time-Online Algorithm



Time-Online Algorithm



Time-Online Algorithm

The following algorithm is 2-competitive

for any f with $\lim_{x \to \infty} f(x) = \infty$

Algorithm 1 T-Online Algorithm

Input :

G': the known evolving graph

time: the current time

u: the current node (here u is always the starting node s)

Let $T_{\max}$ be an optimal space-time travel in G', if it exists, starting at time 0, from node s to node d.

if $T_{\max}$ *does not exist* ***or*** $\mathfrak{f}(time + 1) < cost(T_{\max})$ **then**

wait 1 time instant;

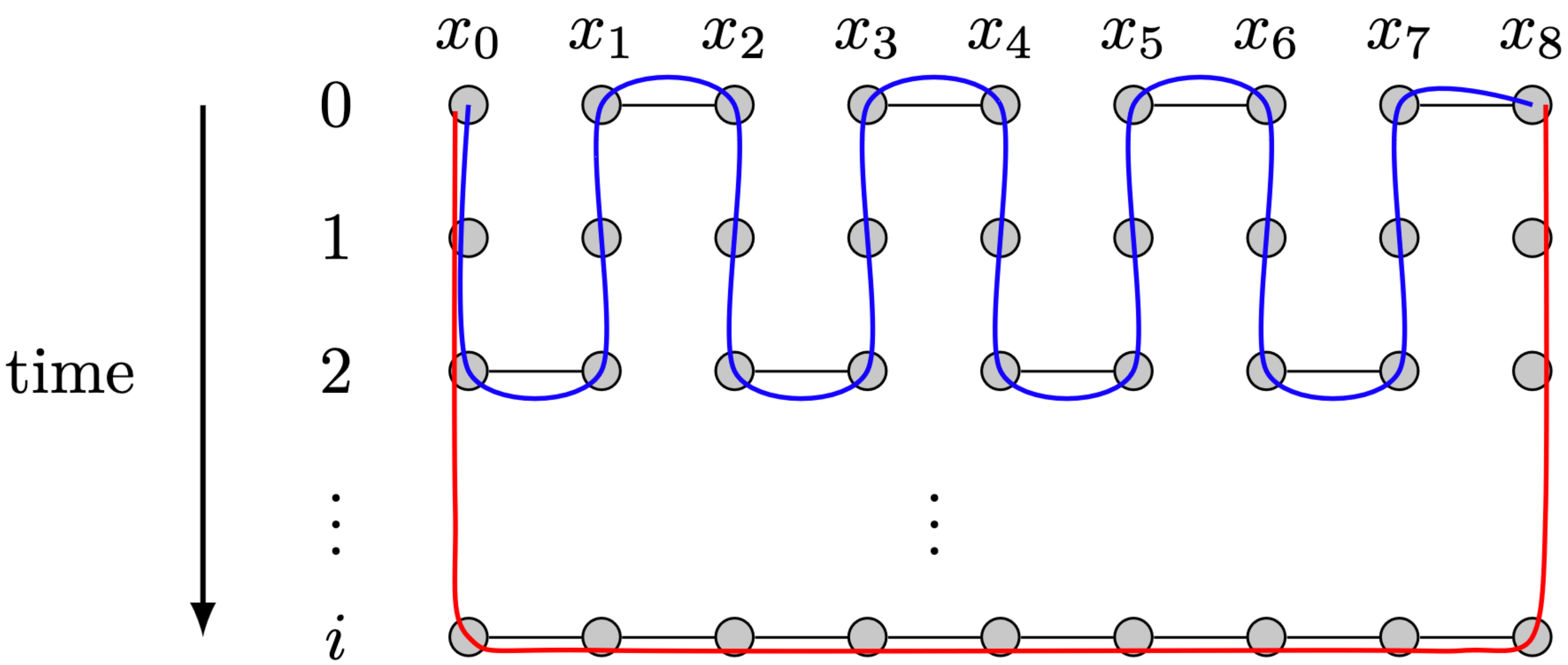
else

go back at time 0, then follow the travel $T_{\max}$;

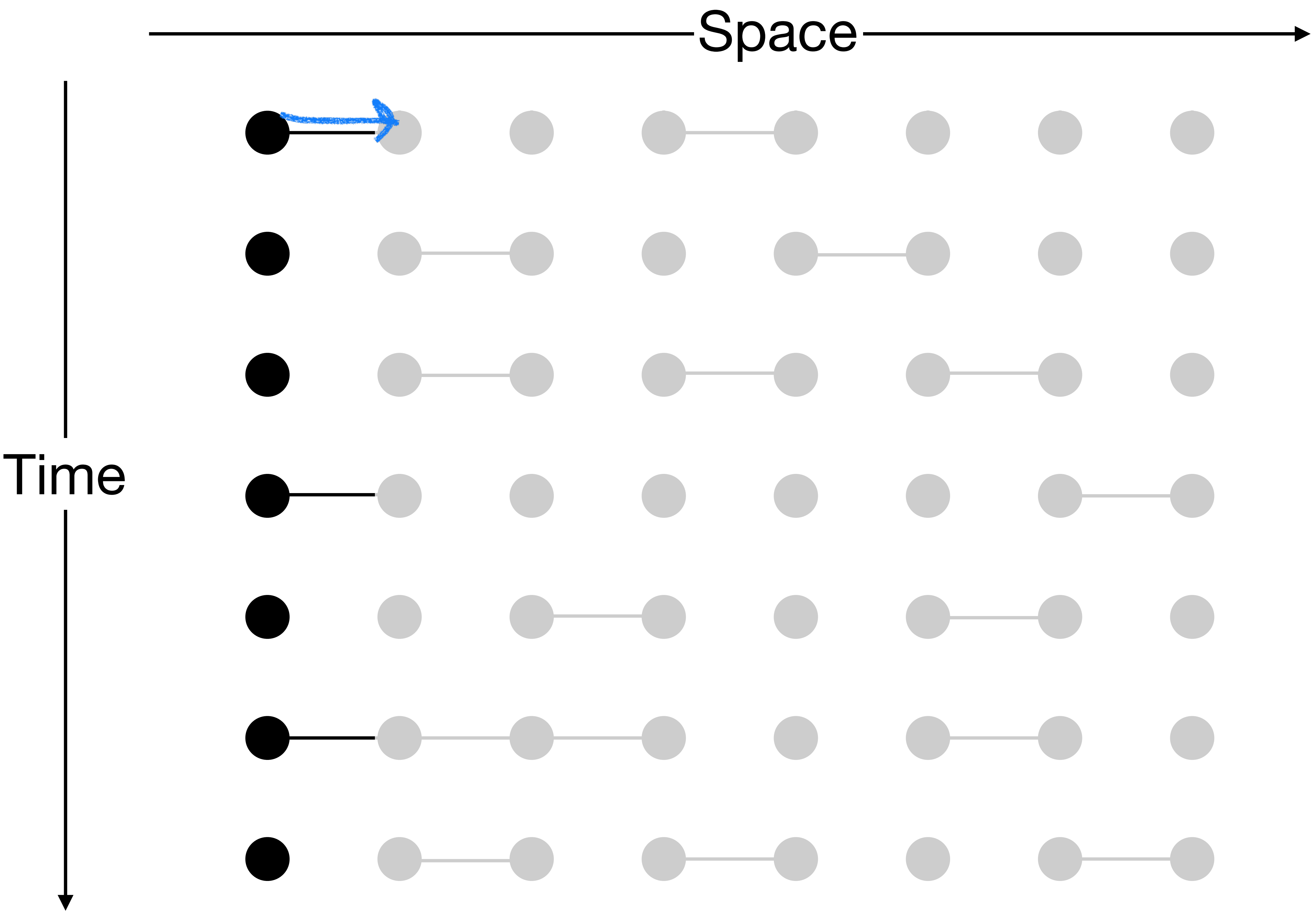
This is optimal

Time-Online Algorithm

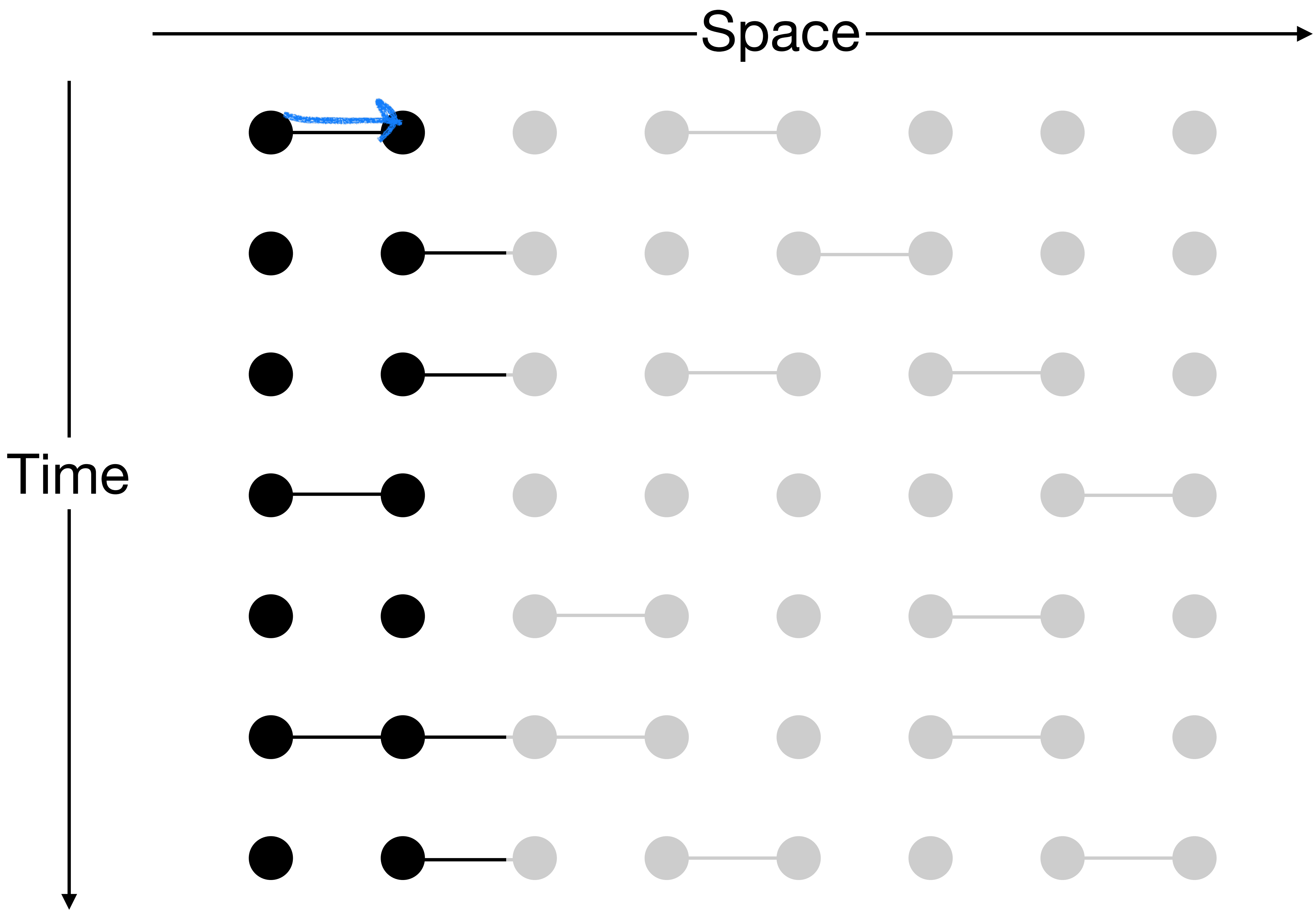
Let G_i be



Space-Online Algorithm

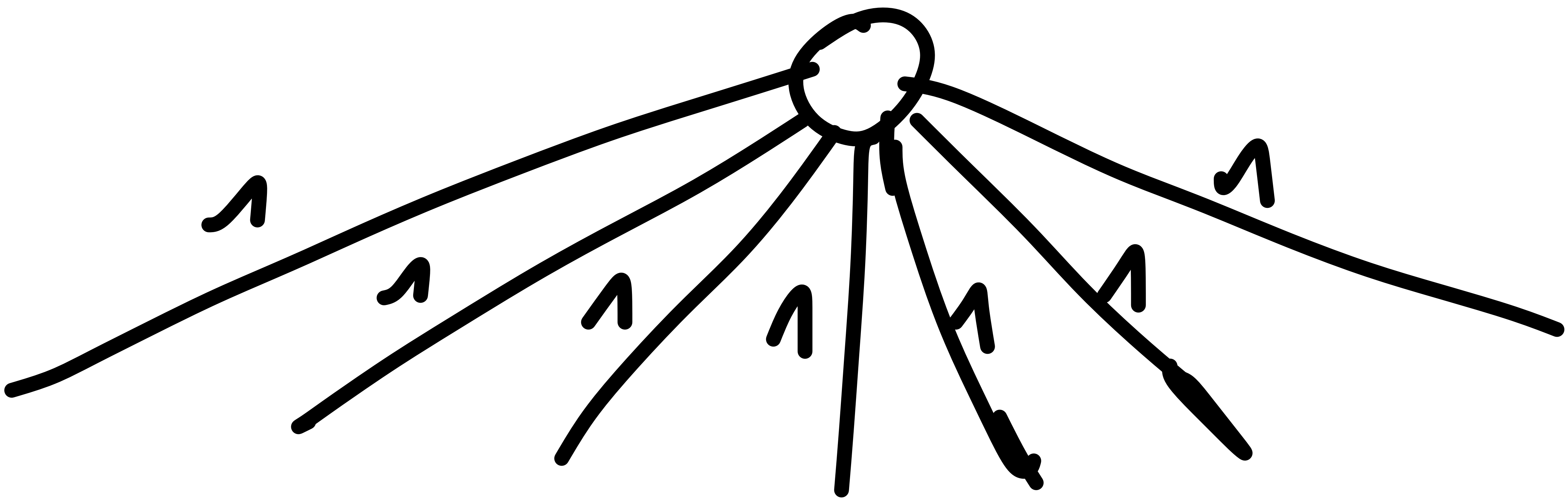


Space-Online Algorithm

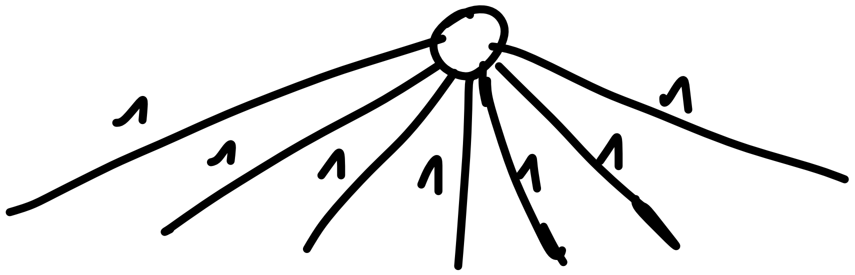


Space-Online Algorithm Lower Bound

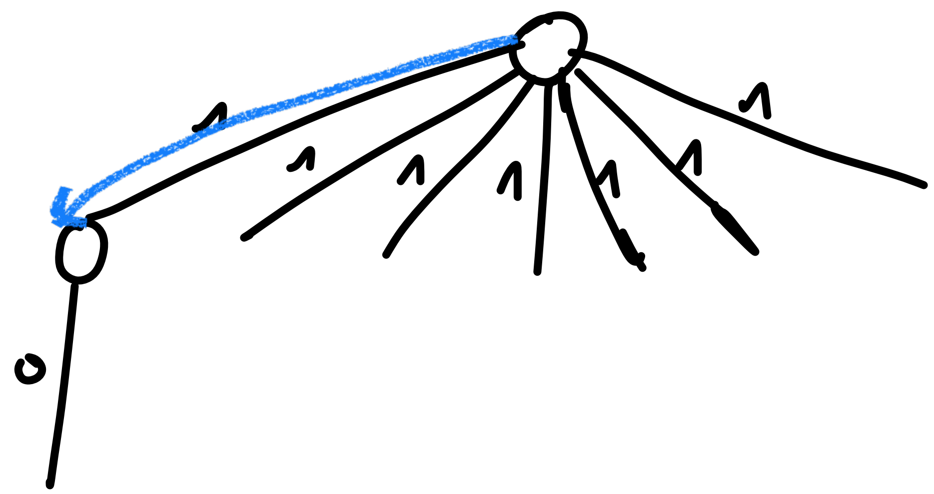
Space-Online Algorithm Lower Bound



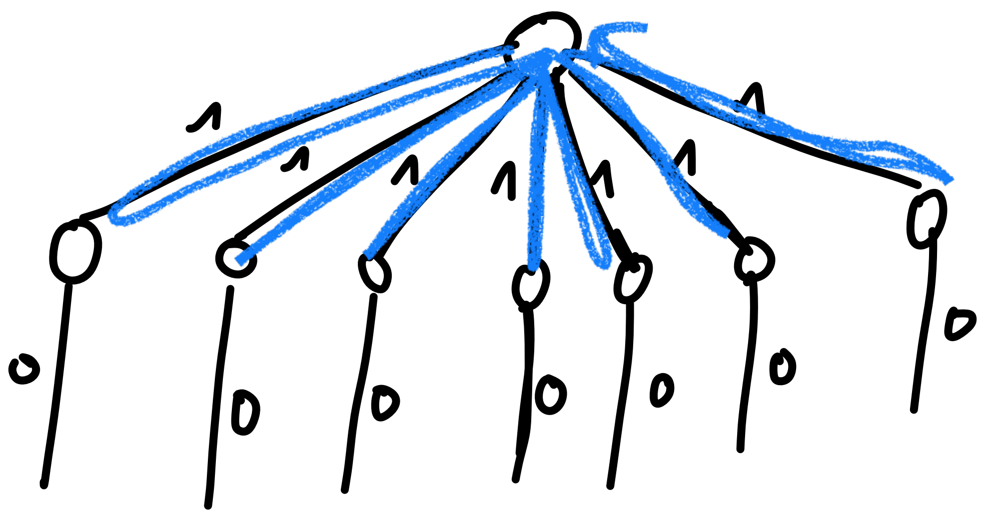
Space-Online Algorithm Lower Bound



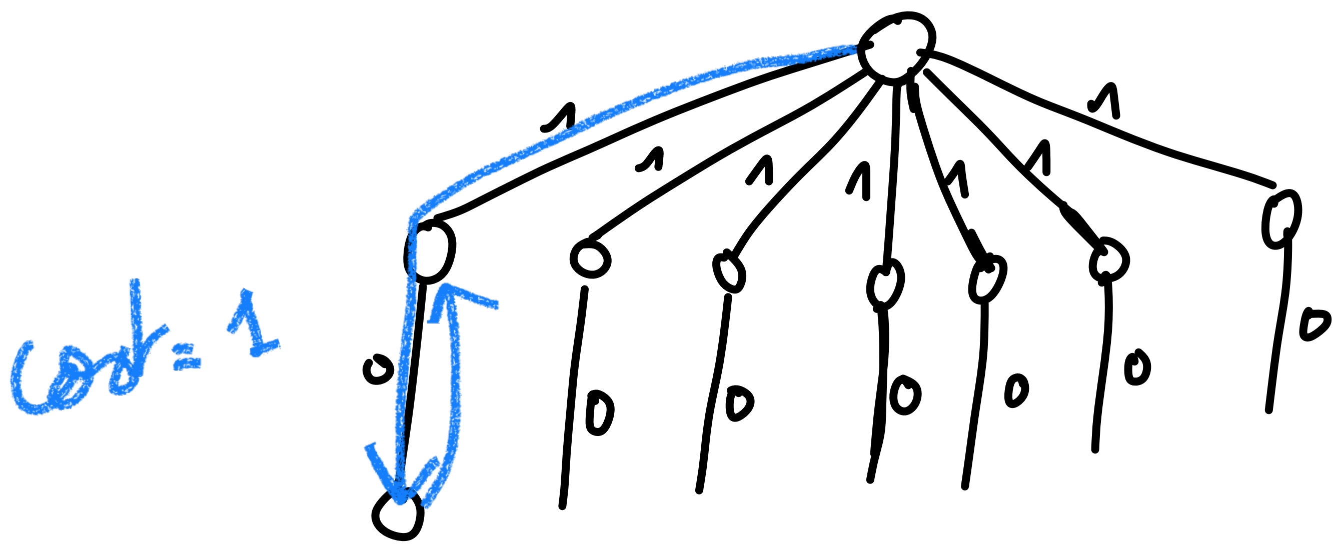
Space-Online Algorithm Lower Bound



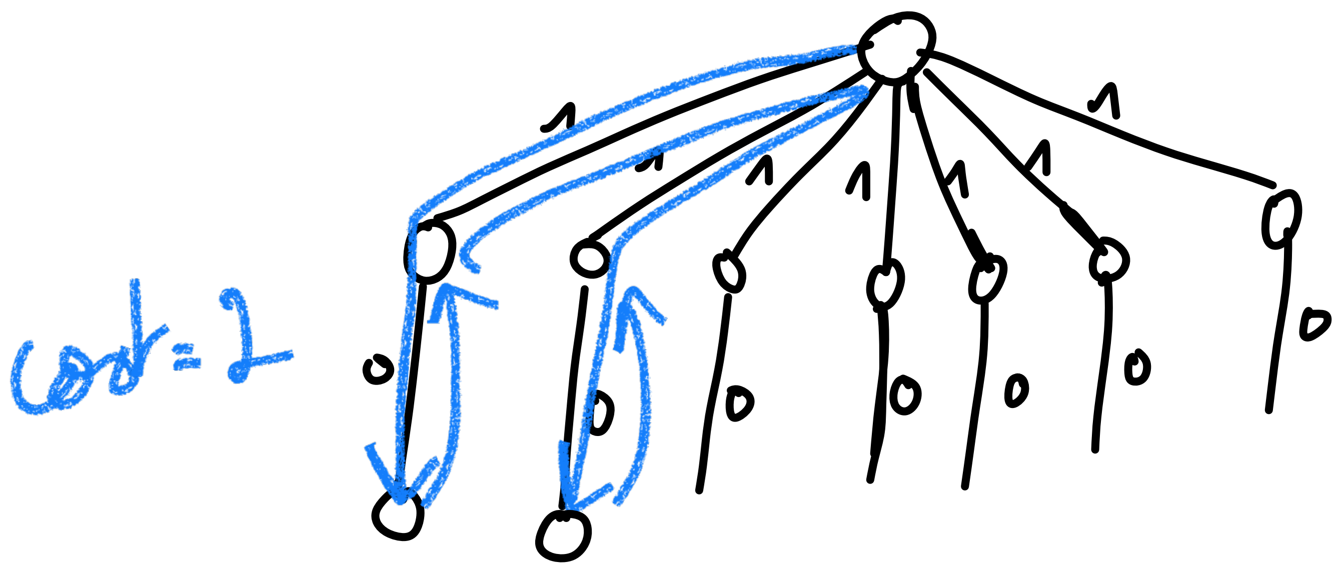
Space-Online Algorithm Lower Bound



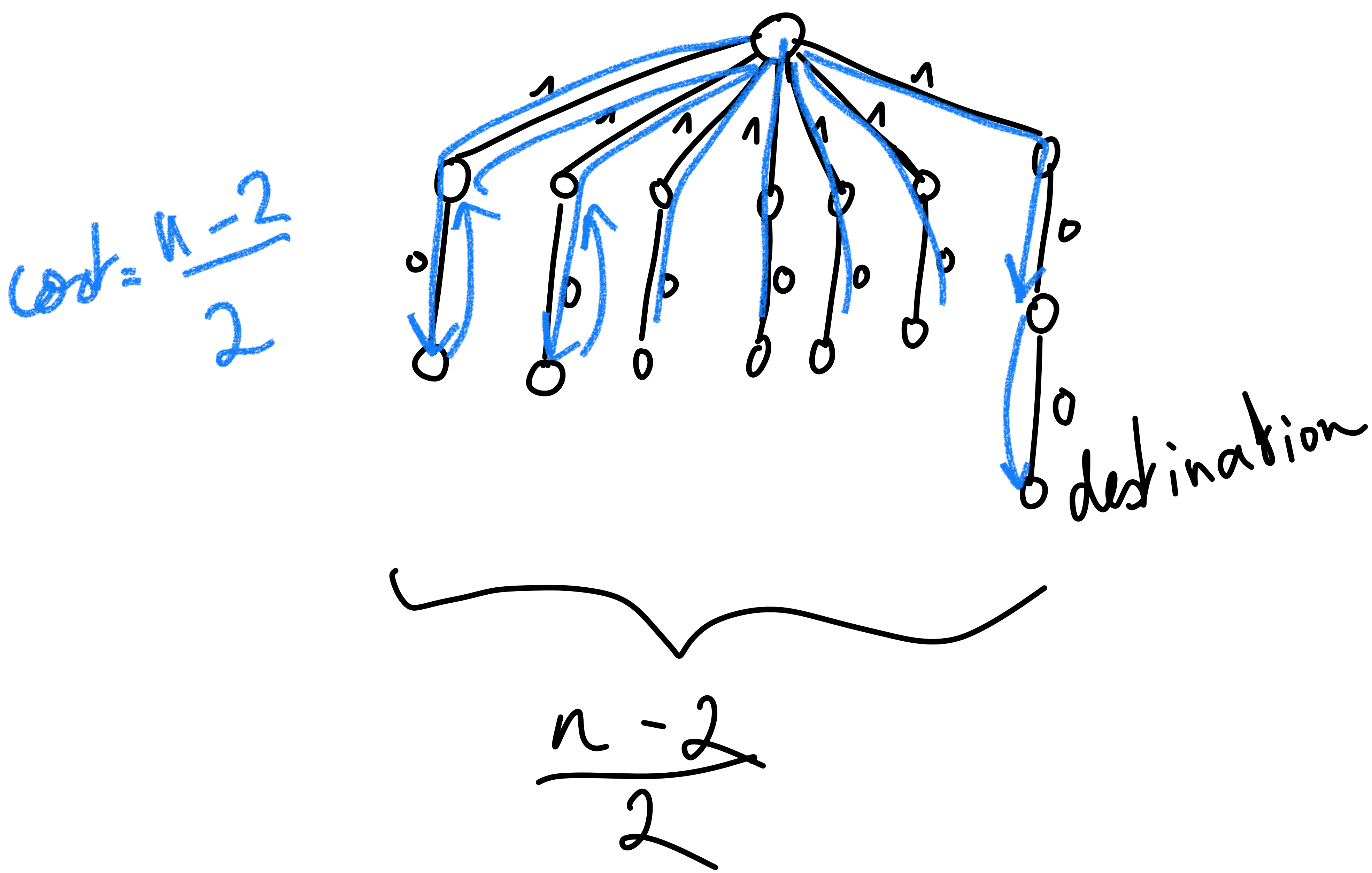
Space-Online Algorithm Lower Bound



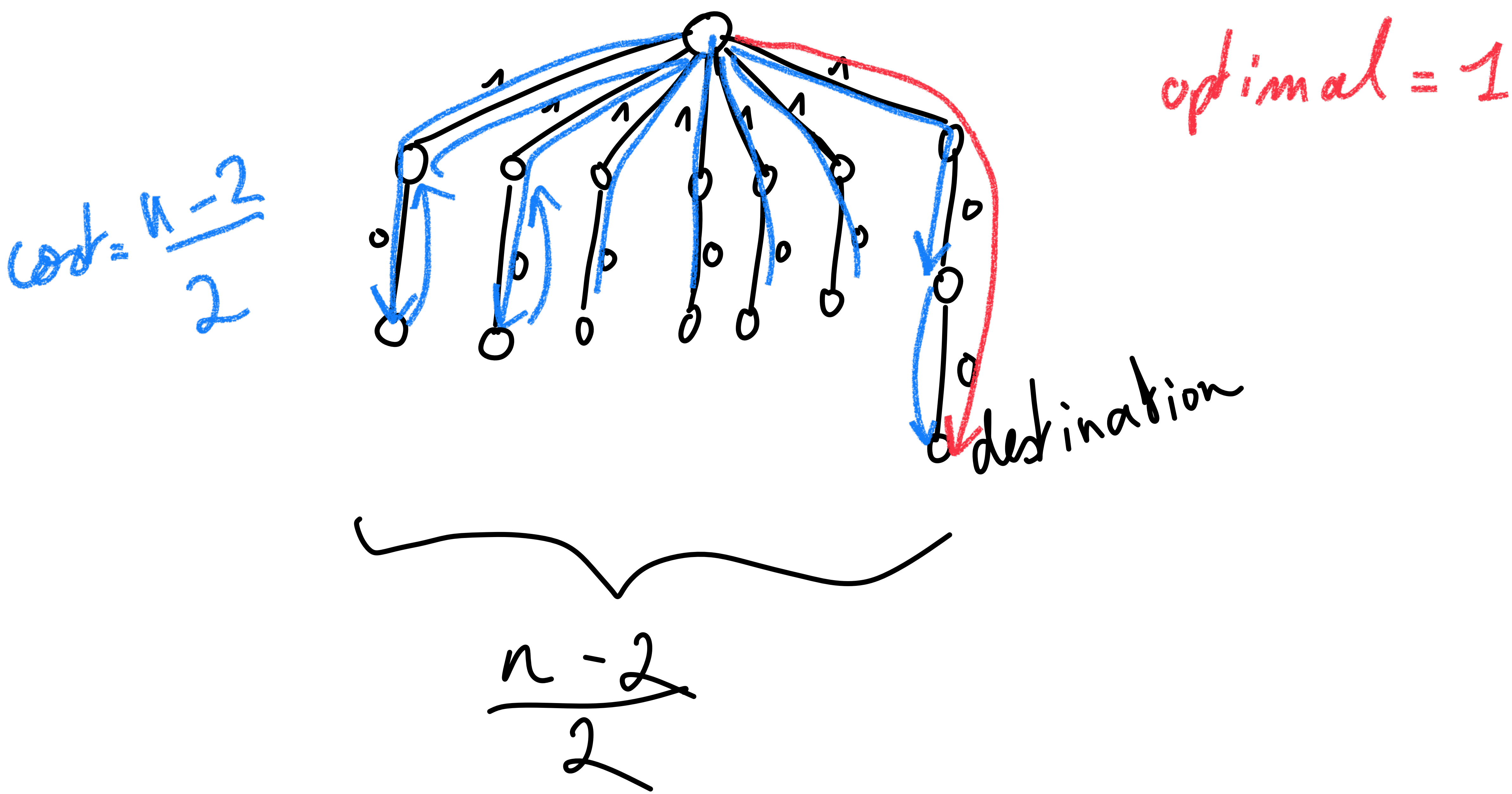
Space-Online Algorithm Lower Bound



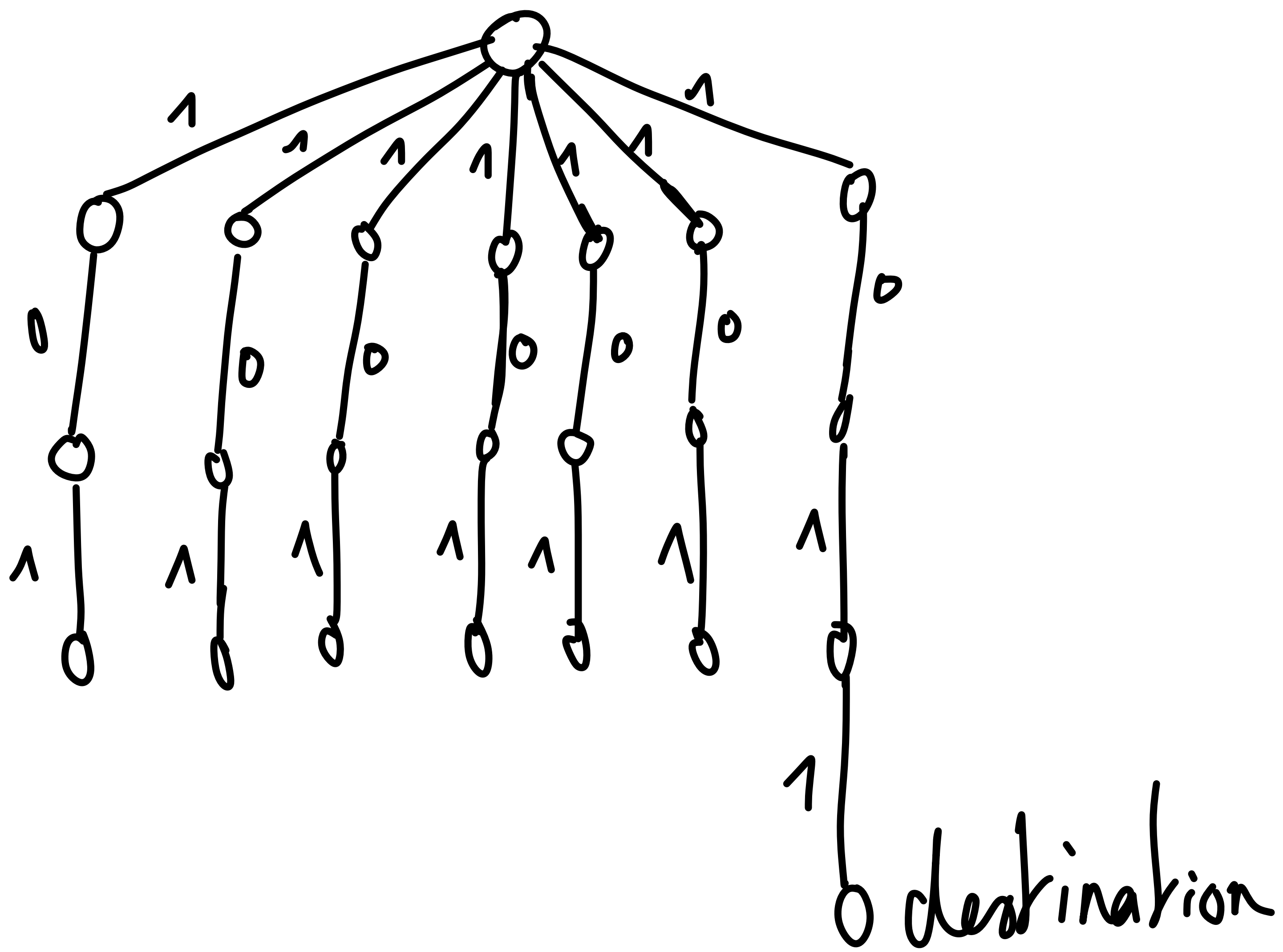
Space-Online Algorithm Lower Bound



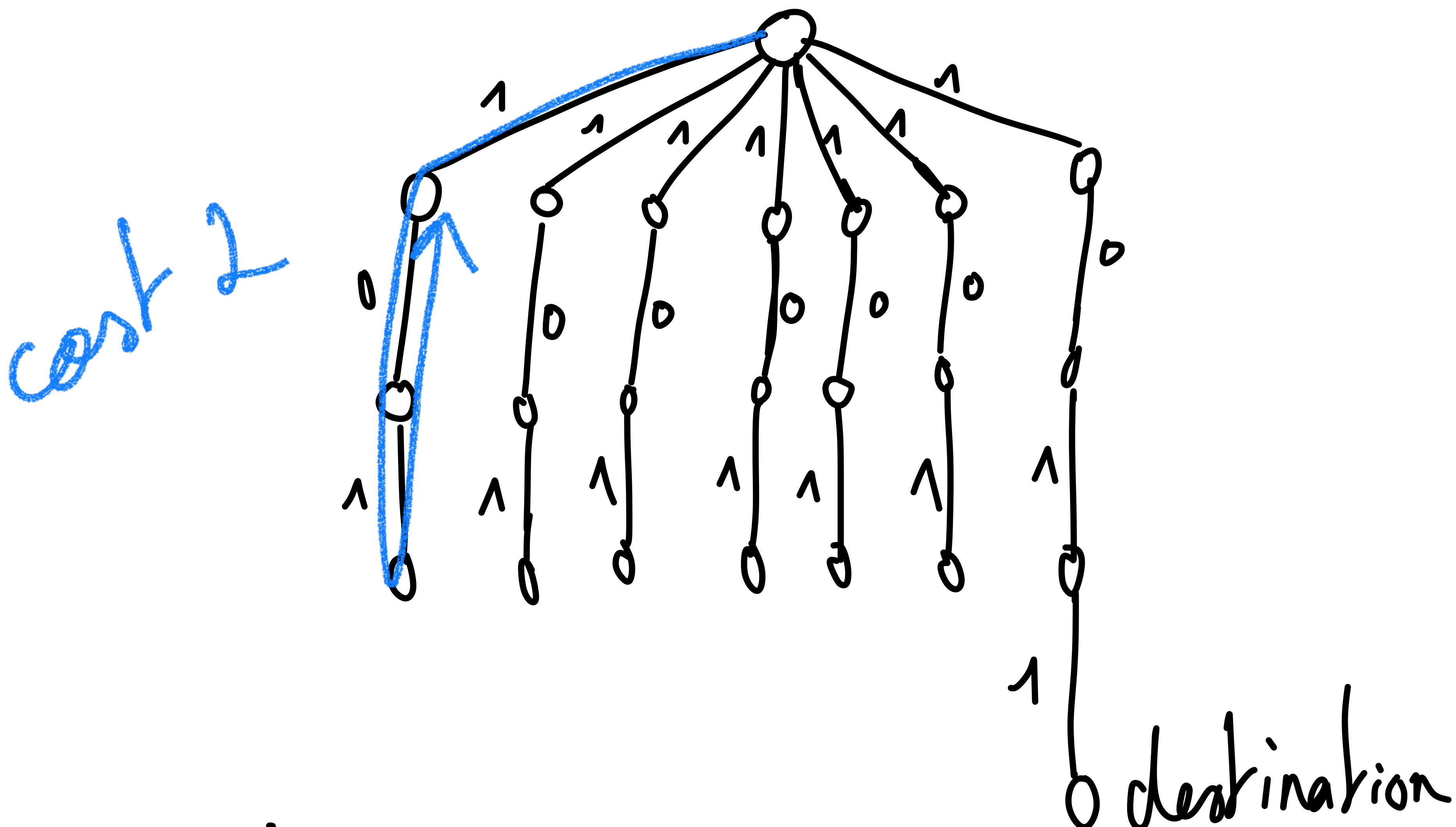
Space-Online Algorithm Lower Bound



Space-Online Algorithm Lower Bound

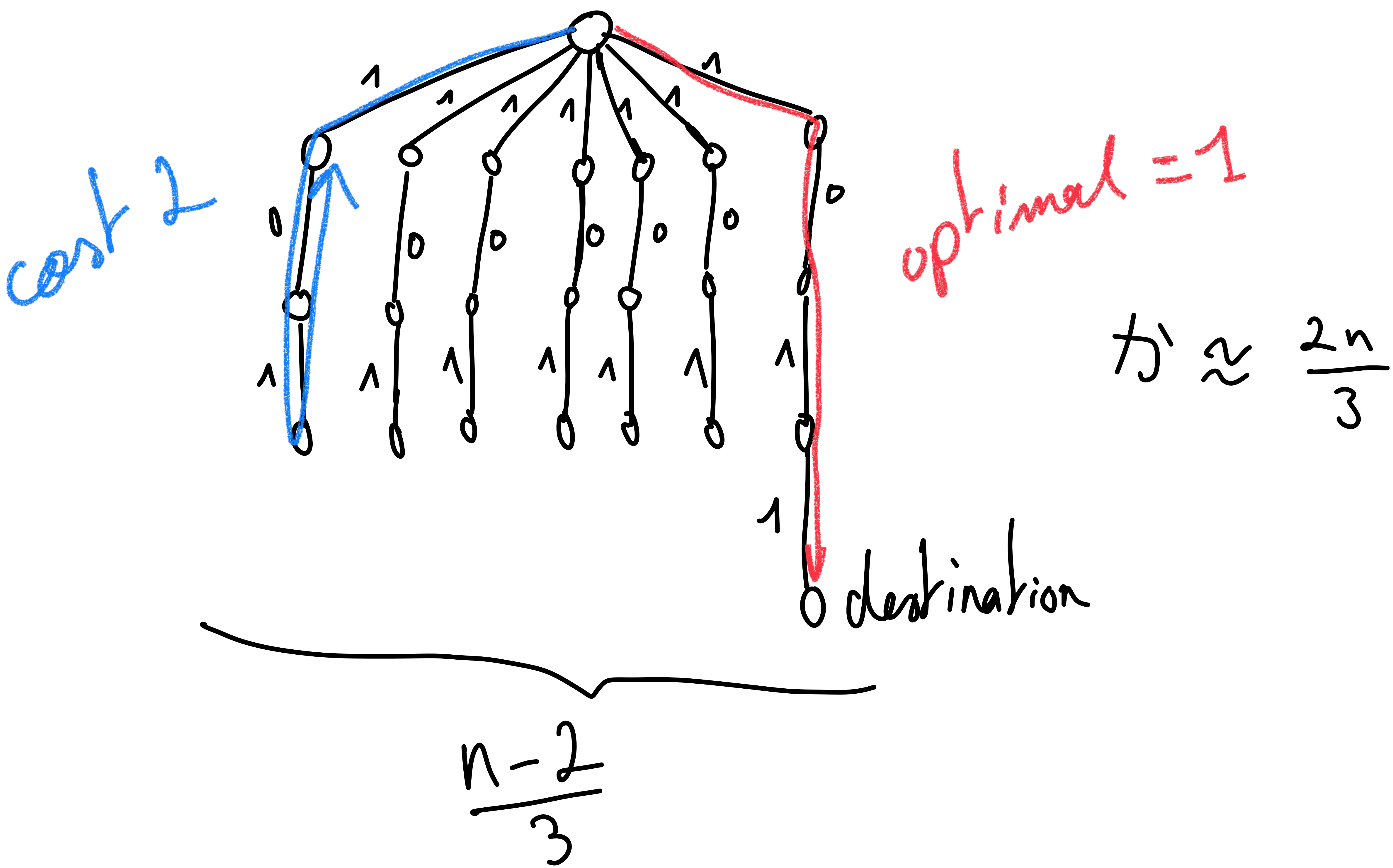


Space-Online Algorithm Lower Bound



$\frac{n-2}{3}$

Space-Online Algorithm Lower Bound



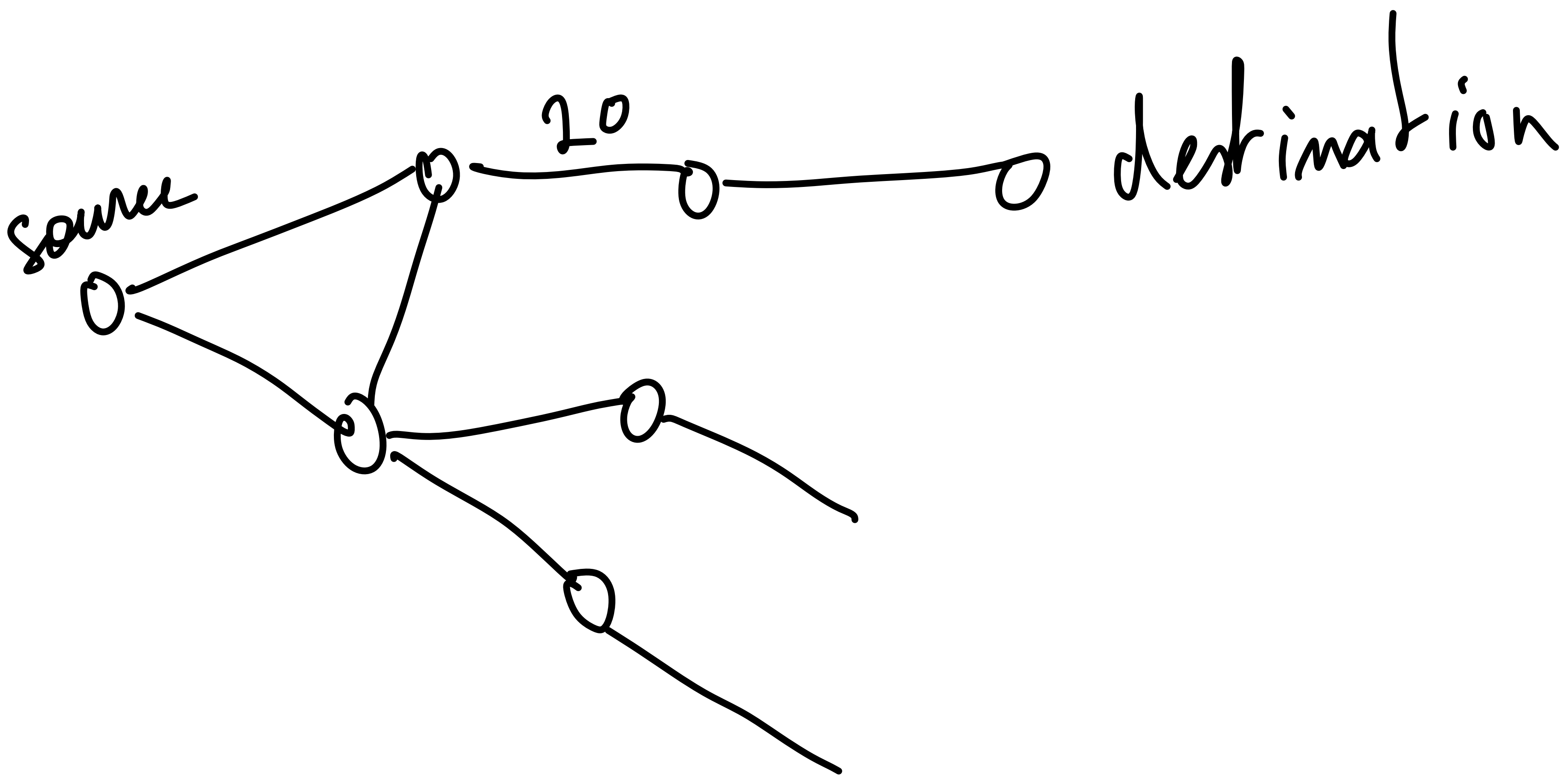
Space-Online Algorithm Lower Bound

No!

$\frac{2n}{3}$ is our best current LB

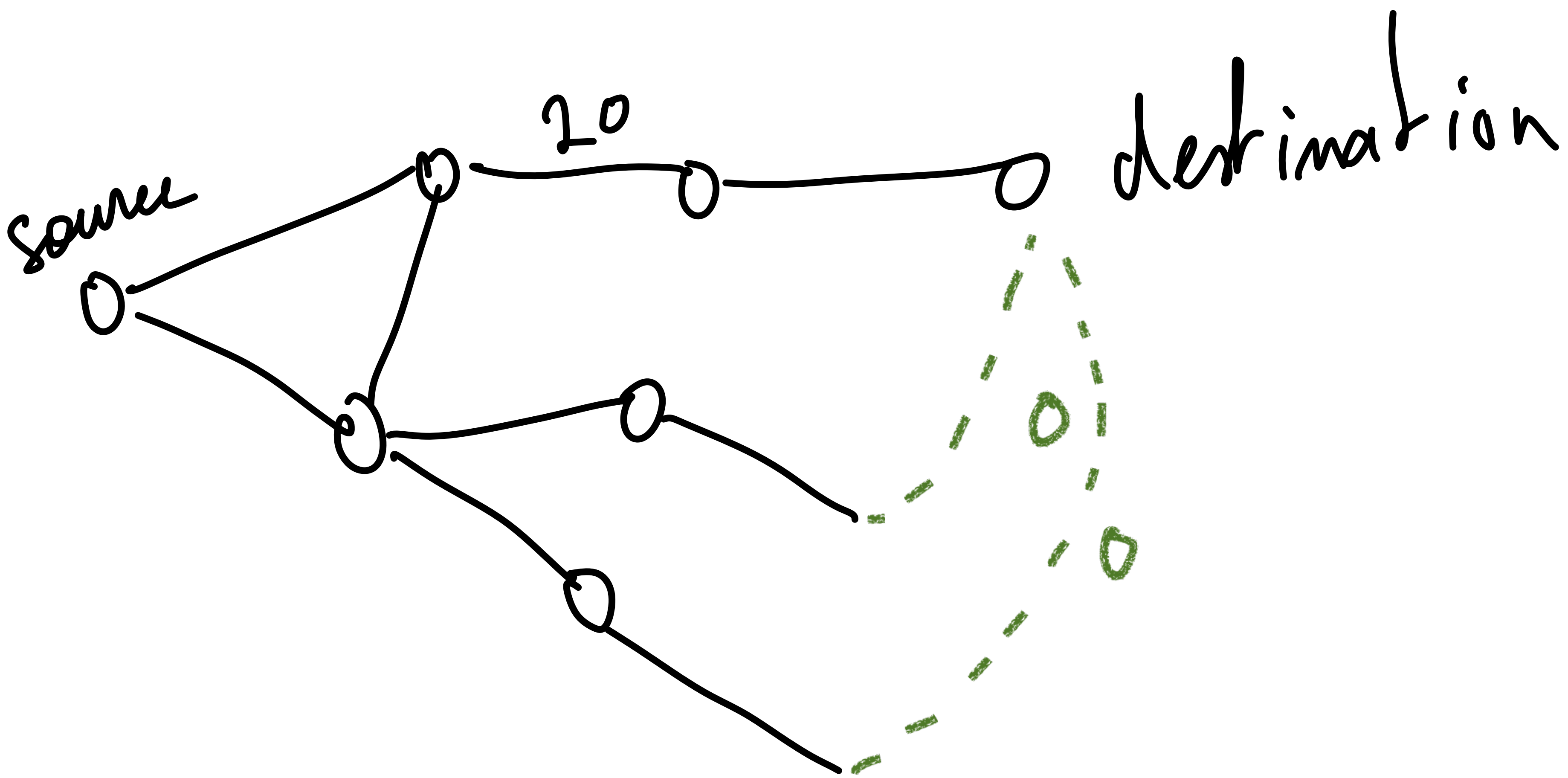
Our Space-Online Algorithm

Algo S-online



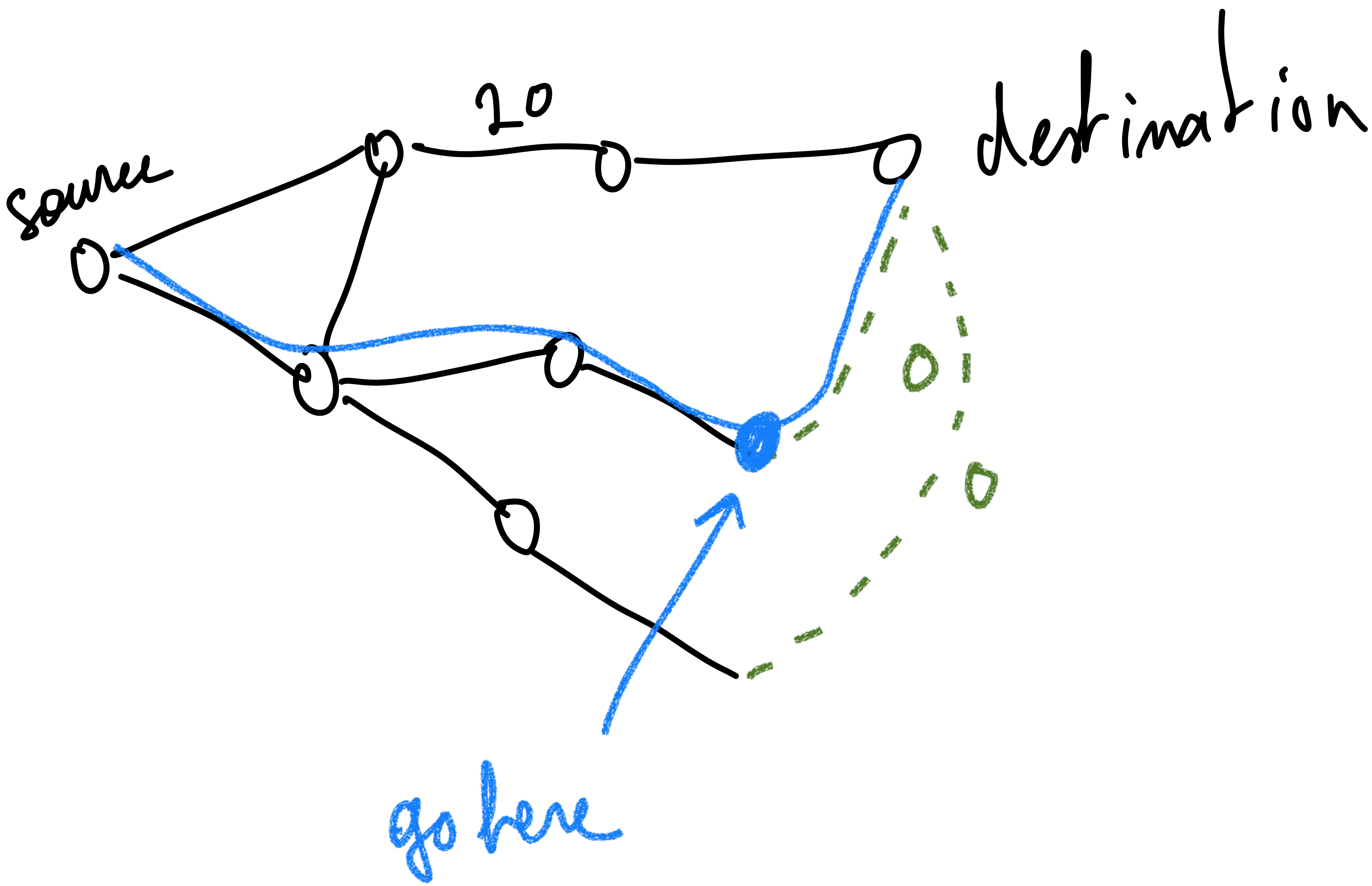
Our Space-Online Algorithm

Algo S-online



Our Space-Online Algorithm

Algo S-online



Our Space-Online Algorithm

Algo S-online

Our Space-Online Algorithm

Algo S-online

Each step cost $\leq$ optimal

Our Space-Online Algorithm

Algo S-online

* Each step cost $\leq$ optimal
* At most n-1 steps

Our Space-Online Algorithm

Algo S-online

* Each step cost $\leq$ optimal
* At most $n-1$ steps

力 $\leq n-1$

Conclusion

- Optimal Time-Online algorithm
- Upper and lower bound for the Space-Online problem

- The punchline comes before the question
- What’s the problem with Time Travel jokes ?

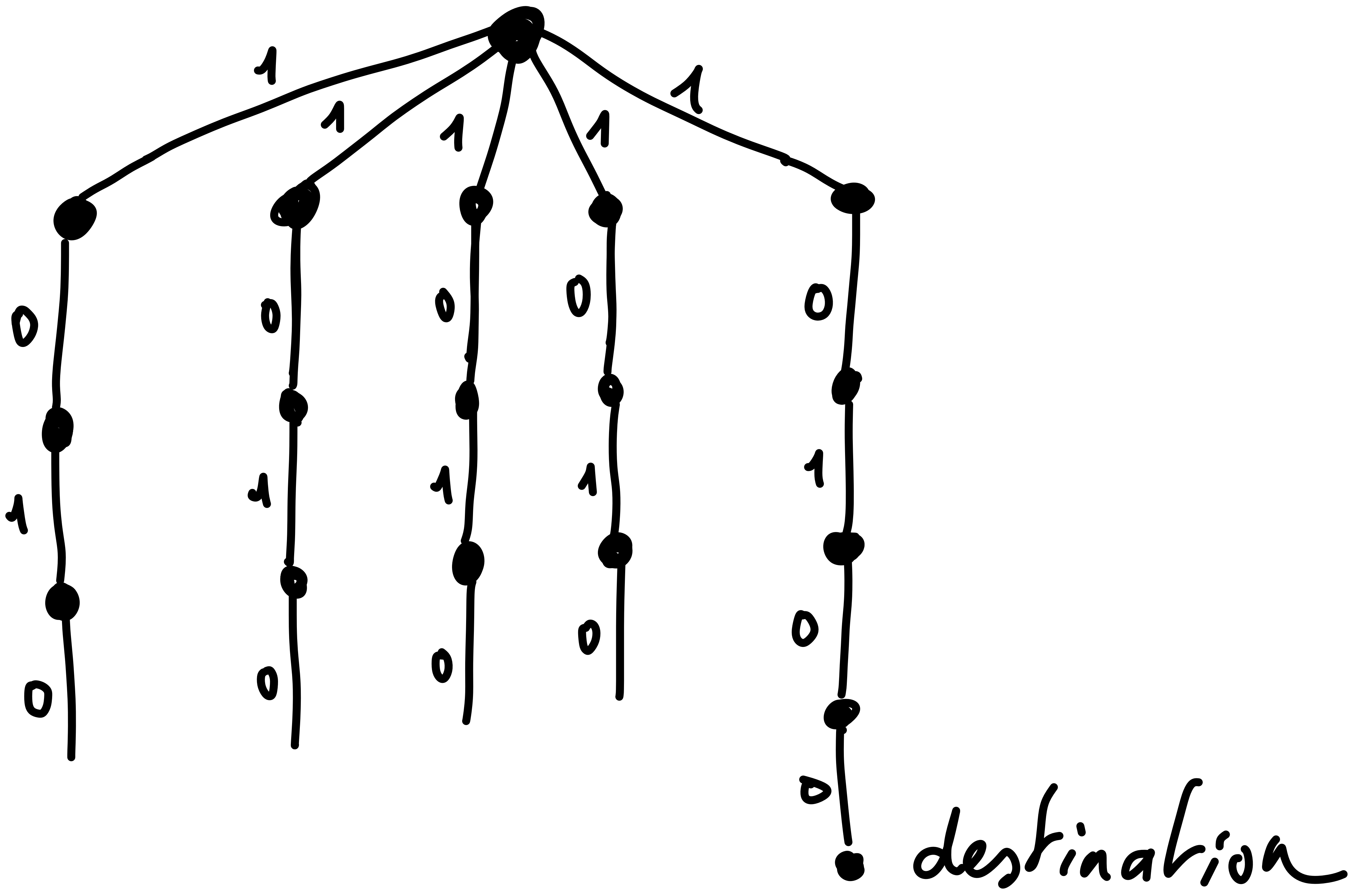
Conclusion

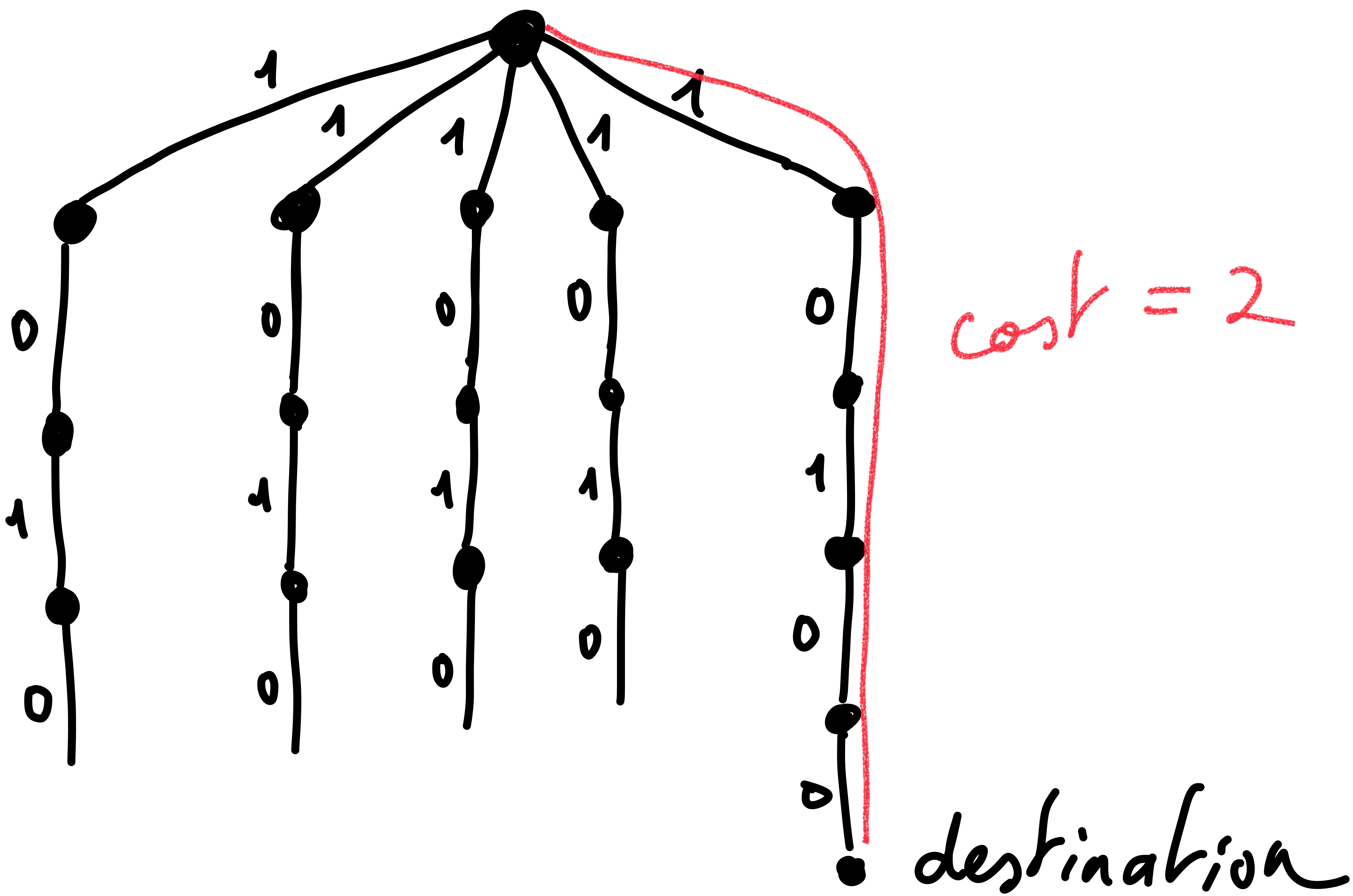
- Optimal Time-Online algorithm
- Upper and lower bound for the Space-Online problem

- The punchline comes before the question
- What’s the problem with Time Travel jokes ?

Thank you!

NO


1
1
1
1
1
0
0
0
0
0
1
1
1
1
1
0
0
0
0
0
0
destination


1
1
1
1
1
0
0
0
0
0
1
1
1
1
1
0
0
0
0
0
0
cost = 2
destination