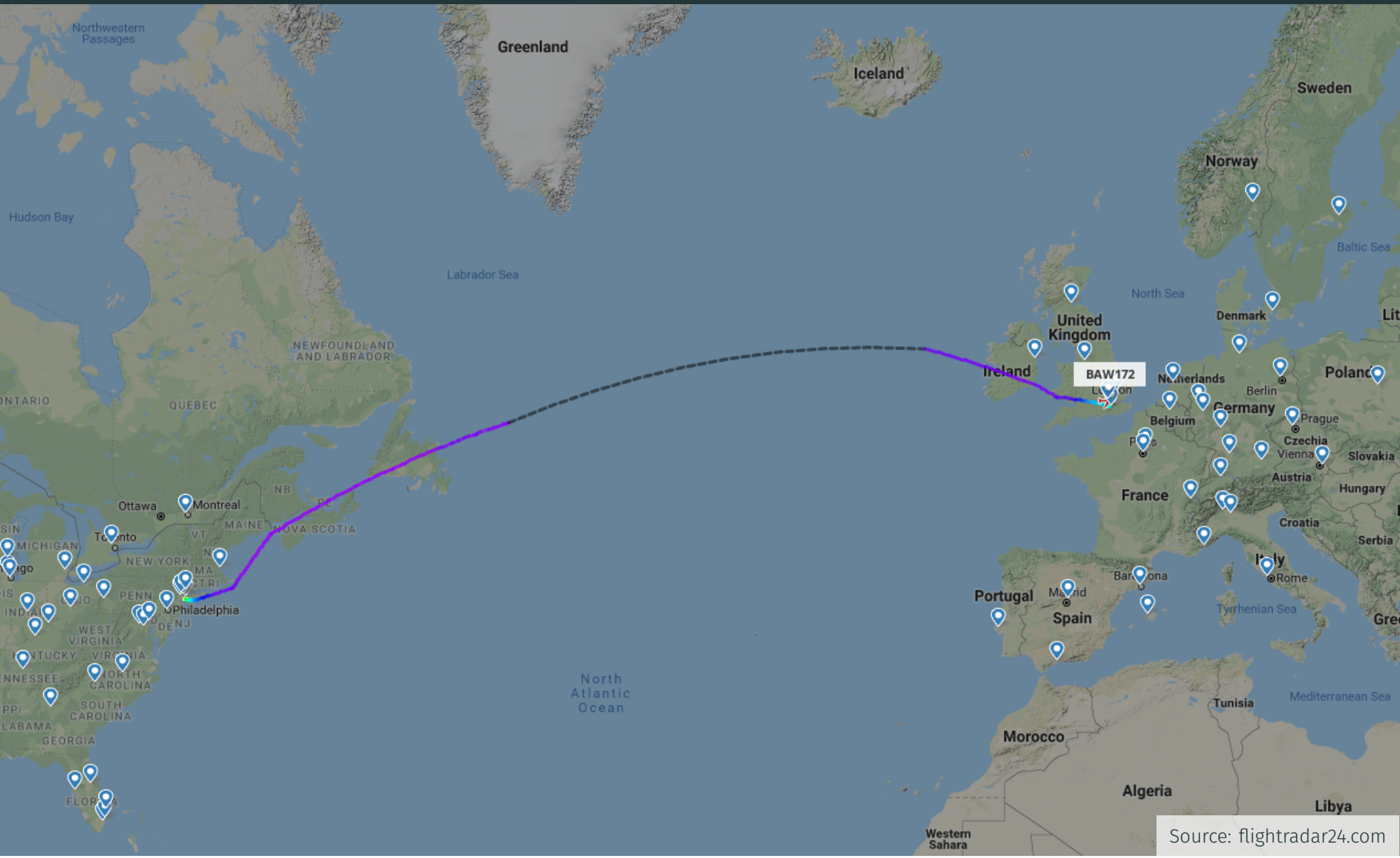
Large-Scale Search in Flight Planning

Adam Schienle

05 October 2018

Zuse Institute Berlin

The Flight Planning Problem



Source: flightradar24.com

The Flight Planning Problem

The Flight Planning Problem

Objective
Compute a feasible, minimum cost, 3D trajectory

The Flight Planning Problem

Objective
Compute a feasible, minimum cost, 3D trajectory

Given:

- Origin and Destination airports s, t

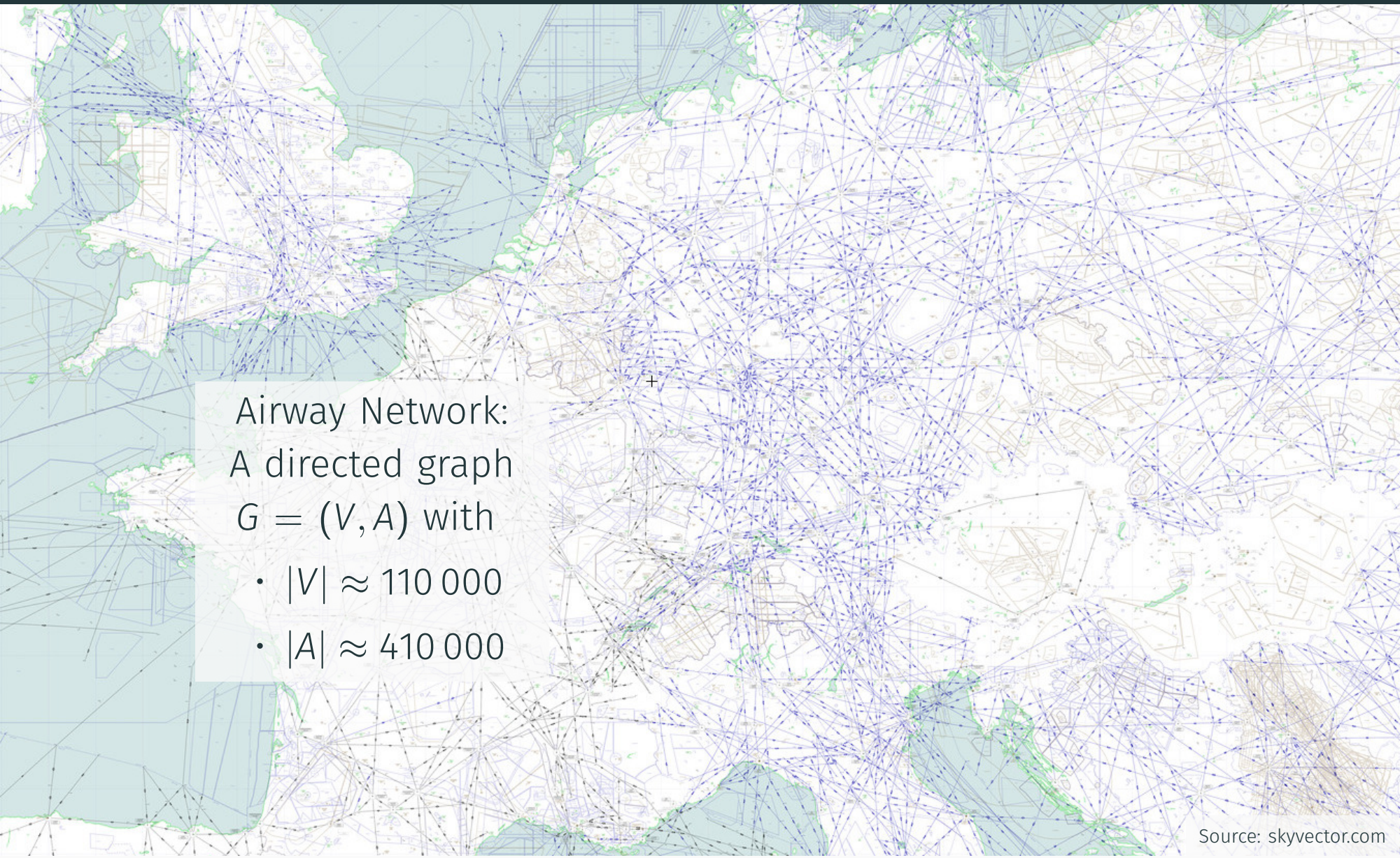
The Flight Planning Problem

Objective
Compute a feasible, minimum cost, 3D trajectory

Given:

- Origin and Destination airports s, t
- Airway Network and possible flight altitudes

The Airway Network



Airway Network:
A directed graph
$G = (V, A)$ with

- $|V| \approx 110\,000$
- $|A| \approx 410\,000$

Source: skyvector.com

The Airway Network - on Different Levels

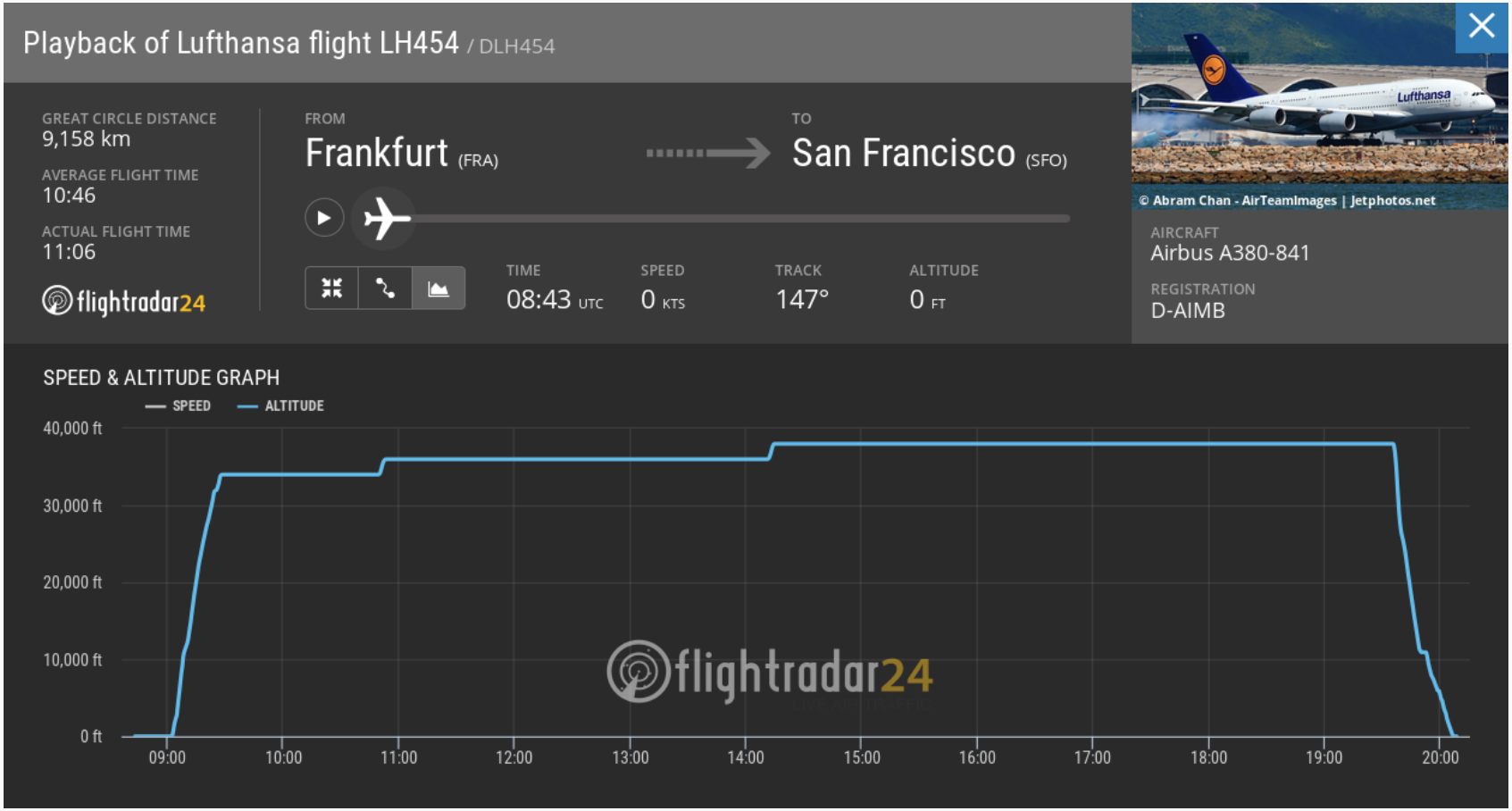
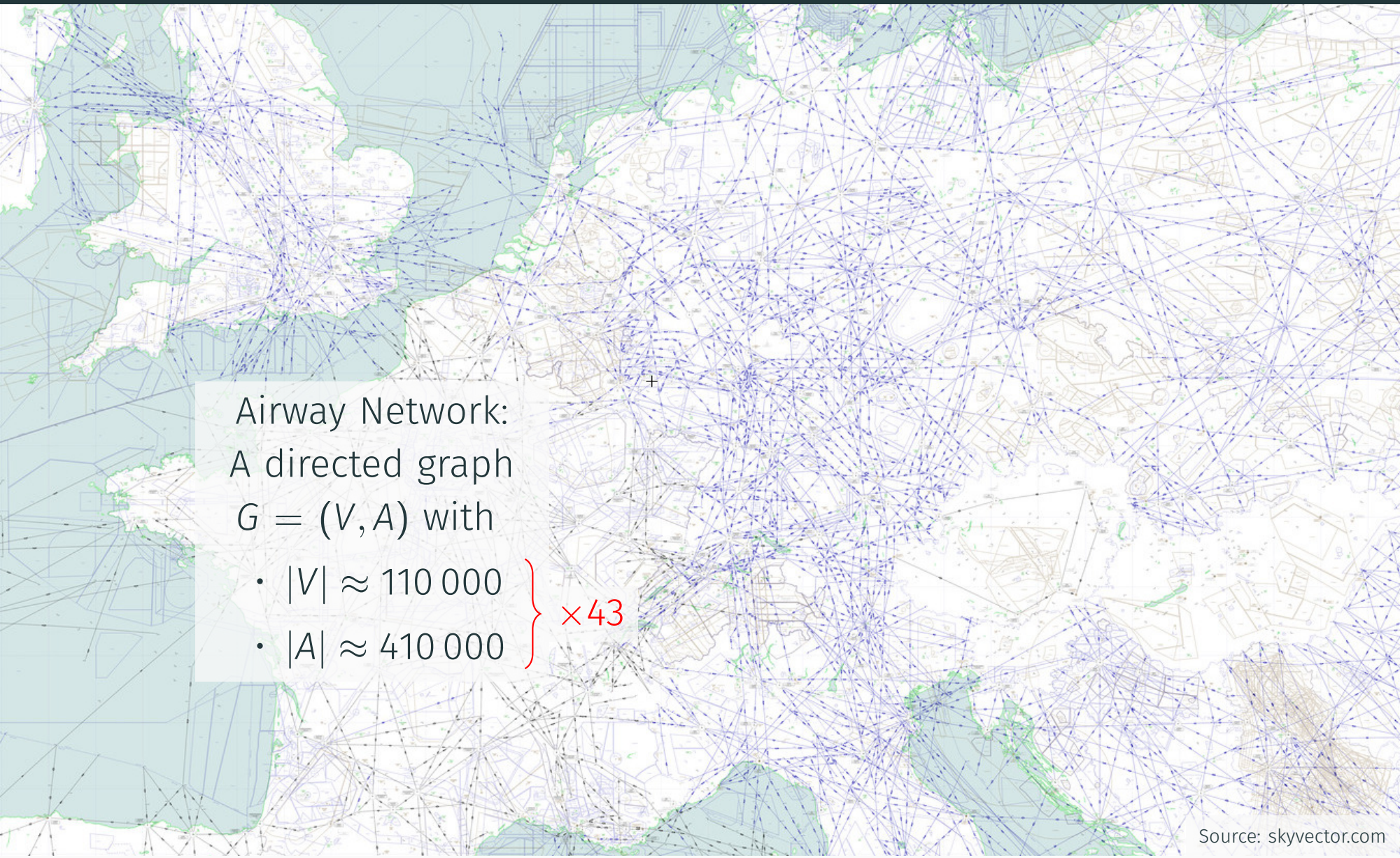


Figure 1: A typical flight profile. Source: flightradar24.com

The Airway Network



Airway Network:
A directed graph
$G = (V, A)$ with

- $|V| \approx 110\,000$
- $|A| \approx 410\,000$

$\times 43$

Source: skyvector.com

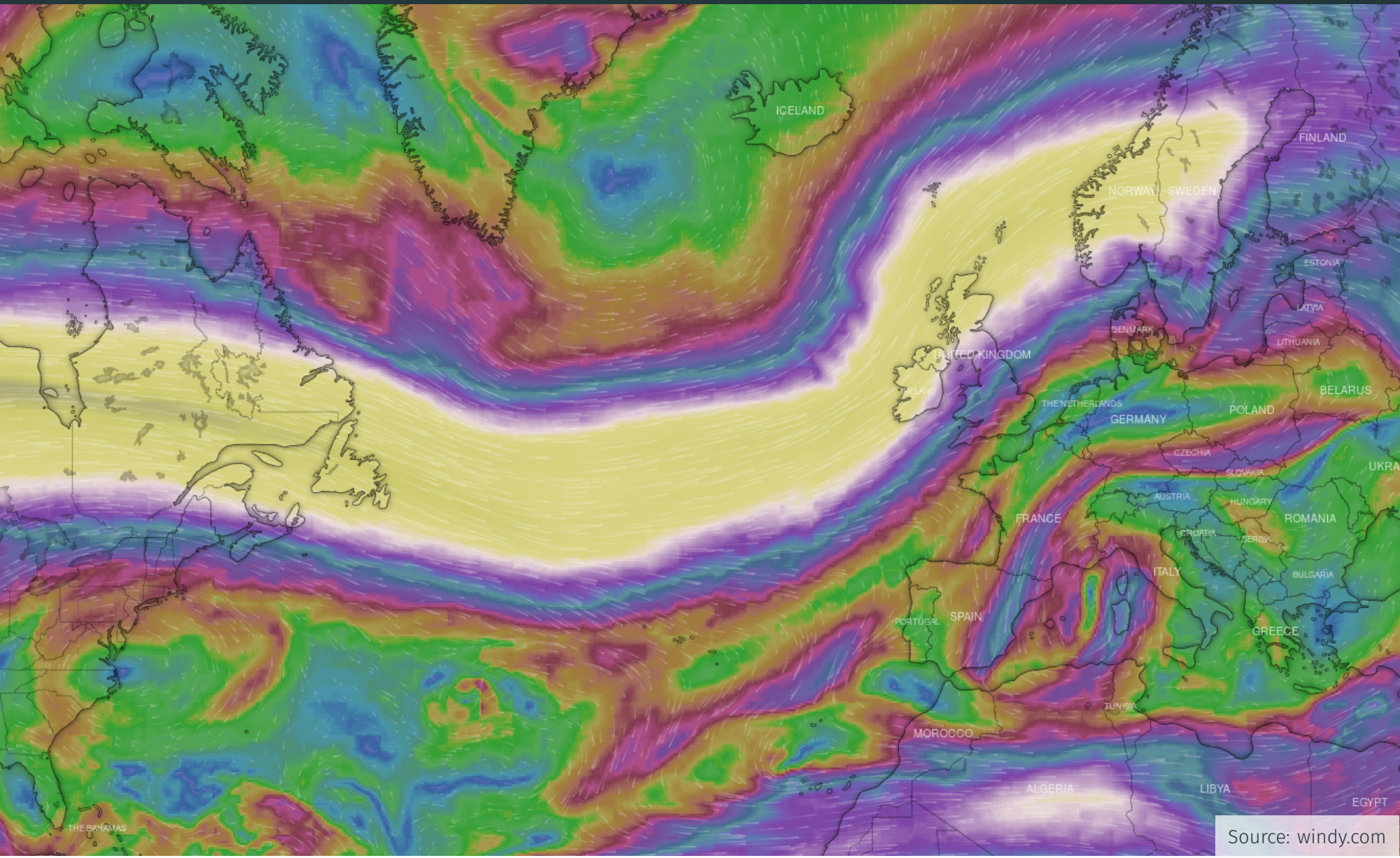
The Flight Planning Problem

Objective
Compute a feasible, minimum cost, 3D trajectory

Given:

- Origin and Destination airports s, t
- Airway Network and possible flight altitudes
- Weather forecast

Weather Forecast



The Flight Planning Problem

Objective
Compute a feasible, minimum cost, 3D trajectory

Given:

- Origin and Destination airports s, t
- Airway Network and possible flight altitudes
- Weather forecast
- Departure time and constant flight speed

The Flight Planning Problem

Objective
Compute a feasible, minimum cost, 3D trajectory

Given:

- Origin and Destination airports s, t
- Airway Network and possible flight altitudes
- Weather forecast
- Departure time and constant flight speed
- Overflight Costs

Overflight Costs

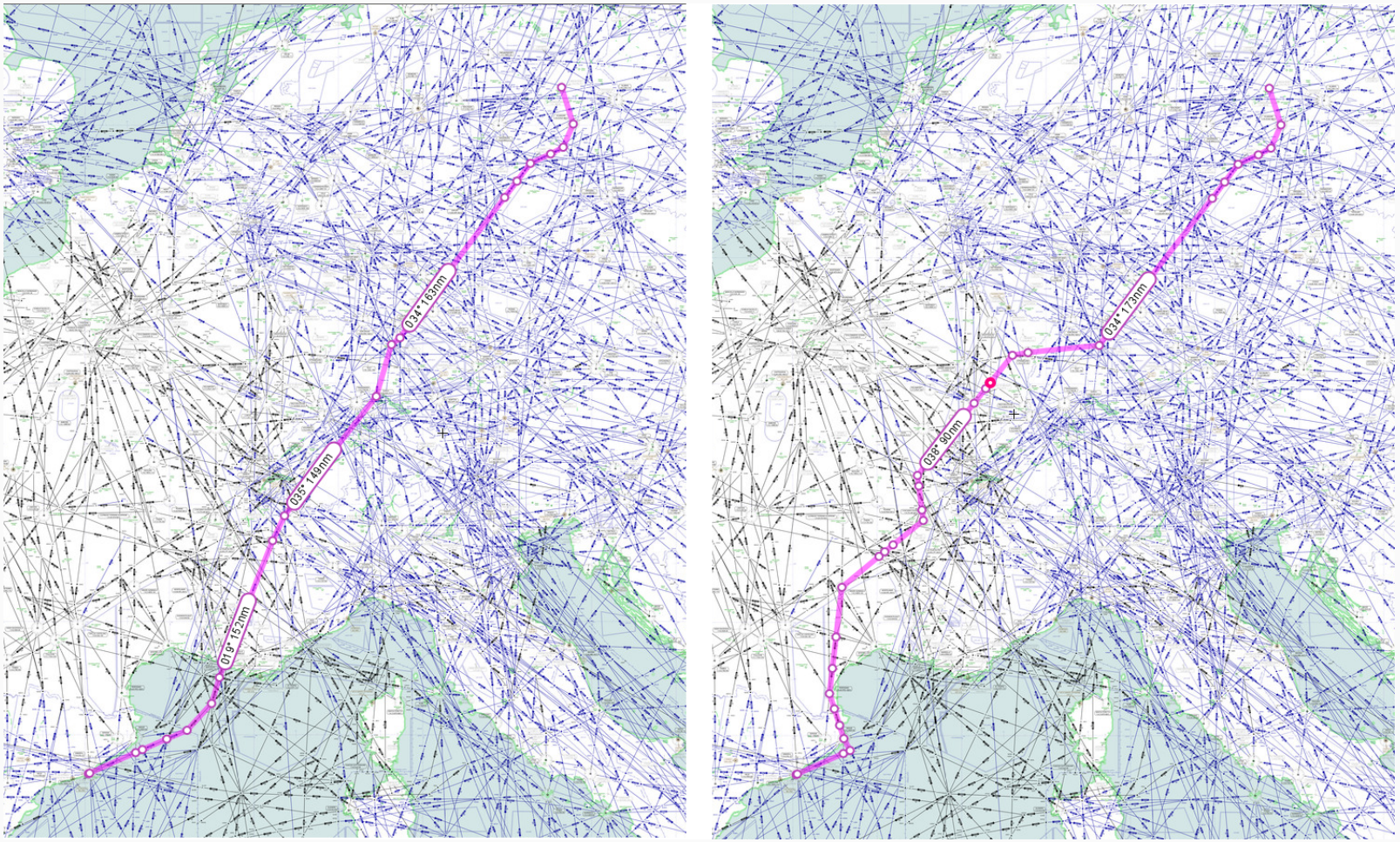


Figure 2: Route BCN-TXL without Overflight Costs and with Overflight Costs.
Source: skyvector.com

The Flight Planning Problem

Objective
Compute a feasible, minimum cost, 3D trajectory

Given:

- Origin and Destination airports s, t
- Airway Network and possible flight altitudes
- Weather forecast
- Departure time and constant flight speed
- Overflight Costs
- Fuel Prices

The Flight Planning Problem

Objective
Compute a feasible, minimum cost, 3D trajectory

Given:

- Origin and Destination airports s, t
- Airway Network and possible flight altitudes
- Weather forecast
- Departure time and constant flight speed
- Overflight Costs
- Fuel Prices
- Aircraft specifications, weight, ...

The Planning Process starts when

- the aircraft type is known

[1]

[1] Source: Wikipedia
[2] Source: www.shareicon.net

The Planning Process starts when

- the aircraft type is known

[1]

- the weather forecast is available

[1]

[1]Source: Wikipedia
[2]Source: www.shareicon.net

The Planning Process starts when

- the aircraft type is known

[1]

- the weather forecast is available

[1]

- we know the approximate passenger count

[2]

[1] Source: Wikipedia
[2] Source: www.shareicon.net

The Planning Process starts when

- the aircraft type is known

[1]

- the weather forecast is available

[1]

- we know the approximate passenger count

[2]

- we know about extraordinary circumstances

[1]

[1] Source: Wikipedia
[2] Source: www.shareicon.net

The Planning Process starts when

- the aircraft type is known

[1]

- the weather forecast is available

[1]

- we know the approximate passenger count

[2]

- we know about extraordinary circumstances

[1]

The computation must be fast.

[1] Source: Wikipedia
[2] Source: www.shareicon.net

Observations

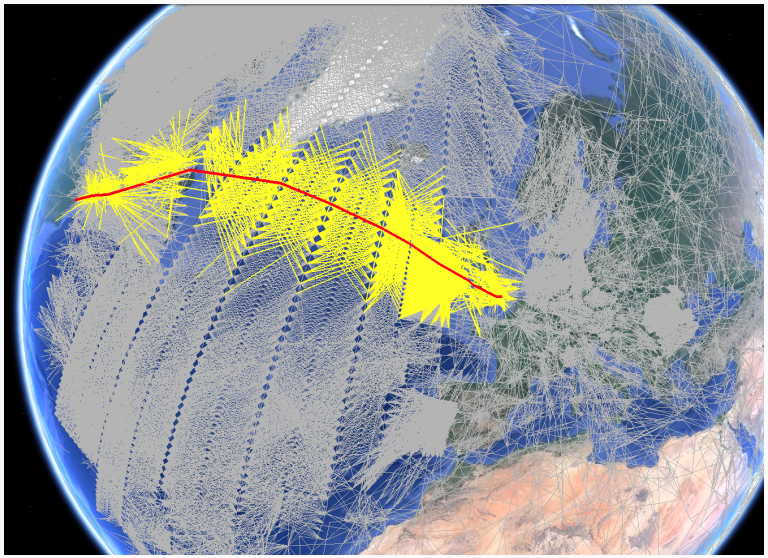


Figure 3: Search spaces for A* (yellow) and Dijsktra's Algorithm (gray) between LHR and JFK. Source: Google Earth

- Only ≈ 1300 nodes are airports.

Observations

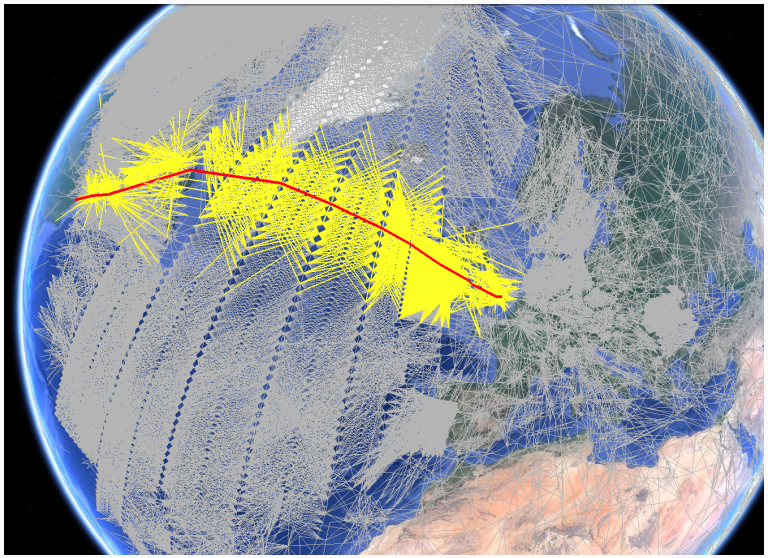


Figure 3: Search spaces for A* (yellow) and Dijsktra's Algorithm (gray) between LHR and JFK. Source: Google Earth

- Only ≈ 1300 nodes are airports.
- We *almost* know the direction of the flight.

Observations

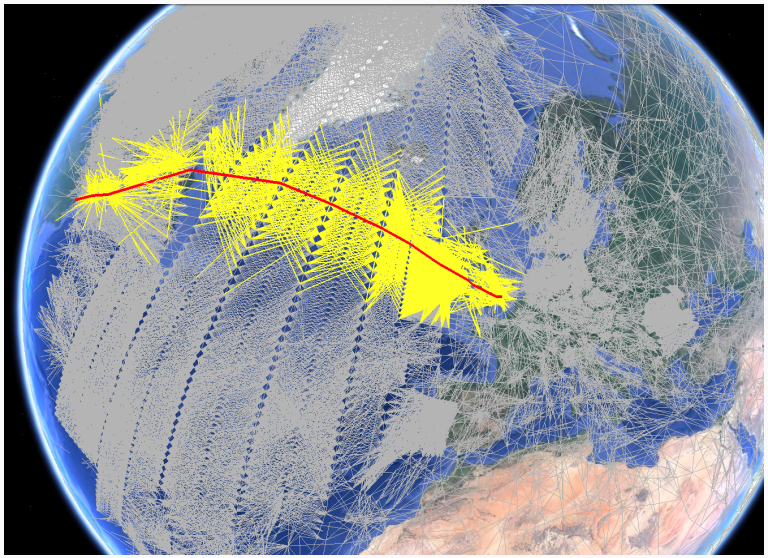


Figure 3: Search spaces for A* (yellow) and Dijsktra's Algorithm (gray) between LHR and JFK. Source: Google Earth

- Only ≈ 1300 nodes are airports.
- We *almost* know the direction of the flight.
- A* may yield a speedup over Dijkstra's Algorithm.

Observations

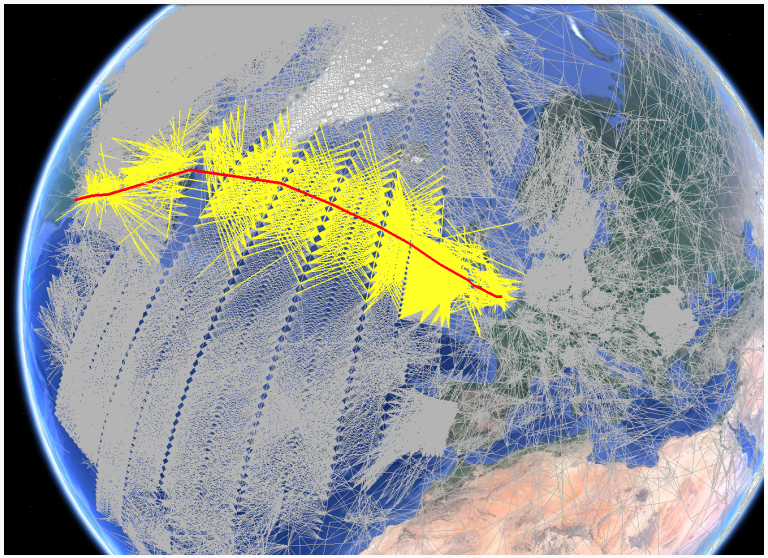


Figure 3: Search spaces for A* (yellow) and Dijsktra's Algorithm (gray) between LHR and JFK. Source: Google Earth

- Only ≈ 1300 nodes are airports.
- We *almost* know the direction of the flight.
- A* may yield a speedup over Dijkstra's Algorithm.
- Due to weather: potential calculation is not straightforward.

Simplification

Simplifications

- Concentrate on one layer of the graph

Simplifications

- Concentrate on one layer of the graph
- Only look at the weather-dependency

Simplifications

- Concentrate on one layer of the graph
- Only look at the weather-dependency
- Disregard overflight costs and restrictions

Simplifications

- Concentrate on one layer of the graph
- Only look at the weather-dependency
- Disregard overflight costs and restrictions
- Optimise for the minimum time (in this case, equivalent to minimum fuel consumption)

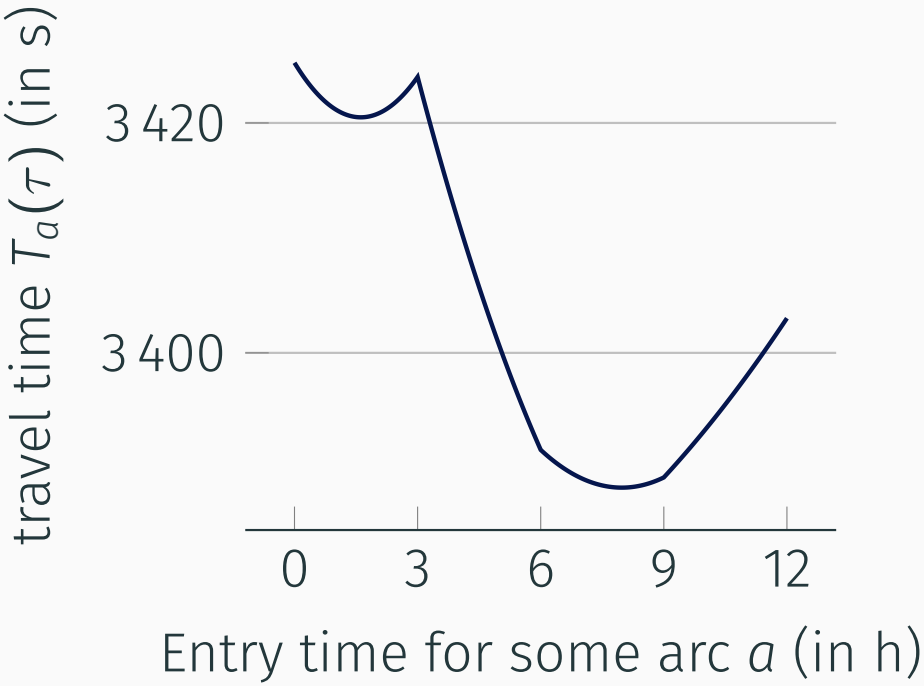
Simplifications

- Concentrate on one layer of the graph
- Only look at the weather-dependency
- Disregard overflight costs and restrictions
- Optimise for the minimum time (in this case, equivalent to minimum fuel consumption)

Examine benefits of A* on this subproblem

A* – The Potential Function π_t

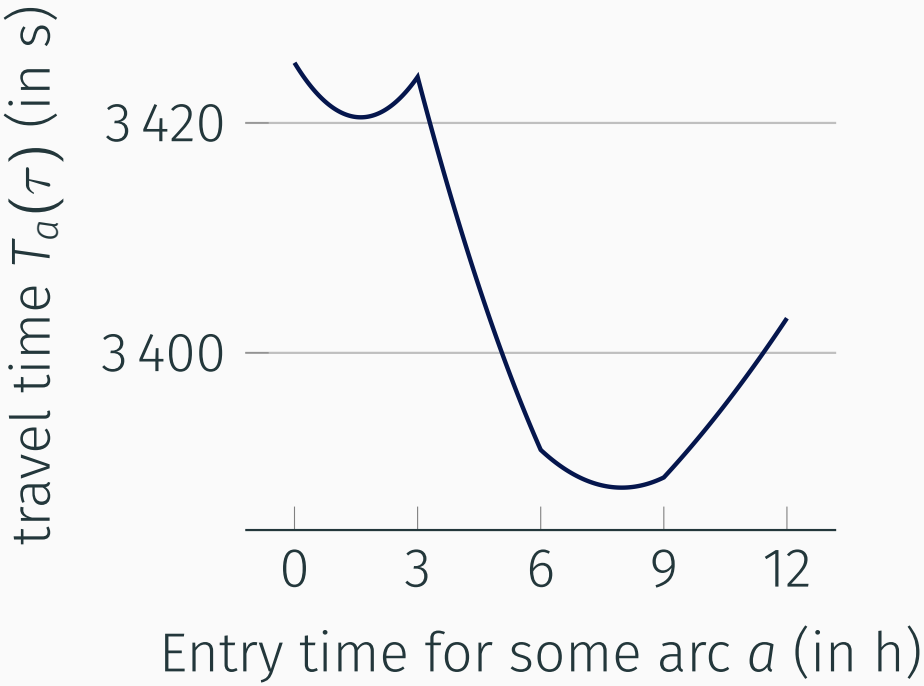
- Given: travel time function T_a for each arc $a \in A$.



A* – The Potential Function π_t

- Given: travel time function T_a for each arc $a \in A$.
- Find minimum travel time

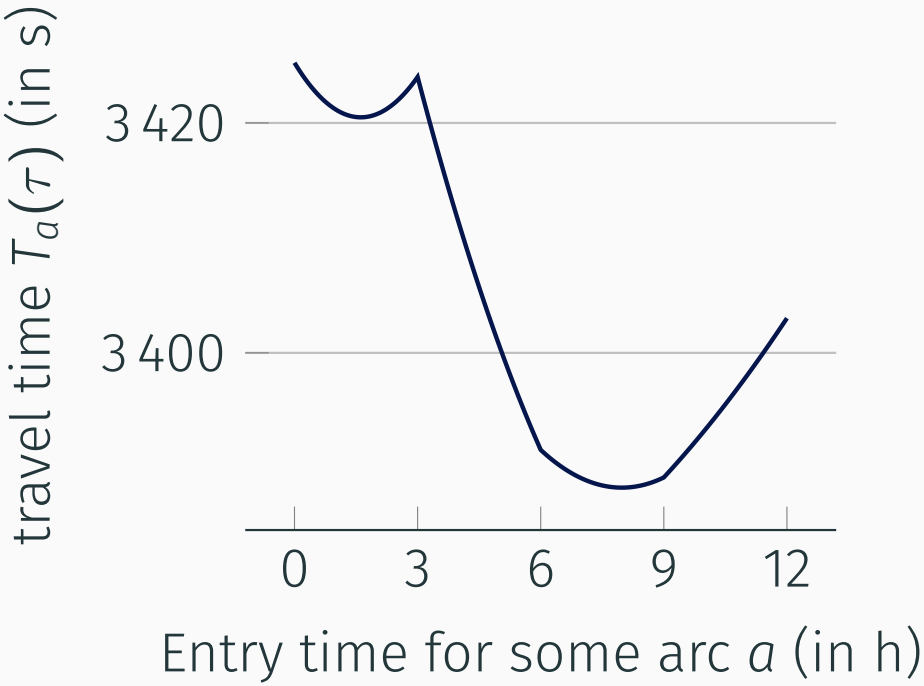
$$\underline{T}(a) := \min_{\tau \in [t_0, t_N]} T_a(\tau).$$



A* – The Potential Function π_t

- Given: travel time function T_a for each arc $a \in A$.
- Find minimum travel time

$$\underline{T}(a) := \min_{\tau \in [t_0, t_N]} T_a(\tau).$$

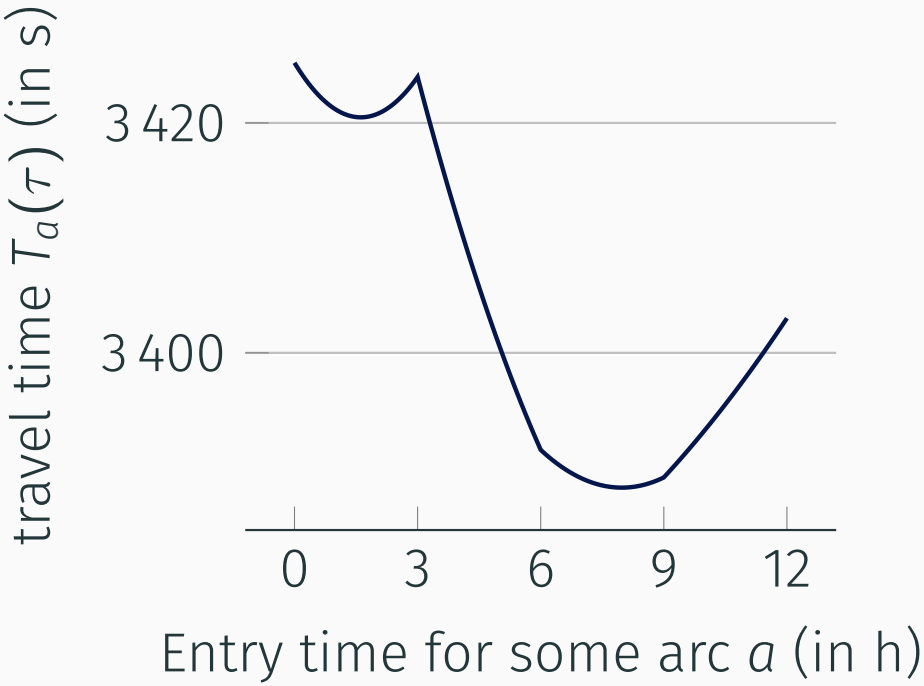


- Compute all-to-one shortest path trees towards all airport nodes ($\approx$ 1300) using $\underline{T}$ as arc costs.

A* – The Potential Function π_t

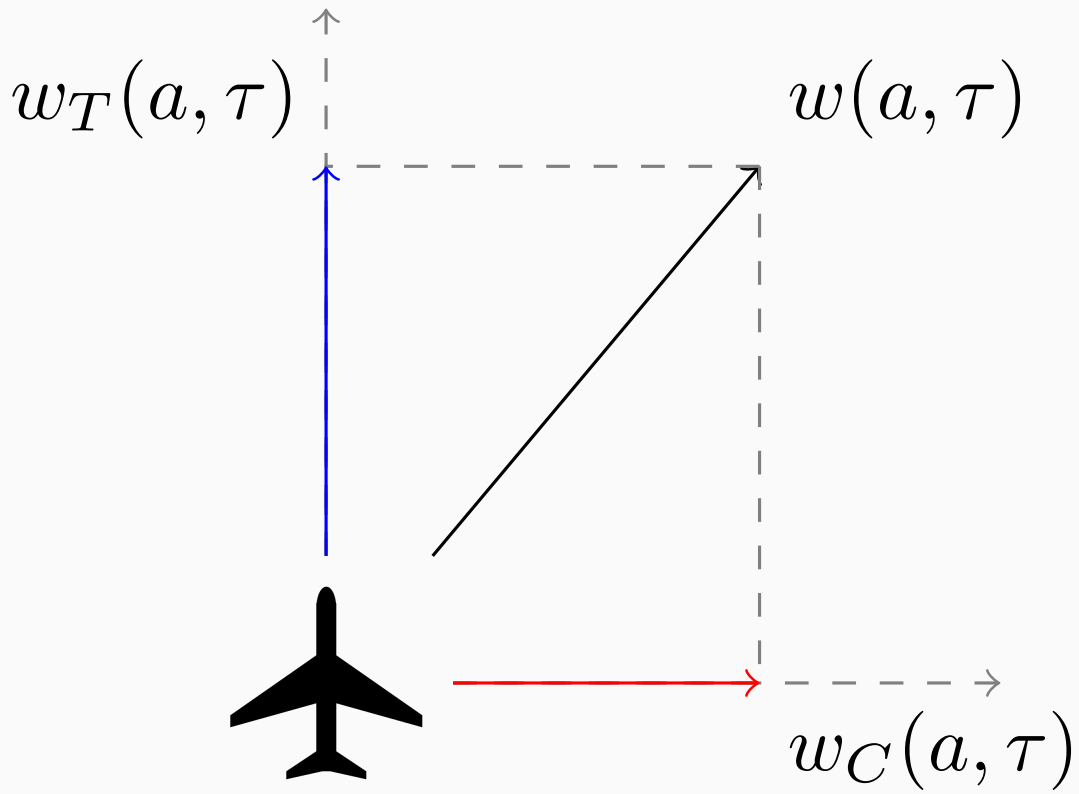
- Given: travel time function T_a for each arc $a \in A$.
- Find minimum travel time

$$\underline{T}(a) := \min_{\tau \in [t_0, t_N]} T_a(\tau).$$

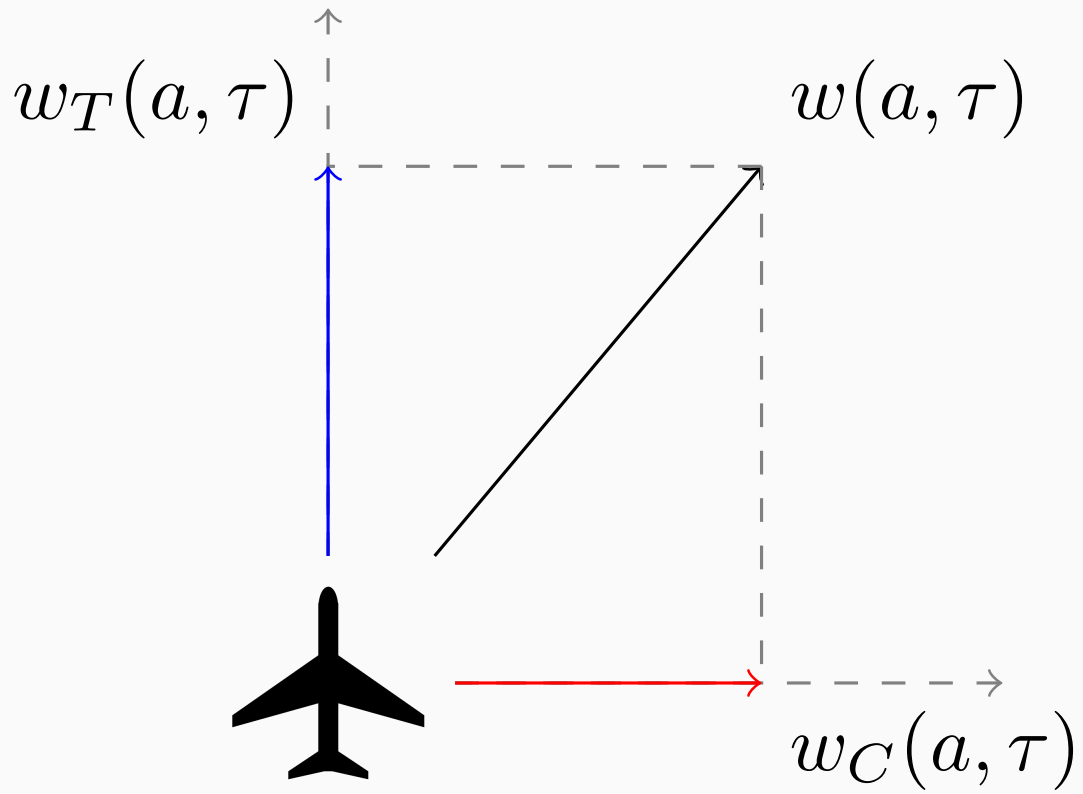


- Compute all-to-one shortest path trees towards all airport nodes ($\approx$ 1300) using $\underline{T}$ as arc costs.
- Define $\pi_t(v) = \underline{T}(P_v^t)$ – the length of the shortest (v, t)-path on $(G, \underline{T})$.

Interlude: Nomenclature



Interlude: Nomenclature



Minimum travel time on $a \in A \leftrightarrow$ “Optimal” wind conditions.

Underestimate the Travel Time: Super-Optimal Wind

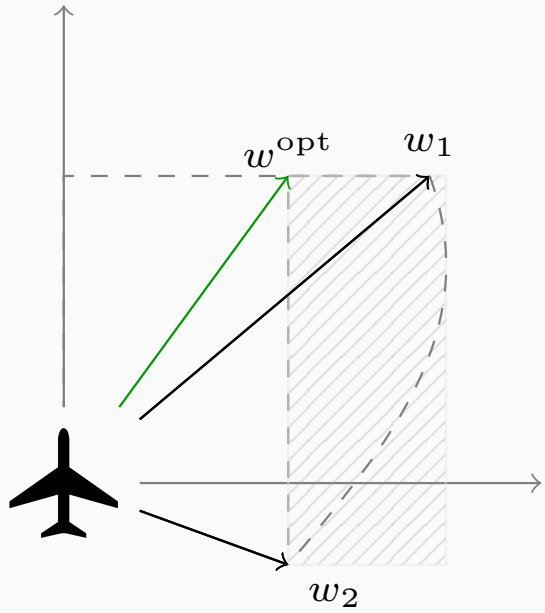
- The general travel time function is piecewise smooth.

Underestimate the Travel Time: Super-Optimal Wind

- The general travel time function is piecewise smooth.
- Instead of finding the minimum travel time, underestimate it.

Underestimate the Travel Time: Super-Optimal Wind

- The general travel time function is piecewise smooth.
- Instead of finding the minimum travel time, underestimate it.



- For a given time interval, superimpose the minimum cross wind and maximum track wind to obtain the **Super-Optimal Wind** vector yielding a lower bound $\underline{T}(a)$ on the travel time for $a \in A$.

Super-Optimal Wind in Theory and Practice

Theorem: Let $T^*(a)$ be the minimum travel time on $a \in A$. *Under reasonable assumptions on the wind*, there is a constant C such that

$$0 \leq T^*(a) - \underline{T}(a) \leq C\Delta. \qquad \square$$

Super-Optimal Wind in Theory and Practice

Theorem: Let $T^*(a)$ be the minimum travel time on $a \in A$. *Under reasonable assumptions on the wind*, there is a constant C such that

$$0 \leq T^*(a) - \underline{I}(a) \leq C\Delta. \qquad \square$$

Theorem: If $\pi_t(v) = \underline{I}(P_v^t)$, then π_t is admissible and consistent. $\square$

Super-Optimal Wind in Theory and Practice

Theorem: Let $T^*(a)$ be the minimum travel time on $a \in A$. *Under reasonable assumptions on the wind*, there is a constant C such that

$$0 \leq T^*(a) - \underline{T}(a) \leq C\Delta. \qquad \square$$

Theorem: If $\pi_t(v) = \underline{T}(P_v^t)$, then π_t is admissible and consistent. $\square$

Super-Optimal Wind in Practice

- Average relative error over all arcs: $0.434 \cdot 10^{-3}$
- No error in 1/3 of all cases

Contraction Hierarchies

- State-of-the-art routing algorithm for road networks[3]
- Speedup *on road networks*: over 40 000 wrt Dijkstra's algorithm
- Also effective in the time-dependent case[4]
- Karlsruhe Institute of Technology provides free source code of their implementation

[3] R. Geisberger et al.: Exact Routing in Large Road Networks Using Contraction Hierarchies, *Transportation Science*, 2012

[4] G. Batz et al.: Minimum Time-Dependent Travel Times with Contraction Hierarchies, *Journal of Experimental Algorithmics*, 2013

A* vs Dijkstra's Algorithm vs Contraction Hierarchies

Instance	DIJKSTRA	CONTRACTION HIERARCHIES			A*		
	query (ms)	prep (min)	query (ms)	speedup ×	prep (min)	query (ms)	speedup ×
I-29-Dec-1	4.91	380.48	4.08	1.20	0.031	0.22	21.51
I-34-Dec-1	4.91	451.82	4.27	1.15	0.030	0.24	20.24
I-39-Dec-1	4.93	195.75	3.23	1.53	0.030	0.16	30.15
I-29-Feb-1	4.90	414.78	3.94	1.25	0.031	0.21	22.96
I-34-Feb-1	4.86	466.95	3.96	1.23	0.028	0.21	22.23
I-39-Feb-1	4.92	184.20	3.01	1.63	0.029	0.15	31.50
I-29-Mar-1	4.55	216.57	2.82	1.61	0.025	0.16	27.27
I-34-Mar-1	4.55	189.18	2.92	1.55	0.026	0.18	24.38
I-39-Mar-1	4.58	127.38	2.52	1.81	0.026	0.15	29.45
I-29-Dec-3	4.36	312.40	2.67	1.63	0.026	0.19	22.03
I-34-Dec-3	4.38	351.70	2.80	1.56	0.026	0.21	20.85
I-39-Dec-3	4.38	160.20	2.30	1.90	0.026	0.14	30.87
I-29-Feb-3	4.31	328.47	2.66	1.62	0.025	0.18	23.09
I-34-Feb-3	4.28	372.15	2.92	1.47	0.027	0.19	21.68
I-39-Feb-3	4.33	155.07	2.20	1.97	0.025	0.13	31.94
I-29-Mar-3	4.22	179.45	2.31	1.82	0.022	0.14	28.39
I-34-Mar-3	4.26	146.52	2.33	1.83	0.023	0.16	26.68
I-39-Mar-3	4.26	96.80	2.03	2.10	0.023	0.13	31.02
Summary							
Average	4.55	262.77	2.94	1.60	0.027	0.18	25.90
Minimum	4.22	96.8	2.03	1.15	0.022	0.13	20.24
Maximum	4.93	466.95	4.27	2.10	0.031	0.24	31.94

Generalisation

Generalisation for 3D

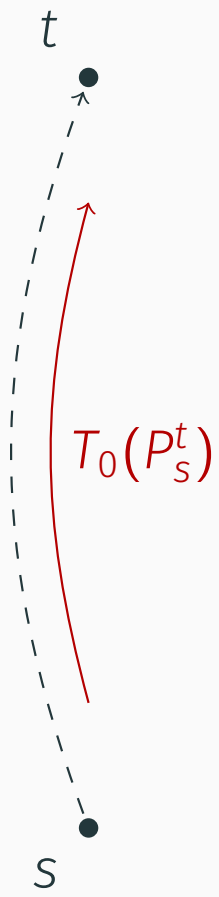
- The previous 2D results hold true in the 3D case, too.

Generalisation for 3D

- The previous 2D results hold true in the 3D case, too.
- Instead of using A*, we compute an a-priori search space:

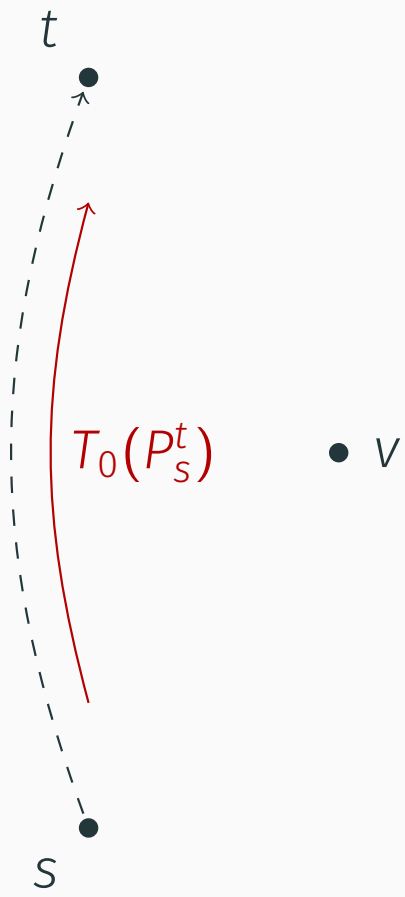
Generalisation for 3D

- The previous 2D results hold true in the 3D case, too.
- Instead of using A*, we compute an a-priori search space:
- Compute an upper bound solution, say its arrival time is T_0.



Generalisation for 3D

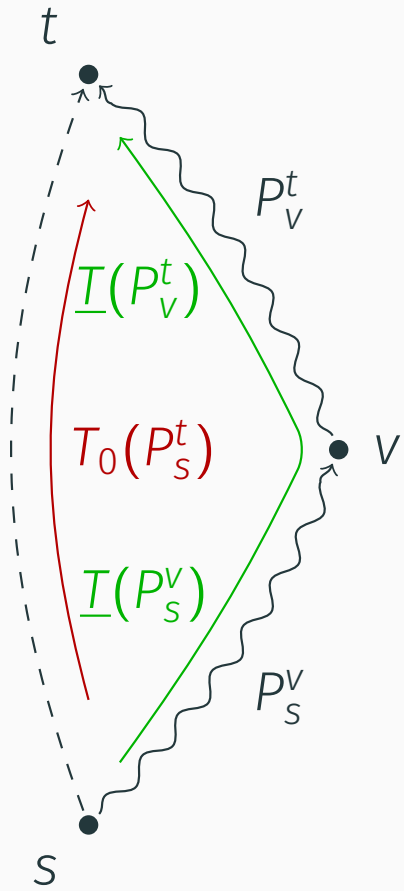
- The previous 2D results hold true in the 3D case, too.
- Instead of using A*, we compute an a-priori search space:
- Compute an upper bound solution, say its arrival time is T_0.
- A node $v \in V$ is active iff it satisfies the inequality



Generalisation for 3D

- The previous 2D results hold true in the 3D case, too.
- Instead of using A*, we compute an a-priori search space:
- Compute an upper bound solution, say its arrival time is T_0.
- A node $v \in V$ is active iff it satisfies the inequality

$$\underline{T}(P_s^v) + \underline{T}(P_v^t) \leq T_0.$$



Generalisation for 3D

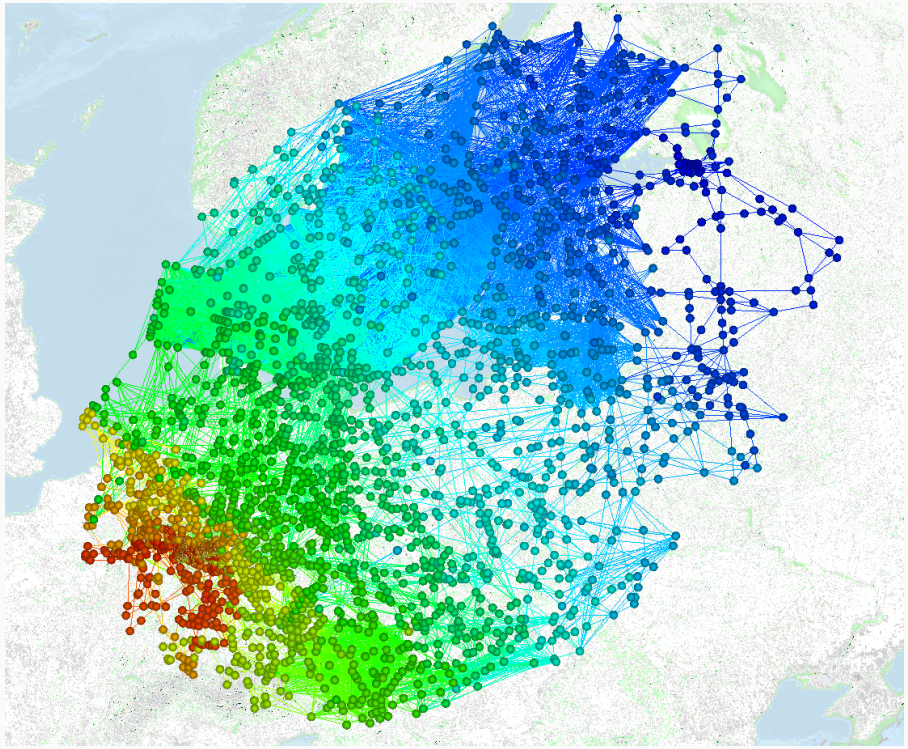
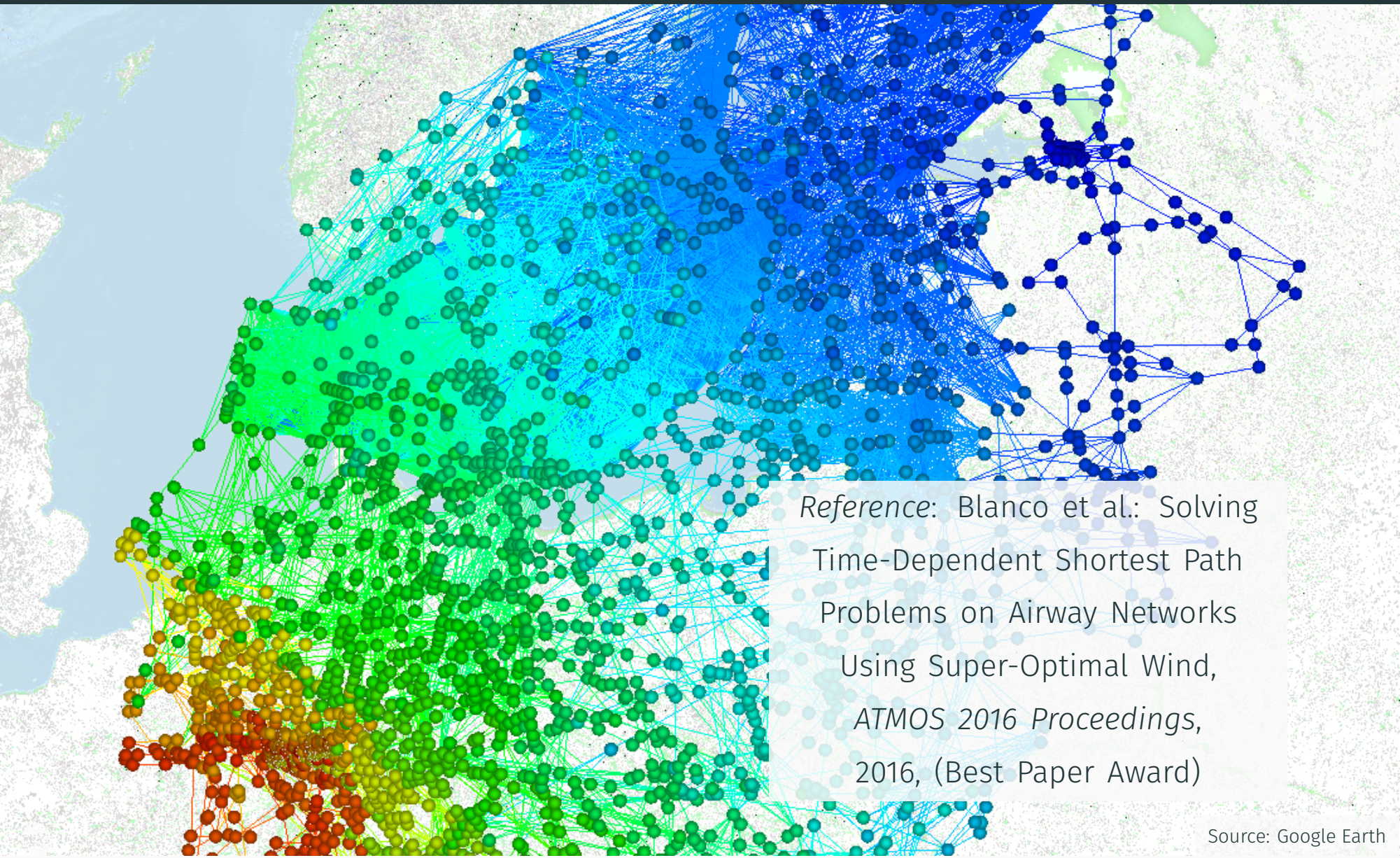


Figure 4: A-priori search space for LED-FRA. Source: Google Earth

- Benefit: this pruning also works for algorithms without heaps.
- Incorporation of overflight costs is also possible.

Thank You!



Reference: Blanco et al.: Solving Time-Dependent Shortest Path Problems on Airway Networks Using Super-Optimal Wind, *ATMOS 2016 Proceedings*, 2016, (Best Paper Award)

Source: Google Earth