How does PISA find optimal locations?

Joaquim Gromicho

Professor of Business Analytics @ UvA

Lead Optimization Architect @ ABW

Science and Education Officer @ ORTEC

Analytics for a Better World

All starts with a good accessibility metric

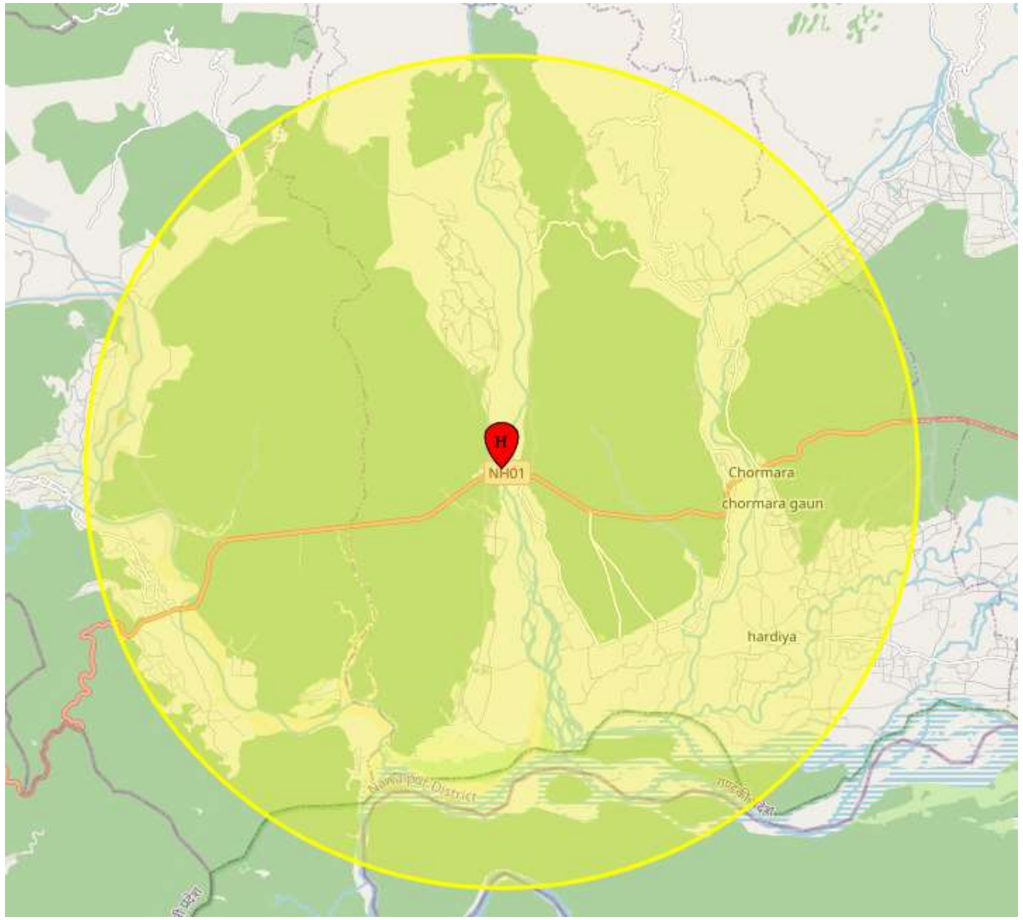
Why is that important?

Example: Madhyabindu Hospital, Nepal.

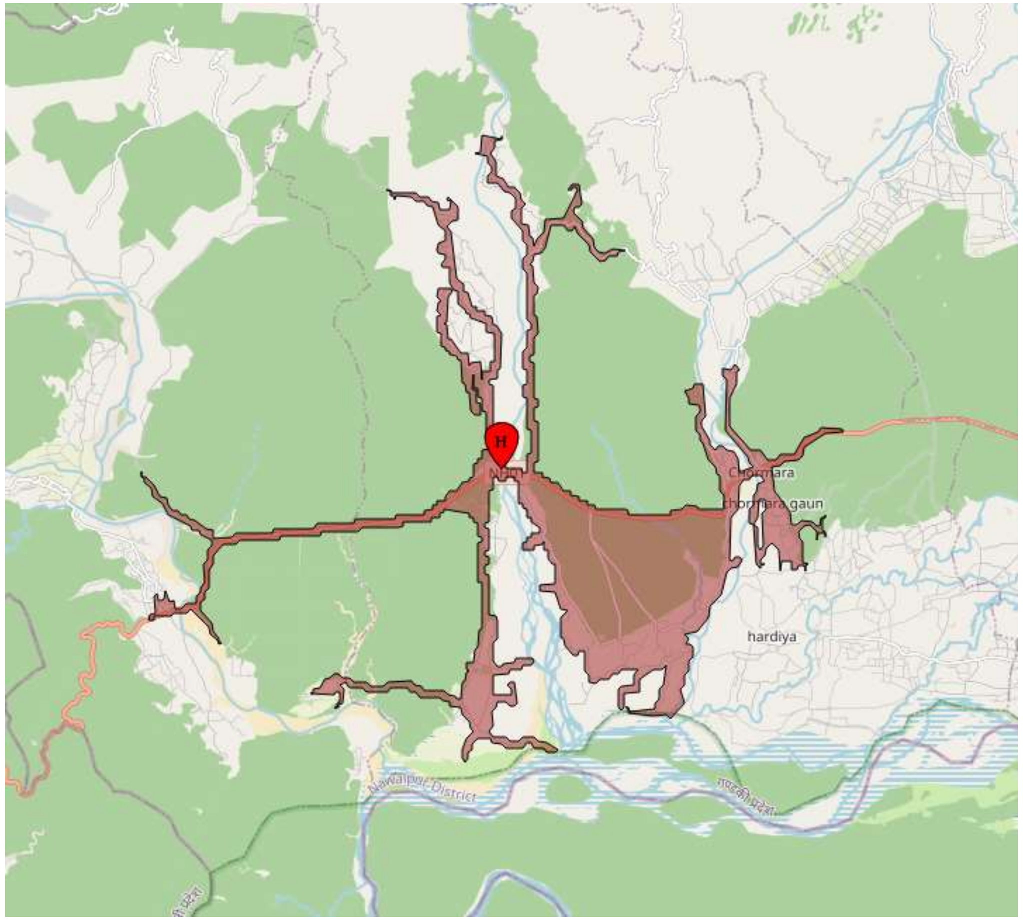
Picture from https://madhyabinduhospital.gandaki.gov.np/

Example: Madhyabindu Hospital, Nepal. From Python using folium and OSM

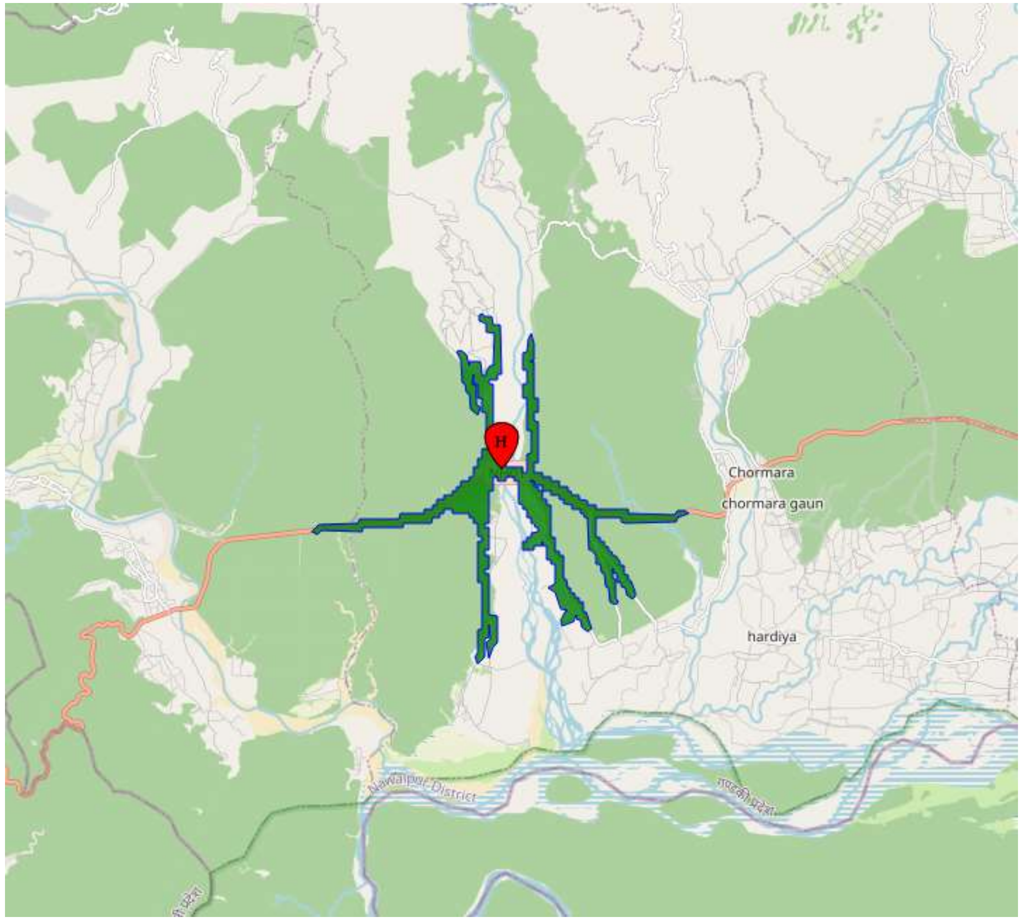
Example: Madhyabindu Hospital, Nepal. 10 km Euclidean radius



Example: Madhyabindu Hospital, Nepal.
10 km actual travel distance using roads

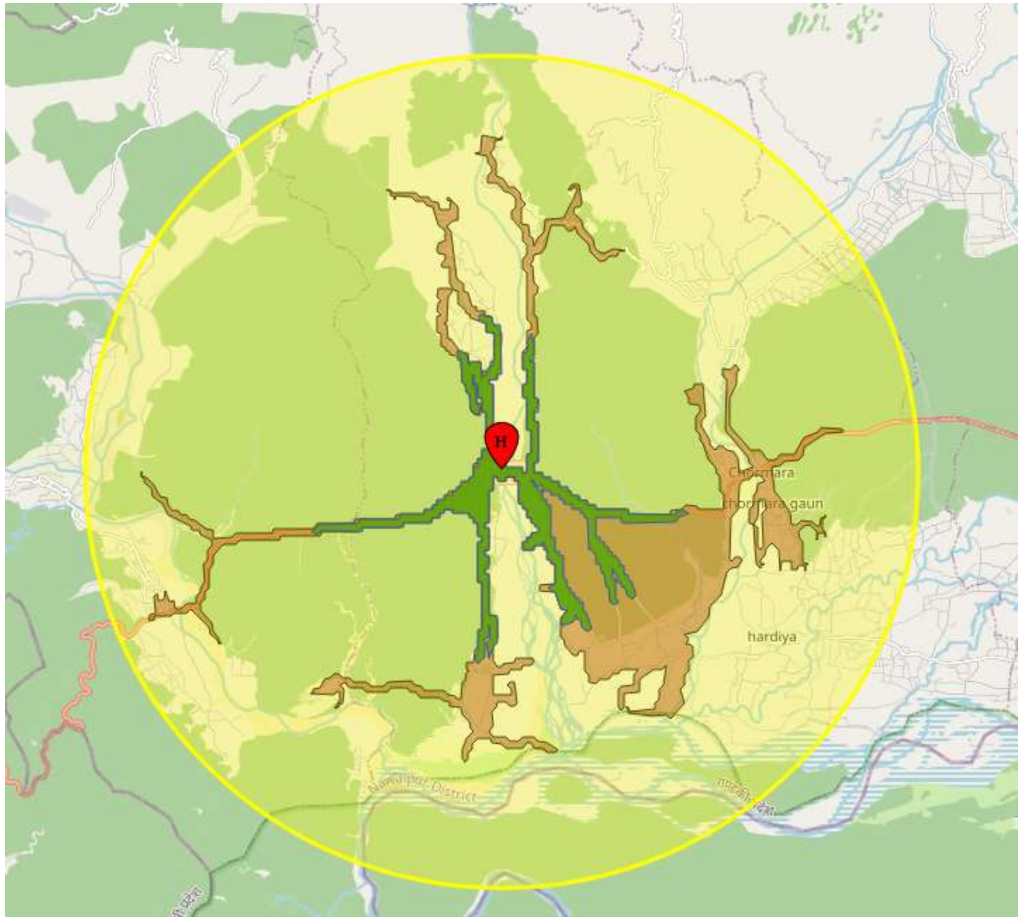


Example: Madhyabindu Hospital, Nepal. 1 hour walking using any road or trail



Analytics for a Better World

Take a few seconds to spot the differences…



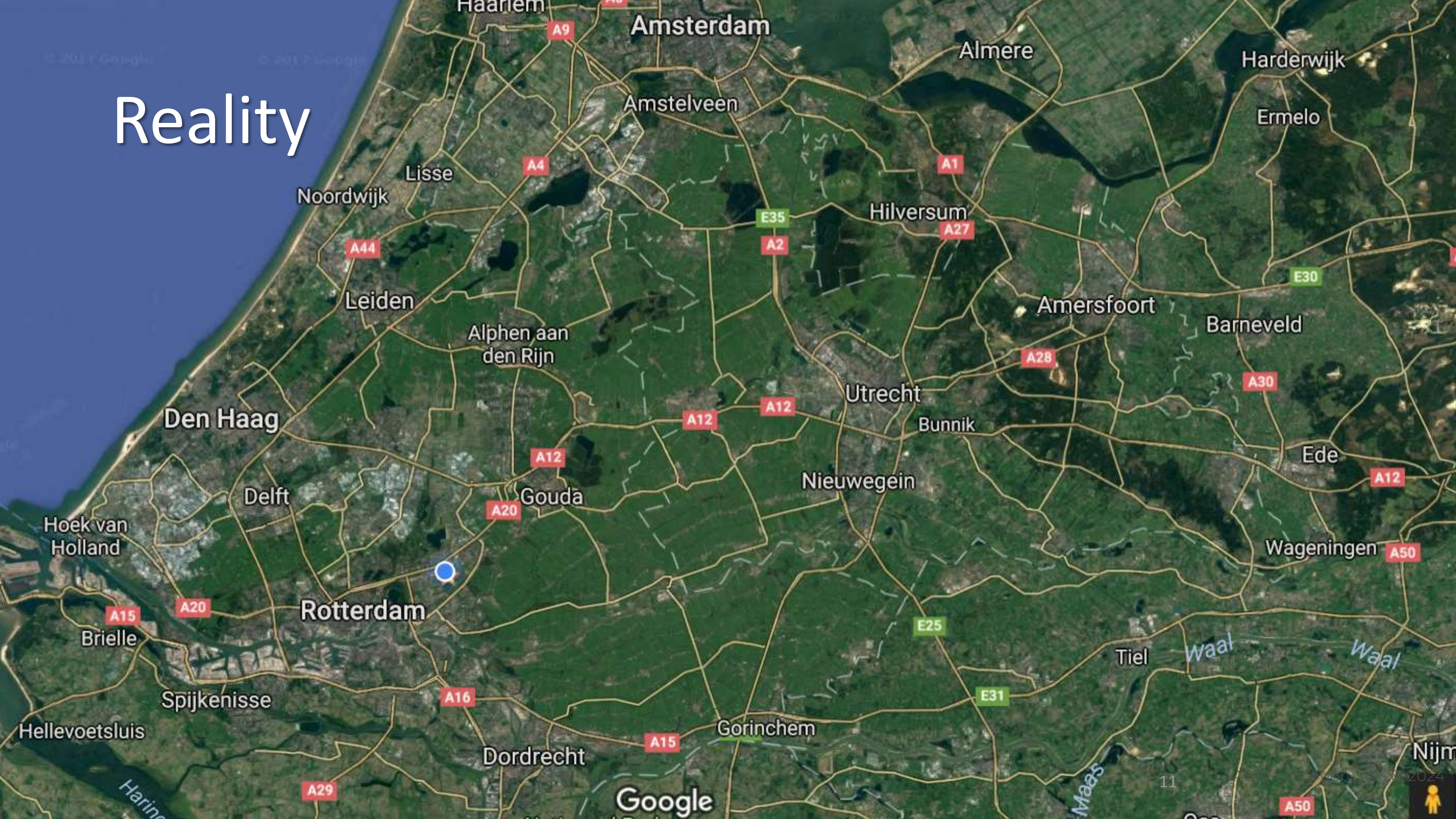
How does one compute such distances and travel times?

A very well-solved optimization problem

The proof of the pudding is in the eating…

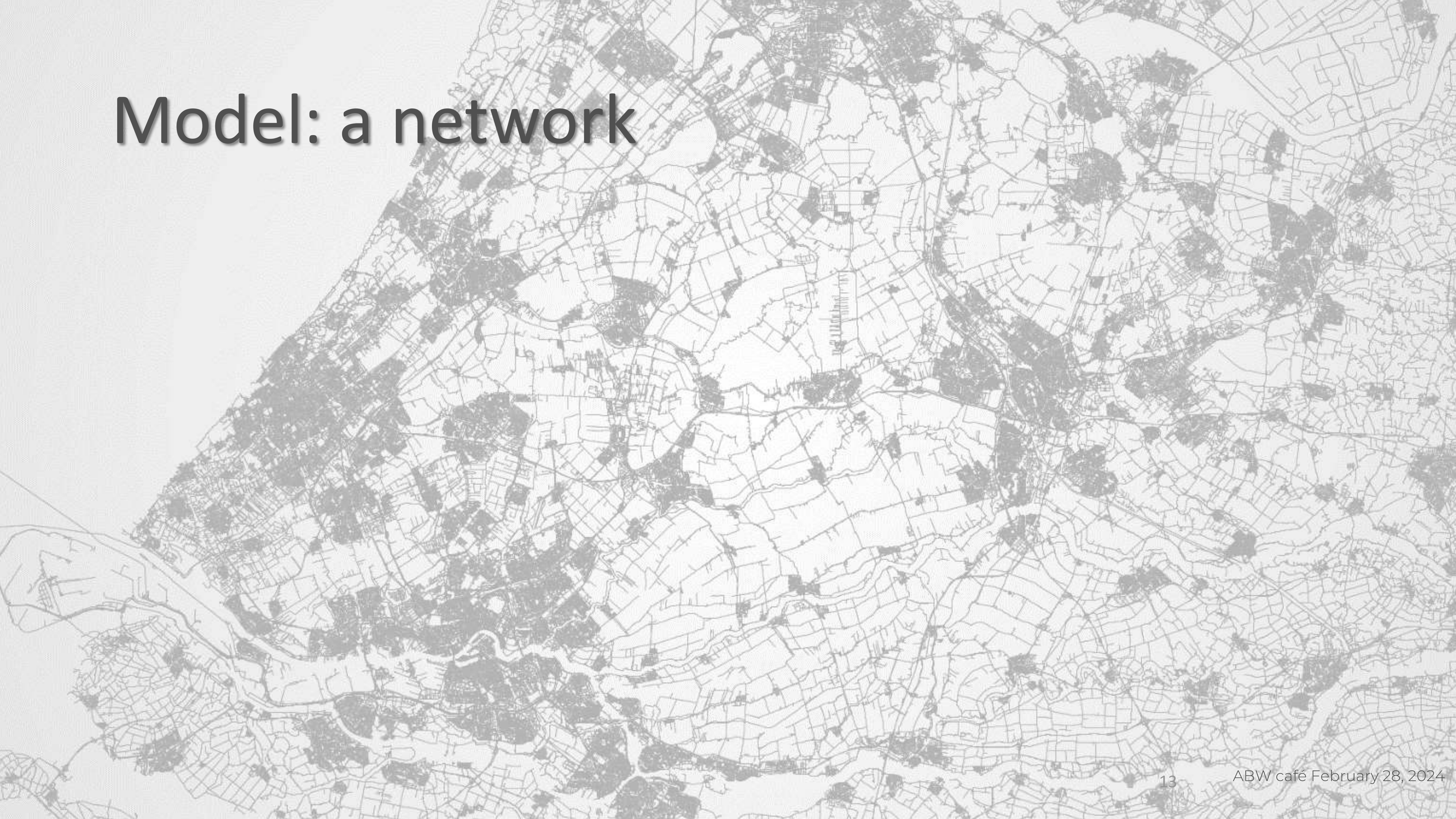
- Consider this book and in particular section 4.3.2 on page 124 and subsequent pages. We use the same as there example later.
- The textbook is complemented by an online companion to help taking steps toward real-world computations. In particular, you may enjoy this notebook.

Reality


Haarlem
A9
Amsterdam
Almere
Harderwijk
Amstelveen
Ermelo
Lisse
A4
A1
Noordwijk
E35
Hilversum
A27
A44
A2
E30
Leiden
Amersfoort
Barneveld
Alphen aan
den Rijn
A28
A30
Utrecht
A12
A12
Den Haag
Bunnik
A12
Ede
A12
Nieuwegein
Delft
Gouda
A20
Hoek van
Holland
Wageningen
A50
A20
A15
Rotterdam
Brielle
E25
Tiel
Waal
Waal
A16
E31
Spijkenisse
Gorinchem
Hellevoetsluis
A15
Nijm
Dordrecht
A29
Maas
Google
A50

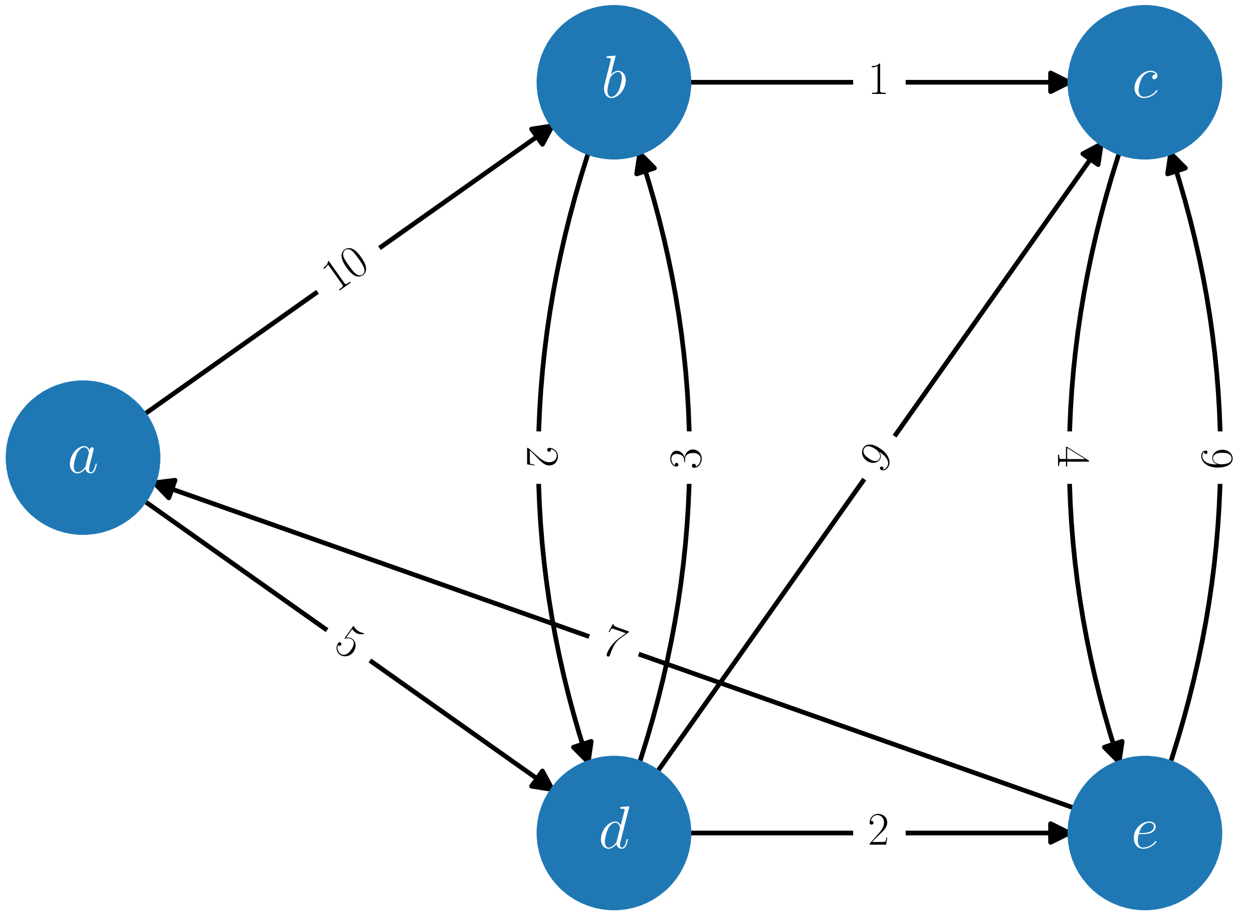
Simplification

Model: a network

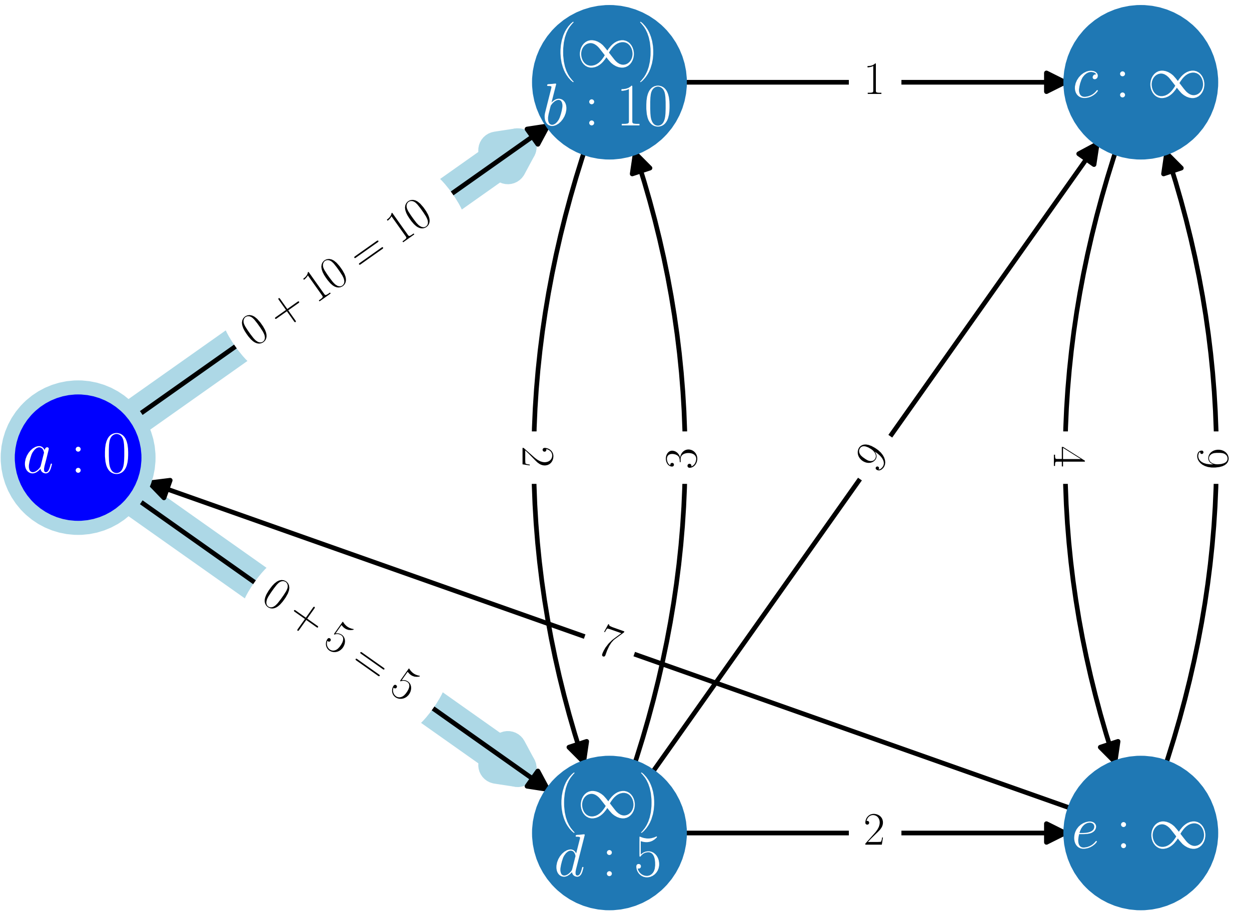


Example: you are at a and want to reach c

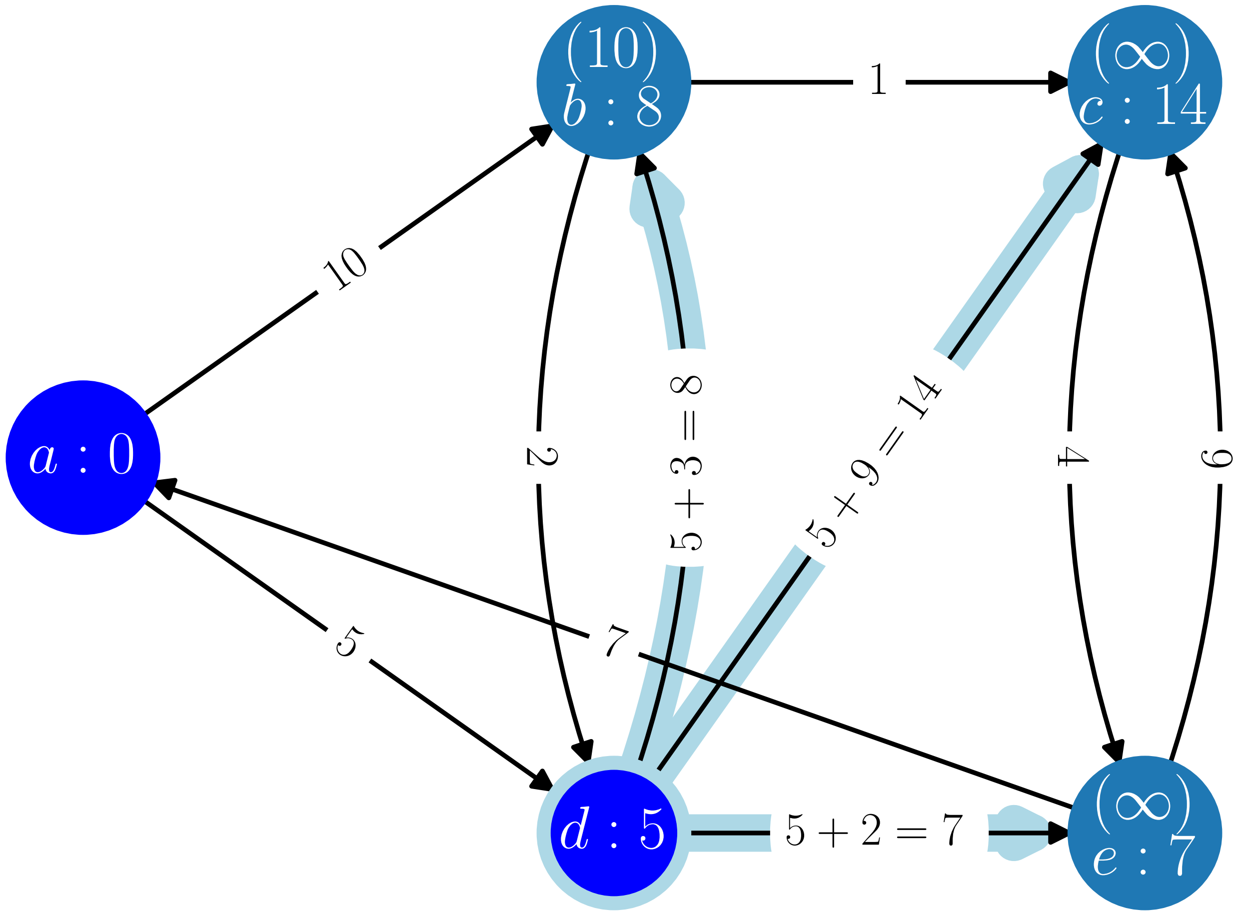
Edsger Wybe Dijkstra (1930 –2002)



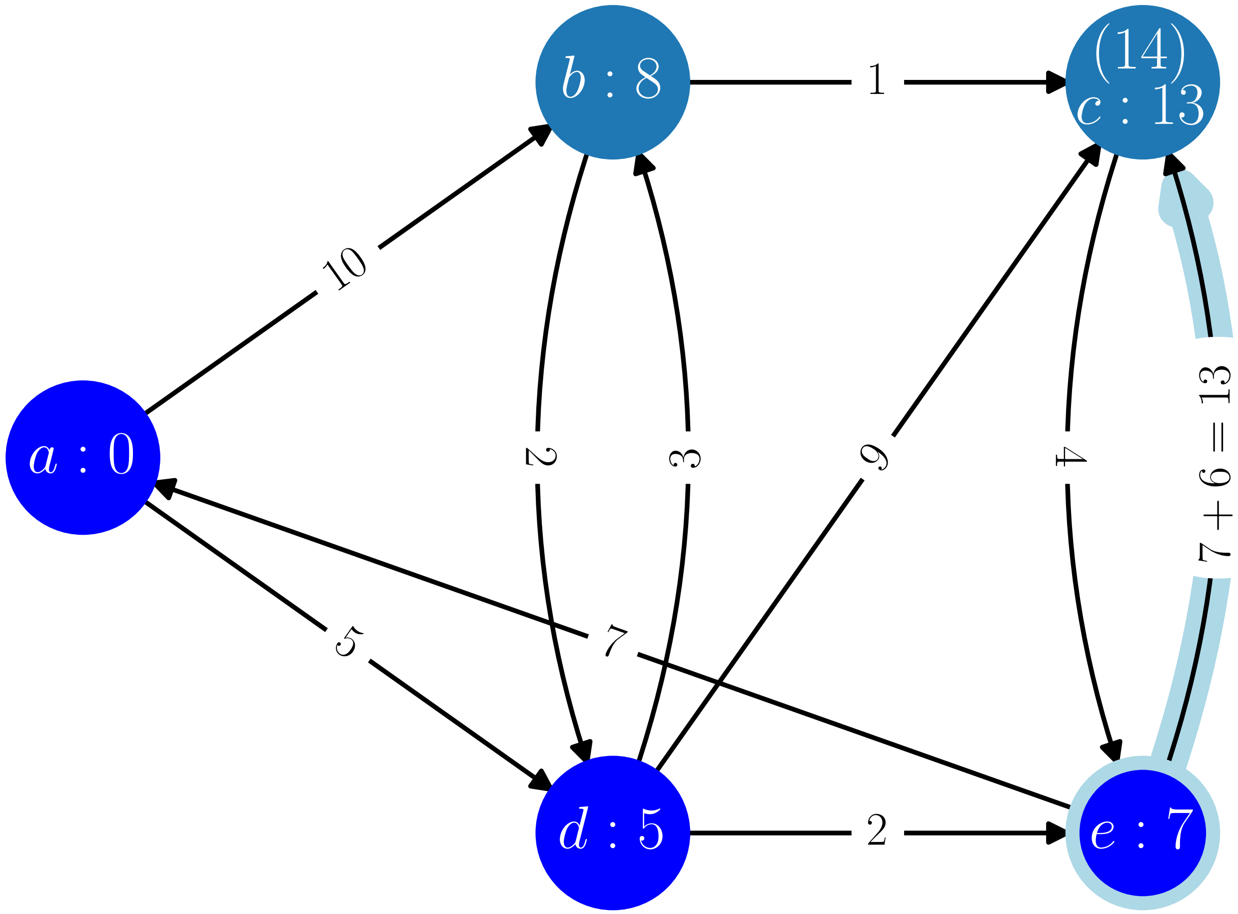
Where it starts: a



What you are certain about: d

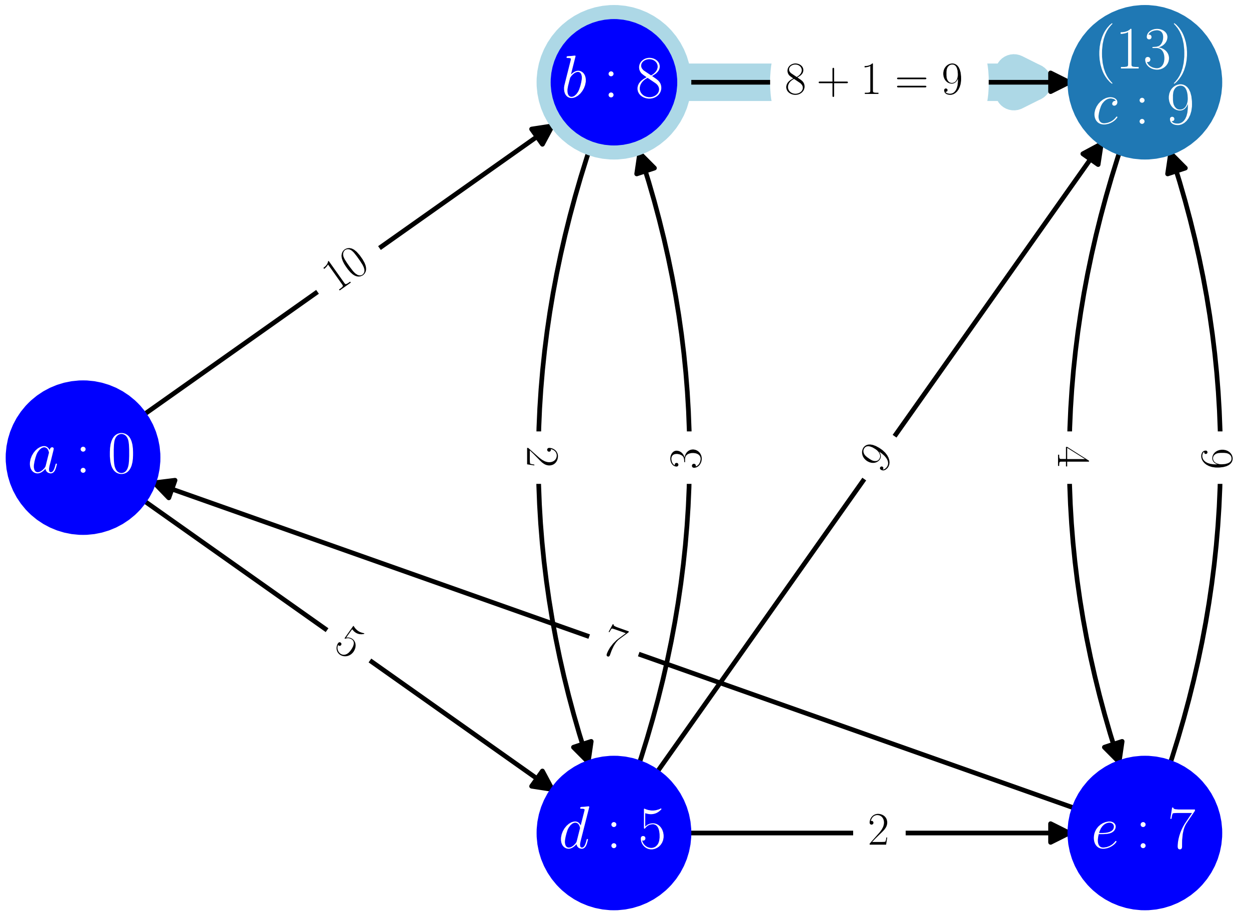


Now e is at its best!

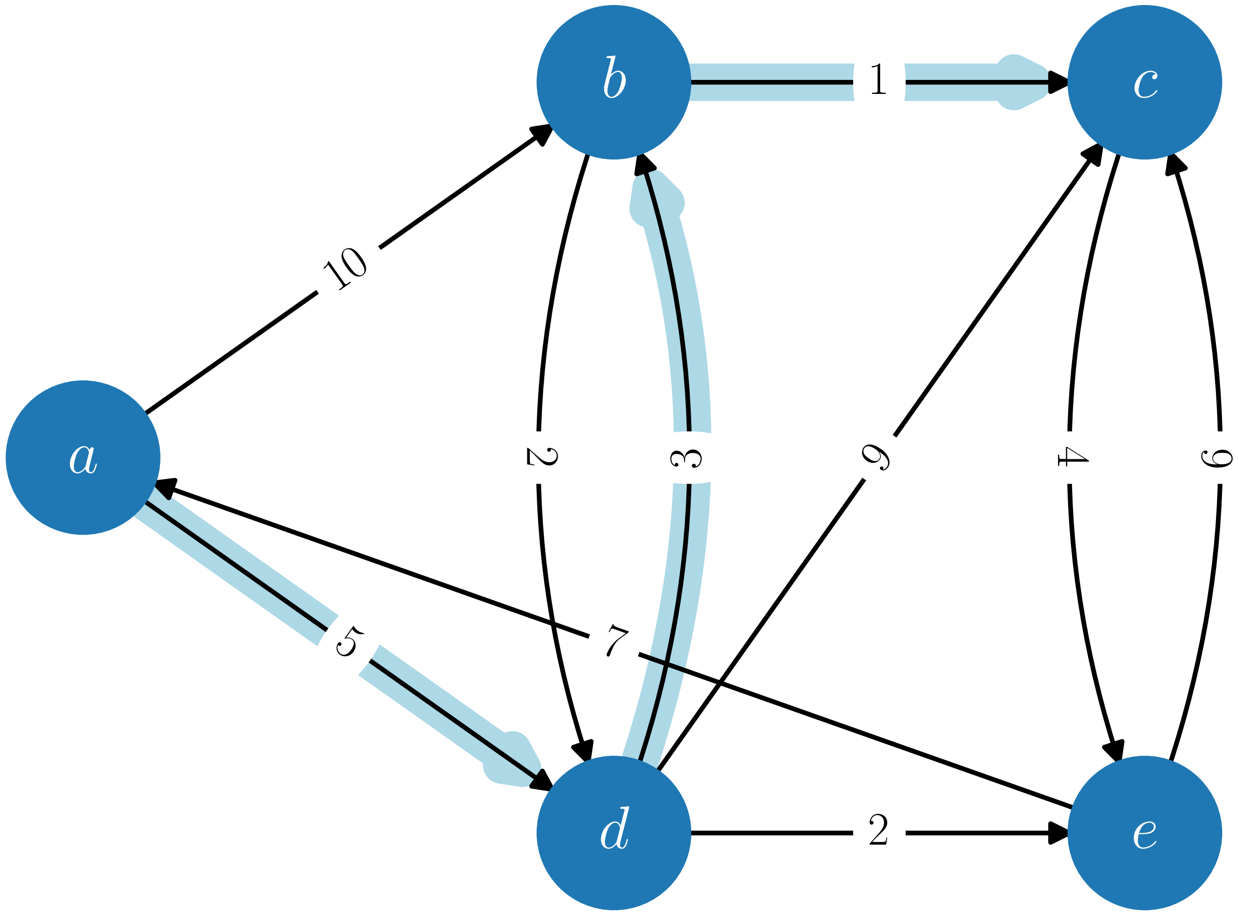


Analytics for a Better World

And now b



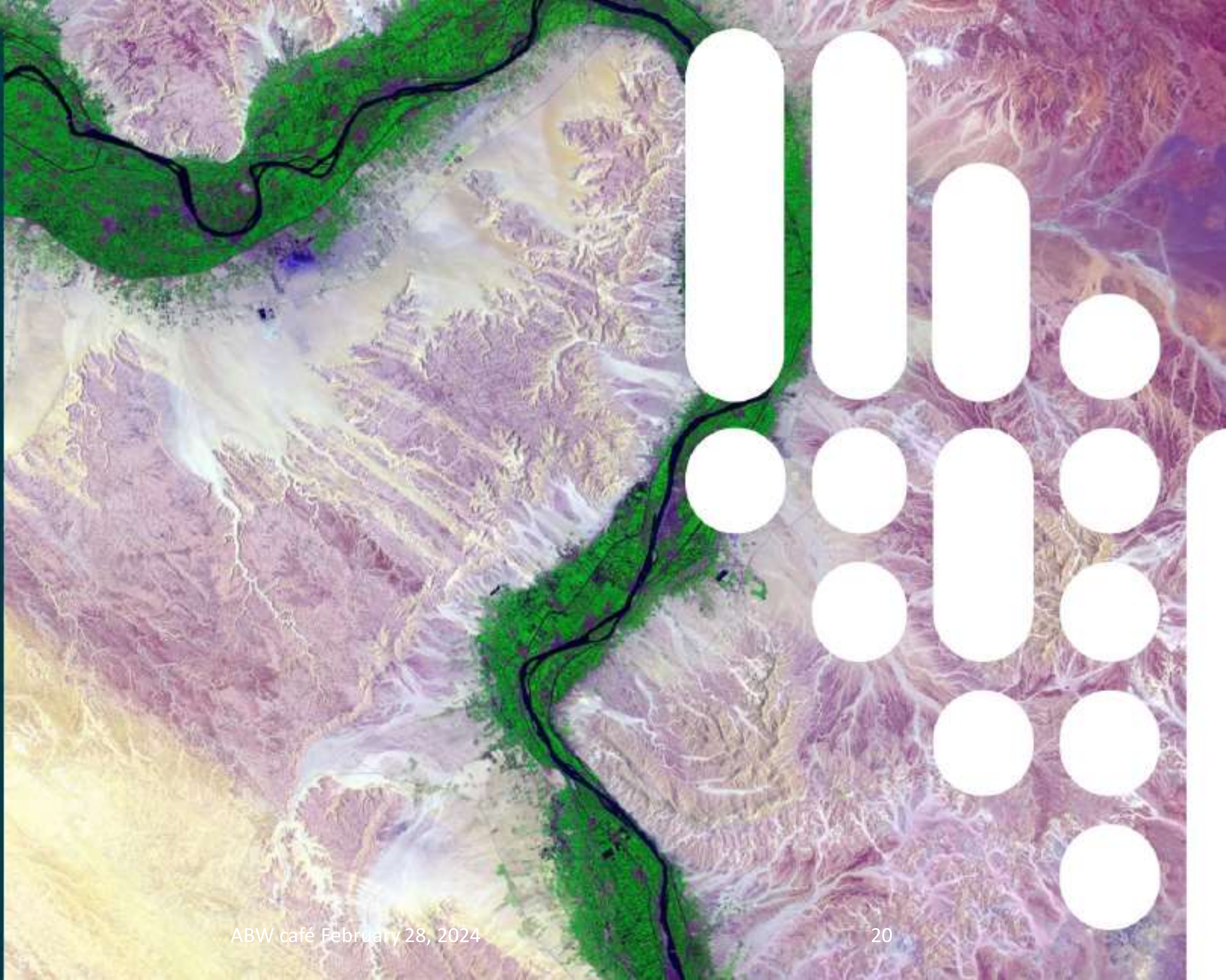
And the magic is done!



Are you inclined for computer science?

Then you may like to check this one out! It may be added to the online companion at some point.

Analytics for a Better World



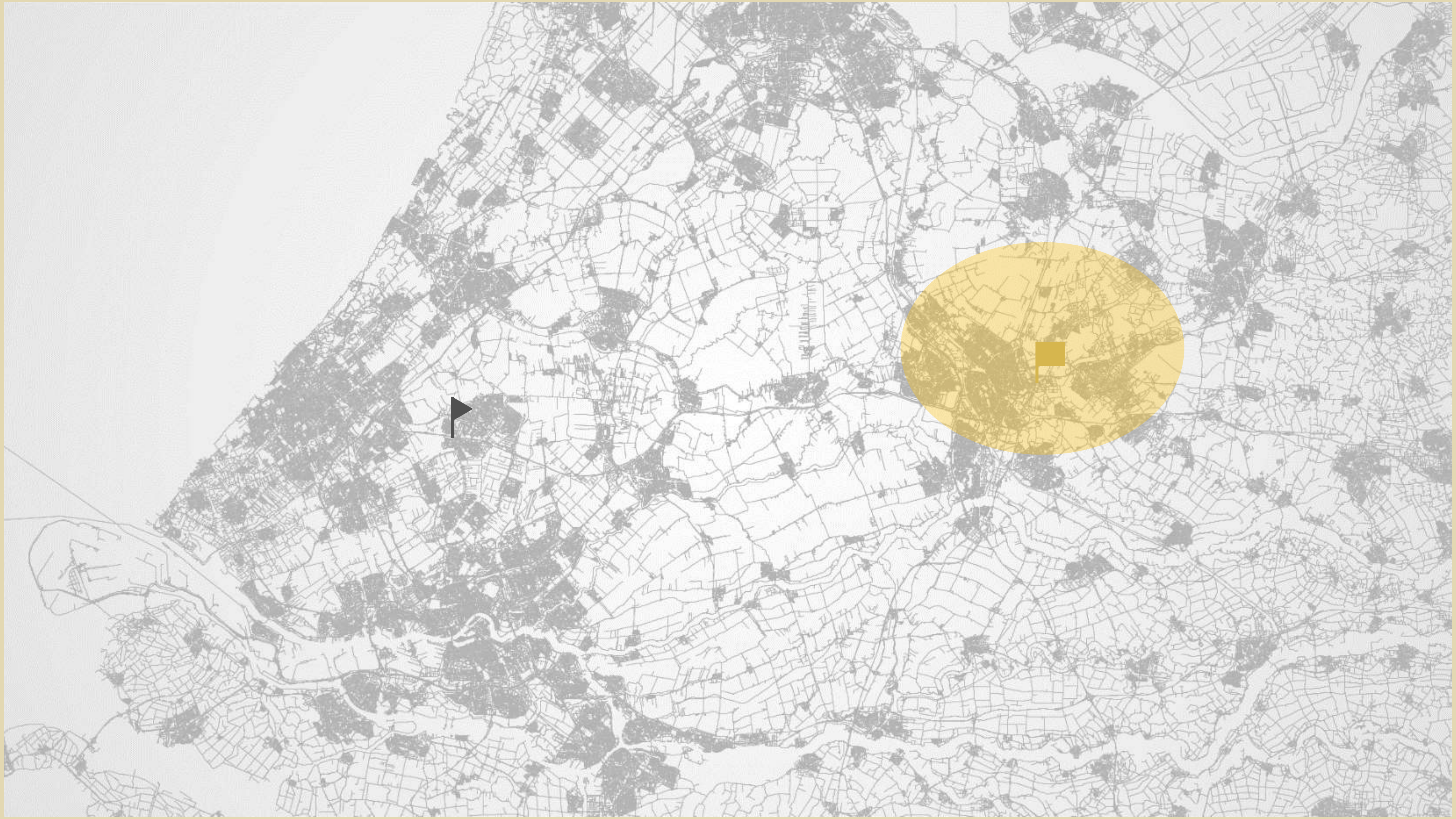
Is Dijkstra version 1959 all what we need?

We have detailed digital maps!

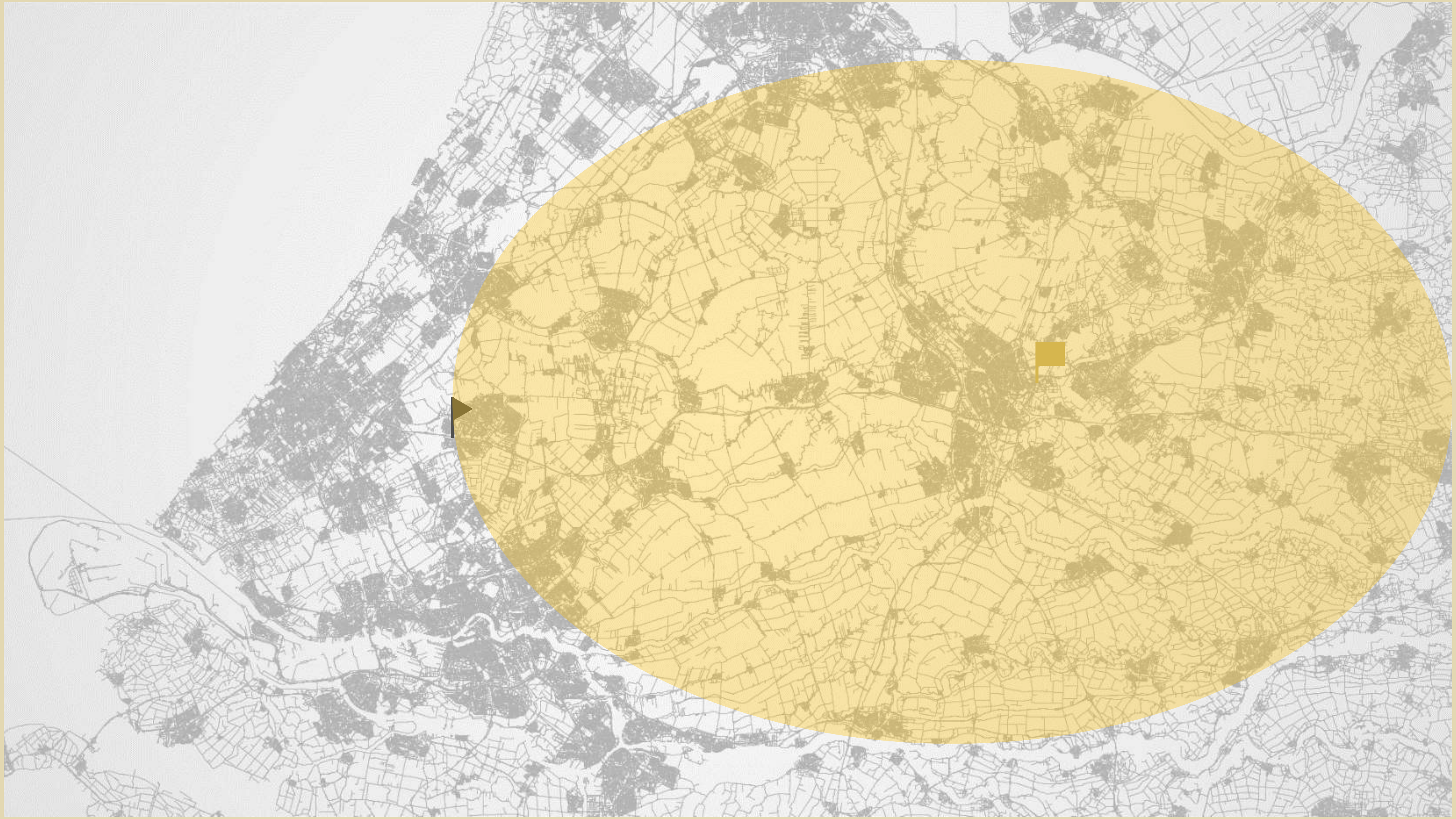
We know an efficient algorithm to compute shortest paths!

We must therefore be able to fill the catchment areas for a whole country very fast, right?

Let us recap how the method of Dijkstra works…



Analytics for a Better World

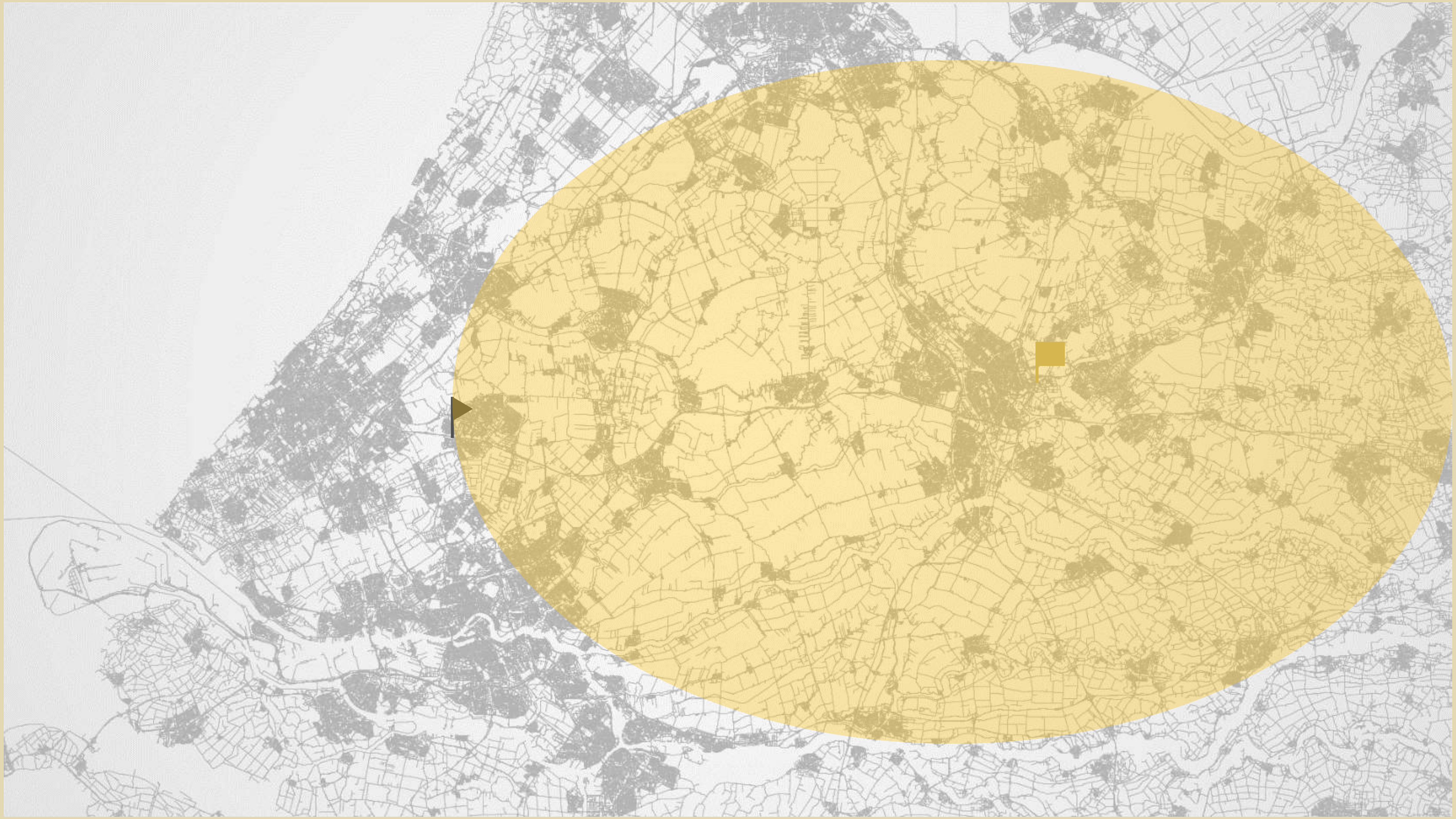


Analytics for a Better World

Challenges

- Unlike a GPS navigation system, we need the distance and travel times for **many** origin-destination pairs at once!
- These number grows with the square of the number of individual locations.
- Suppose that each point-to-point costs a single second:
- Only after having these figures are computed, we can actually start looking for optimal locations!!!

n	n times n	Filling all cells
10	100	0:01:40
50	2’500	0:41:40
100	10’000	2:46:40
1’000	1’000’000	11 days 13:46:40
5’000	25’000’000	289 days 8:26:40



One to many is as expensive as one to one

n	n times n	Cell by cell	One row and column at the time
10	100	0:01:40	0:00:20
50	2500	0:41:40	0:01:40
100	10000	2:46:40	0:03:20
1000	1000000	11 days 13:46:40	0:33:20
5000	25000000	289 days 8:26:40	2:46:40

Above Google on SciAm50 in 2008

- **The Fastest Way to Get There**
- Novel ways of calculating routes and predicting traffic jams promise less time in the car
- By [Peter Sergo]
- Providing directions instantly online has until recently meant that navigational mapping programs, such as MapQuest and Google Maps, often simplify the problem by not considering every possible route to a destination. Scientists at the University of Karlsruhe in Germany have designed a computer application that can quickly calculate the most expedient of all possible driving routes without the need for excessive computation.
- **Dominik Schultes**, one of the project's scientists, designed the program around a simple premise: driving somewhere usually requires crossing major intersections that are sparsely interconnected. Figuring the best route occurs by precomputing the connections between a starting point (or destination) and its nearest major intersections and between all locations where major routes cross each other's paths—so-called transit nodes.

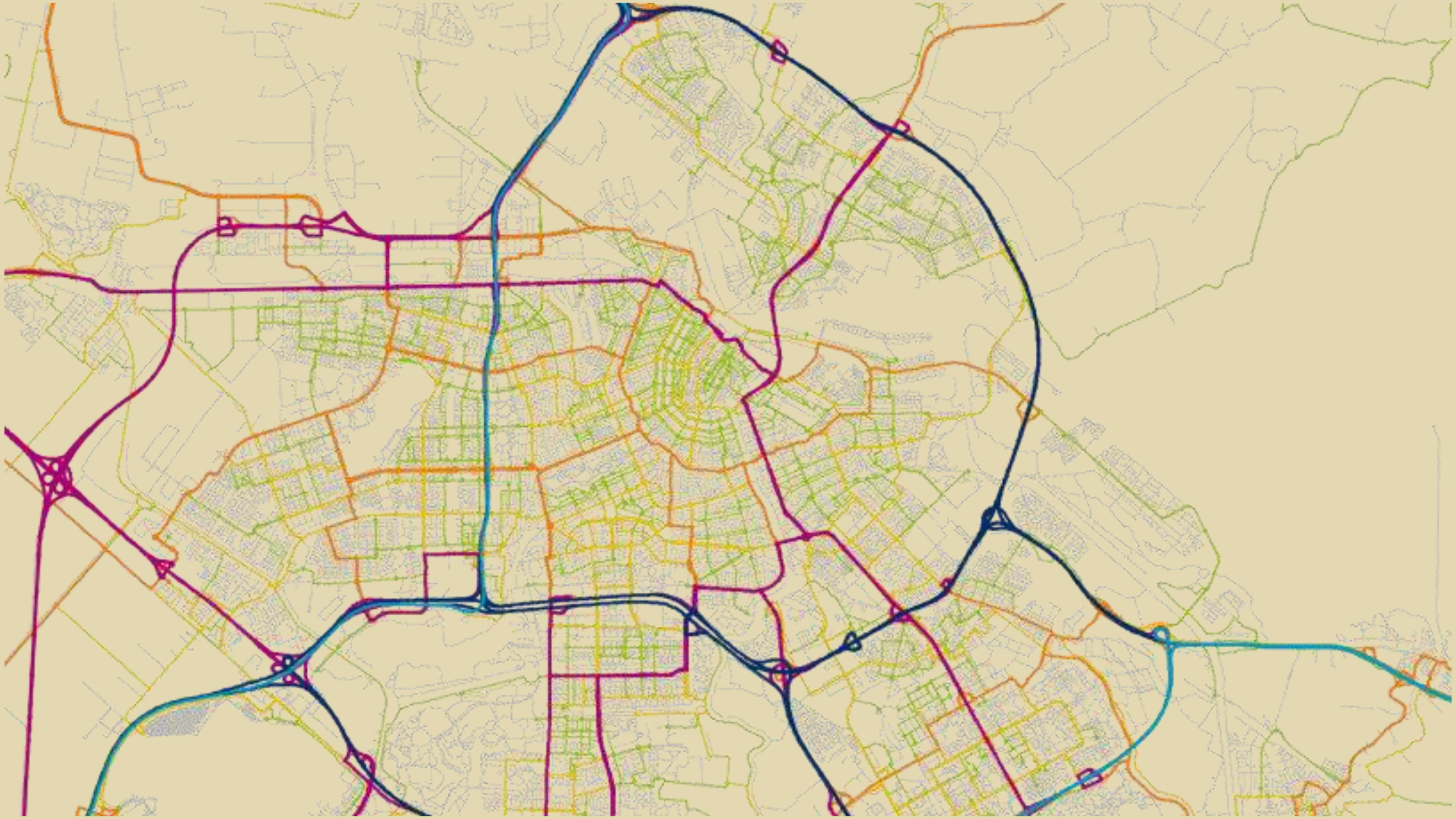
Amazing query times!

n	n times n	HNR
10	100	0:00:01
50	2500	0:00:08
100	10000	0:00:13
1000	1000000	0:01:05
5000	25000000	0:03:13

Amsterdam

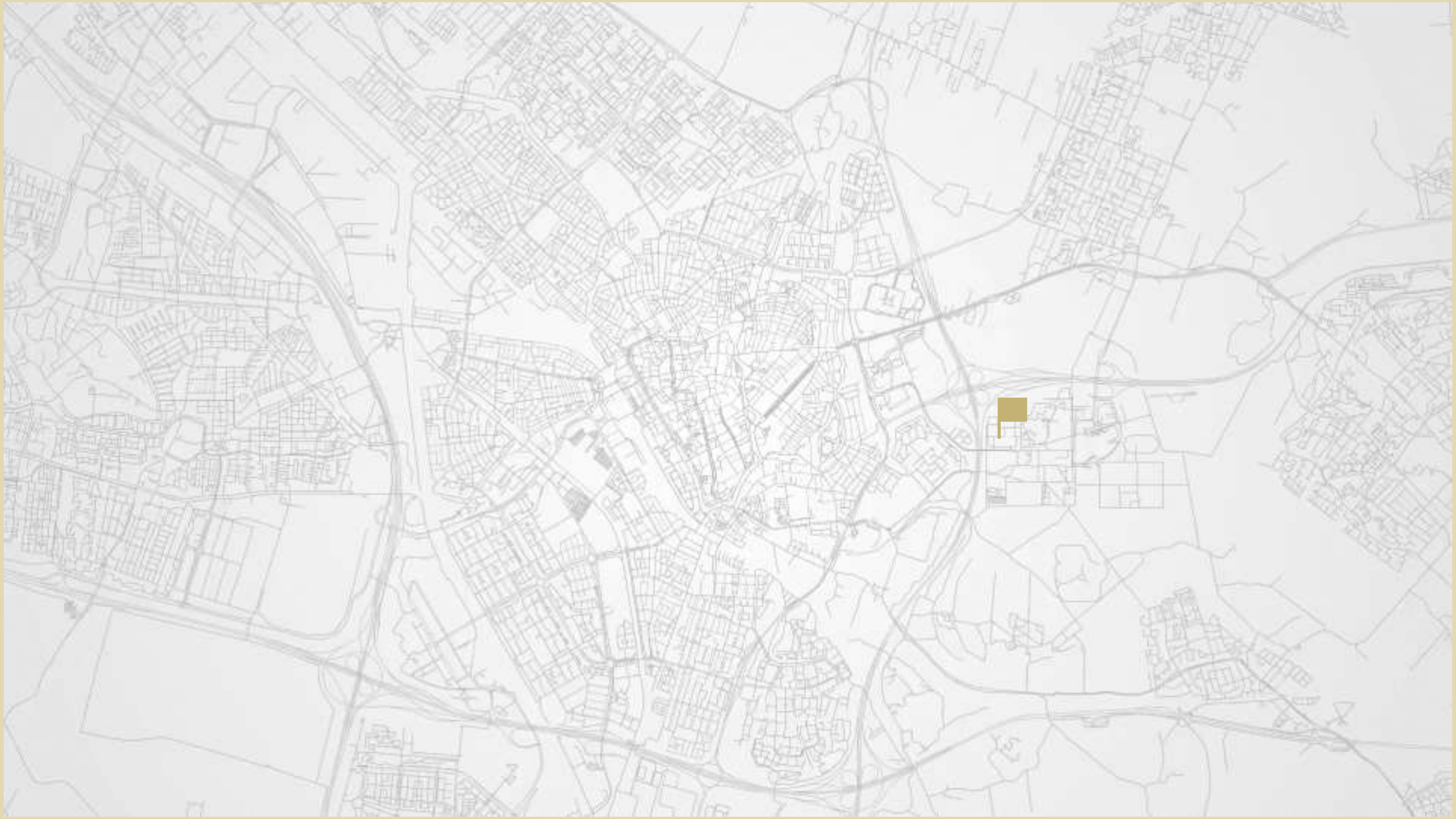
Analytics for a Better World

Amsterdam’s HNR



Amsterdam’s administrative

Analytics for a Better World



Even better: contraction hierarchies!

- Brought to the Python ecosystem as pandana.
- Used by PISA when we must create and manage our own road network, as we do for example in Timor-Leste.
- Illustrated in this notebook.

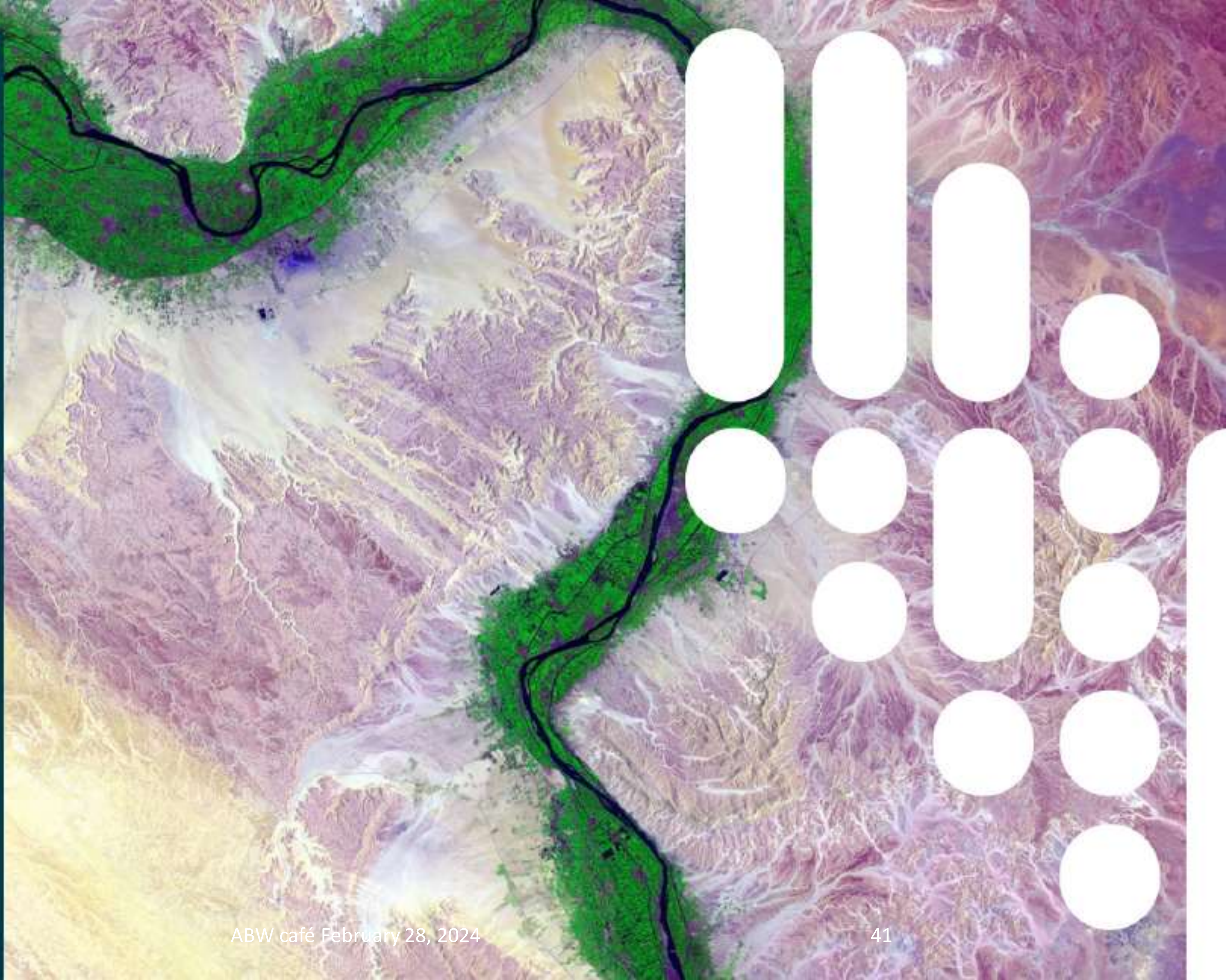
When professionals suffice…

When quality is sufficient, we leverage on the services of, for instance, MapBox.

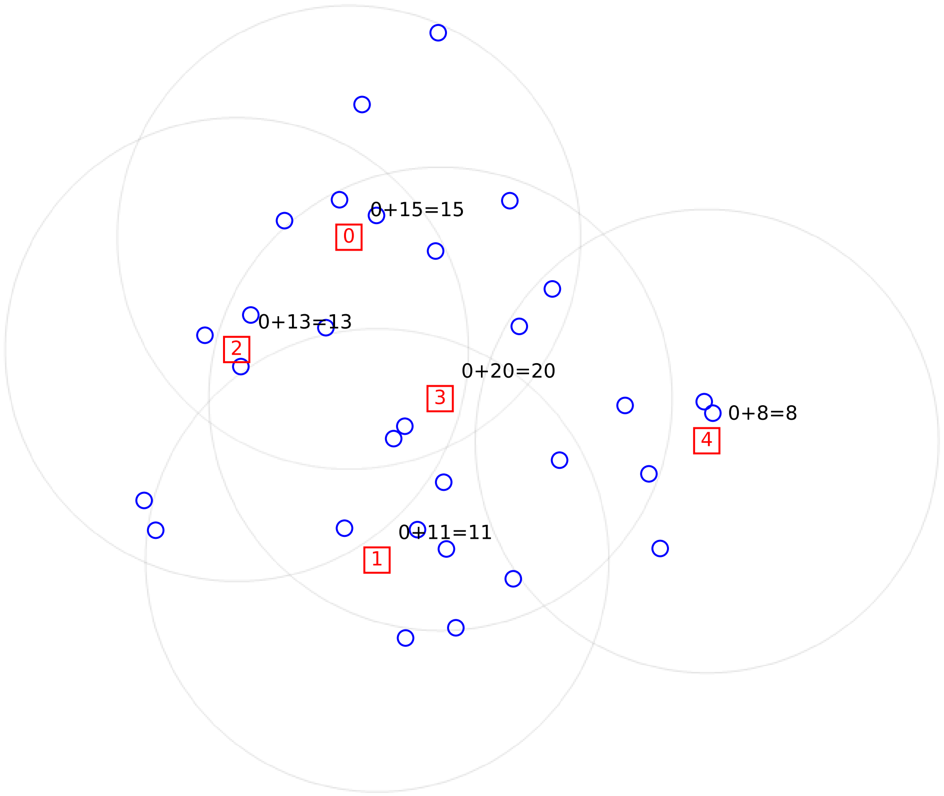
We have access metrics!

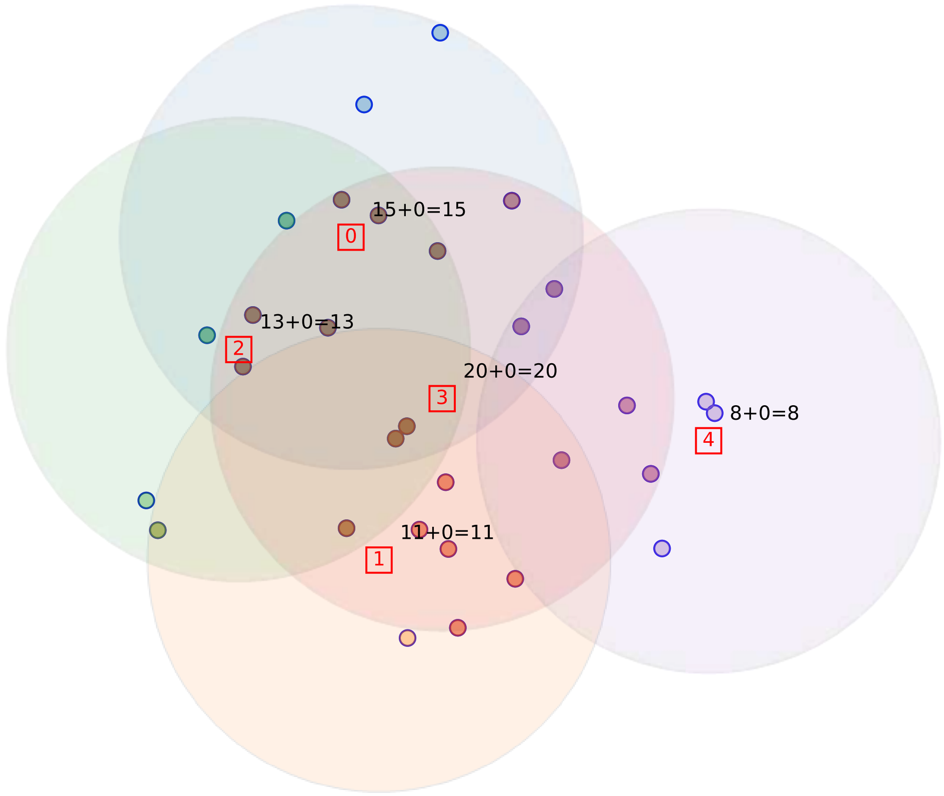
Now what?

Analytics for a Better World

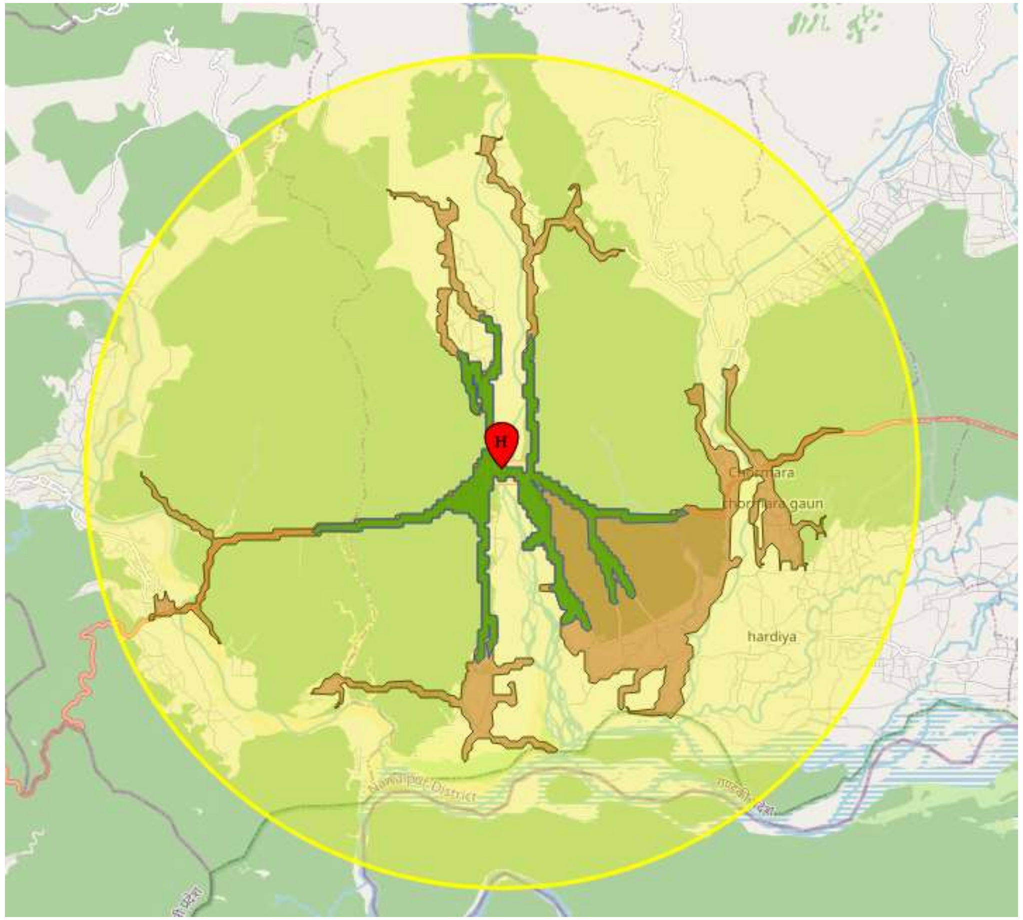


Some random instance



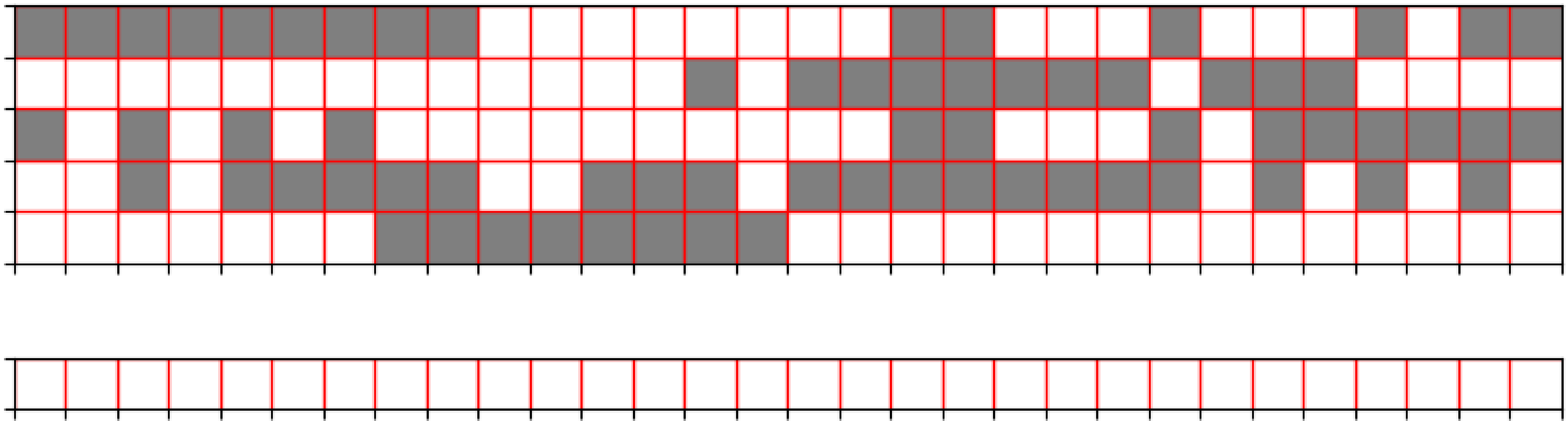


In reality

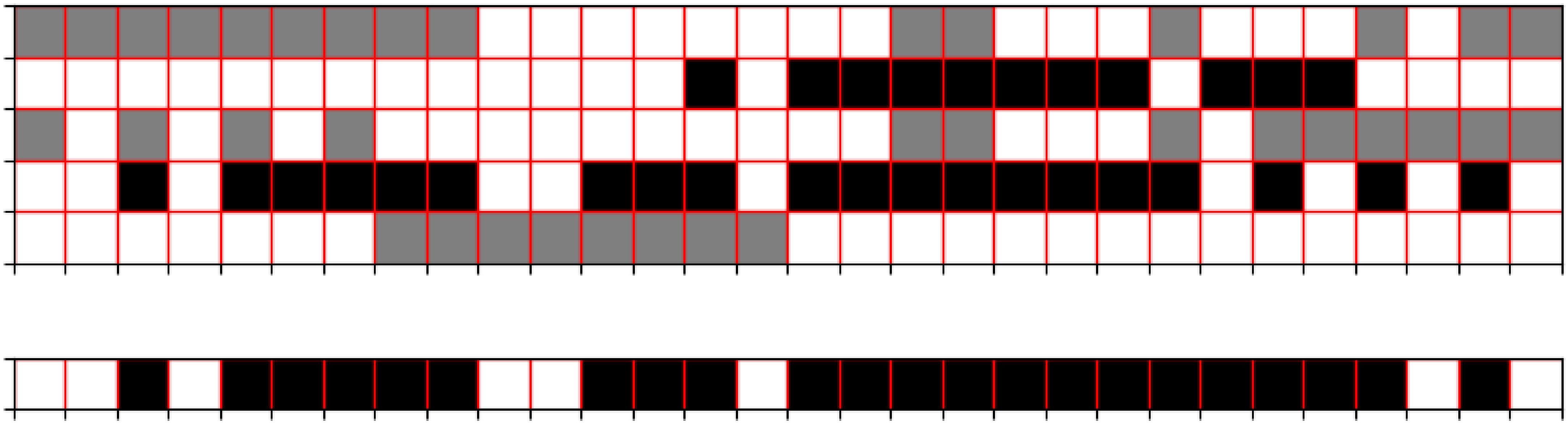


Analytics for a Better World

Many ways to look at this puzzle



Many ways to look at this puzzle



Analytics for a Better World

In all cases, the relevant data is just…

assuming constant unitary headcounts

0: [0, 1, 2, 3, 4, 5, 6, 7, 8, 17, 18, 22, 26, 28, 29]

1: [13, 15, 16, 17, 18, 19, 20, 21, 23, 24, 25]

2: [0, 2, 4, 6, 17, 18, 22, 24, 25, 26, 27, 28, 29]

3: [2, 4, 5, 6, 7, 8, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 24, 26, 28]

4: [7, 8, 9, 10, 11, 12, 13, 14]

Reverting the logic

0: [0, 2]
1: [0]
2: [0, 2, 3]
3: [0]
4: [0, 2, 3]
5: [0, 3]
6: [0, 2, 3]
7: [0, 3, 4]
8: [0, 3, 4]
9: [4]
10: [4]
...
27: [2]
28: [0, 2, 3]
29: [0, 2]

Analytics for a Better World

Time has come to revert you to an exercise…

- On this book (mentioned at the beginning) you may enjoy reading Example 3.10 about location models, complemented by this notebook.
- However, Exercise 3.1 (solved at the end of the book) concerns the model that is in the heart of PISA’s maximal covering optimizations!
- Again, all started with a classic!

There is a way to express such puzzles in ways enable solvers to find optimal solutions!

$$\max \quad \sum_{i \in I} w_i y_i \tag{1}$$

$$\text{subject to} \quad y_i \leq \sum_{j \in J_i} x_j \qquad \forall i \in I \tag{2}$$

$$\sum_{j \in J} x_j \leq b \tag{3}$$

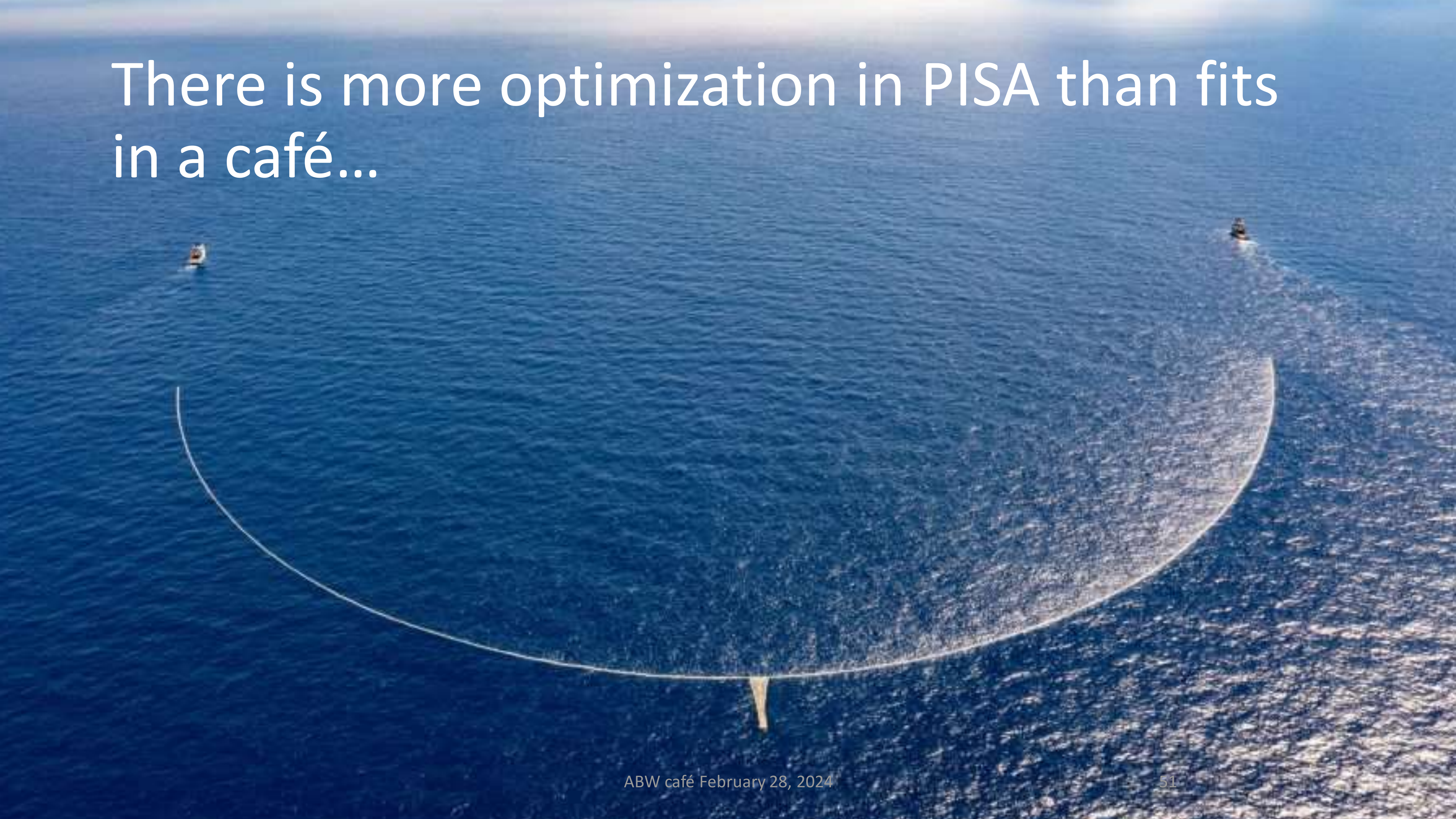
$$x_j \in \{0, 1\} \qquad \forall j \in J \tag{4}$$

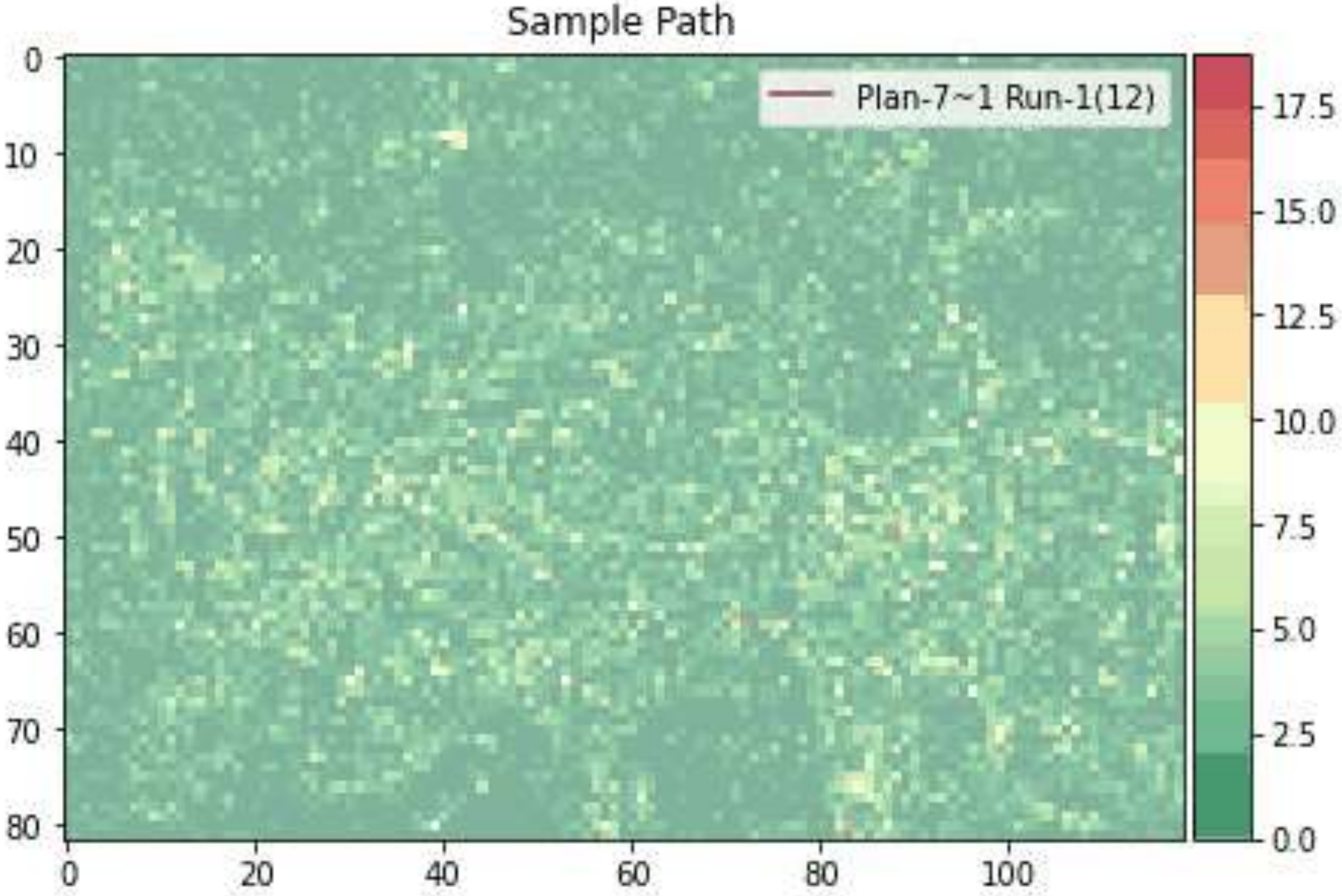
$$y_i \in \{0, 1\} \qquad \forall i \in I \tag{5}$$

The state of the art of solvers

- Gurobi is able to solve instances with many thousands households and potential facilities in minutes to optimality!
- Handling the data requires and instancing the model requires some careful coding.
- Beyond that: heuristics! But that we must reserve for another edition!

There is more optimization in PISA than fits in a café…





Sample Path
Plan-7~1 Run-1(12)
0
10
20
30
40
50
60
70
80
0
20
40
60
80
100
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0

How did they do that?

- If you are interested on how the locations to be at are optimized in those fascinating routes, consider reading this paper.

- There you will be introduced to yet another powerful optimization paradigm: Dynamic Programming!

- There are many great references, I like this very well written PhD thesis.


THANK YOU

Analytics for a Better World