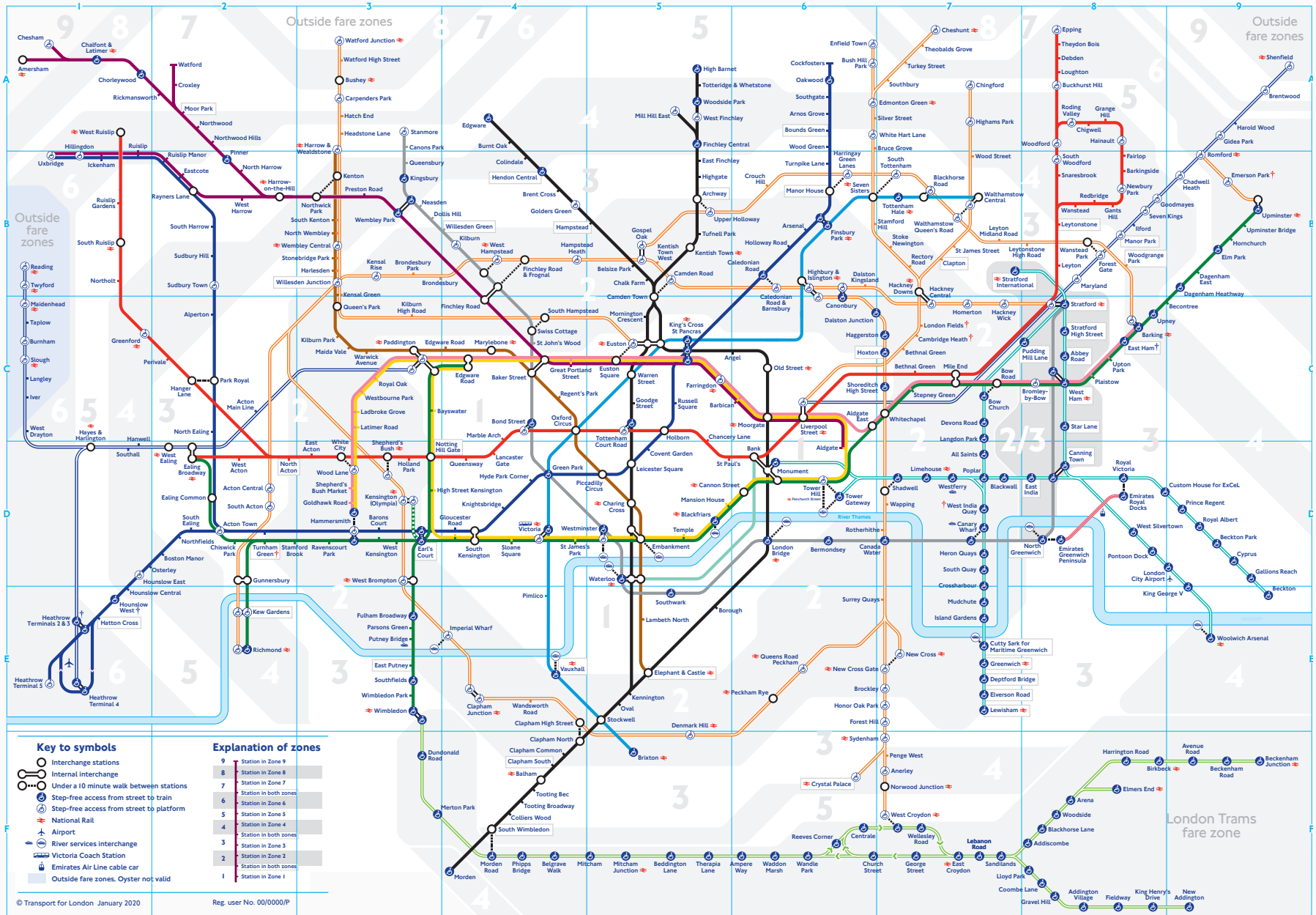
Metro Maps on Octilinear Grid Graphs

Hannah Bast [1], Patrick Brosi [1] and Sabine Storandt [2]

[1] University of Freiburg
[2] University of Konstanz

EuroVis 2020 - Norrköping, Sweden

Motivation - Official London Tube Map

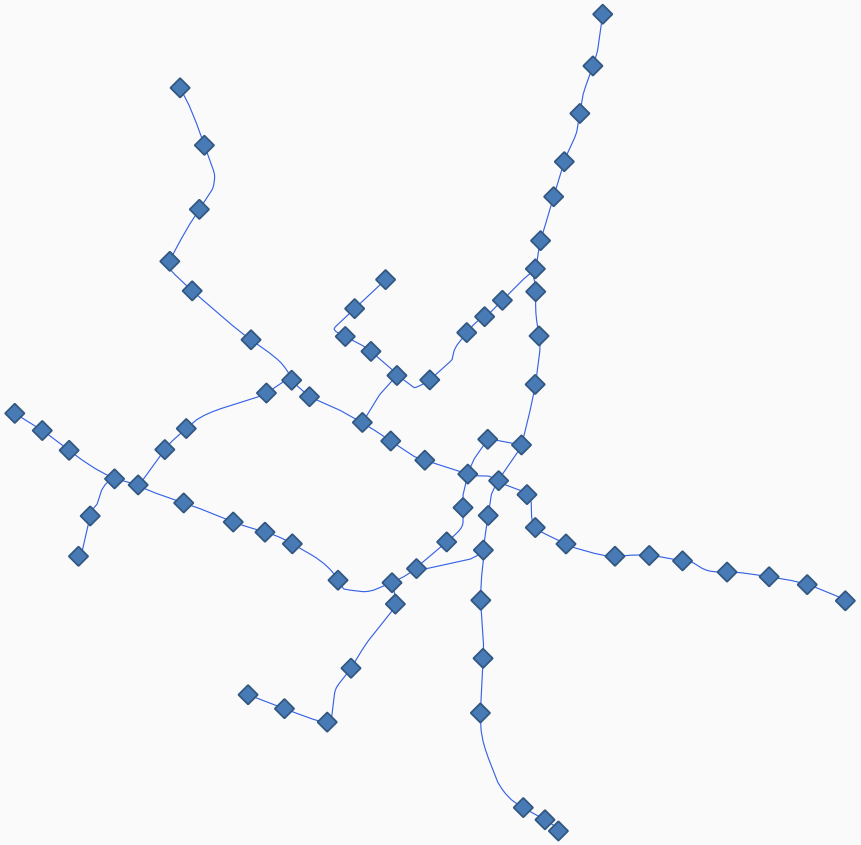


Goals

Given an input line graph $G = (V, E, L)$ with edge lines $L(e)$, render an **octilinear** drawing of G automatically and fast

Goals

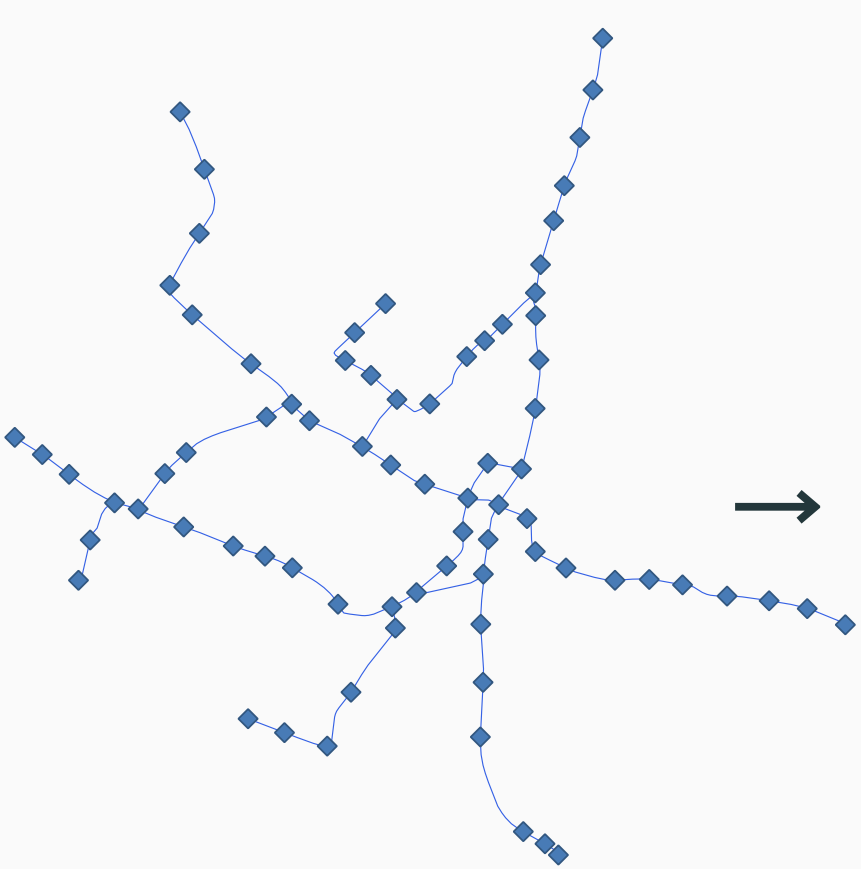
Given an input line graph $G = (V, E, L)$ with edge lines $L(e)$, render an **octilinear** drawing of G automatically and fast



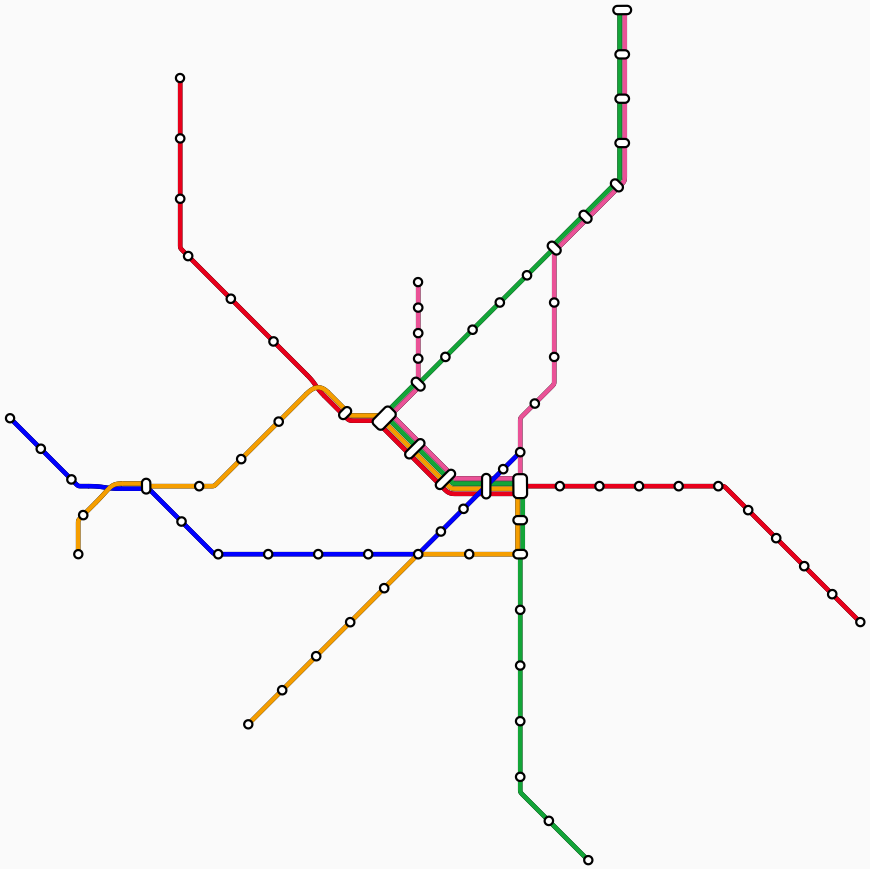
Input line graph G

Goals

Given an input line graph $G = (V, E, L)$ with edge lines $L(e)$, render an **octilinear** drawing of G automatically and fast



Input line graph G



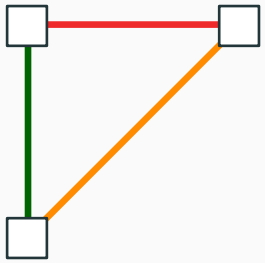
Octilinear drawing of G

Goals (ctd.)

Allow arbitrary (but optimal) number of edge bends to circumvent obstacles and approximate geographical courses

Goals (ctd.)

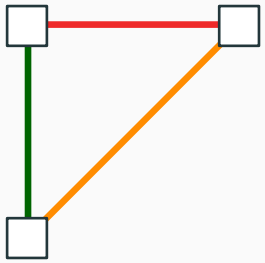
Allow arbitrary (but optimal) number of edge bends to circumvent obstacles and approximate geographical courses



Octilinear embedding

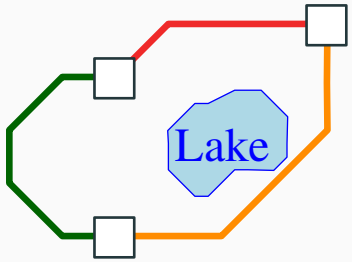
Goals (ctd.)

Allow arbitrary (but optimal) number of edge bends to circumvent obstacles and approximate geographical courses



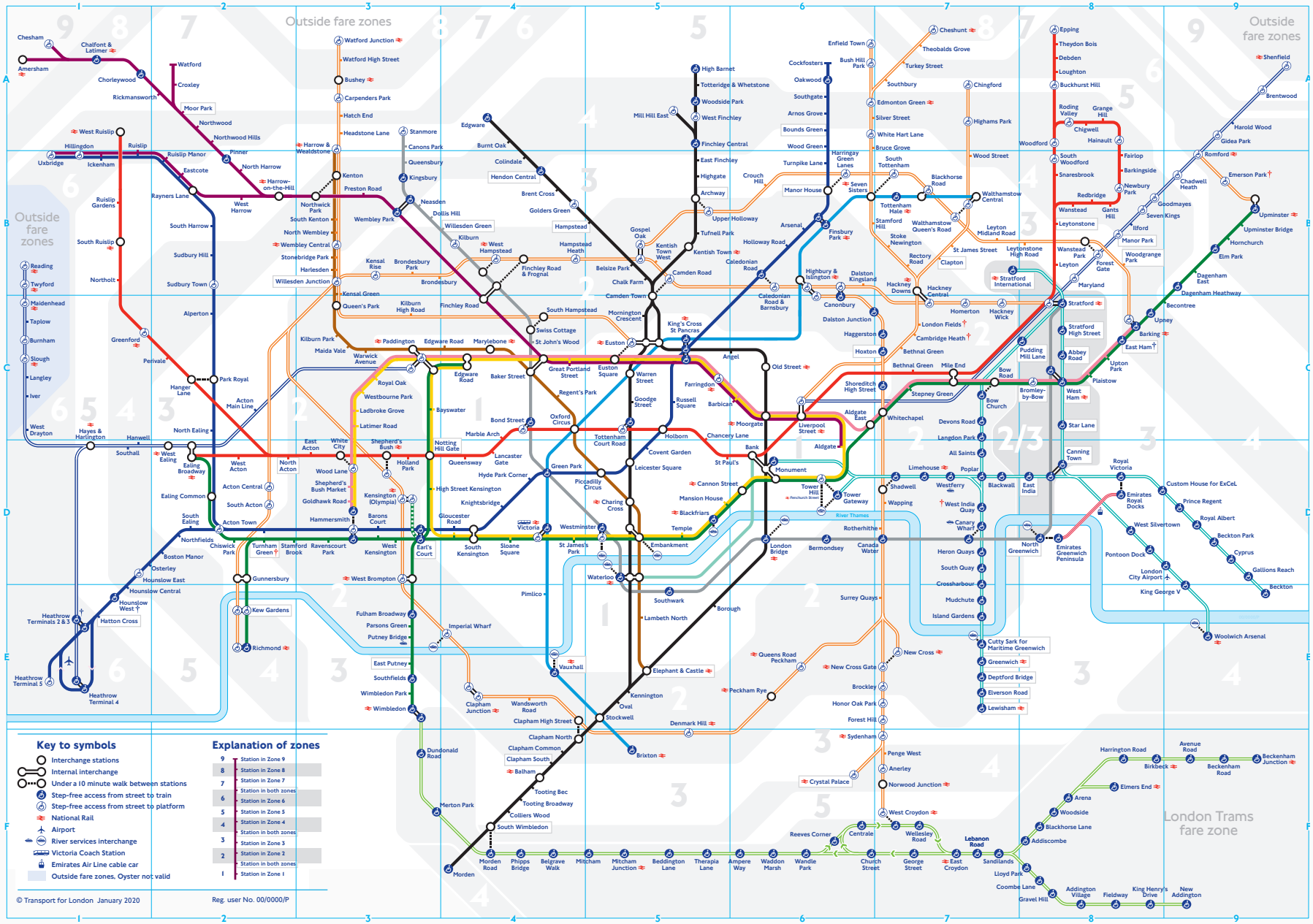
Octilinear embedding

vs.



Octilinear drawing

Octilinear Embedding vs. Octilinear Drawing



Octilinear Embedding vs. Octilinear Drawing

Excerpt of regional train map of southwestern Germany

Existing Approaches

1. Local search / Hill climbing (Stott 2004)

Existing Approaches

1. Local search / Hill climbing (Stott 2004)
2. Linear Programming (Nöllenburg, Wolff, 2005)

Existing Approaches

1. Local search / Hill climbing (Stott 2004)
2. Linear Programming (Nöllenburg, Wolff, 2005)
3. Least Squares Optimization (Wang, Chi, 2011)

Existing Approaches

1. Local search / Hill climbing (Stott 2004)
2. Linear Programming (Nöllenburg, Wolff, 2005)
3. Least Squares Optimization (Wang, Chi, 2011)

Existing Approaches

1. Local search / Hill climbing (Stott 2004)
2. Linear Programming (Nöllenburg, Wolff, 2005)
3. Least Squares Optimization (Wang, Chi, 2011)

Problems:

- Octilinearity not guaranteed [1]

Existing Approaches

1. Local search / Hill climbing (Stott 2004)
2. Linear Programming (Nöllenburg, Wolff, 2005)
3. Least Squares Optimization (Wang, Chi, 2011)

Problems:

- Octilinearity not guaranteed [1]
- Produce octilinear embeddings, so no bends along edges (can be mitigated by adding explicit bend nodes) [1, 2, 3]

Existing Approaches

1. Local search / Hill climbing (Stott 2004)
2. Linear Programming (Nöllenburg, Wolff, 2005)
3. Least Squares Optimization (Wang, Chi, 2011)

Problems:

- Octilinearity not guaranteed [1]
- Produce octilinear embeddings, so no bends along edges (can be mitigated by adding explicit bend nodes) [1, 2, 3]
- Are often too slow for practical purposes (on-demand maps, editors) [1, 2]

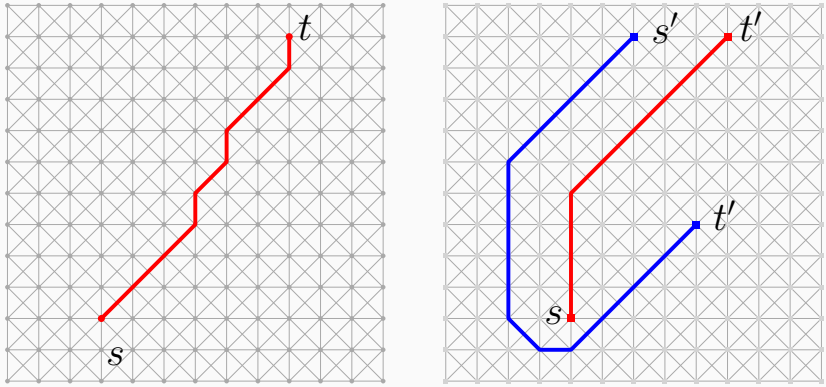
Our Approach

Basic idea: Formulate as a set of shortest path problems.

Our Approach

Basic idea: Formulate as a set of shortest path problems.

1. Build octilinear grid graph on which edge bends in paths are penalized

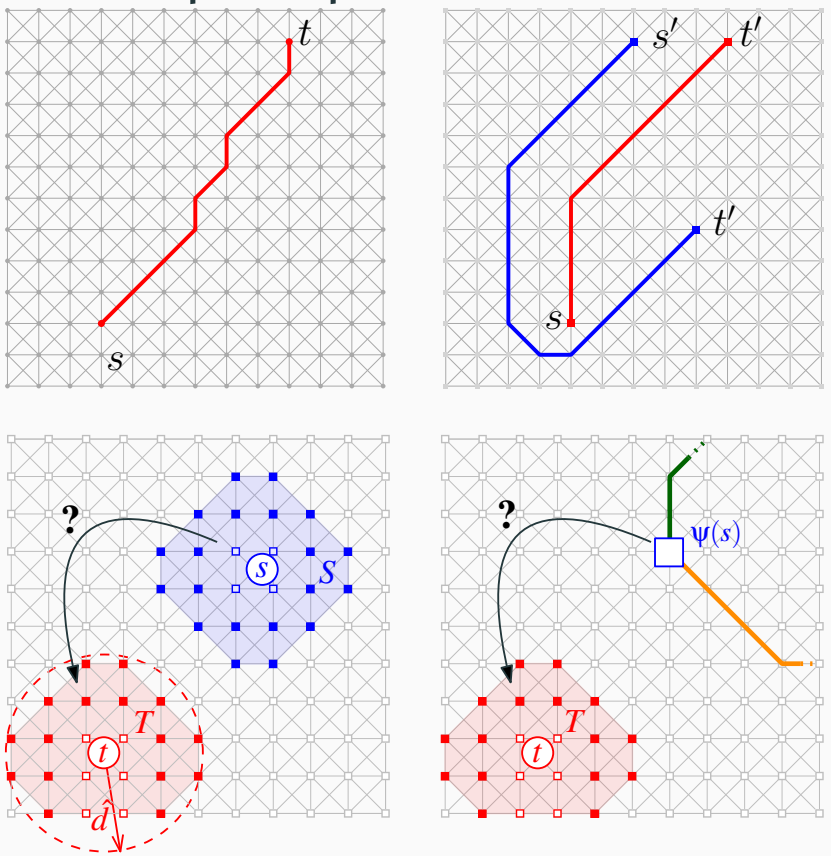


Our Approach

Basic idea: Formulate as a set of shortest path problems.

1. Build octilinear grid graph on which edge bends in paths are penalized

2. For each $e \in E$, $e = (s, t)$: define start and target grid nodes S, T and find the shortest path from S to T. Make path an obstacle.

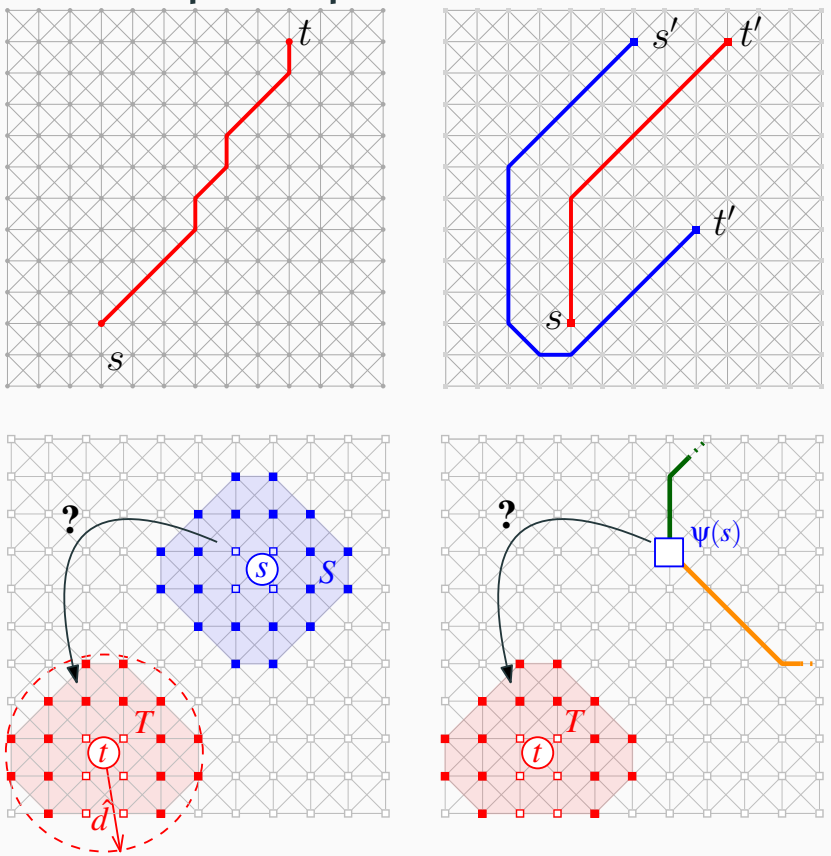


Our Approach

Basic idea: Formulate as a set of shortest path problems.

1. Build octilinear grid graph on which edge bends in paths are penalized

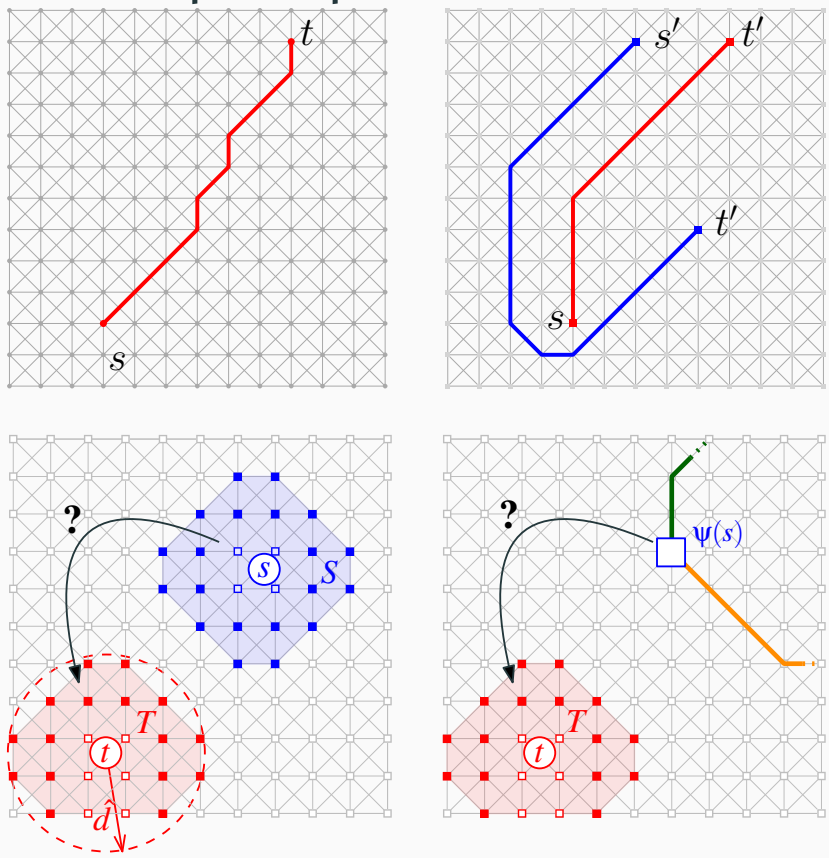
2. For each $e \in E$, $e = (s, t)$: define start and target grid nodes S, T and find the shortest path from S to T. Make path an obstacle.



Our Approach

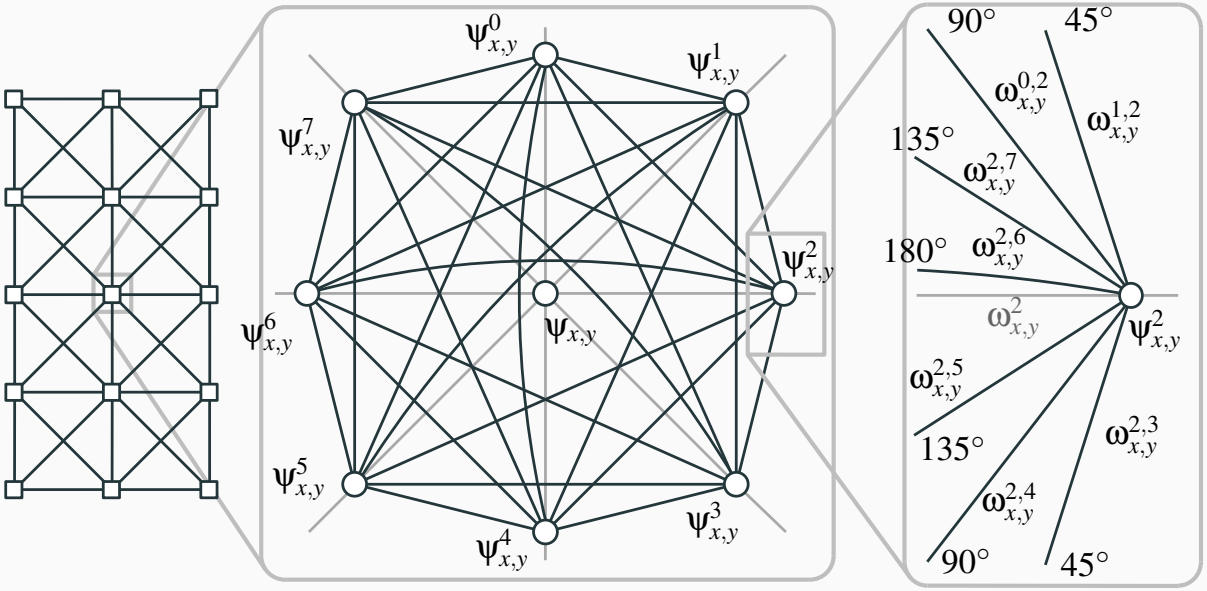
Basic idea: Formulate as a set of shortest path problems.

1. Build octilinear grid graph on which edge bends in paths are penalized
2. For each $e \in E$, $e = (s, t)$: define start and target grid nodes S, T and find the shortest path from S to T. Make path an obstacle.

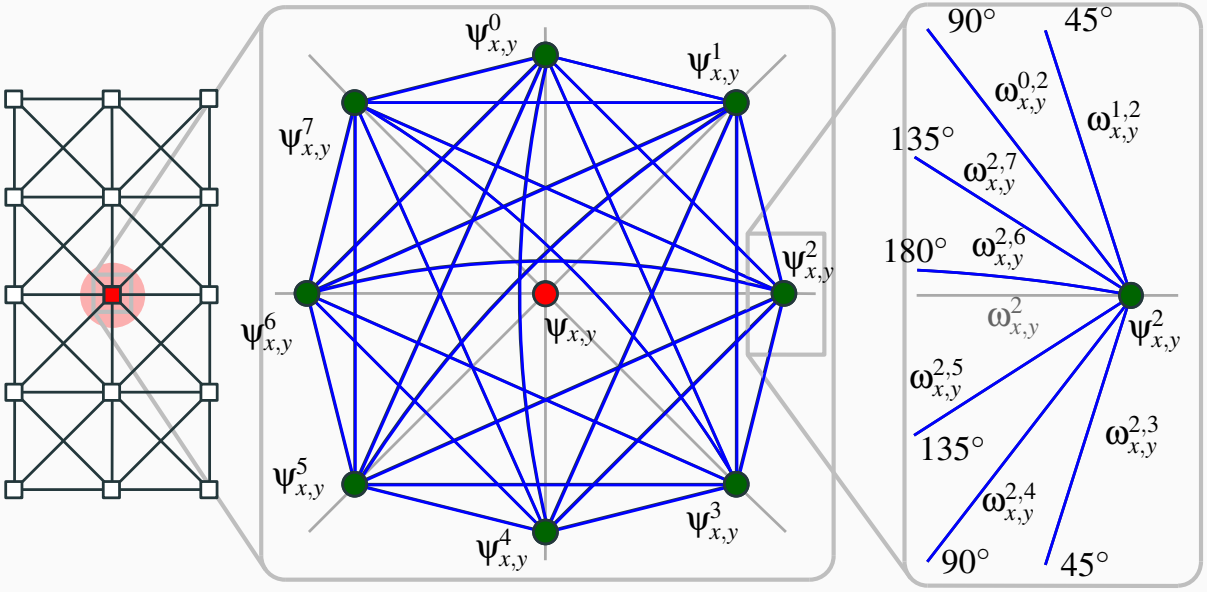


Key question: Should the shortest paths be determined simultaneously or iteratively (then: in which order)?

Graph Modeling - Detail

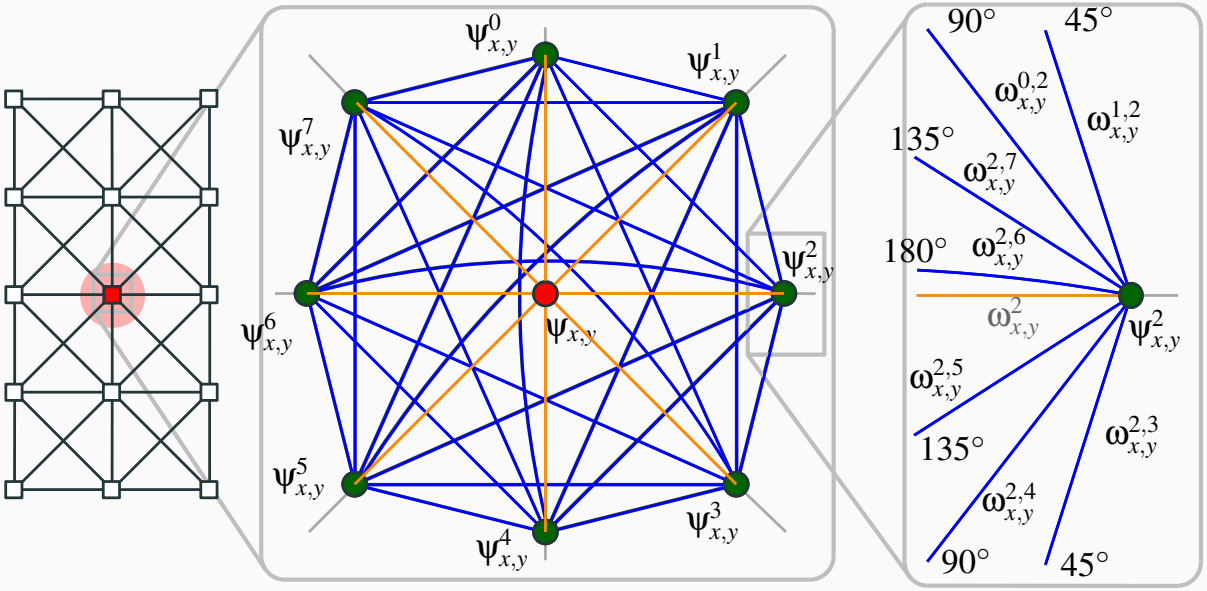


Graph Modeling - Detail



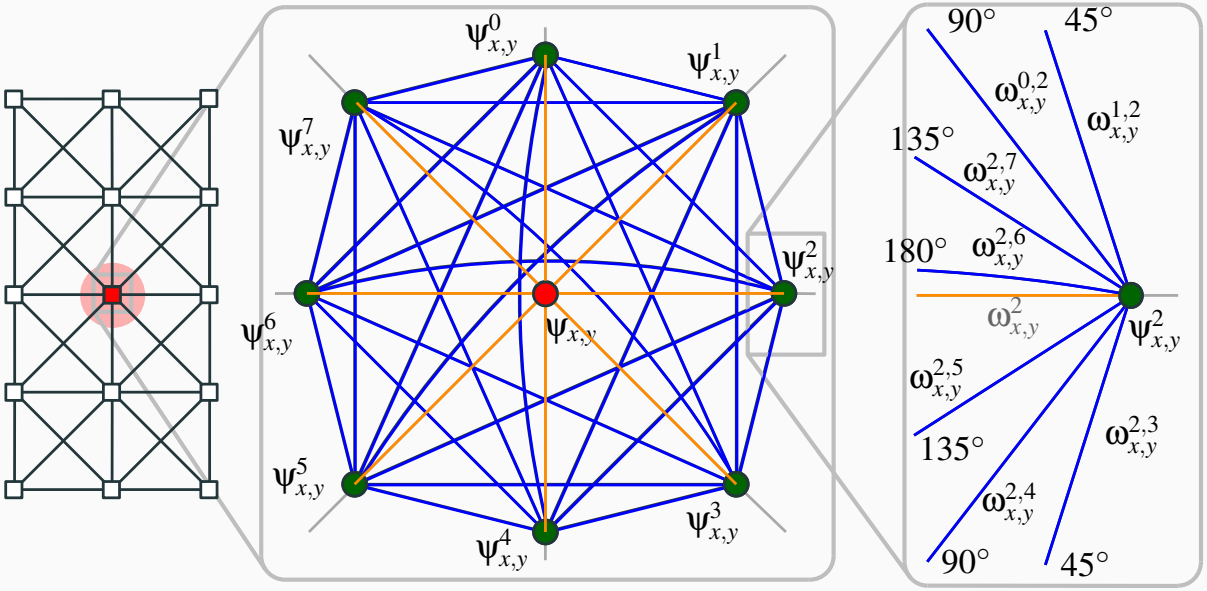
- Each grid node is extended by explicit bend edges between port nodes. Edge costs reflect the angle

Graph Modeling - Detail



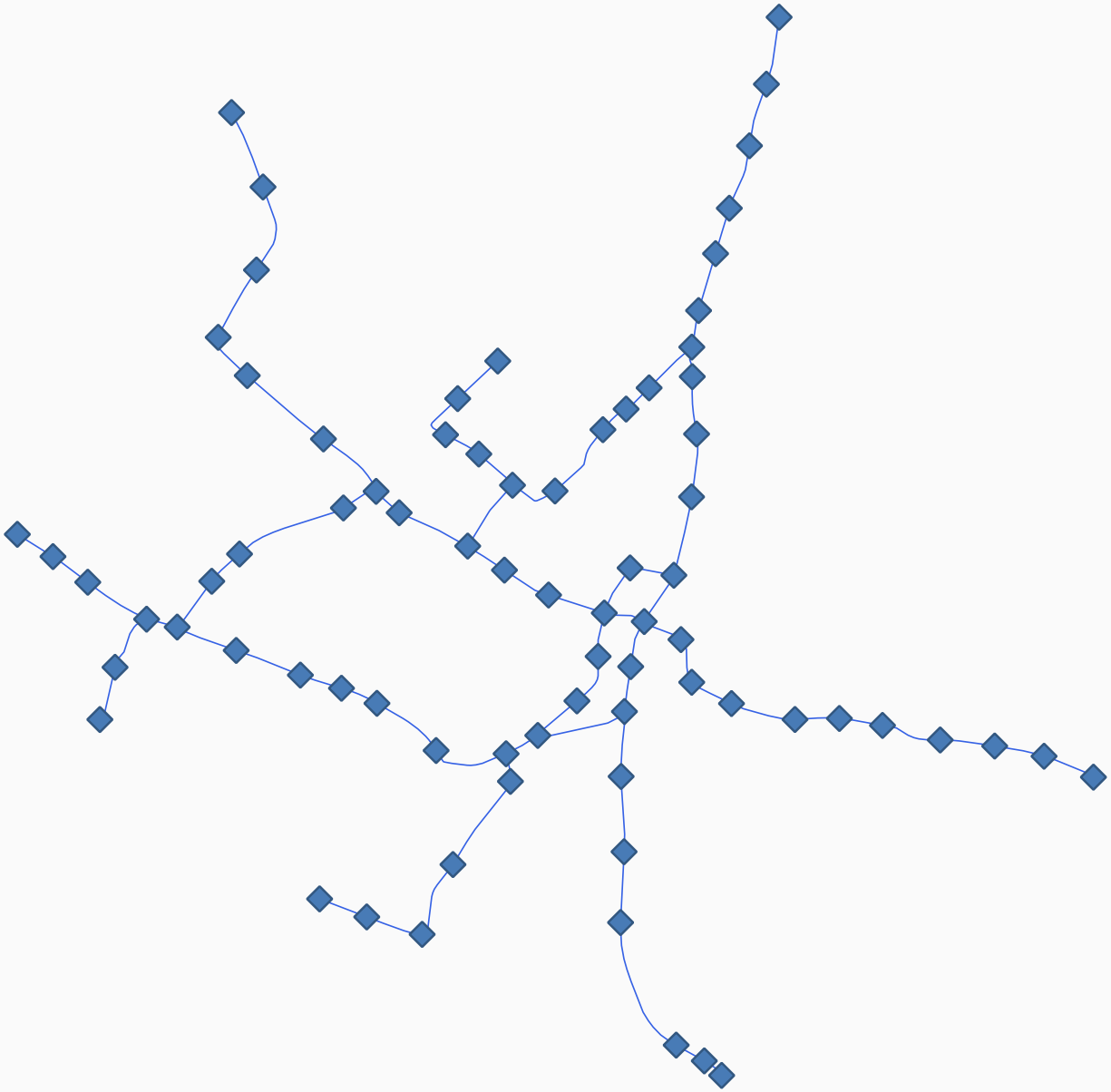
- Each grid node is extended by explicit bend edges between port nodes. Edge costs reflect the angle
- To arrive at a grid node, we add sink edges

Graph Modeling - Detail

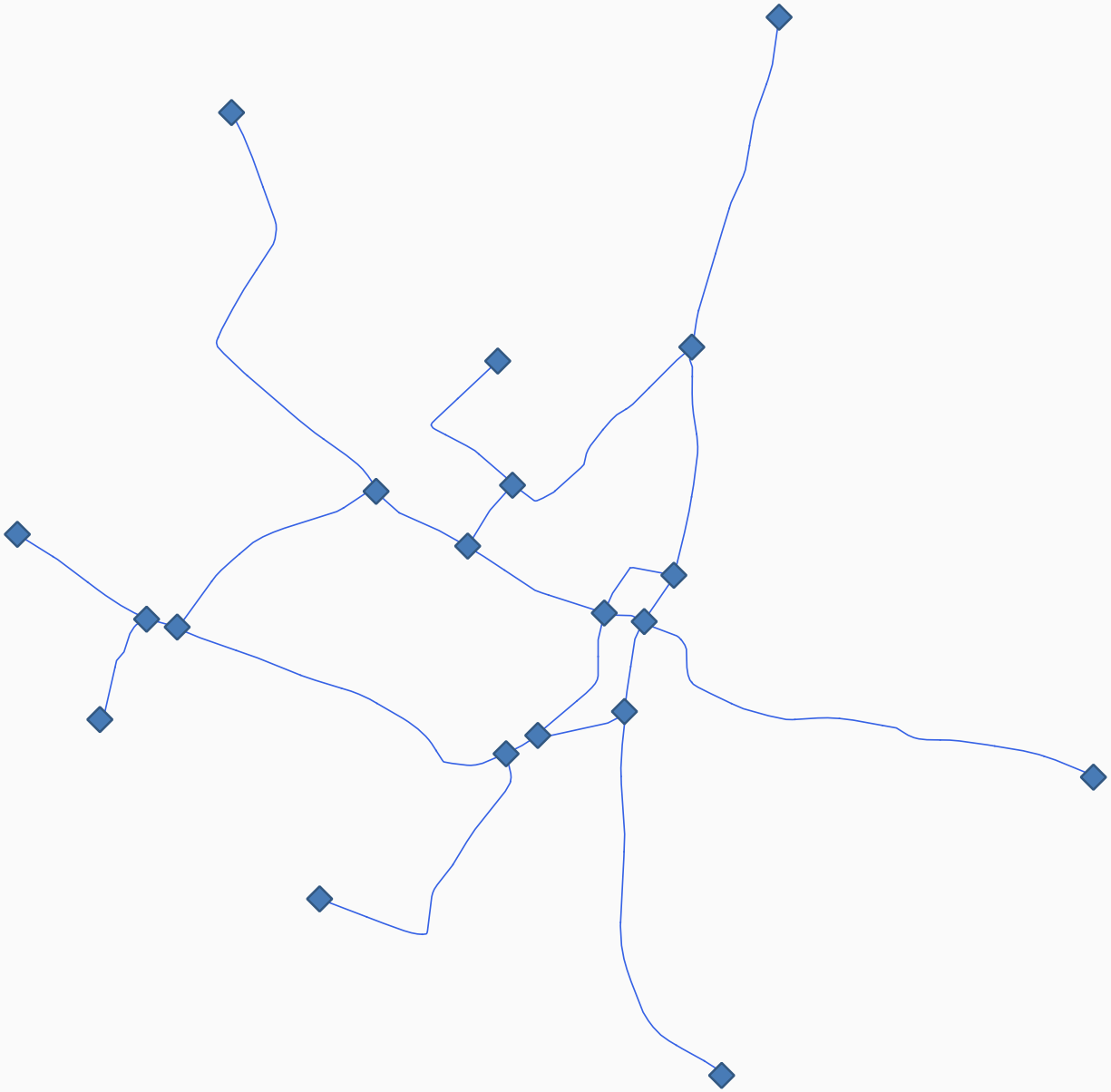


- Each grid node is extended by explicit bend edges between port nodes. Edge costs reflect the angle
- To arrive at a grid node, we add sink edges
- Sink edge costs reflect (1) the angle between the sink and continued lines on already settled sinks and (2) a node move penalty

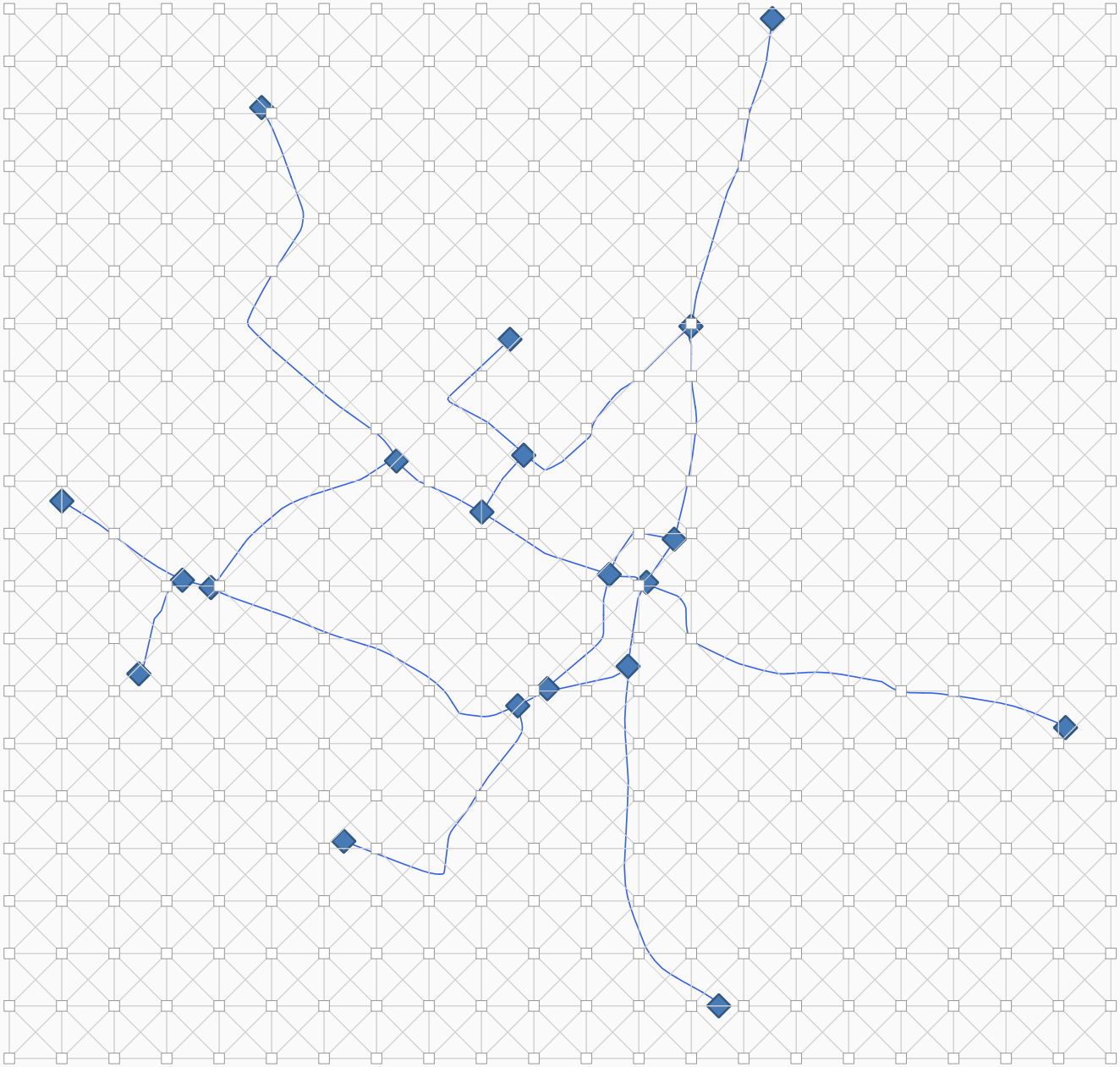
Iterative Calculation



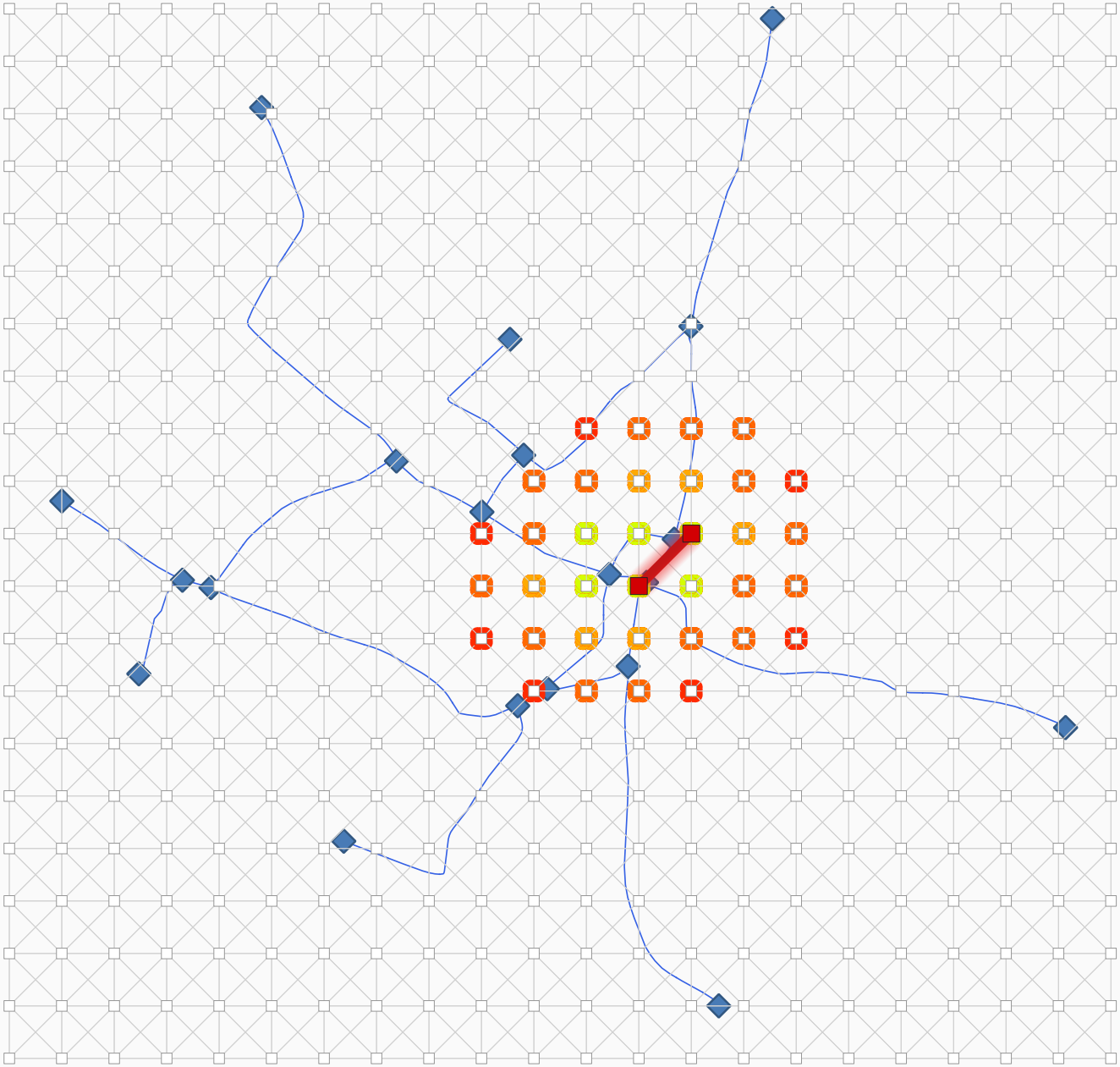
Iterative Calculation



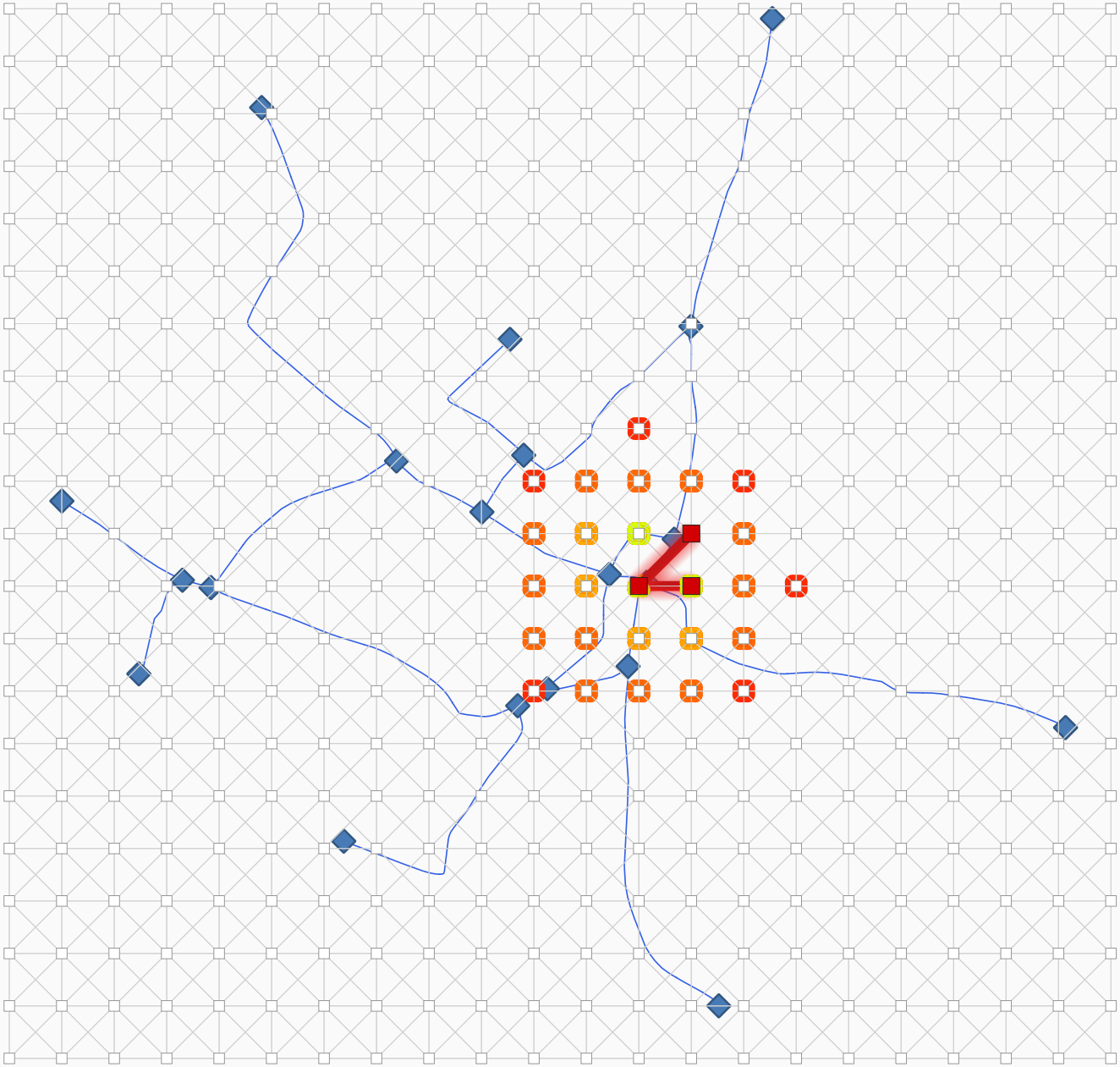
Iterative Calculation



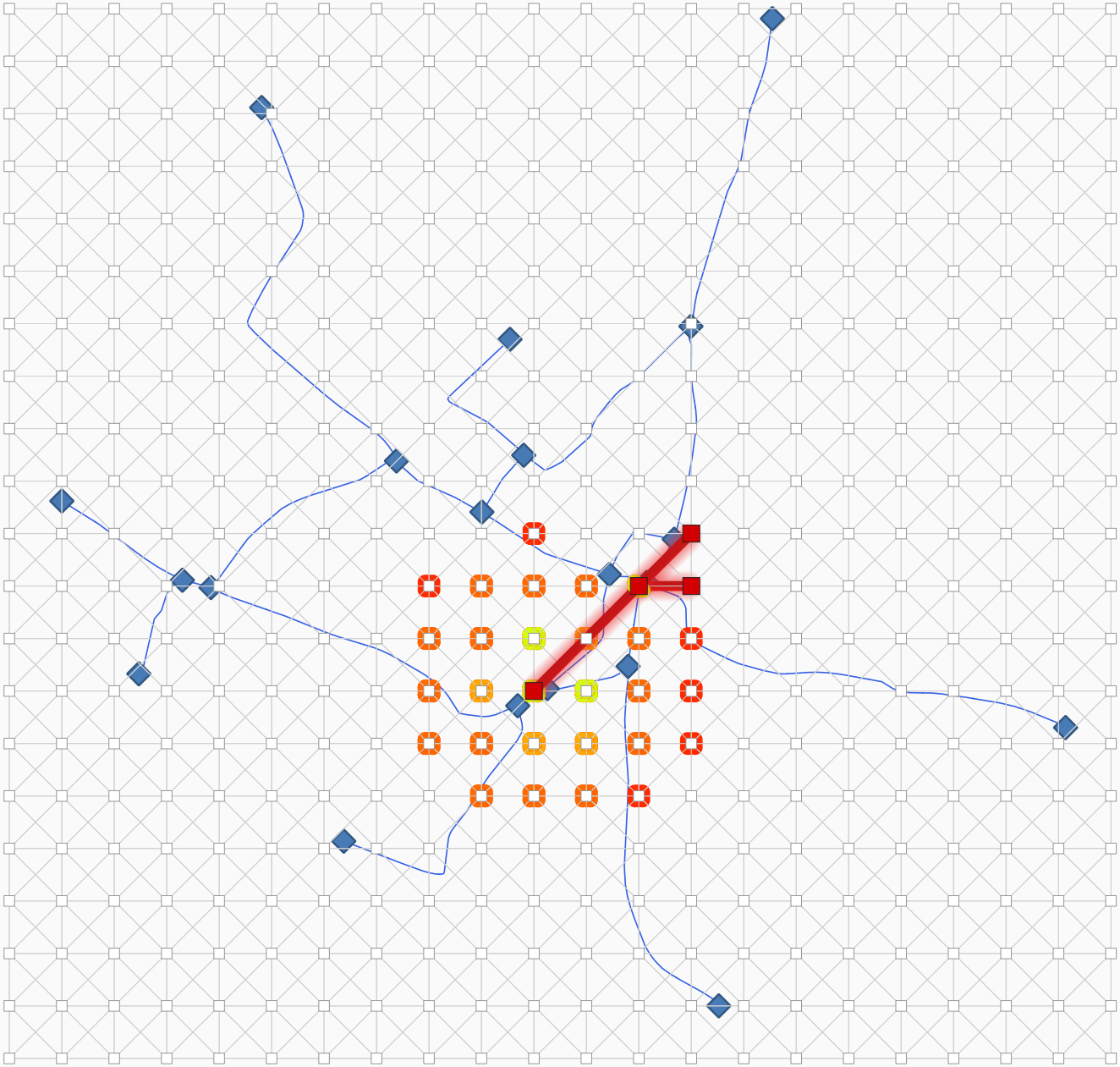
Iterative Calculation



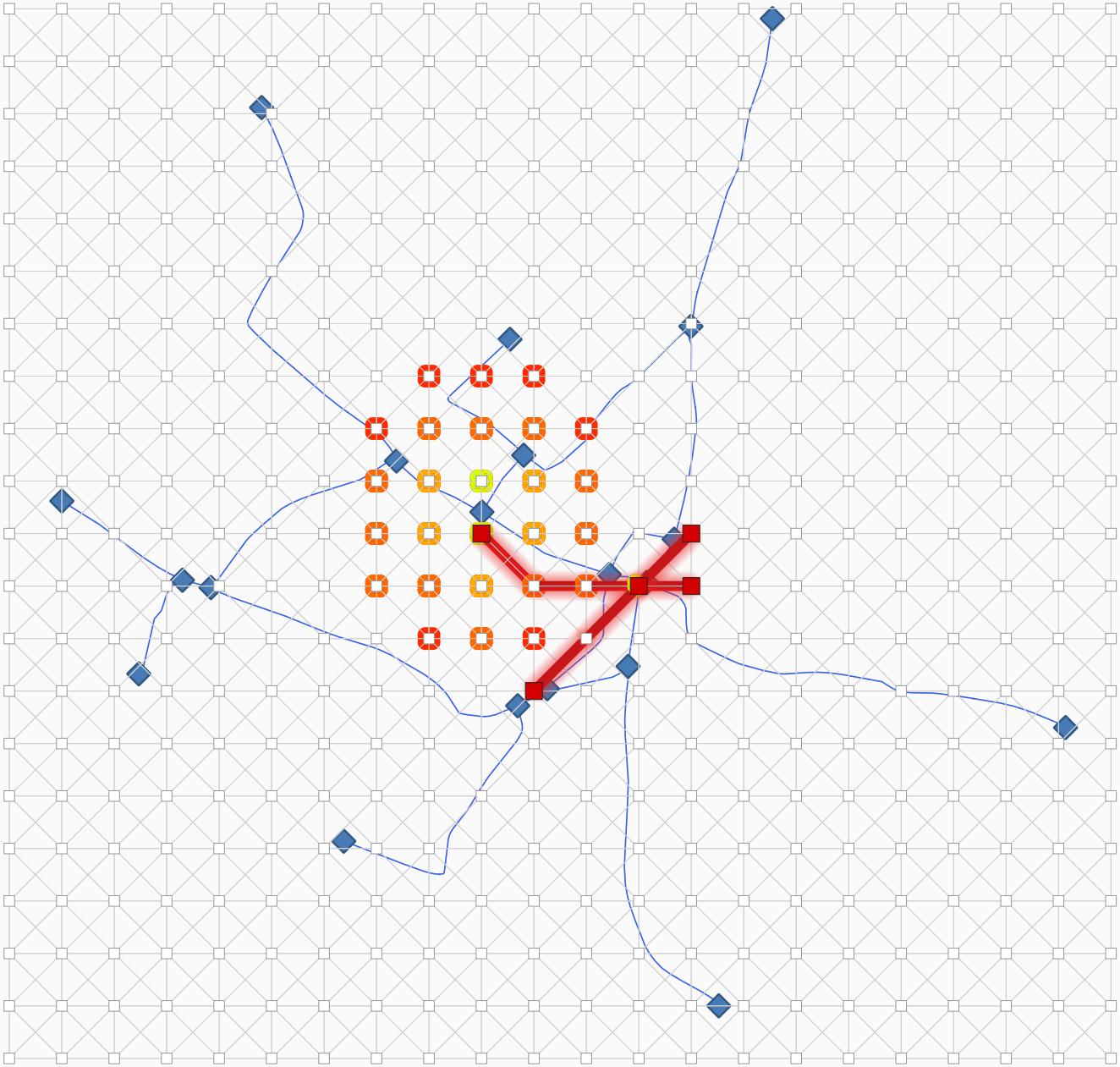
Iterative Calculation



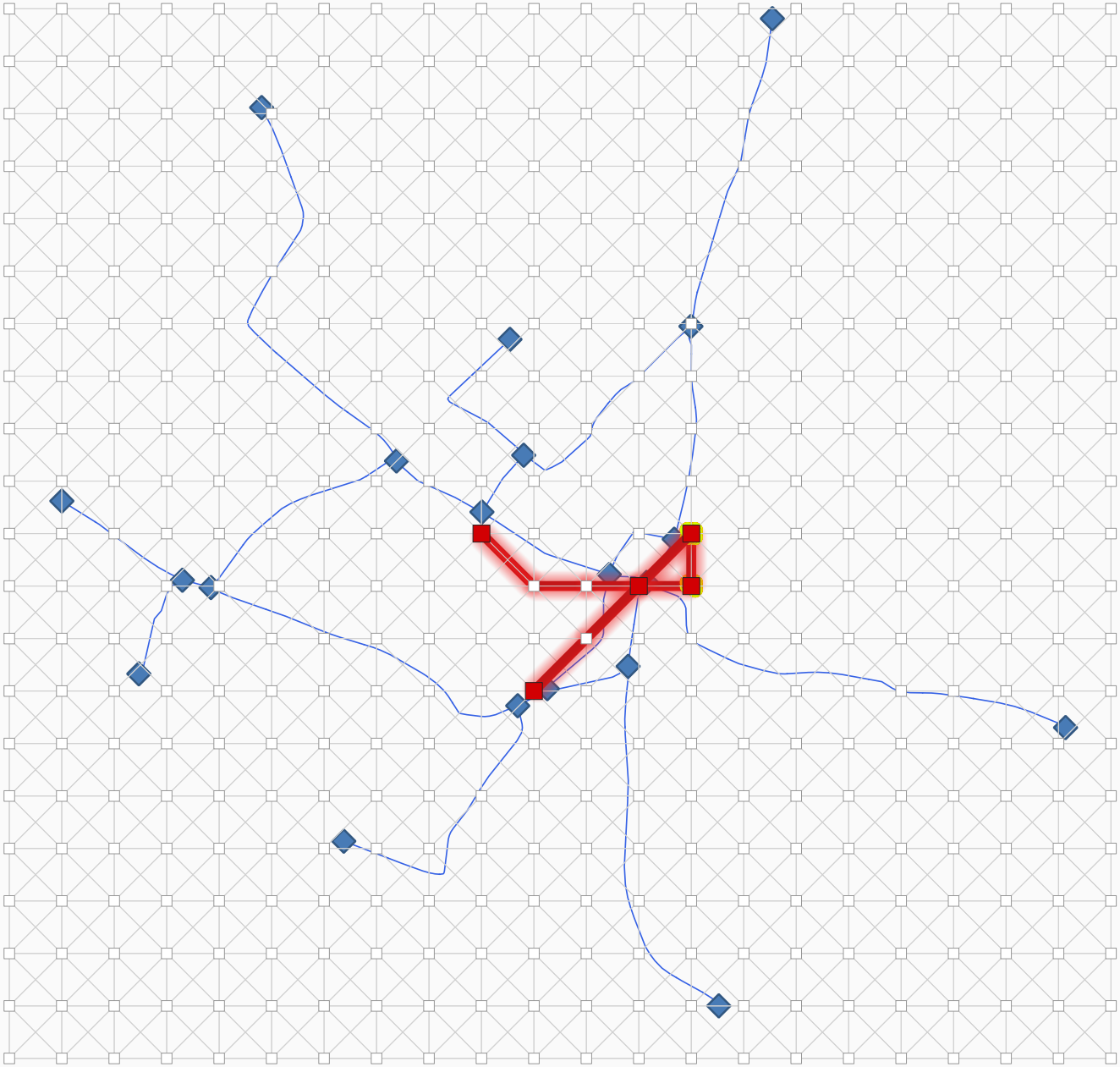
Iterative Calculation



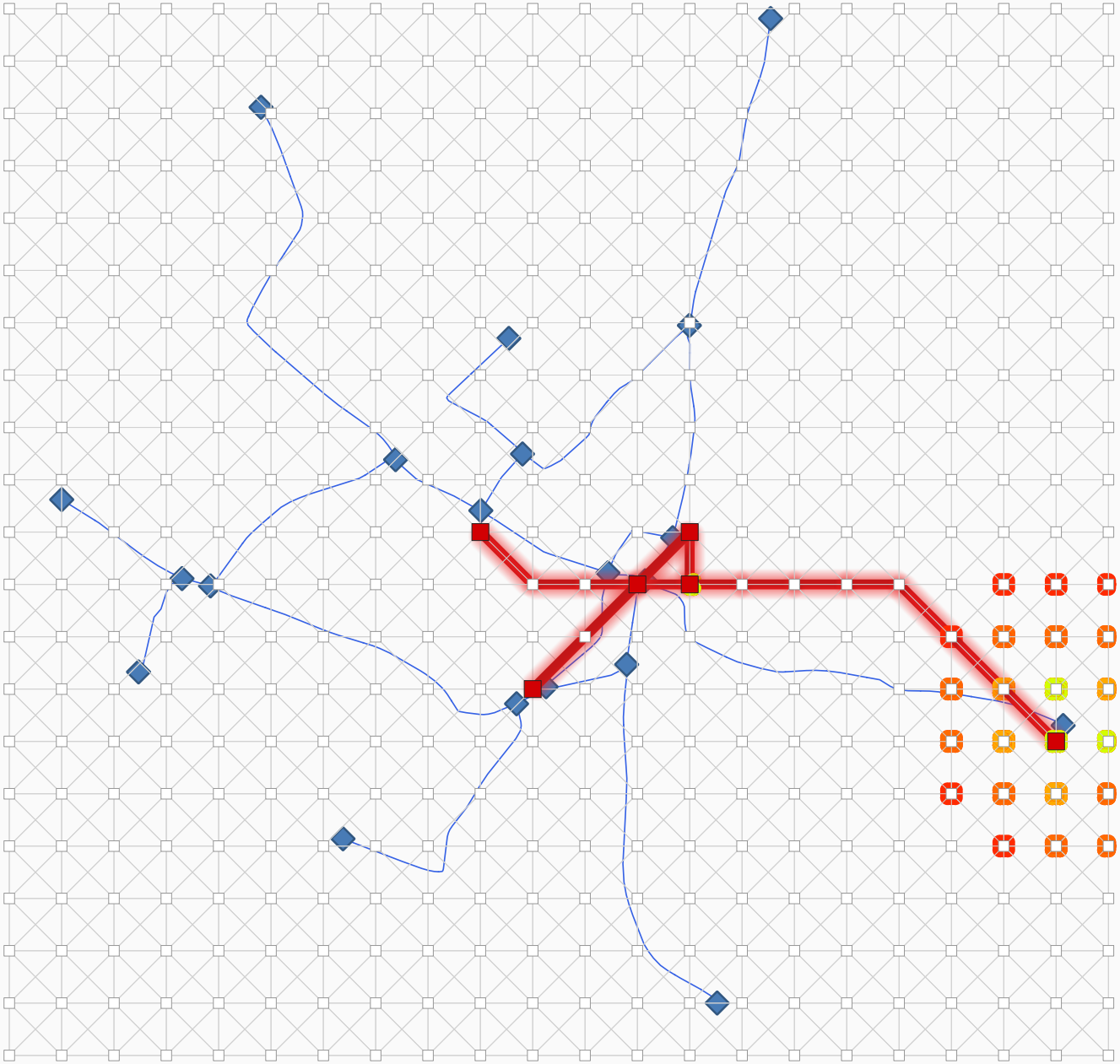
Iterative Calculation



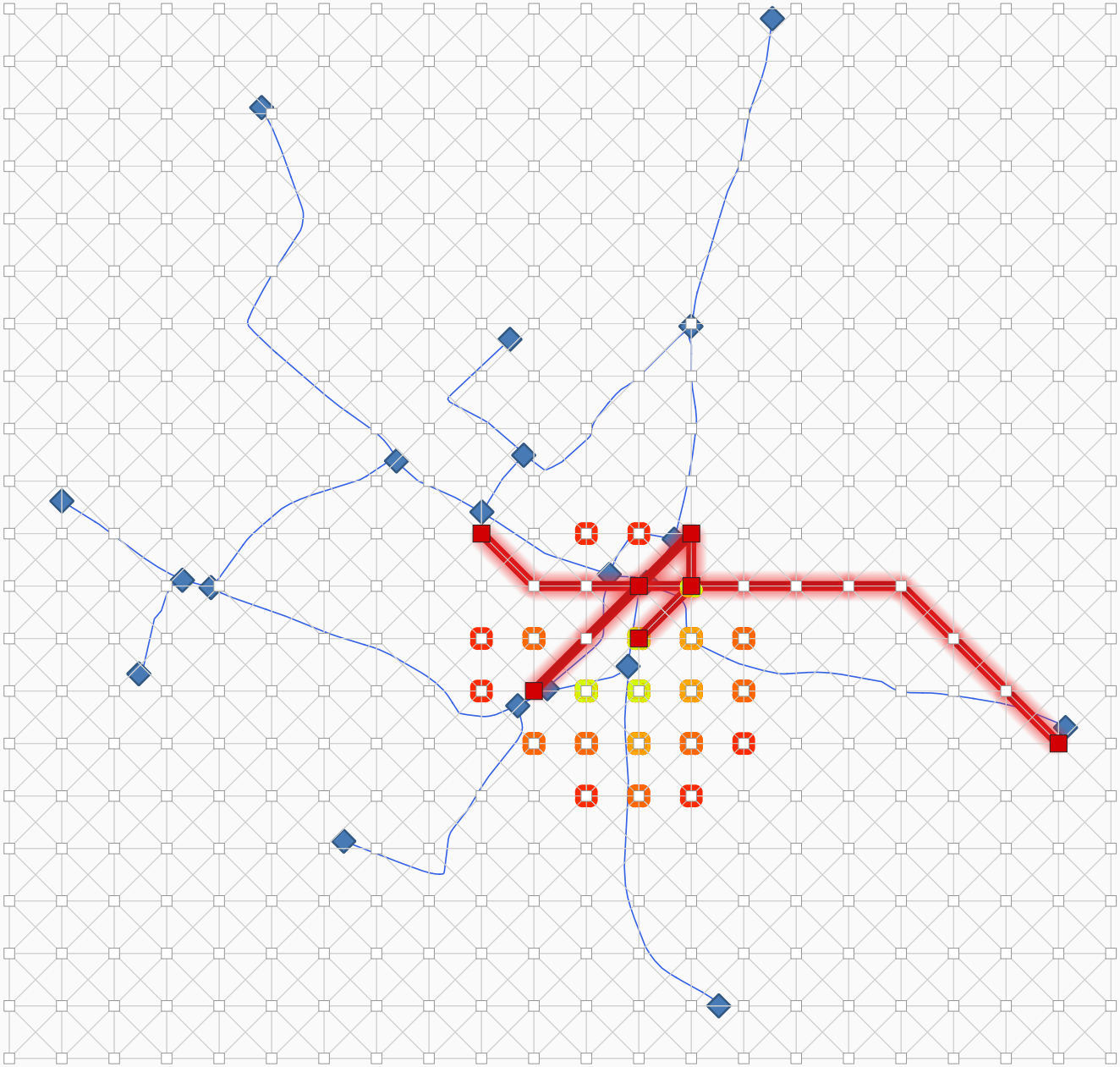
Iterative Calculation



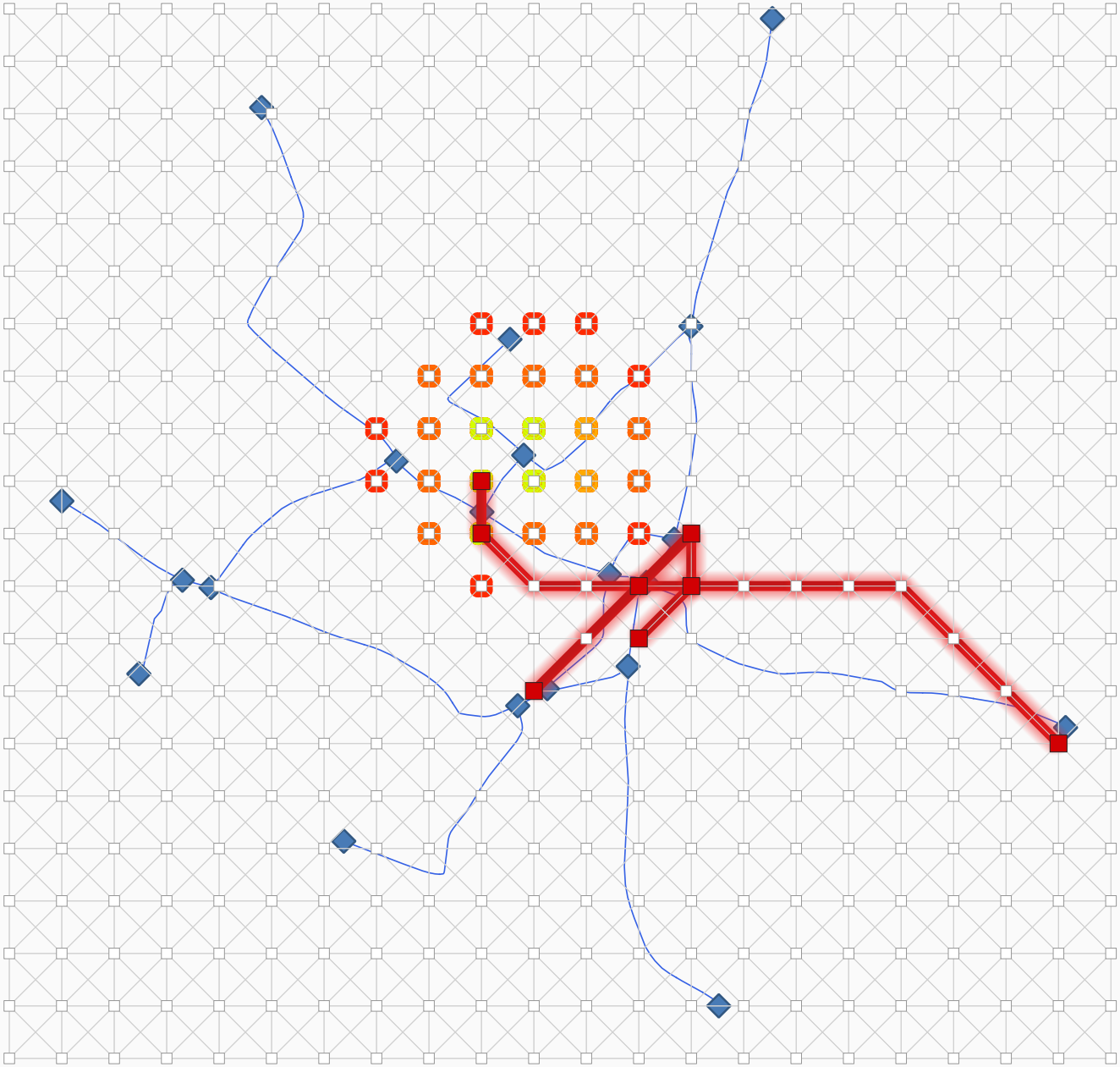
Iterative Calculation



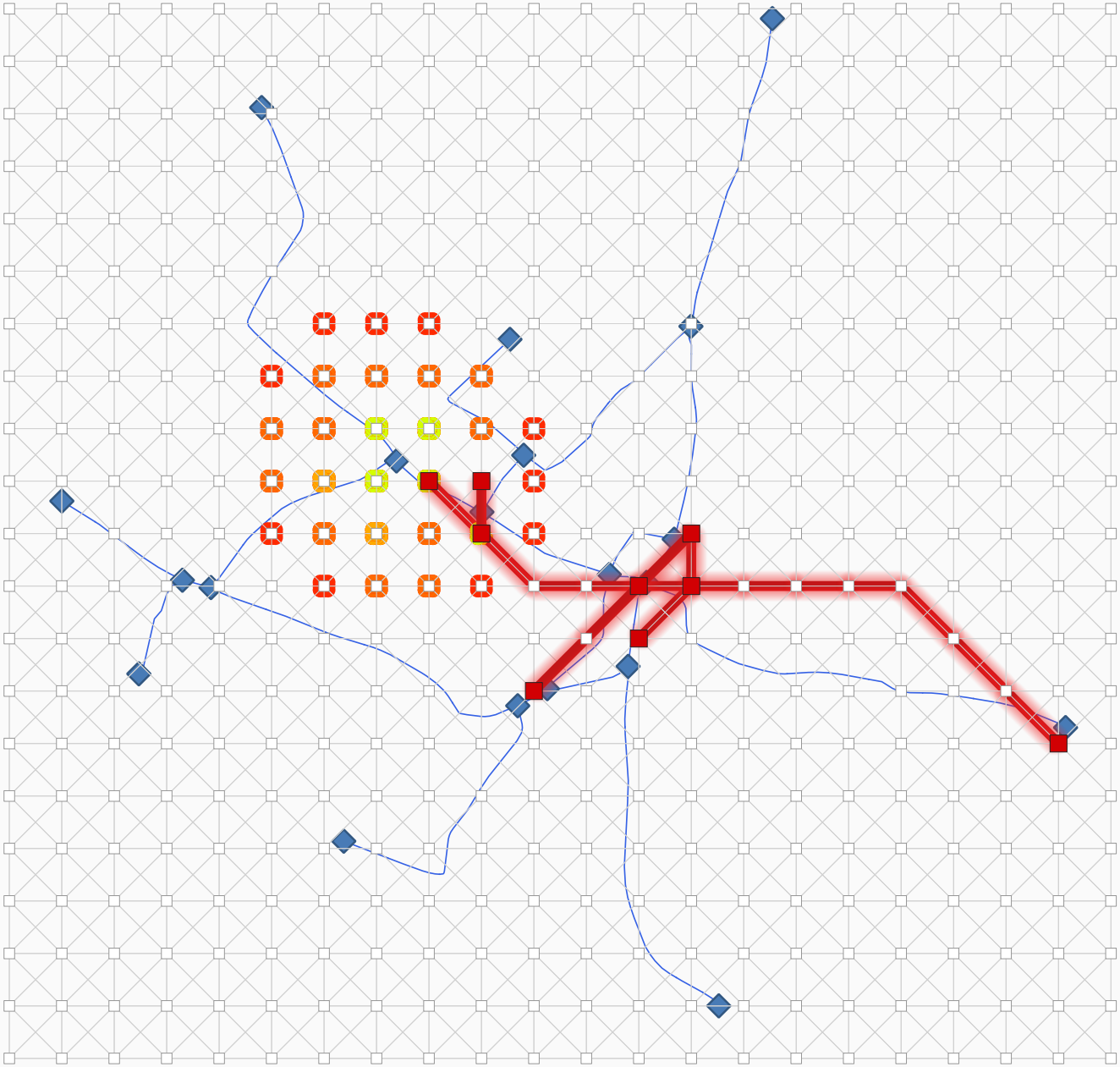
Iterative Calculation



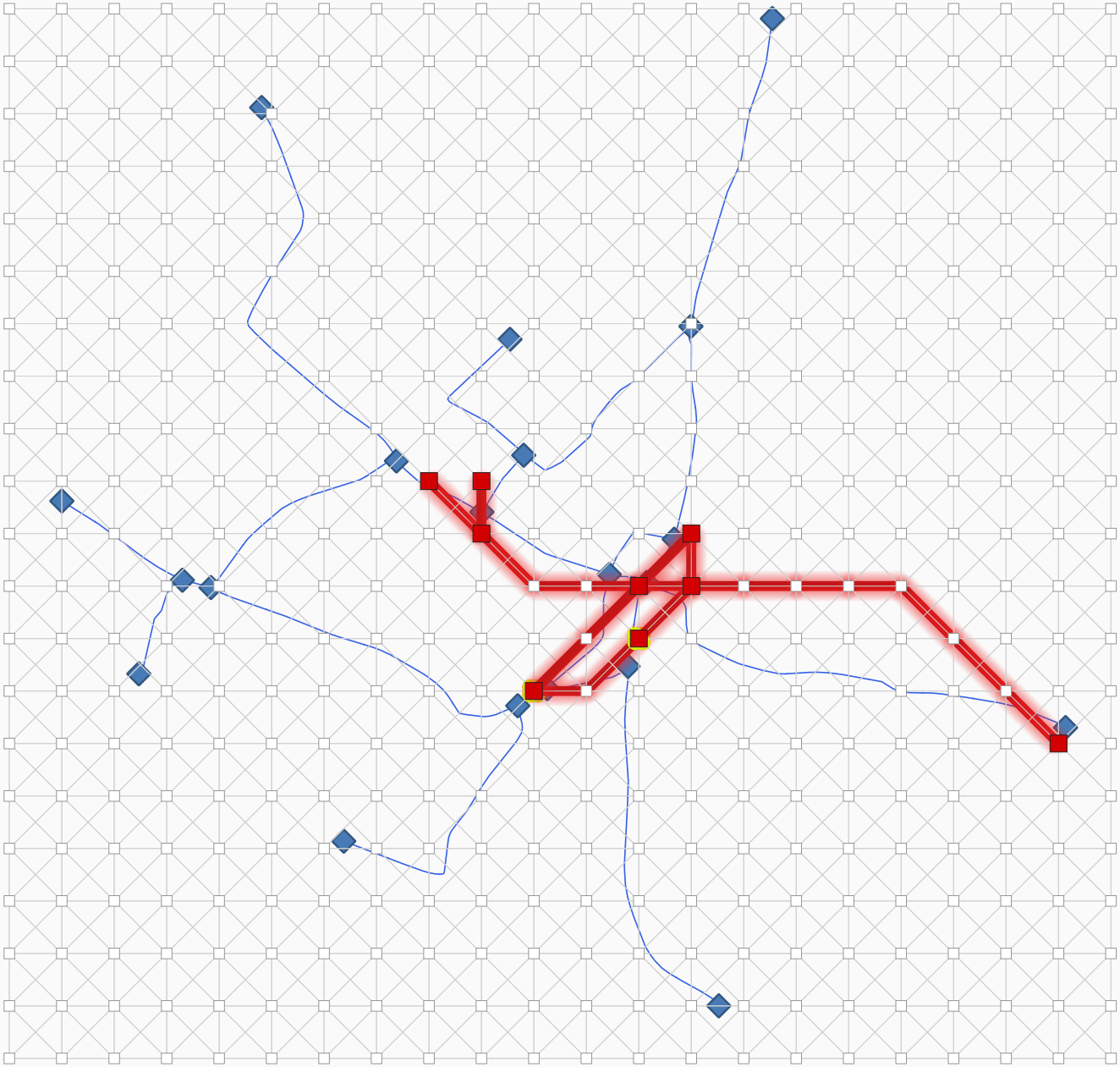
Iterative Calculation



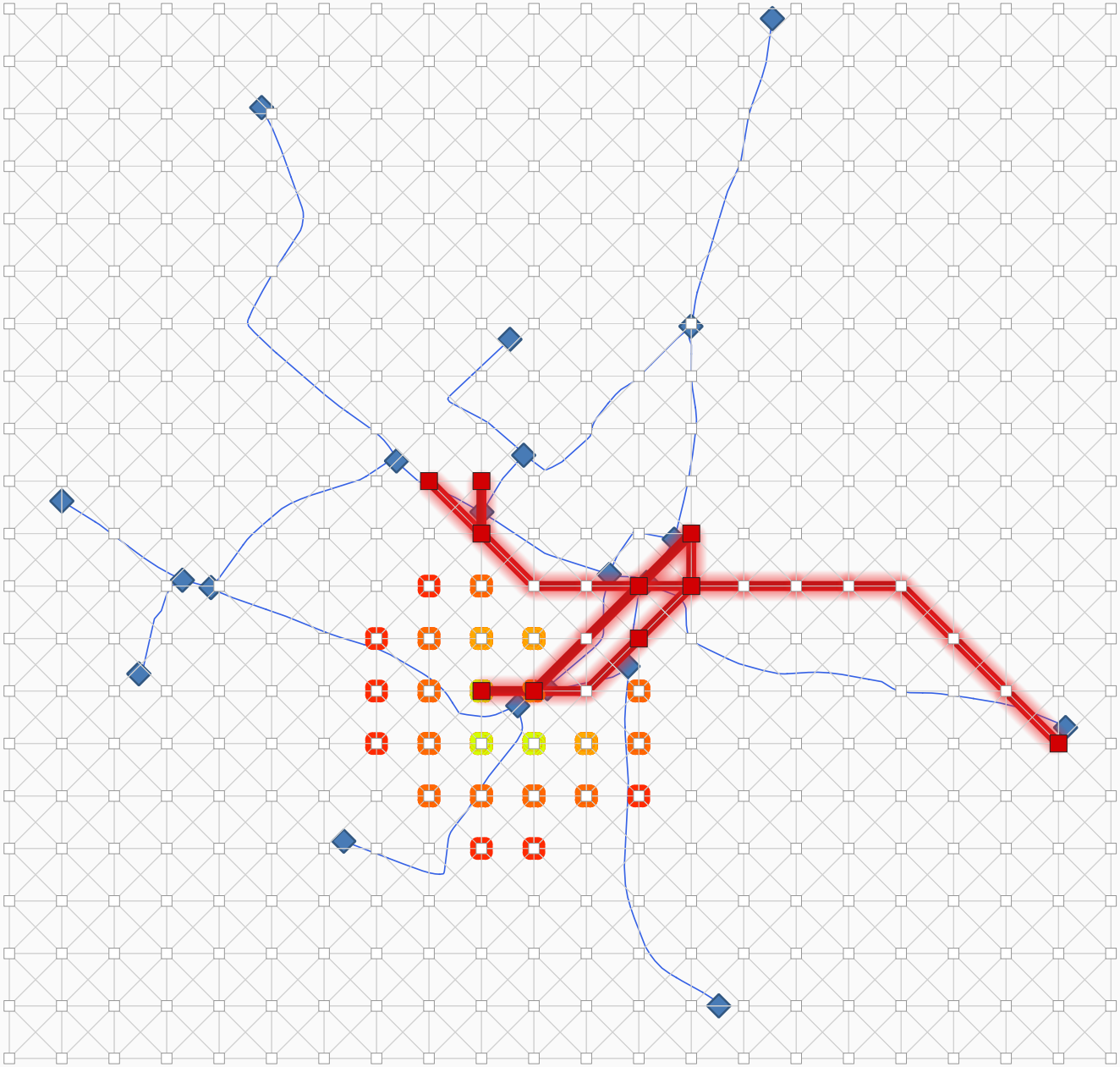
Iterative Calculation



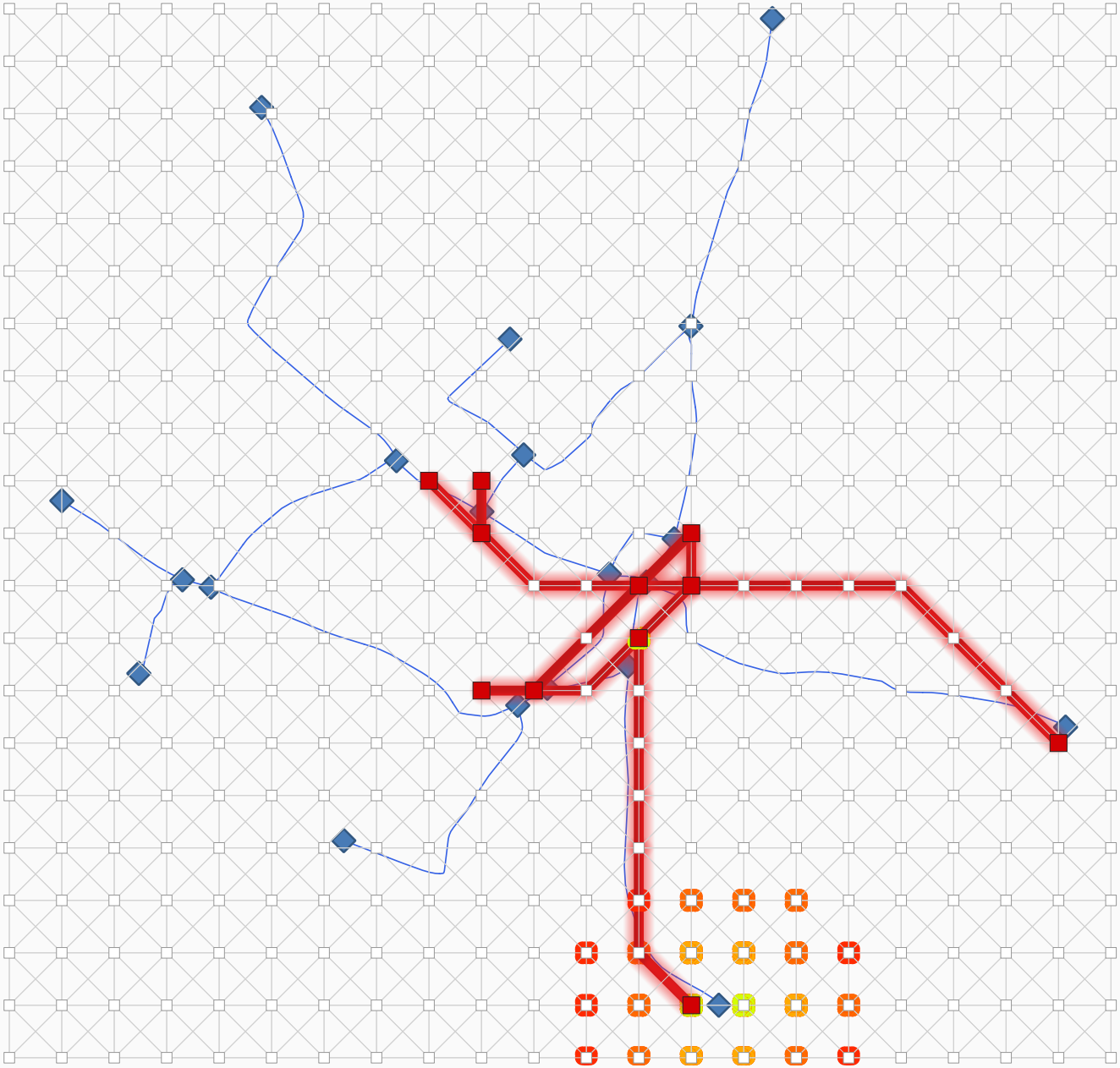
Iterative Calculation



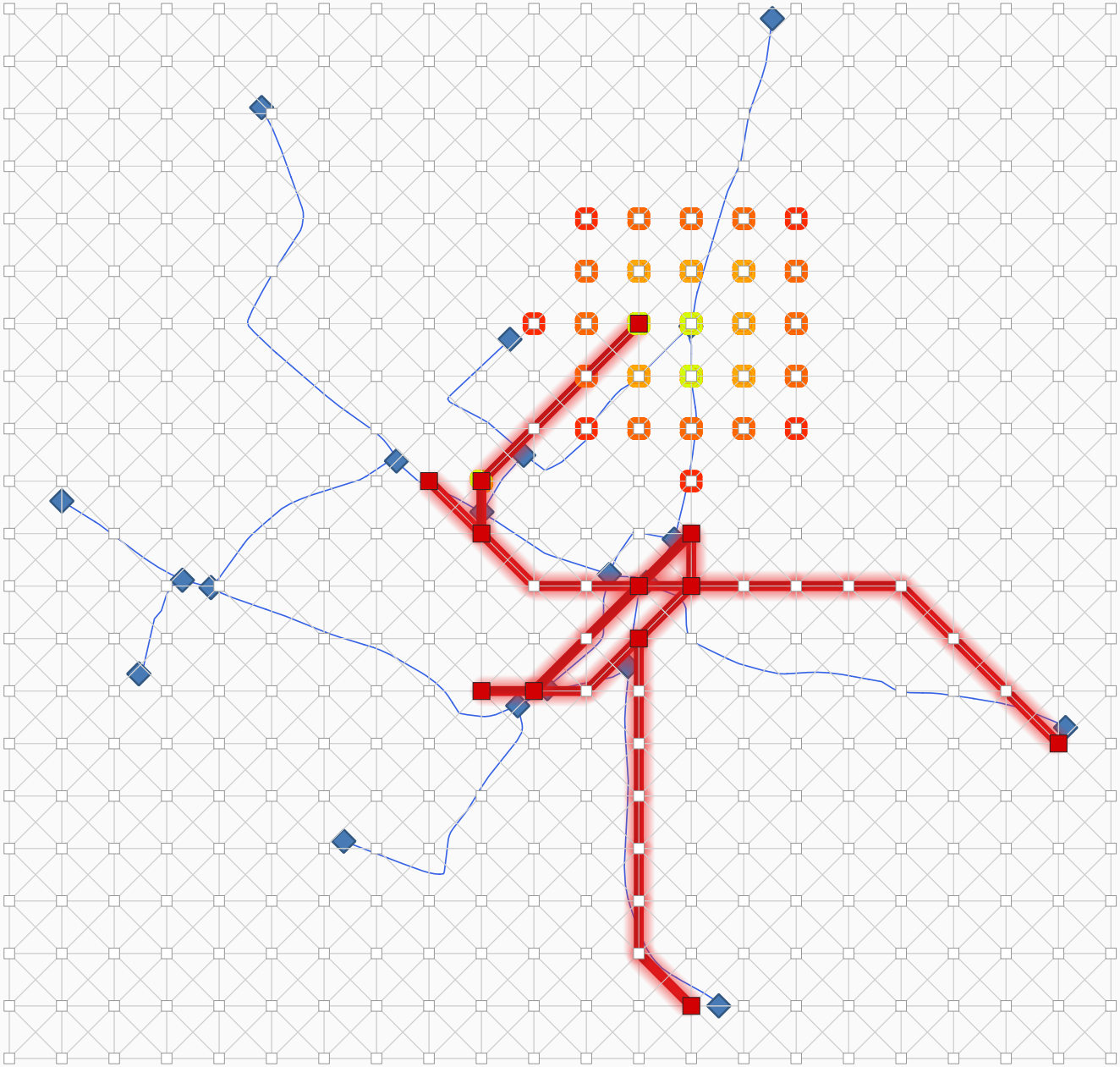
Iterative Calculation



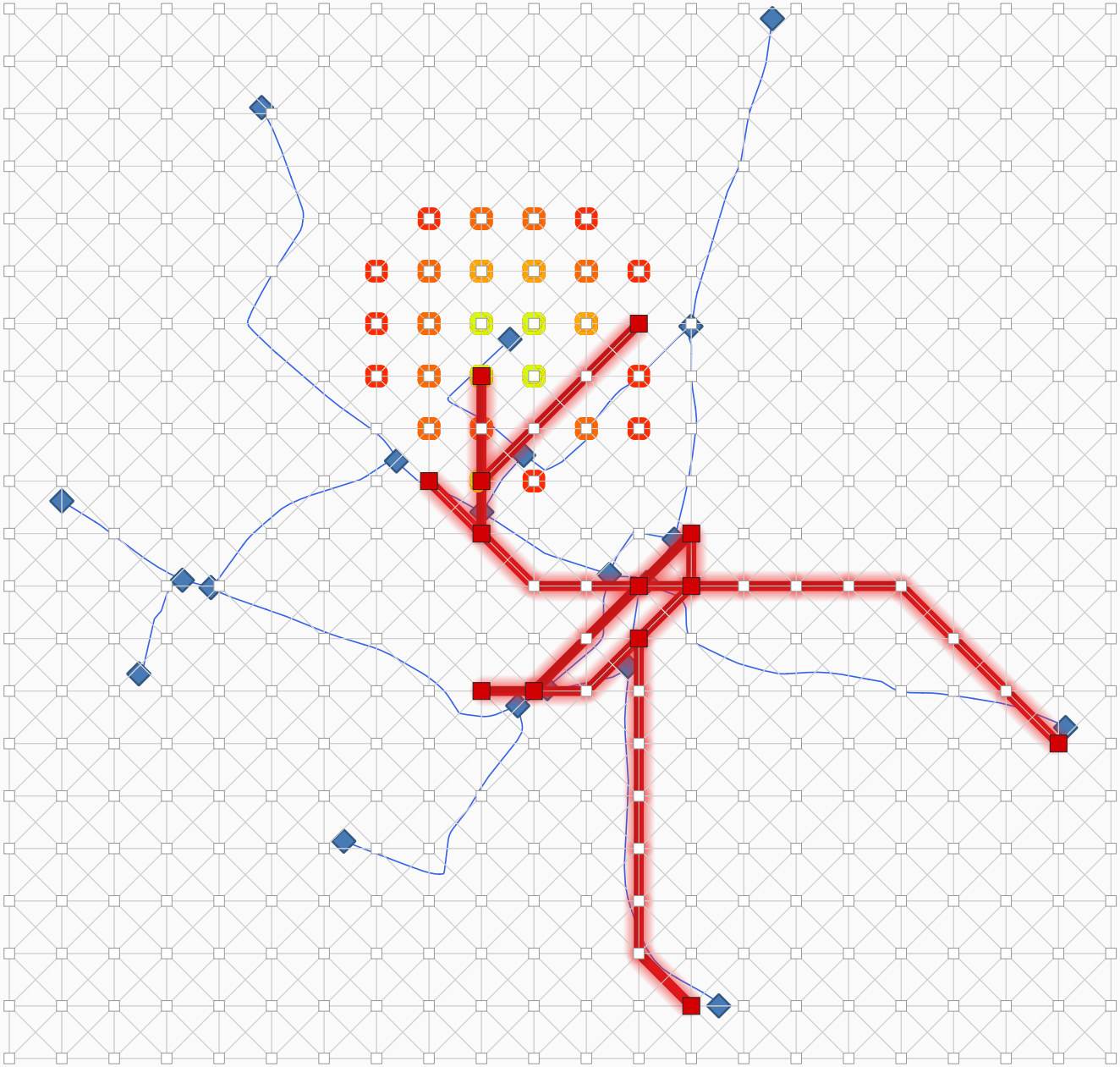
Iterative Calculation



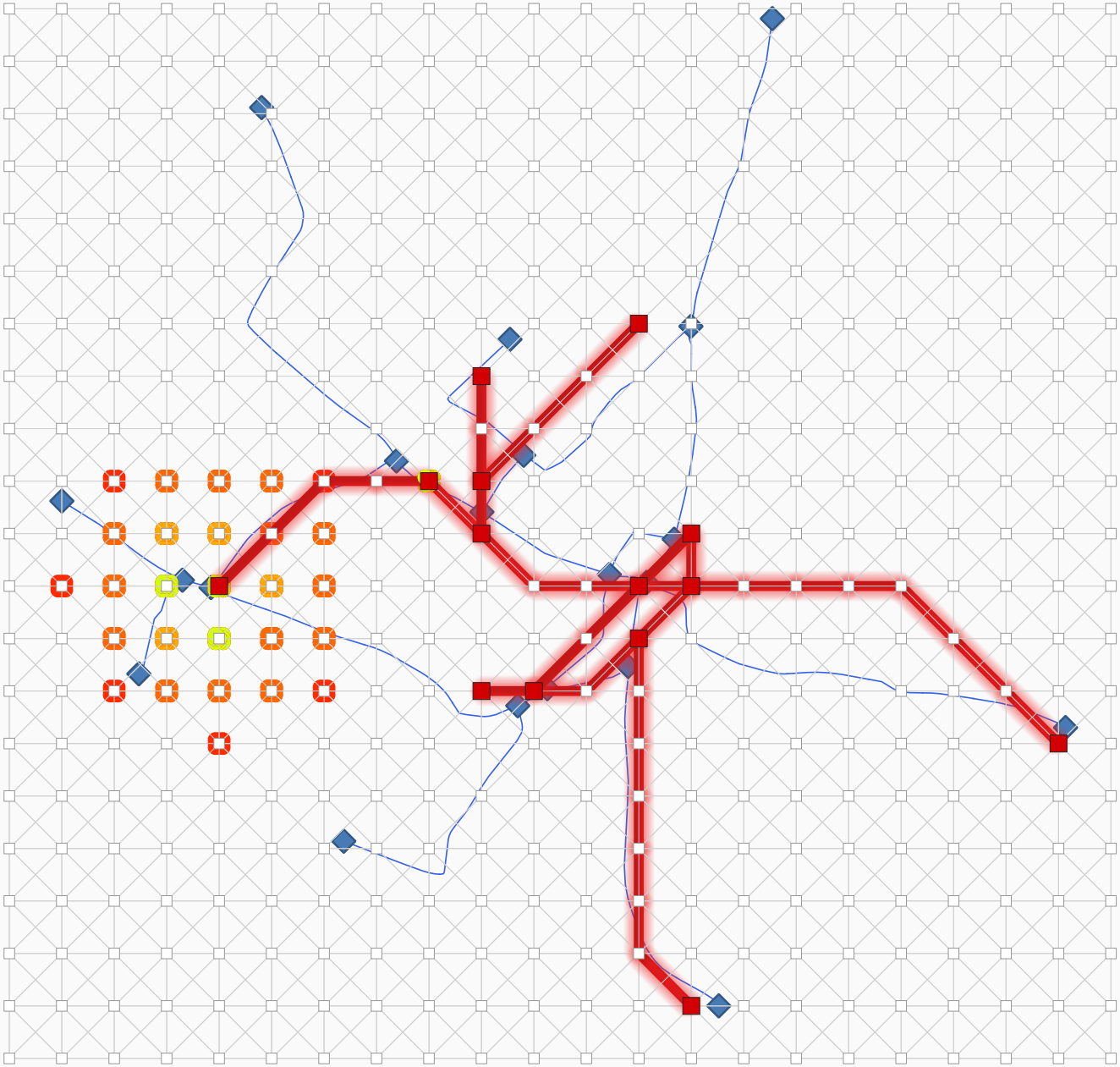
Iterative Calculation



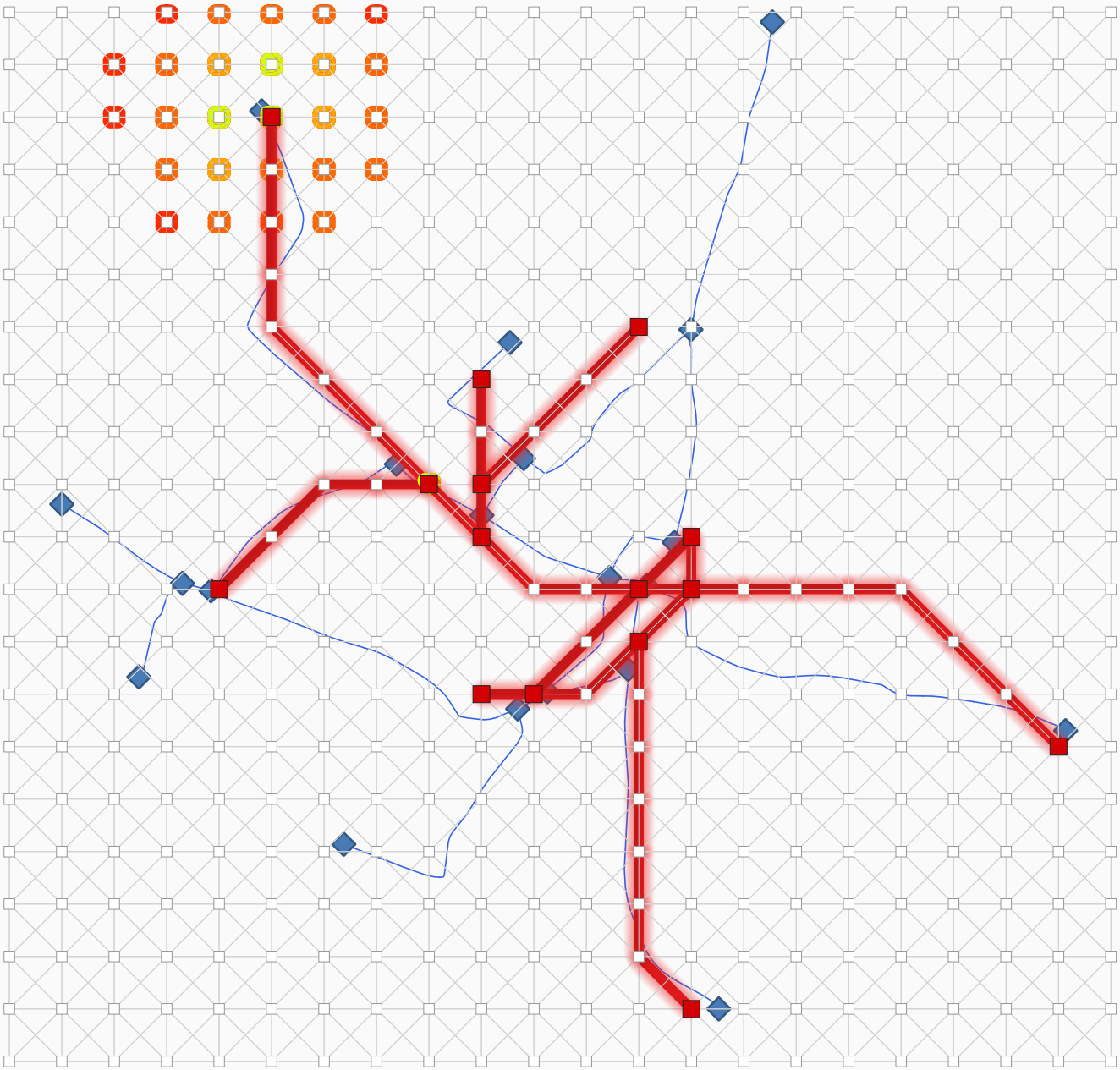
Iterative Calculation



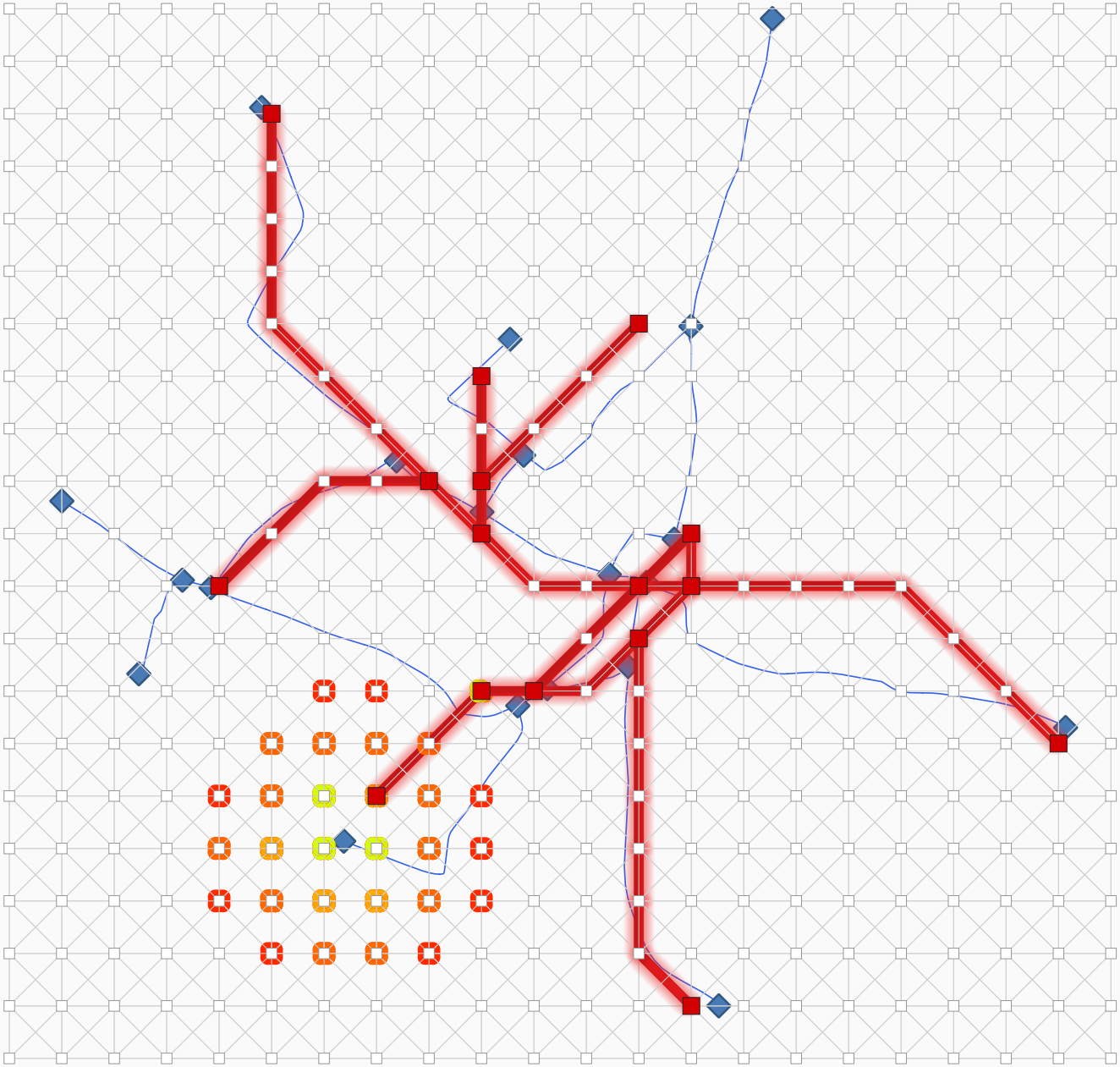
Iterative Calculation



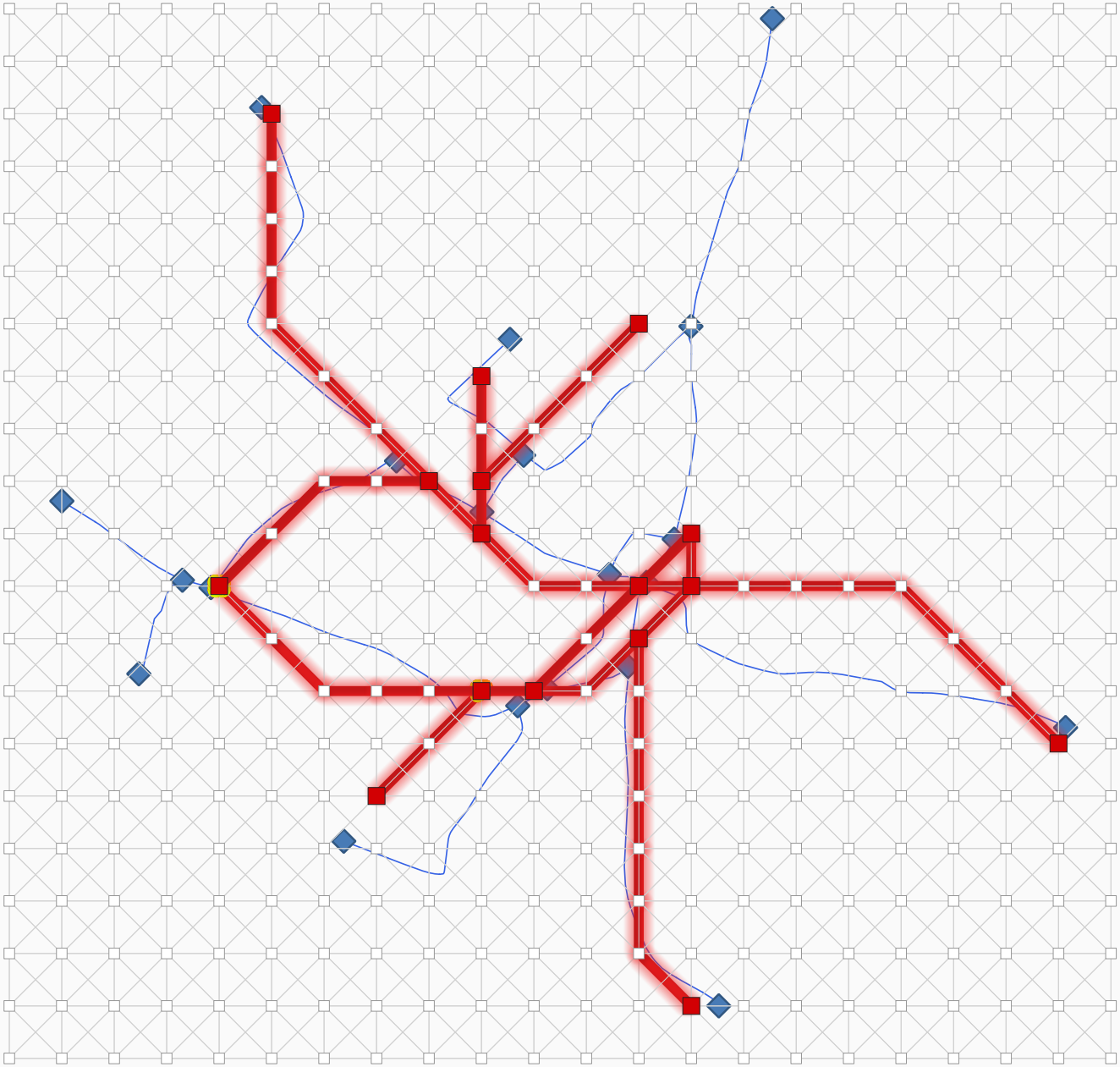
Iterative Calculation



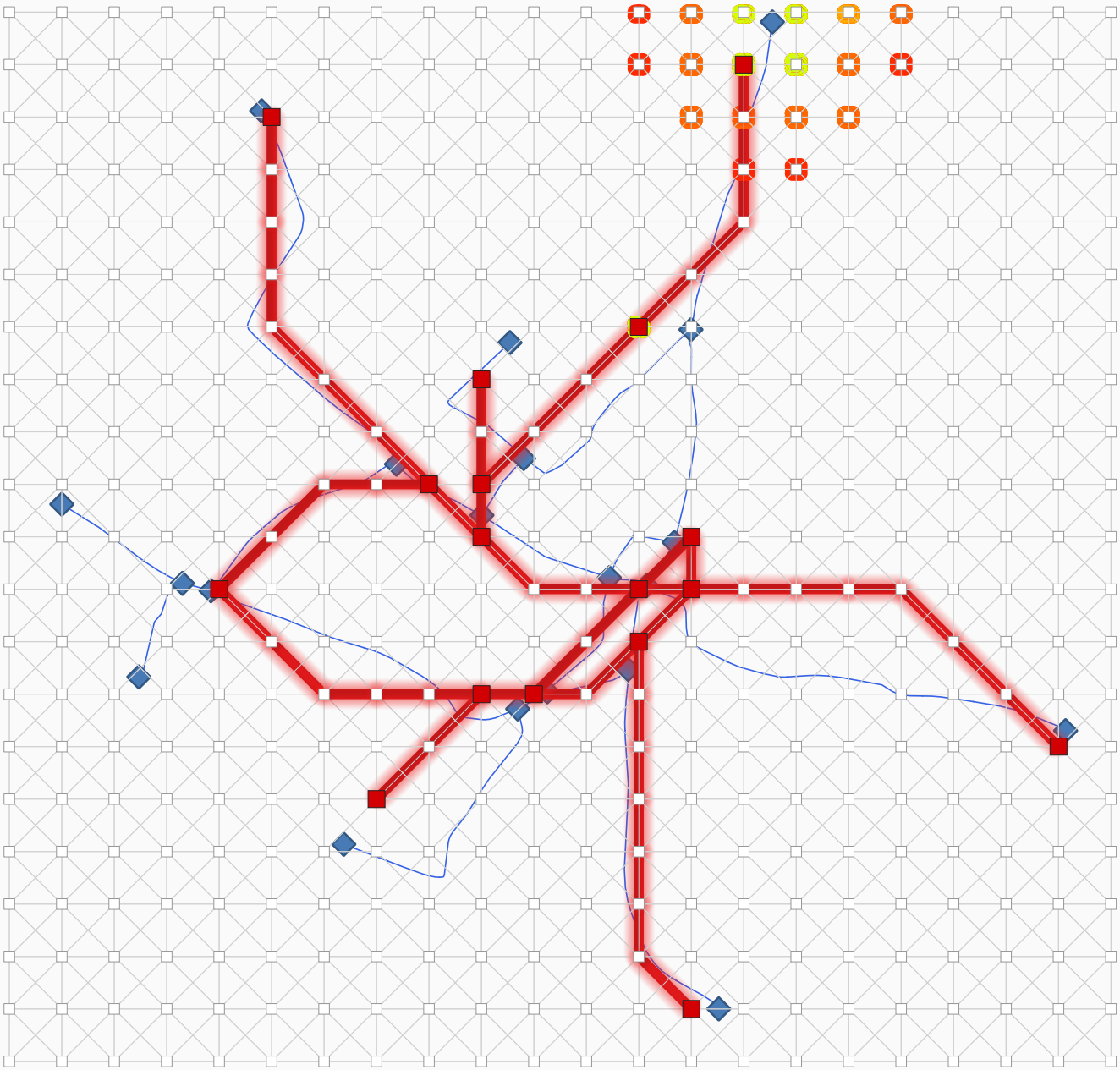
Iterative Calculation



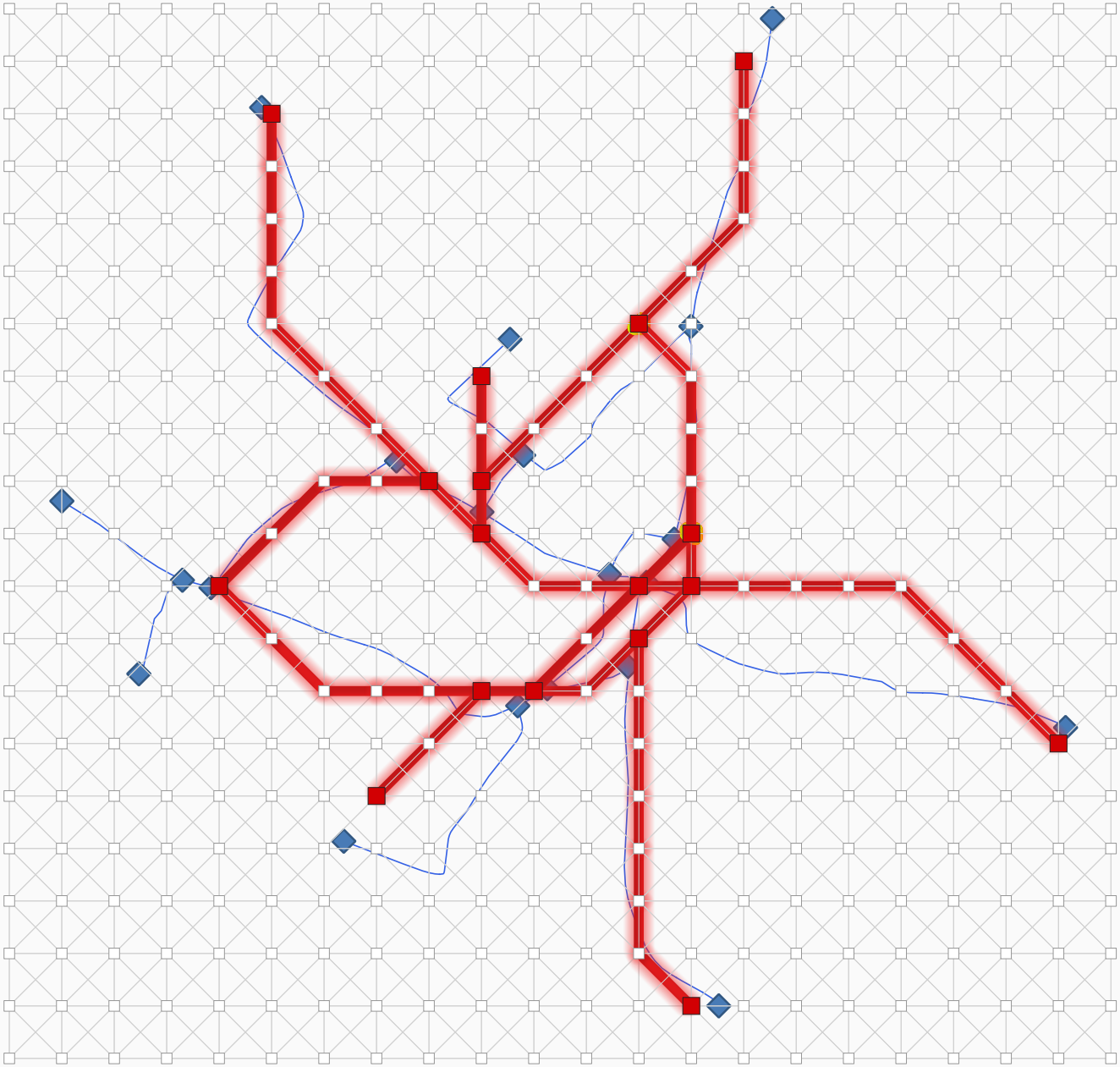
Iterative Calculation



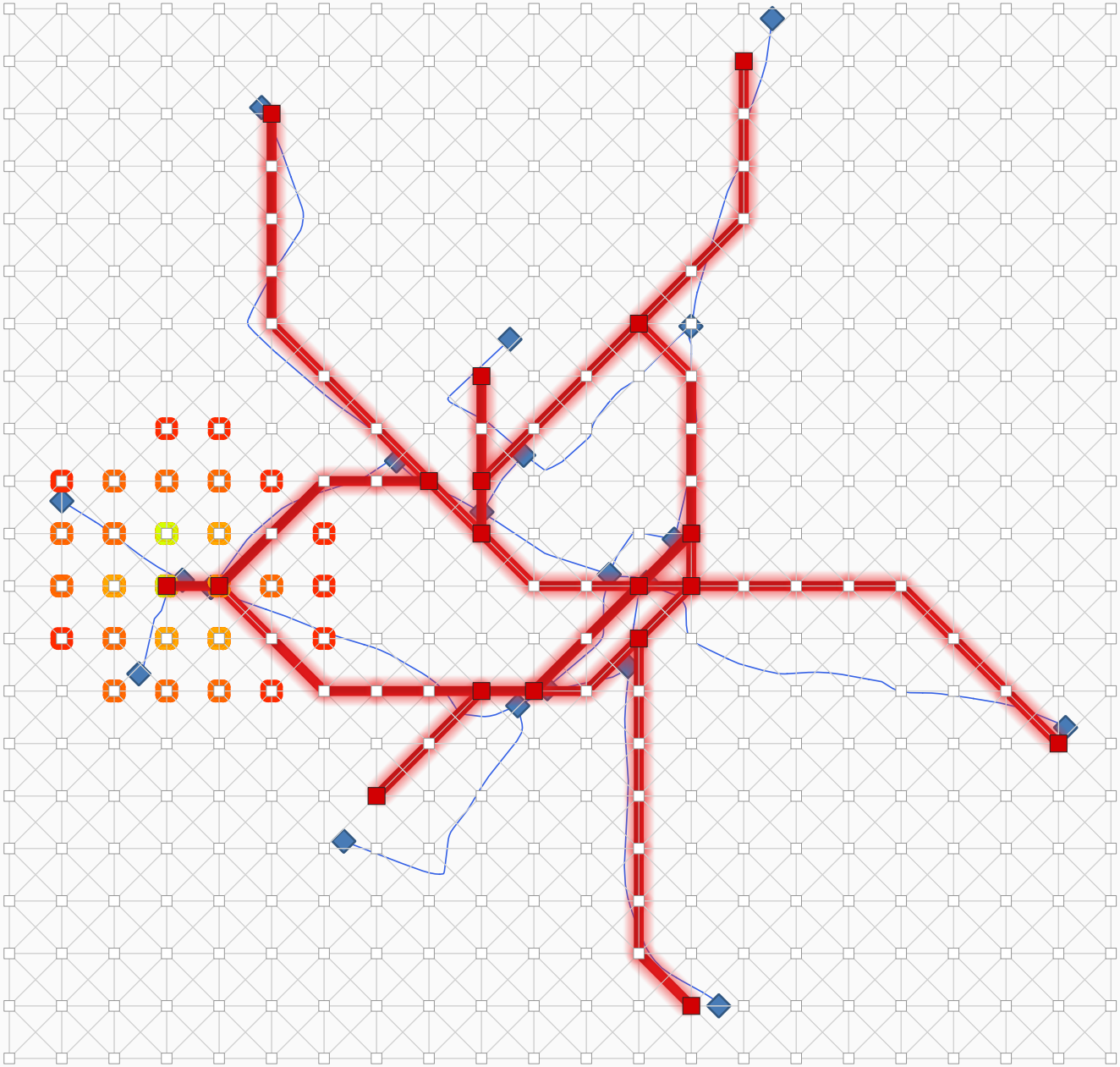
Iterative Calculation



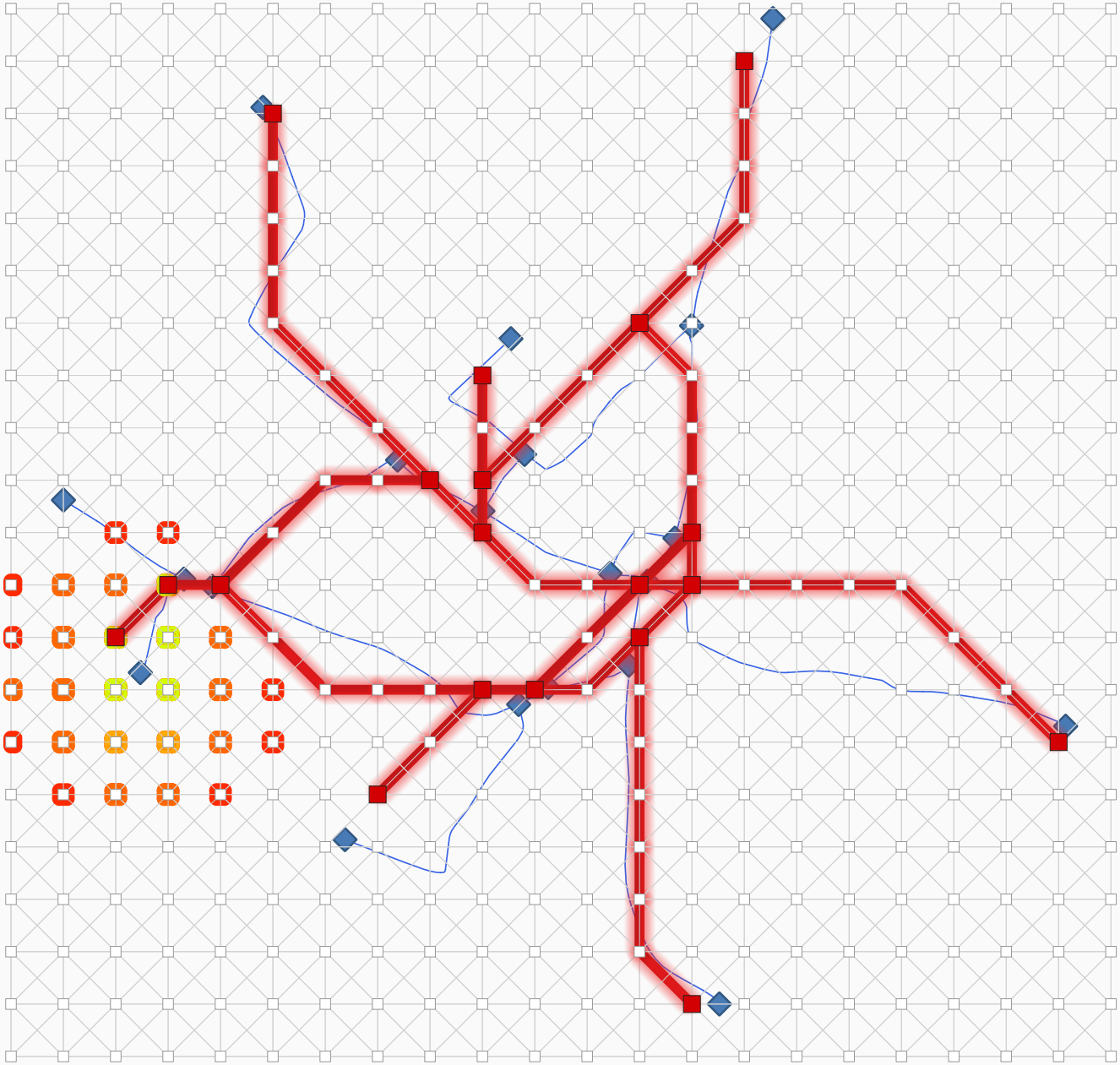
Iterative Calculation



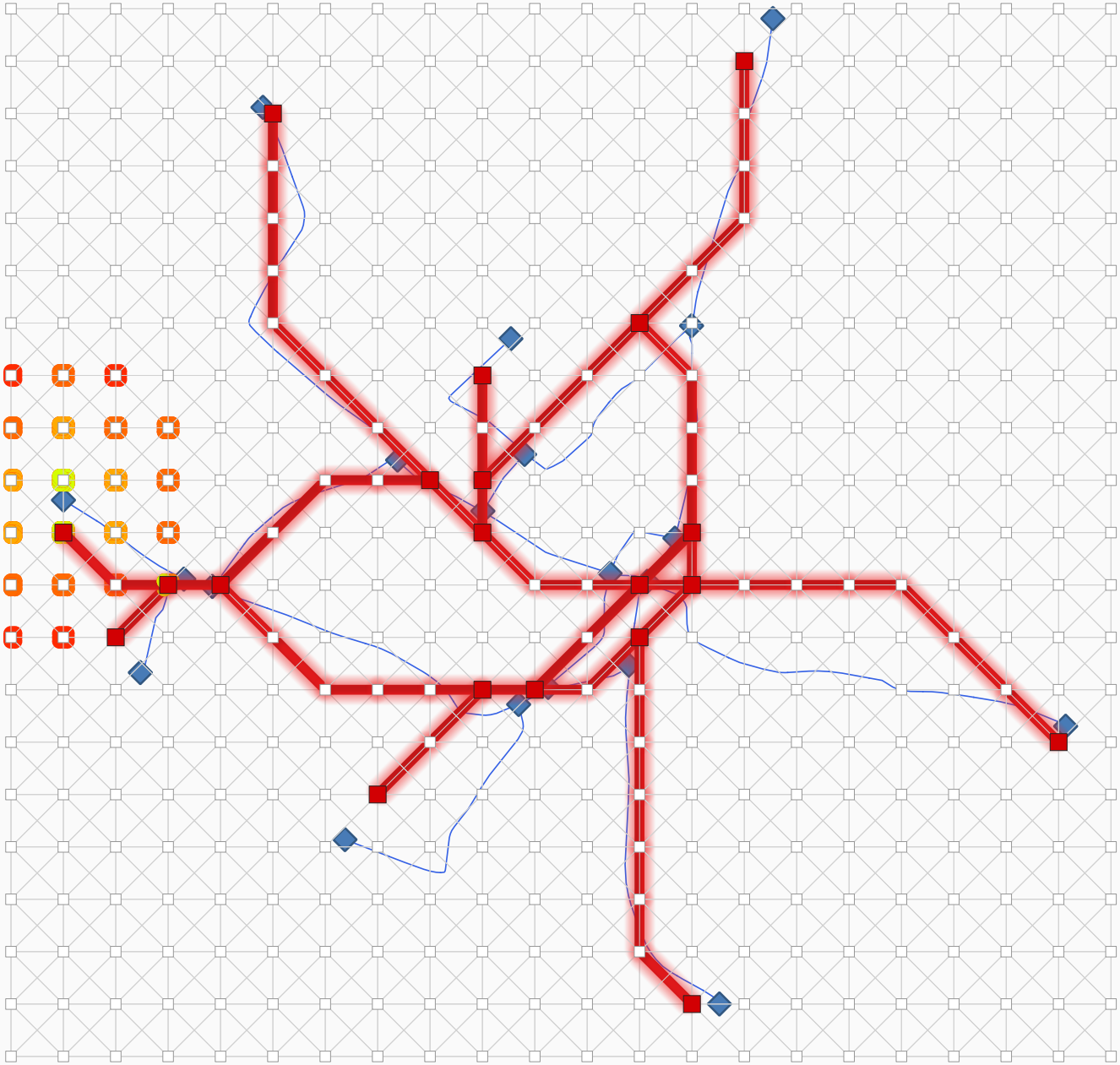
Iterative Calculation



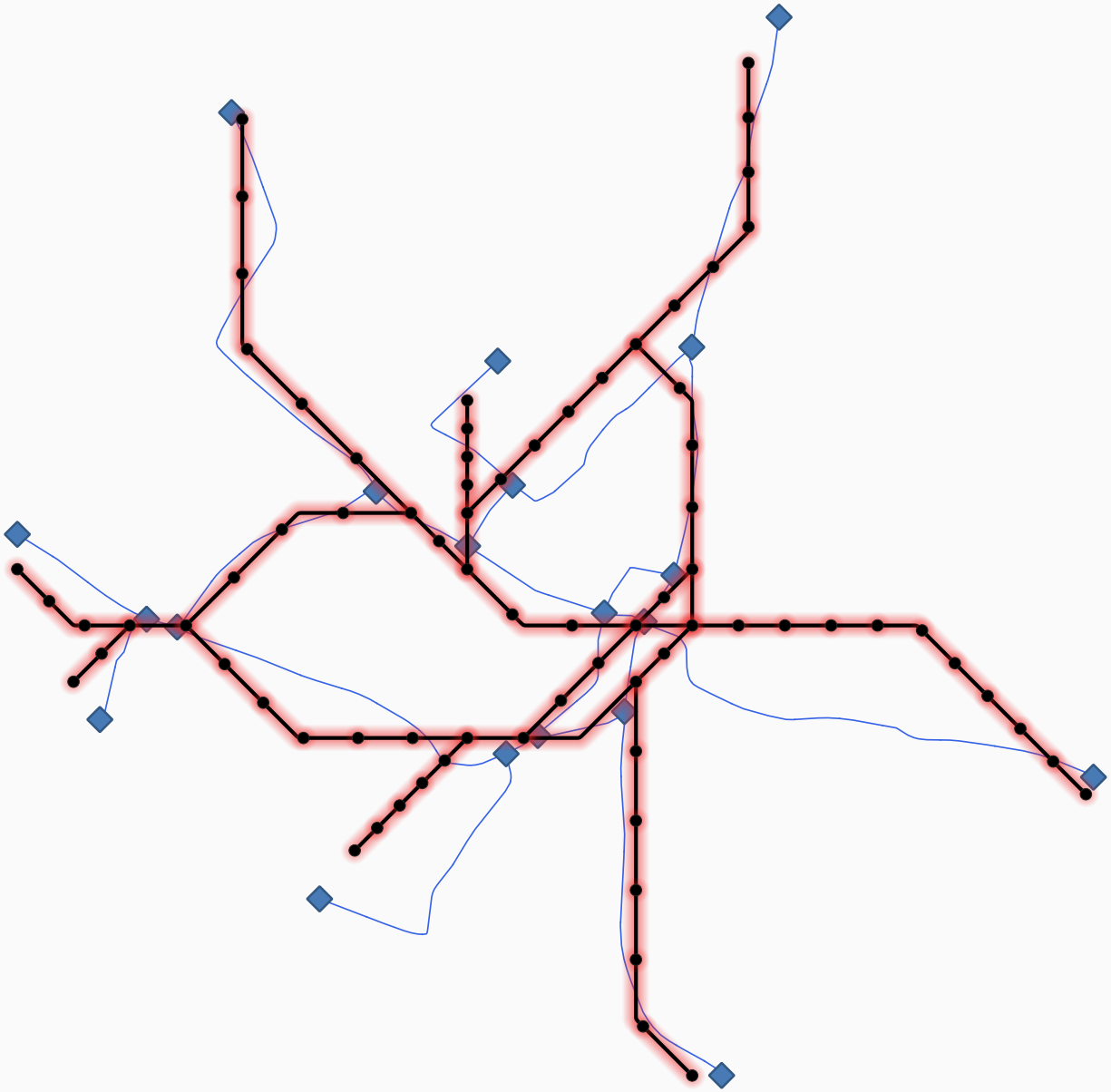
Iterative Calculation



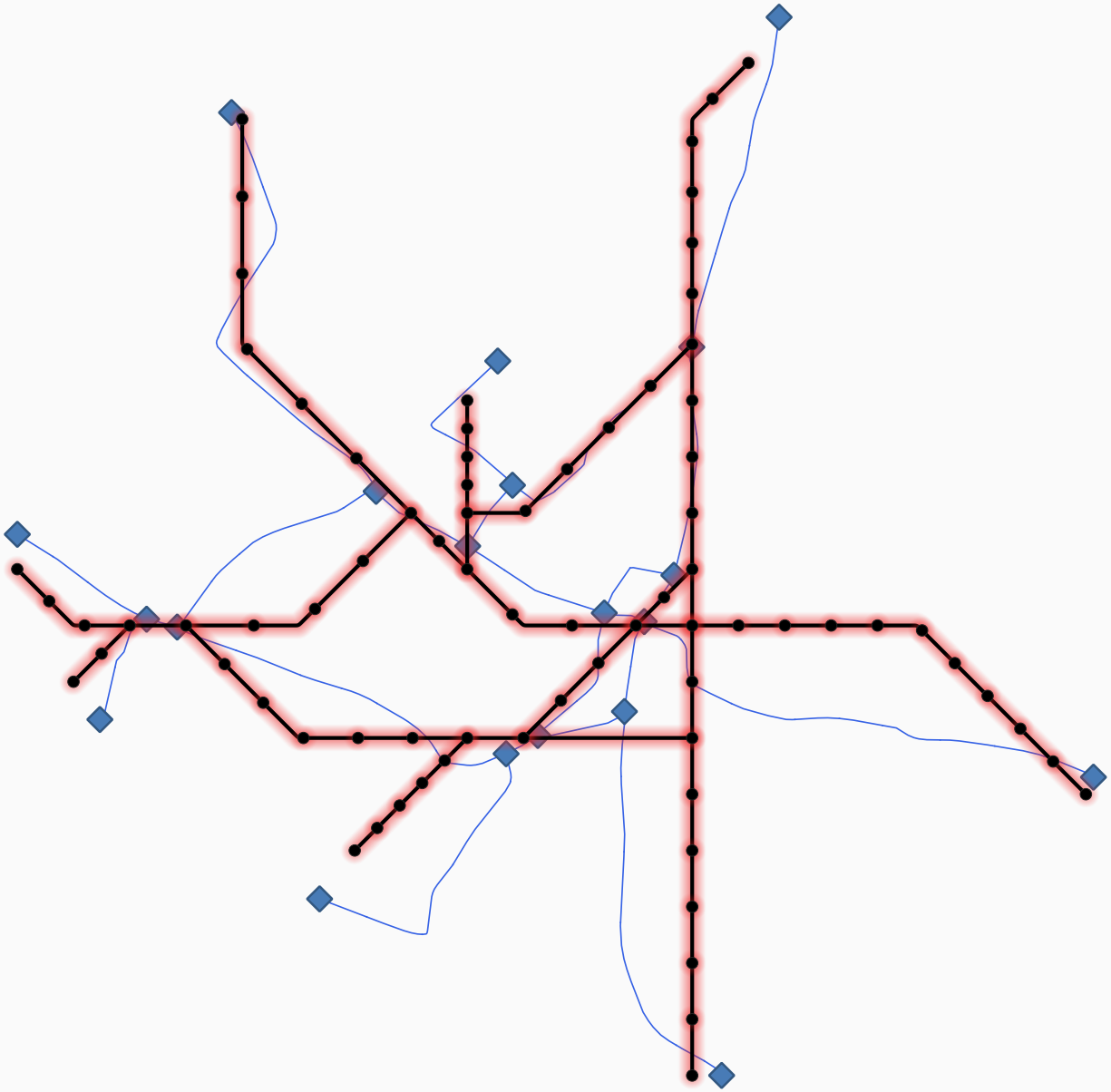
Iterative Calculation



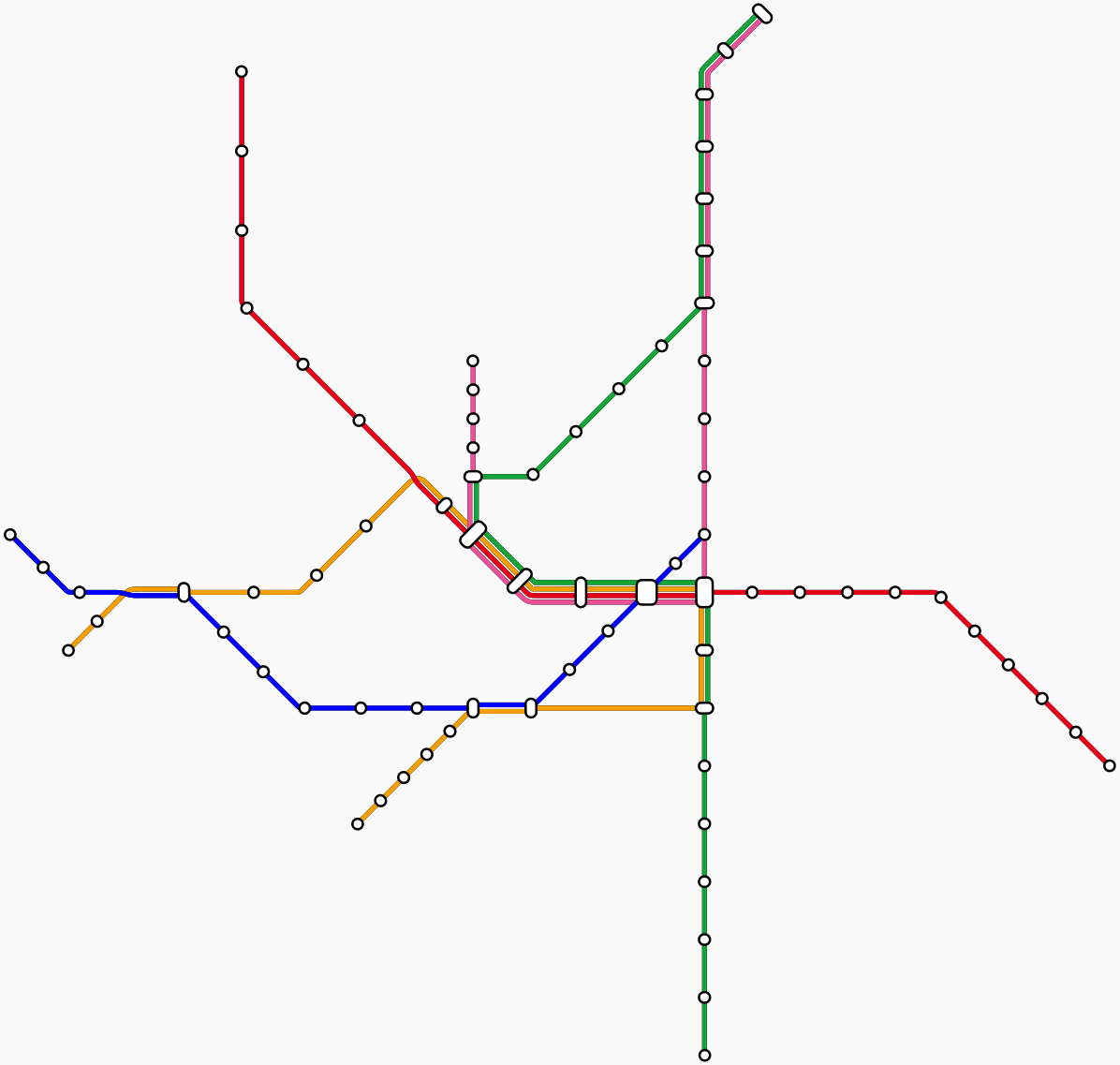
Iterative Calculation



Iterative Calculation



Iterative Calculation

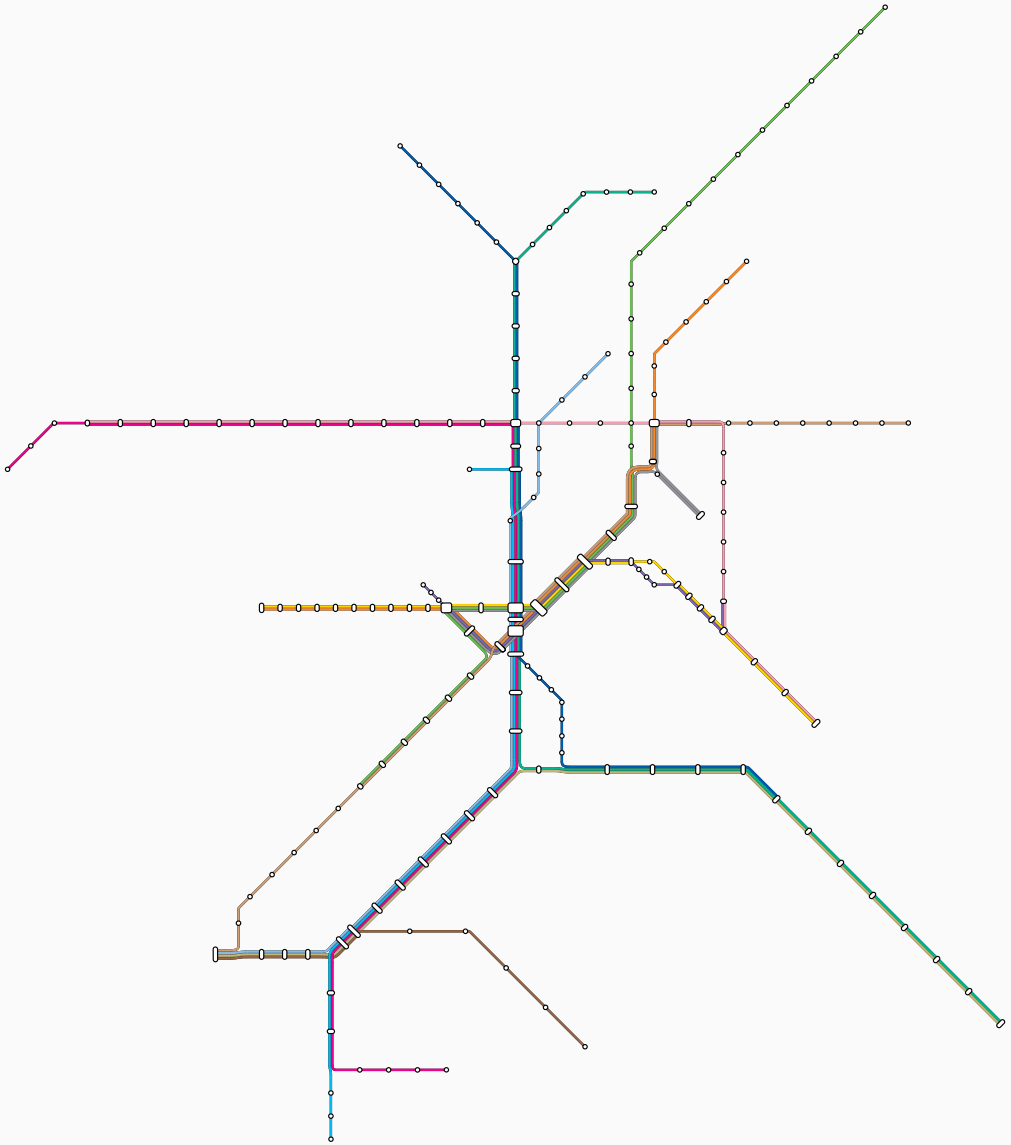


Linear Program (LP) or Approximate Approach (A)?

Final target values and approximation error δ of linear program (LP-2) and approximate approach (A-2) when degree 2 nodes where contracted first

	LP-2	t	A-2	t	δ
Freiburg	144.6	11m	146.5	73ms	**1.3%**
Vienna	170.5	13h	175.1	171ms	**2.7%**
Stuttgart	383.2	12h	399.2	510ms	**4.1%**
Berlin	315.4	20h	326.0	513ms	**3.4%**
Sydney	360.6	7h	361.4	250ms	**0.2%**
London	$\geq$669.2	—	758.3	2.1 s	$\leq$**14%**

Linear Program (LP) or Approximate Approach (A)?



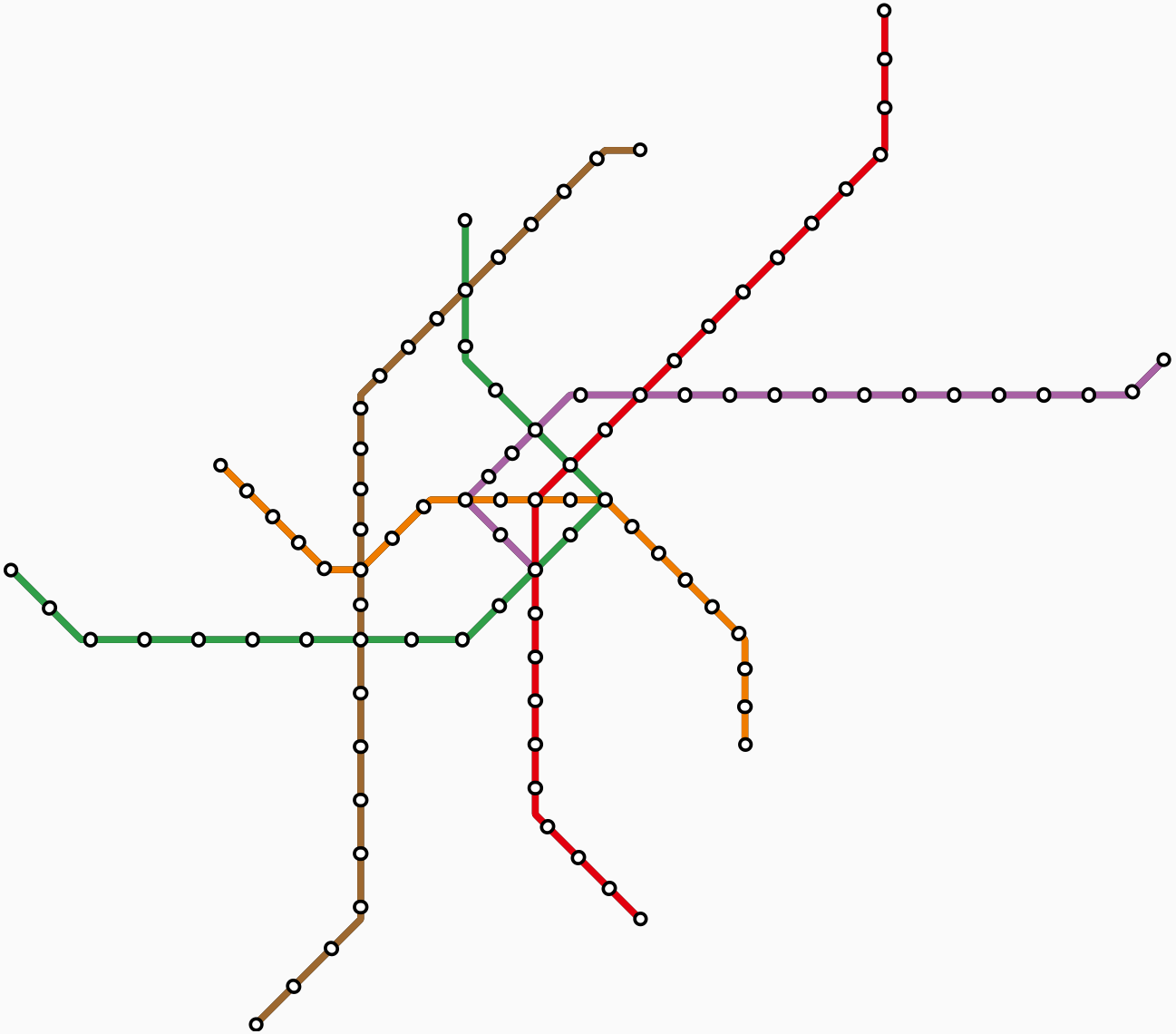
Linear Program (LP)

Linear Program (LP) or Approximate Approach (A)?

Linear Program (LP)

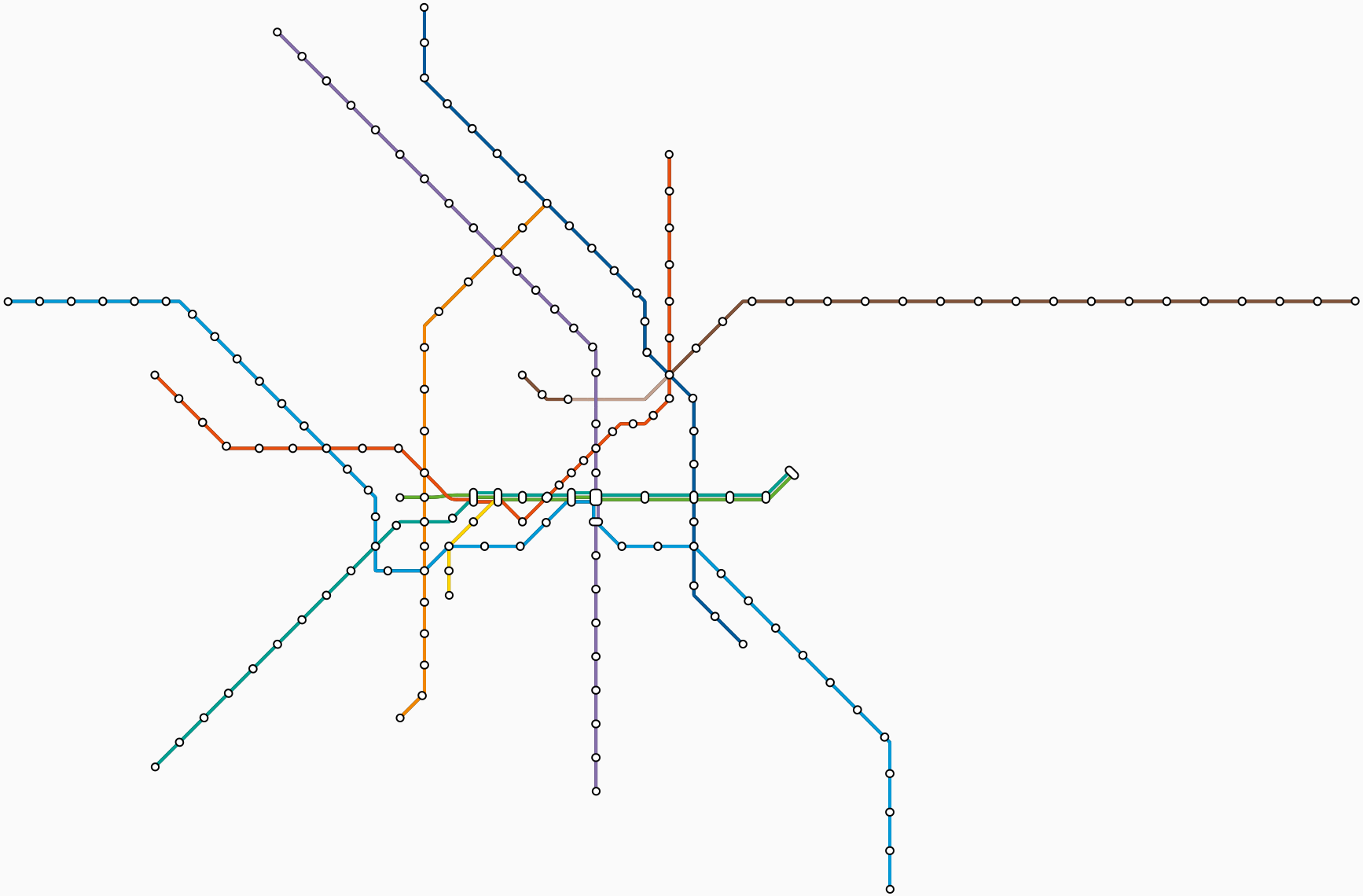
Approximate Approach (A)

Results



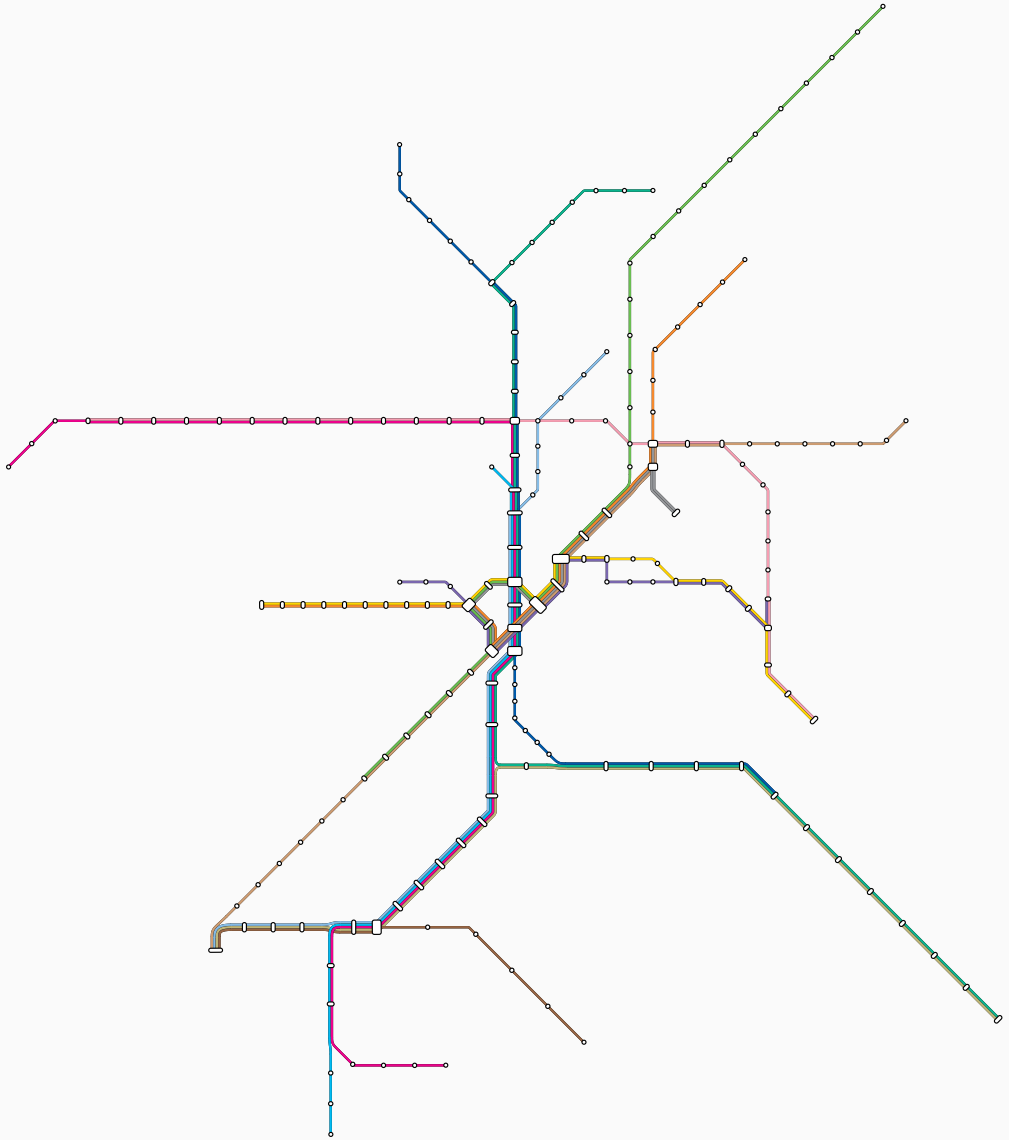
Vienna, drawn with approx. approach in **202 ms**.

Results



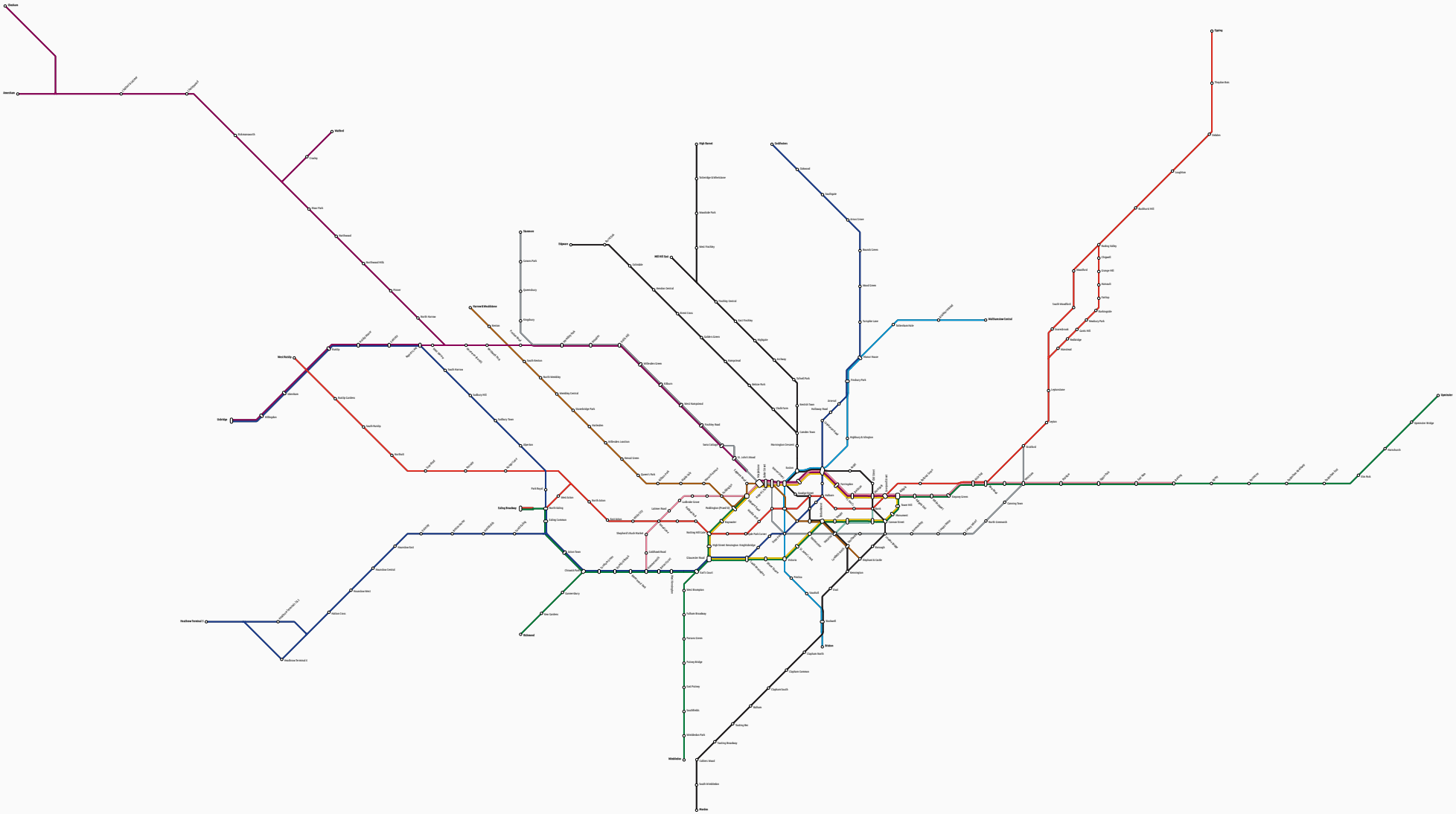
Berlin, drawn with approx. approach in **764 ms**.

Results



Stuttgart, drawn with approx. approach in **843 ms**.

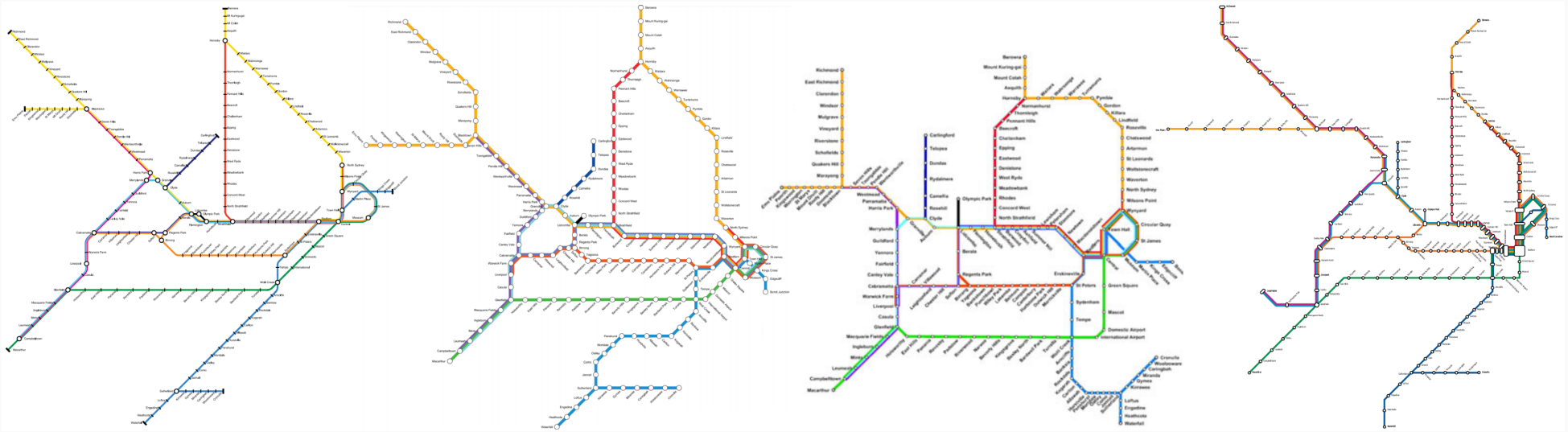
Results



London (labeled), drawn with approx. approach in **2.7 s**.

Comparison to other work

Sydney light rail network



Nöllenburg et al.
$t = 23\text{m}$

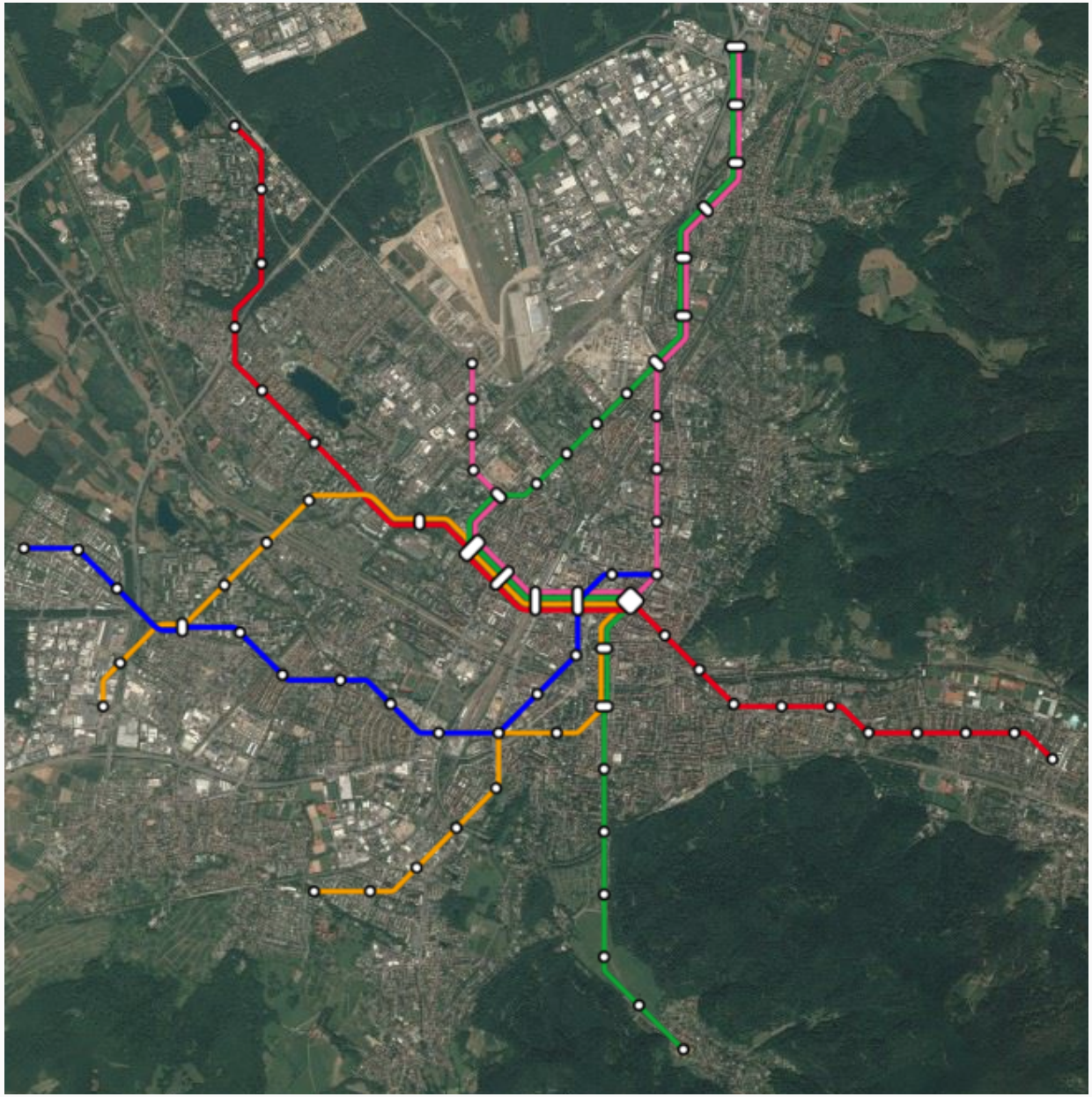
Wang et al. (2011)
$t = 816\text{ms}$

Wang et al. (2016)
$t \approx 150\text{ms}$ *

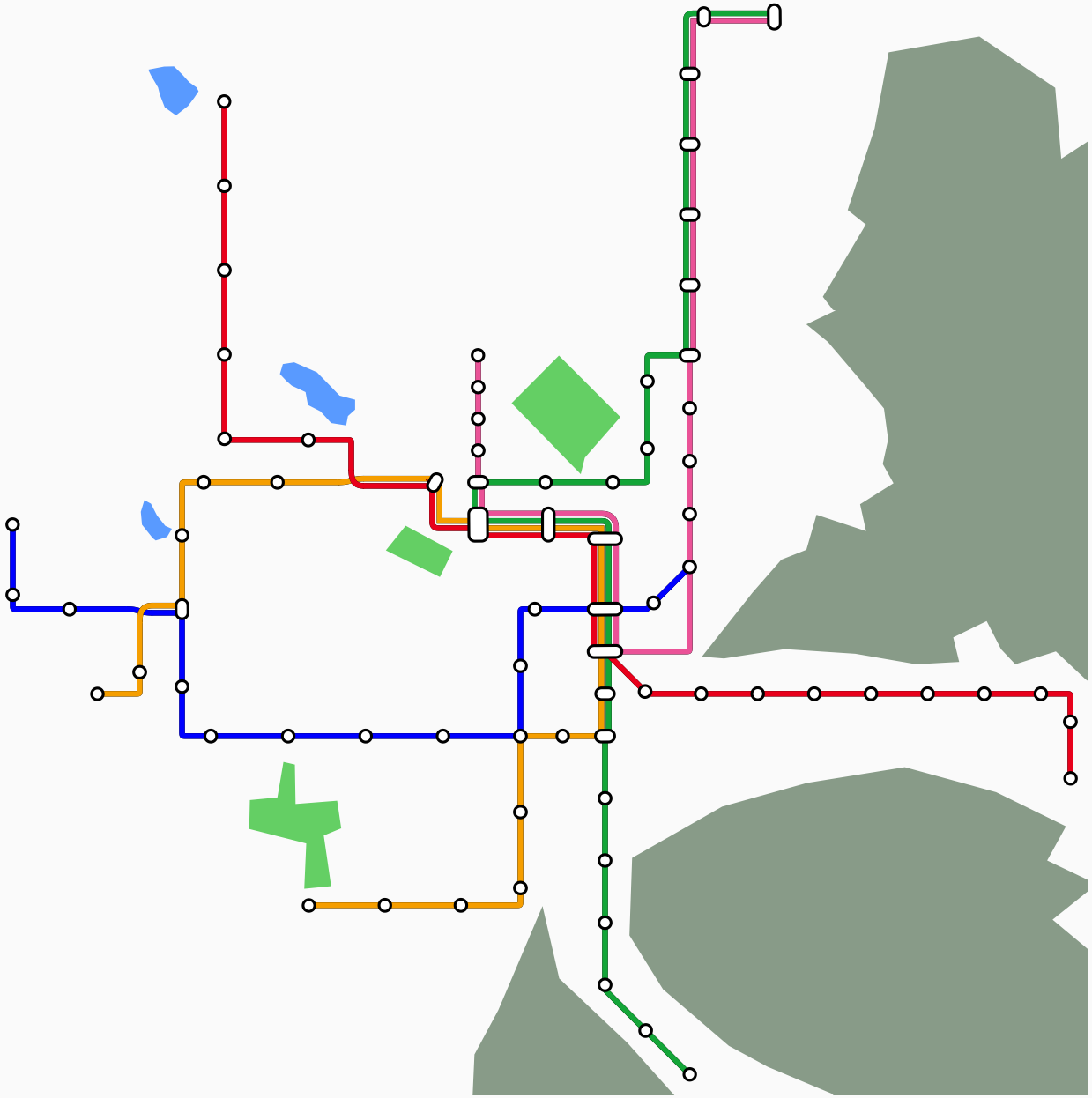
Our approach
$t = 370ms$

* No time was reported, given time is for a network of similar size (Berlin)

Results



Results



Future Work

Our approach allows for several refinements and improvements:

- Labeling only rudimentary so far and optimized separately

Future Work

Our approach allows for several refinements and improvements:

- Labeling only rudimentary so far and optimized separately
- Use grid graphs with different node densities

Future Work

Our approach allows for several refinements and improvements:

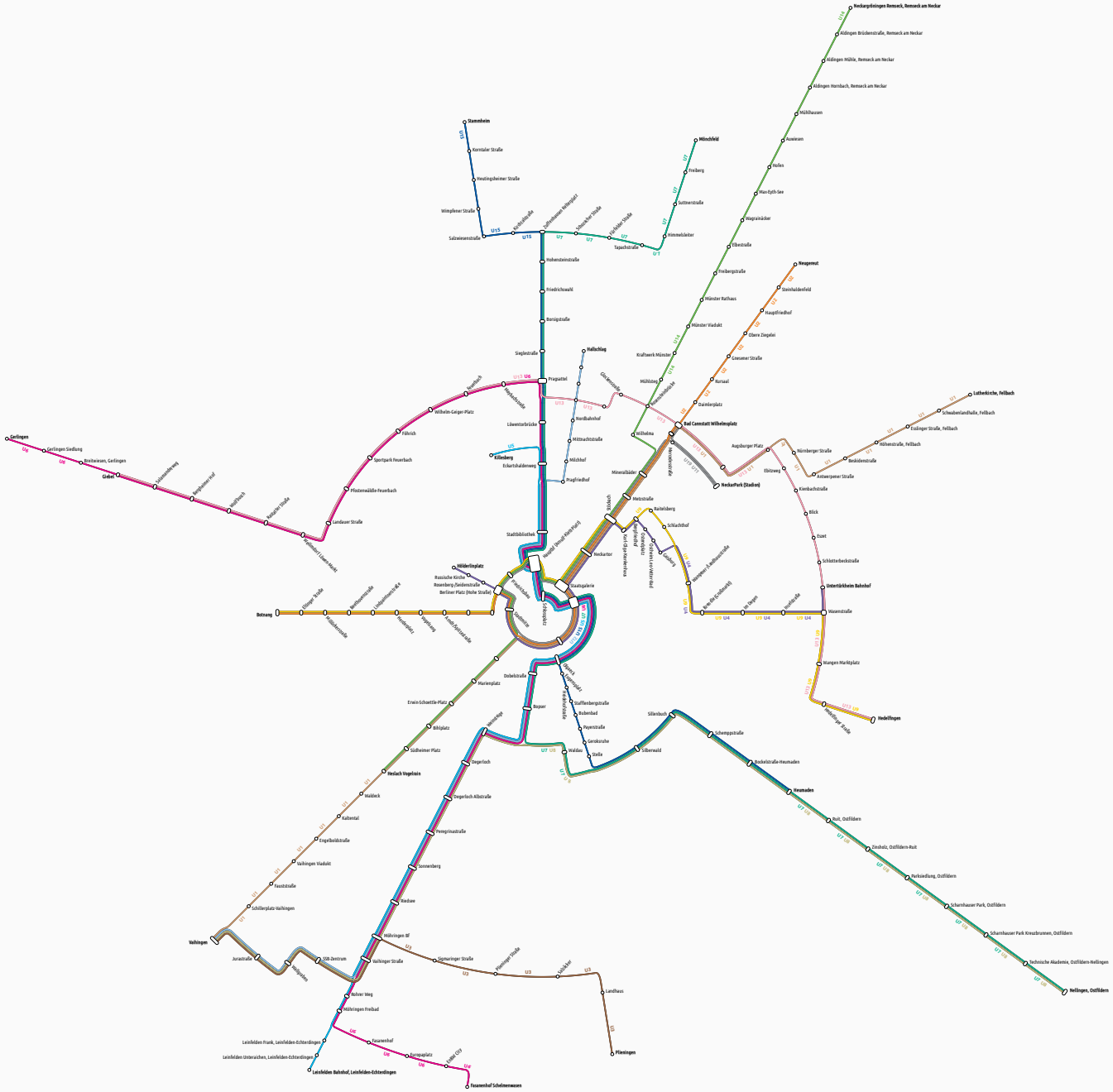
- Labeling only rudimentary so far and optimized separately
- Use grid graphs with different node densities
- Include previously developed clustering techniques for local search

Future Work

Our approach allows for several refinements and improvements:

- Labeling only rudimentary so far and optimized separately
- Use grid graphs with different node densities
- Include previously developed clustering techniques for local search
- Use different base grids

Outlook: Orthoradial Base Grids



Thank you!

http://octi.informatik.uni-freiburg.de