Metro Maps on Flexible Base Grids

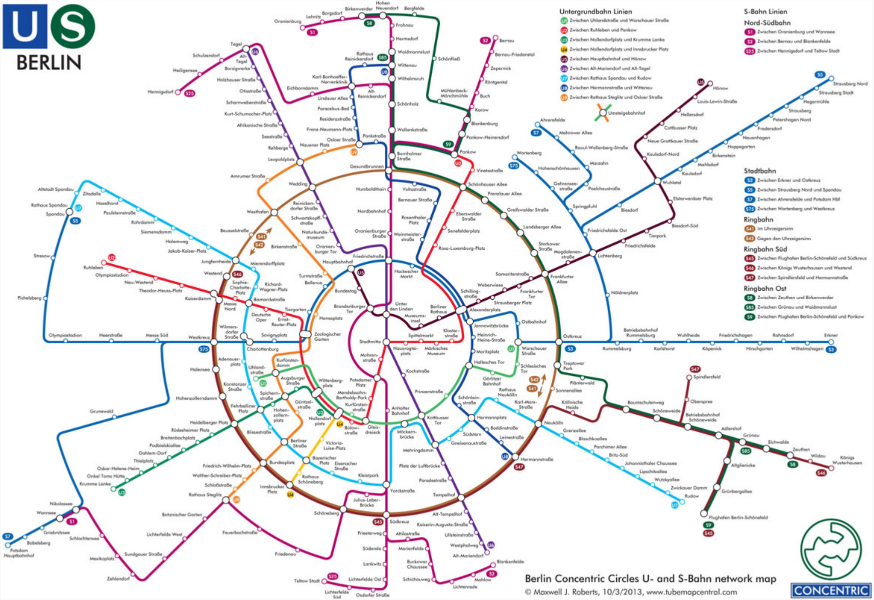
Hannah Bast [1], Patrick Brosi [1] and Sabine Storandt [2]

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

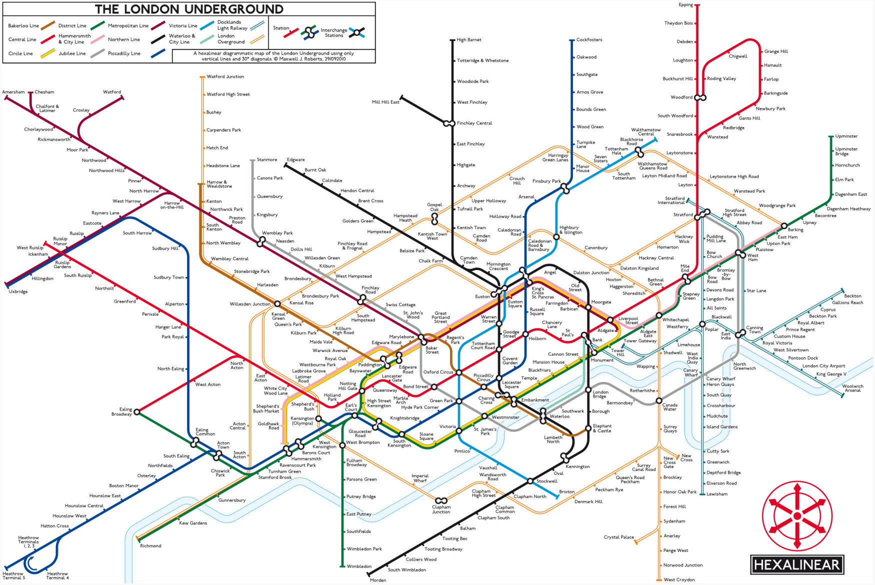
17th International Symposium on Spatial and Temporal Databases
August 23-25, 2021

Motivation - Octilinear Berlin Subway Map

Motivation - Orthoradial Berlin Subway Map (incl. local trains)

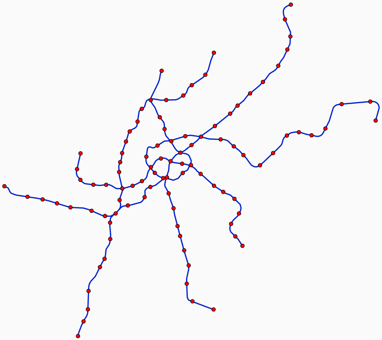


Motivation - Hexalinear London Subway Map

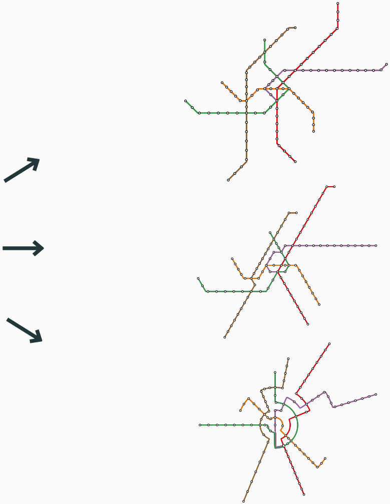


Goal

Given a line graph $G = (V, E, L, \mathcal{L})$ with edge lines $L(e) \subseteq \mathcal{L}$, render a schematic drawing of G following a predefined layout



Input line graph G



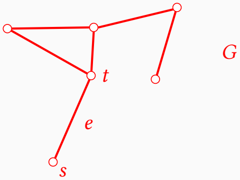
Goal (ctd.)

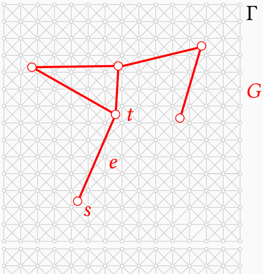
- Allow arbitrary number of edge bends to circumvent obstacles and approximate geographical courses
- Preserve the topology of the input graph (no crossings, preserve the (circular) edge order at nodes)
- Maintain a minimim distance between nodes
- Optimize number and accutenes of bends, node displacement, segment length

Existing Approach

Basic idea: Find a minimum-cost image of G in an octilinear grid graph Γ covering the (padded) bounding box of G.

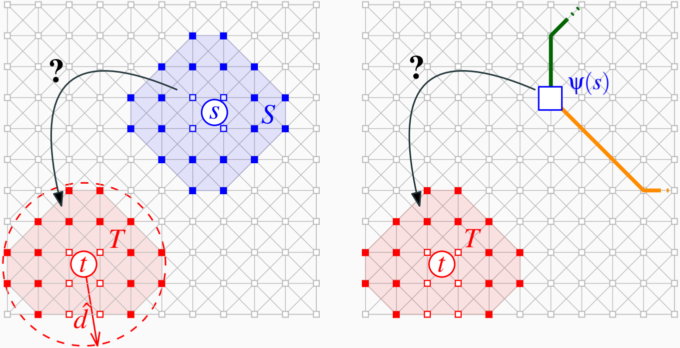
1. Build (octilinear) grid graph on which edge bends in paths are penalized
2. For each $v \in V$, find image node $\mathcal{V}(v)$ in Γ.
3. For $e \in E, e = (s, t)$, find image **path** $\mathcal{P}(e)$ through Γ.
4. Image paths must start at corresponding image nodes and must be node-disjoint.





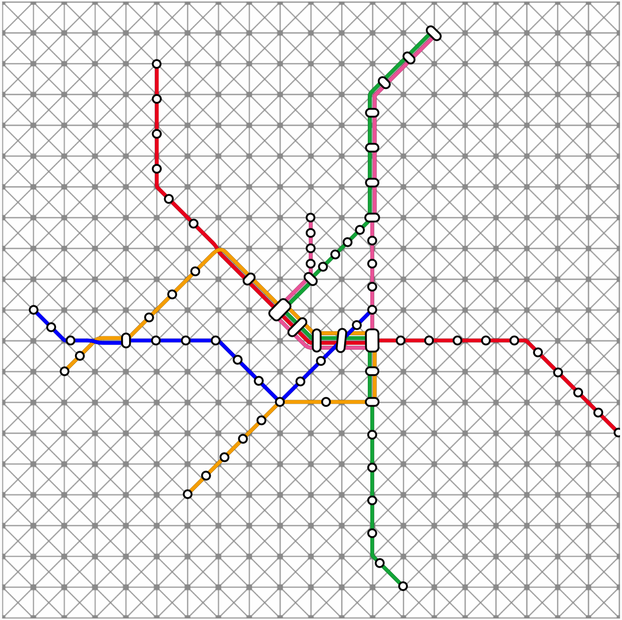
Existing Approach

Optimization either globally using integer linear programming (ILP) or an approximate approach (A): greedily route image path through the grid, local search on neighboring node positions for final polish.



Near-optimal drawing typically found in under 1 second.

Example (before local search)

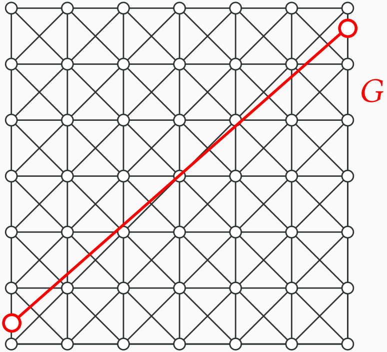


Problems

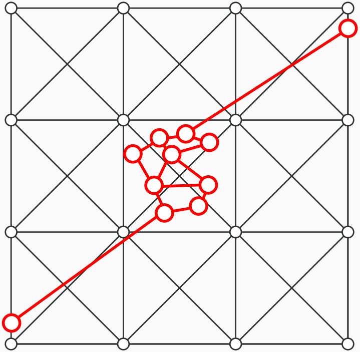
- Only node degrees ≤ 8 supported **see paper**
- Only octilinear layouts **this presentation**
- Inadequate grid graph density **this presentation**
- Stalling of the approximate approach **see paper**

Problem: Grid Graph Density Problems

Grid density too high



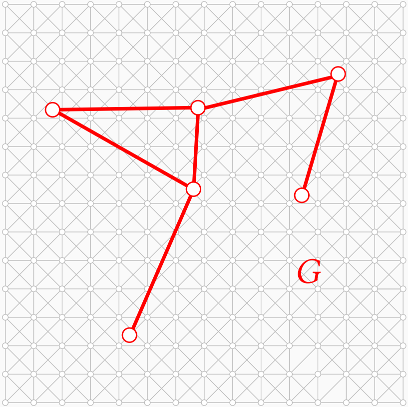
Grid density too low

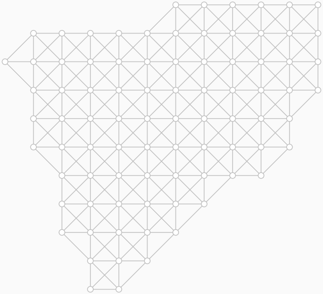


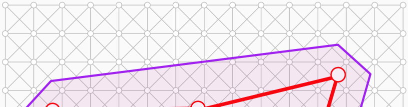
We would like to have adaptive sparse base grids.

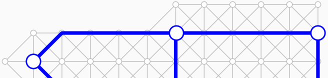
Grid Graph Density: Convex Hull Grids

Simple idea: Crop the grid graph to the (padded) convex hull of the input graph nodes.





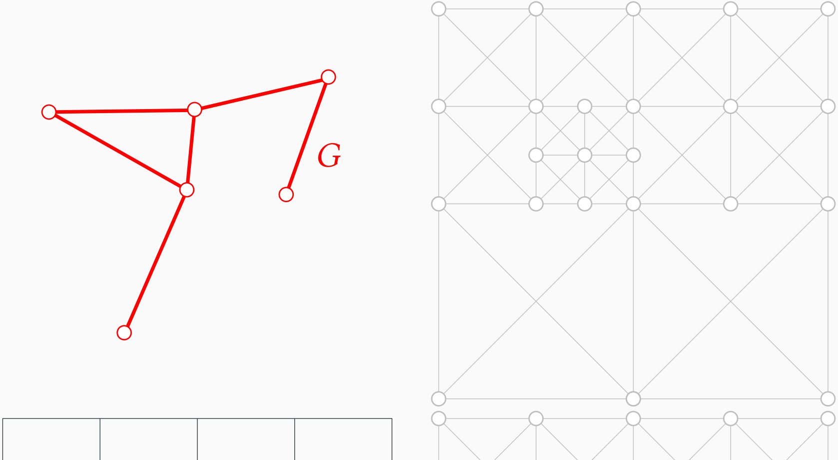




Quadtree Grids

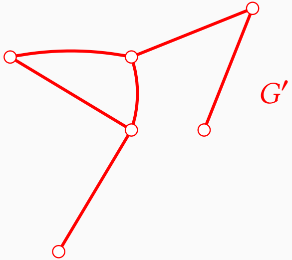
Idea: Build a quadtree from the input nodes.

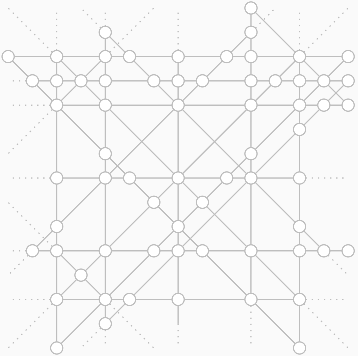
Ensure that each cell only contains one input node, but maintain a minimum cell size.

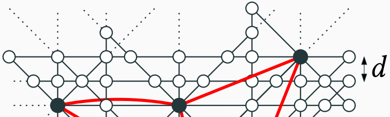


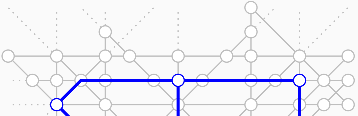
Octilinear Hanan Grids

Idea: Build an octilinear Hanan grid (vertical, horizontal, and diagonal lines through each input node, add nodes at intersection points)



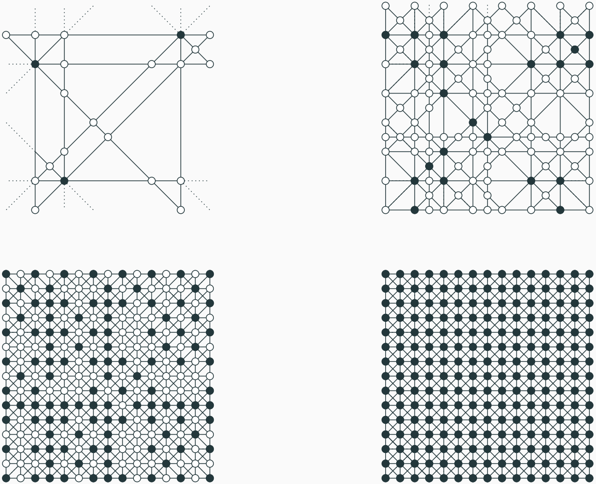






Hanan Iterations

Idea: Take the nodes of the octilinear Hanan grid as input points for another octilinear Hanan grid)



Results - Sparse Grid Sizes

Grid graph size reductions (measured in number of edges) when compared to the full grid

	Convex Hull	Quadtree	OHG-1	OHG-2*
Freiburg	51%	75%	68%	0%
Vienna	57%	81%	74%	0%
Stuttgart	43%	80%	62%	0%
Berlin	45%	83%	67%	0%
Sydney	37%	87%	74%	0%
avg	47%	**80%**	69%	0%

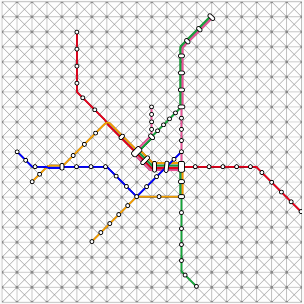
* After second iteration, all but one OHG were already the full grid.

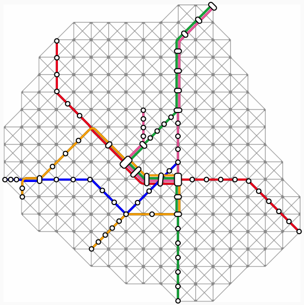
Results - Sparse Grid Quality Loss

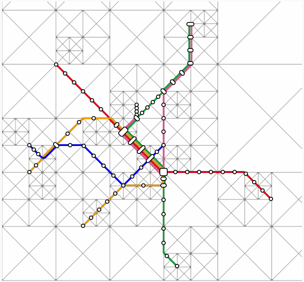
Average additional approximation error of sparse grids

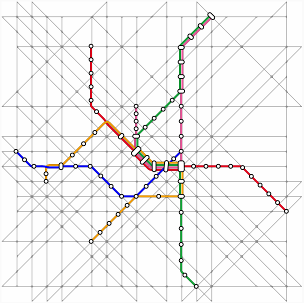
	Convex Hull	Quadtree	OHG-1	OHG-2
ILP	1%	10%	2%	1%
A	-1%	9%	2%	0%
A+D	3%	40%	8%	0%

Results - Visual Comparison









Results - Sparse Grid Speed Up

Solution time reduction by sparse grids, on average

	Convex Hull	Quadtree	OHG-1
ILP	-169%	-66%	-136%
A	7%	-90%	9%
A+D	14%	-32%	-13%

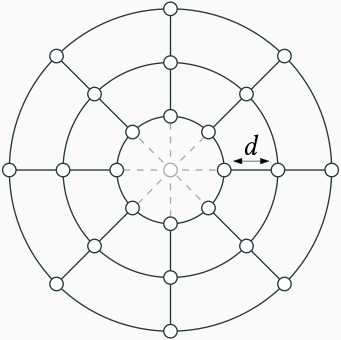
Results - Sparse Grid Speed Up

Solution time reduction by sparse grids, maximum time reduction

	Convex Hull	Quadtree	OHG-1
ILP	35%	77%	84%
A	57%	-1%	34%
A+D	57%	-2%	40%

Problem: Layout Flexibility

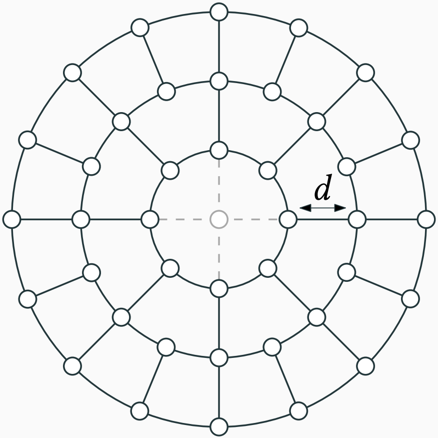
- **Ortholinear** maps: Use a classic grid graph (without diagonal edges)
- **Hexalinear** maps: Use a triangular grid
- **Orthoradial** maps: Radial grid?



Problem: Node-density decreases with distance to center.

Pseudo-Orthoradial Grids

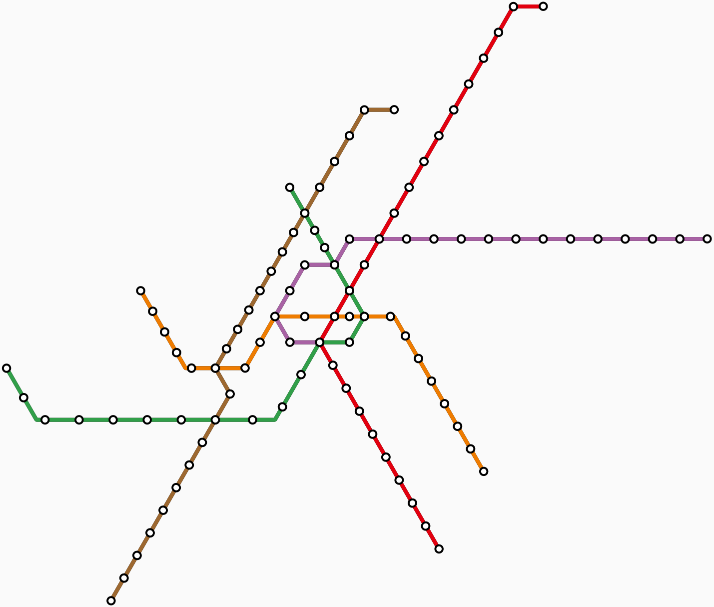
- Double number of beams each time the radius doubles
- Center at node of highest line degree



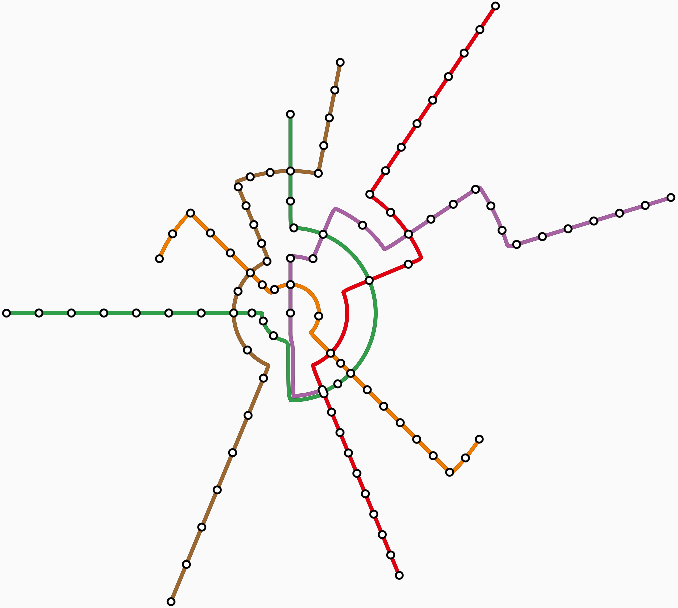
Results - Average solution times for different layouts

	Hexalinear Grid			Pseudo Orthorad. Grid		
	ILP	A	A+D	ILP	A	A+D
F	3.8m	138ms	313ms	50s	234ms	283ms
V	6.5m	146ms	654ms	18.2m	145ms	351ms
ST	43.6m	616ms	1.3s	28.7m	706ms	1.9s
B	1.8h	470ms	1.4s	8.4h	2.4s	2.5s
SD	23.5m	1.6s	2.2s	23.2m	468ms	1.7s
avg.	0.6h	594ms	1.2s	1.9h	791ms	1.3s

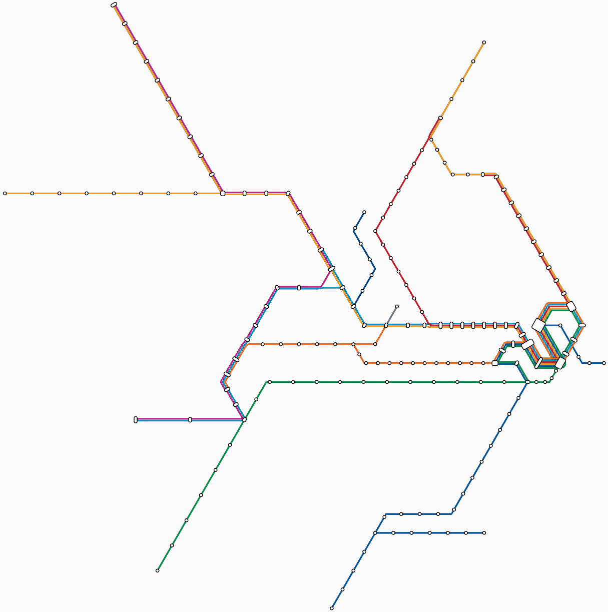
Results - Visual Inspection



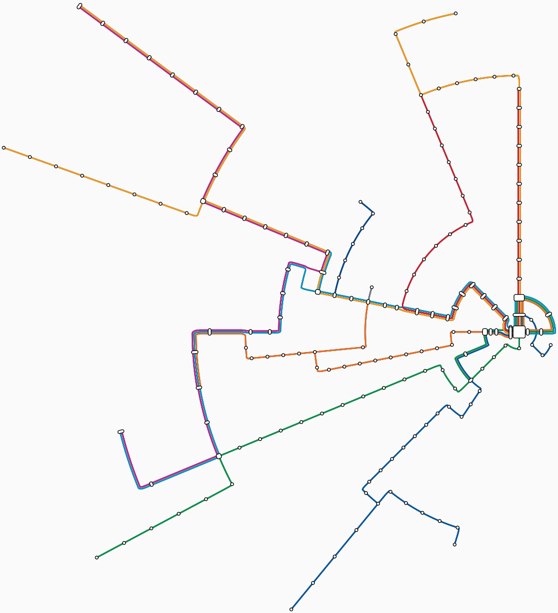
Results - Visual Inspection



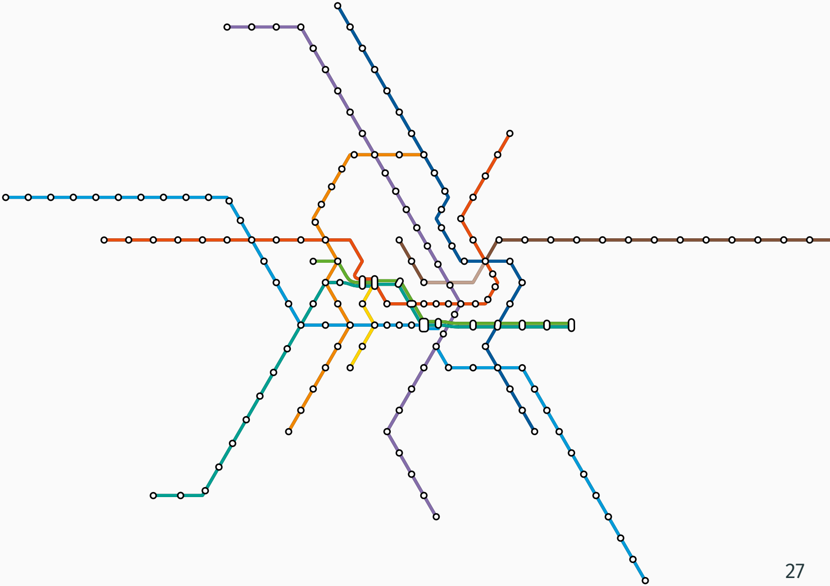
Results - Visual Inspection



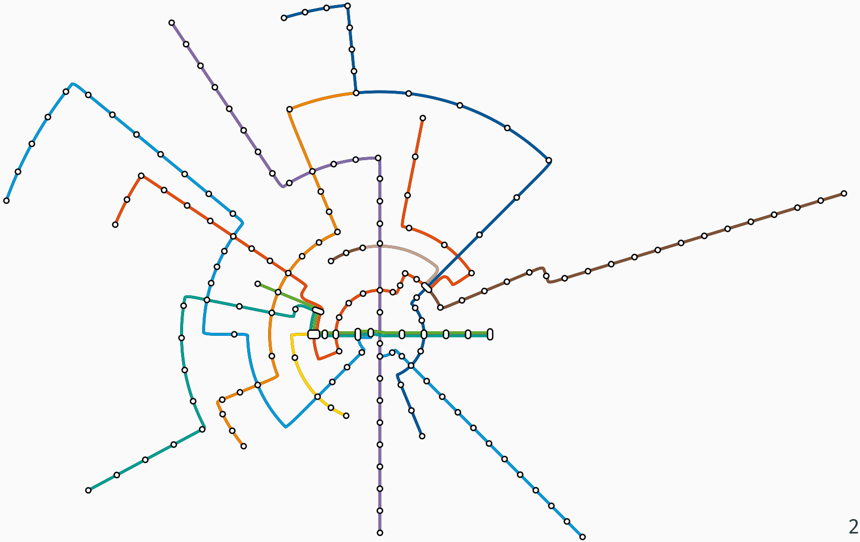
Results - Visual Inspection



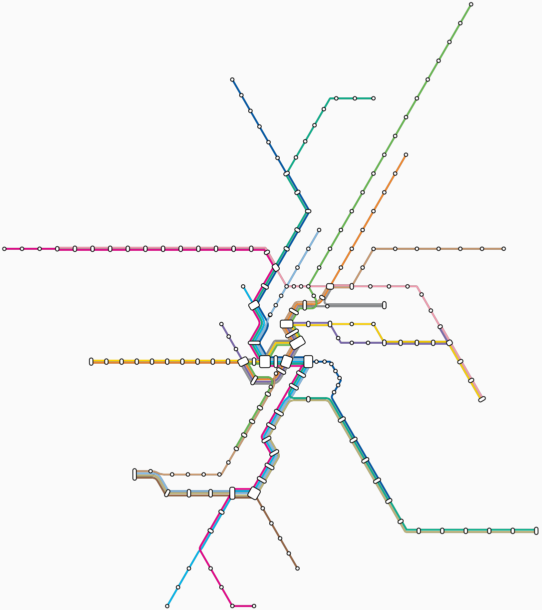
Results - Visual Inspection



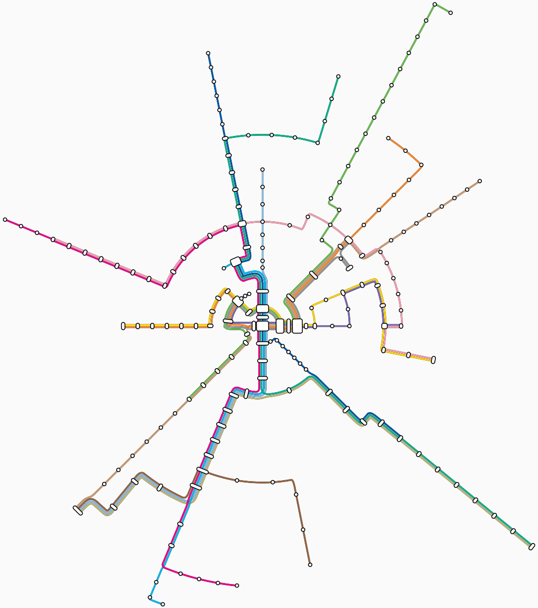
Results - Visual Inspection



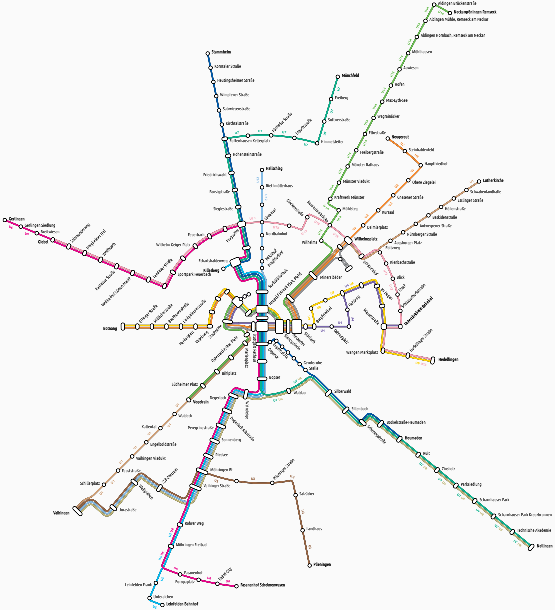
Results - Visual Inspection



Results - Visual Inspection



Results - Visual Inspection



Summary

- Sparse grids are able to reduce the problem instance size with only small effects on optimality, octilinear Hanan grids work best
- This size reduction does not always lead to speedups. Often, it takes considerably longer to find a solution.
- Our original method is able to render orthoradial, and hexalinear maps fast (< 2.5s), and in great quality.

Future Work

- Improve labeling
- Sparse grids for alternative layouts?
- Local enlargement of high-density areas without distorting the rest of the map
- Better assessment of the esthetic quality of the maps, and different target function weights

Thank you!

http://octi.cs.uni-freiburg.de/flexmaps