Towards Automated Multilevel Schematic Maps

Julian Haumer
Algorithms and Complexity
TU Wien
julian.haumer@tuwien.ac.at

Soeren Nickel
Algorithms and Complexity
TU Wien
soeren.nickel@ac.tuwien.ac.at

Martin Nöllenburg
Algorithms and Complexity
TU Wien
noellenburg@ac.tuwien.ac.at

Markus Wallinger
Algorithms and Complexity
TU Wien
mwallinger@ac.tuwien.ac.at

***Abstract*—Automatic transit map schematization focuses on creating schematic maps of real world transit networks. Usually, the scope of such schematic maps is constrained to one predefined level of the network, for example, a city's metro system or a continental rail network. In this paper we take a more holistic, multilevel view on transit networks by imposing a hierarchy that can be used to differentiate between multiple levels of the network. We present a formal model of multilevel transit networks, an extension of existing design criteria to multilevel transit maps, and a preliminary exploration of an algorithmic approach based on mixed-integer linear programming to create octolinear schematic maps of two-level networks.**

I. Introduction

Schematic maps of public transit networks are abstract visual representations that simplify the underlying real-world geography of complex commuter networks with the purpose of making them more accessible to travellers. Such maps assist in navigating the network by facilitating common tasks such as finding the shortest or most comfortable path between departure and destination, counting how many stations are left before arriving at the destination, or deciding which lines and interchange stations are necessary to arrive at a destination. This is based on the assumption that travellers start and end their journey in one transit network. However, travellers might start their journey in city A and use mixed modes of transportation to arrive in city B. For example, they might take the S-Bahn to the central station in Salzburg before transferring onto a long-distance train to Vienna, where another transfer to the metro system is necessary to arrive at the destination; see Fig. 1a and Fig. 1b. Most schematic maps represent only one part of a much wider public transport network, e.g., a city's metro network, which is in fact connected to the (inter-)national railway network. Due to this limited view, a combination of different maps is required to achieve the intended navigation task. This becomes problematic and could lead to suboptimal decisions being made when different maps use different design principles and visual languages or only central hubs of connections between maps are depicted.

In this paper we take a more holistic, multilevel view on transit networks by imposing a hierarchy on the overall network that can be used to differentiate between multiple levels of the network. The hierarchy is able to model the fact that geographic distances between stations have a different magnitude on different levels, or that certain subnetworks are physically disconnected. For example, the lowest level of a public transit network could consist of multiple, disconnected, metro networks of different cities, whereas the next level represents long-distance trains connecting the different cities via the central stations in their respective local networks. Note that this hierarchical view is conceptually different from jointly mapping two distinct but tightly intertwined networks like a bus and a metro network in the same city.

There are some hand-crafted maps incorporating multiple levels of transit networks into one schematic map, e.g., see Figs. 1b and 1c. To the best of our knowledge, automatic schematization of transit networks has not been studied for multilevel transit networks.

Our contributions are a formal model of multilevel transit networks, an extension of existing design criteria to multilevel transit maps, and a preliminary exploration of an algorithmic approach based on mixed-integer linear programming to create octolinear schematic maps of two-level networks.

II. Related Work

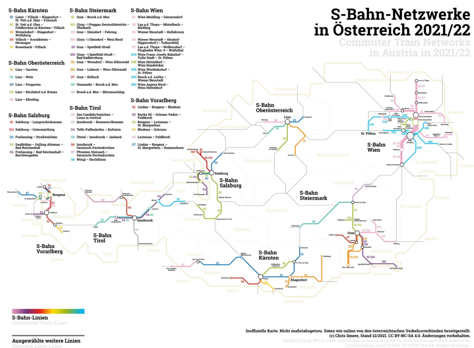
The existing research on automating transit map layout, which is summarized in two survey papers by Nöllenburg [8] and Wu et al. [16], has focused on schematizing individual metropolitan transit networks. More recently, Wu et al. [17] published a broader state-of-the-art report that combines human, machine, and design perspectives on transit map layout. While most automated methods share a common set of design rules and optimization criteria, they apply different algorithmic techniques, each with their own trade-off between computation time and layout quality. Prominent examples of algorithmic techniques used in the last decade include mixed-integer linear programming (MILP) [7], [9], [10], grid-based path routing methods [1], [2], force-directed algorithms [4], [5], or multi-criteria local optimization heuristics [12]–[15].

III. Multilevel Metro Maps

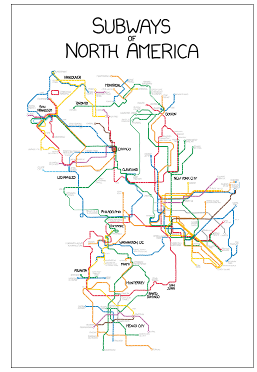
So far, no formal definition of multilevel metro maps has been introduced in the literature. We are primarily interested in extending existing models, as those surveyed by Wu et al. [17], with the concept of multiple levels of a transit network.

The input data for computing a multilevel metro map consists of (i) a connected graph $G = (S, C)$ modeling a transit network with vertex set S representing the stations and edge set C representing the physical connections between stations, (ii) an initial layout $\mathcal{L}$ mapping each vertex $s \in S$ to a point $p \in \mathbb{R}^2$ and each edge $c \in C$ to a curve between its incident vertices, (iii) a set $\mathcal{D}$ of paths (representing the

(a) Salzburg S-Bahn [3]



(b) Austrian S-Bahn [11]



(c) XKCD Metro [6]

Fig. 1: Schematic maps of local transport networks, e.g. S-Bahn network of Salzburg (a), often show only one level of the public transport network, even though they might be connected by long-distance links, as seen in (b) for Austria. Also, artistic works of multilevel transit maps have been published (c).

transit lines) whose union covers all edges in C, and (iv) a hierarchy $\mathcal{H}$ representing the different levels of the network. More specifically, $\mathcal{H} = (h, H)$, where $H = \{C_1, \ldots, C_k\}$ is a set of edge subsets of C and $h\colon H \to \mathbb{N}$ is a function mapping each edge subset in H to a level such that for any C_i, C_j with $h(C_i) = h(C_j)$ the induced subnetworks of C_i and C_j are disjoint. Furthermore, we use the notation H^j to reference all edge-induced subnetworks of level j. Here, we restrict the number of levels to two, i.e., $h\colon H \to \{1, 2\}$. This captures the perspective of a typical national railway network consisting of multiple metropolitan commuter networks that are interconnected by long distance railways.

The algorithmic task is to compute a 2-level hierarchical schematic transit map layout $\mathcal{L}'$ of a given input instance $\mathcal{M} = (G, \mathcal{L}, \mathcal{D}, \mathcal{H})$ that satisfies a set of design criteria.

Design Criteria. Our design criteria are based on the most common principles found in the literature for traditional octolinear metro maps and extend them to multilevel maps. These criteria aim to balance the trade-off between layout simplicity and geographic accuracy. Even though several publications have postulated their own sets of design criteria, e.g. as in [17], we take the more graph theoretical perspective of [9].

- **R1 – Octolinearity.** All edges must have an octolinear orientation (horizontal, vertical, 45°-diagonal).
- **R2 – Topology Preservation.** The schematic map must preserve the topology of the input network.
- **R3 – Straightness** Individual transit lines should have no bends if possible or obtuse bend angles otherwise.
- **R4 – Relative Positions Preservation.** Stations must preserve their relative position to neighbouring stations.
- **R5 – Edge Uniformity.** The distances between pairs of adjacent stations should be as uniform as possible.
- **R6 – Station Separation.** Unrelated transit lines and stations must keep a sufficiently large distance.

As we are concerned with schematizing a hierarchy of networks, several new design principles reflecting intended visual properties need to be introduced. This is guided by the idea that all edges are drawn octolinearly and different networks on the same level must be visually discriminable. Rules R1–R4 need to be respected by the layout regardless of the hierarchy level. The following two design criteria extend the traditional perspective to multilevel networks.

- **H1 – Network Separation.** Subnetworks, e.g., metro networks of different cities, should be visually discernable. Stations of different subnetworks should have larger distances between them than between other stations of their respective subnetwork.
- **H2 – Network Distortion.** The relative distances between subnetworks should represent the true geographic distances between the networks in the input graph.

IV. Algorithm

As previously stated the input to our algorithm is a tuple $\mathcal{M} = (G, \mathcal{L}, \mathcal{D}, \mathcal{H})$ which we process using a multi-step pipeline. In summary, (i) first the input is pre-processed, (ii) then schematic layouts for each subnetwork in H^1 are computed individually, which are (iii) post-processed to create a hull of the respective layout shape and (iv) finally a full schematic layout is computed by arranging all subnetworks and inserting the edges connecting them. We assume that every station $s \in S$ is assigned to the vertex set of exactly one edge-induced subnetwork G_i in H^1.

A. *Preprocessing*

Our algorithm starts by planarizing the input network by replacing all pair-wise edge crossings (if any) with dummy vertices at the respective intersection points and connecting them to the endpoints of the crossing pairs. We assign the dummy vertices to a network in H^1 by majority vote of adjacent stations and break ties arbitrarily. Afterwards, we continue processing every subnetwork $G_i \in H^1$ individually. First, we simplify segments of metro lines, i.e., sequences of consecutive edges between interchange stations or terminals, by contracting stations with degree-2 and their incident edges until only two stations are left on each segment. It is important to keep stations that connect to different subnetwork in H^2, therefore they are not contracted. Consequently, we increase the minimal length of the remaining edges to reflect the number of contractions. For line segments with less than two stations we add up to two dummy vertices to allow for line bends. After contraction we add *external stations* to each network G_i for all edges that connect to some other network G_j in H^2. The added external stations represent hub stations that constrain the layout of G_i to ensure we leave enough space when connecting to network G_j in a later stage of the pipeline. We denote the augmented graph as G_i^a.

B. First Level MILP Model

In this stage we apply the MILP model introduced by Nöllenburg and Wolff [9] to each augmented subnetwork, with a minor extension for our use-case. In summary, we define linear equations as constraints that model the previously discussed design criteria and define a weighted cost function to be minimized. The constraints reflect design criteria that we consider hard constraints, i.e. octolinarity (R1), topology preservation (R2), relative position preservation (R4) and station separation (R6). The cost function reflects soft constraints to be optimized, which are uniform edge length (R5), straightness (R3), and relative position preservation (R4). The assigned weight for each cost factor reflects the impact it should have on the computed layout. For example, a low weight for uniform edge length might lead to a layout that has a higher straightness but uses more area and has a higher variation of edge lengths.

The extension we introduce is necessary to properly prepare a subnetwork of H^1 to be integrated into the overall network later on. As we augmented the subnetwork with external stations, the edges incident to them need to be laid out in conjunction with the remainder of the subnetwork as, e.g., an edge connecting a station in the center of the network to an external station cannot be easily added to the layout later on. Since the external stations lie somewhere outside of the subnetwork, we introduce additional constraints to the mathematical model that enforce external stations to be positioned at least one unit further away in one of the octolinear directions than any other non-external station. At the end of this stage we have a layout $\mathcal{L}_i^a$ for every G_i^a in H^1.

C. Post-processing of Individual Networks

Next, we apply some post-processing on the previously created schematic layouts by creating two copies of layout $\mathcal{L}_i^a$ of G_i^a. For the first copy we remove the external stations and undo the removal of degree-2 stations. Basically, we create a schematic layout $\mathcal{L}_i$ of the unaugmented graph G_i. The second copy of $\mathcal{L}_i^a$ will be repurposed into a simplified version of itself that can be incorporated into a network that represents the structure induced by H^2. First, we compute the convex octolinear hull which is done by moving lines from all octolinear directions towards the layout until we hit the first non-external station. The octolinear convex hull guarantees non-overlapping networks in the final layout as required by design criteria H1. Second, we want to expose the stations of G_i that are adjacent to other subnetworks on the convex hull. In case a neighbor s of an external station does not already lie on the convex hull, we can use the fact that the connecting external station is outside of the hull. Therefore, we compute projections of s that are placed in the intersections of the boundary and the edges that connect s to external stations. In the end, we only keep the convex, octolinear hull and the stations exposed on it. In case stations were projected onto the hull, we keep the original stations, their projections, as well as edges that connect each station to its projections as they influence the optimization in the next section. We denote this simplified graph of G_i^a as G_i^s and its layout as $\mathcal{L}_i^s$.

D. Second Level MILP Model

After individually processing all networks of H^1 we are left to include the structure induced by H^2 to get an intermediate layout $\mathcal{L}^2$. First, we need to position the individual layouts $\mathcal{L}_i^s$. This is done by computing the centroid of each subnetwork's layout and aligning it with the respective centroid in the original layout $\mathcal{L}$. To ensure non-overlapping layouts in $\mathcal{L}^2$ we scale down the layouts until all overlaps are removed.

Next, we process and add edges connecting different subnetworks G_i^s via the stations exposed on the boundaries. For each edge we add two dummy vertices. This subdivision permits two line bends during the schematization. After this process we have a graph G^2 representing a network of G_i^s and its associated, non-schematized, layout $\mathcal{L}^2$.

Finally, the graph G^2 is processed with the same MILP model of Sec. IV-B. Boundaries are modelled as metro lines with additional constraints preventing that their respective shape changes. Additionally, we introduce constraints that prevent line bends over projections of internal stations. Informally, the model we created allows subnetworks to be moved as one unit while edges between subnetworks can bend. With this model we mainly optimize for design rules H1 and R1-R5.

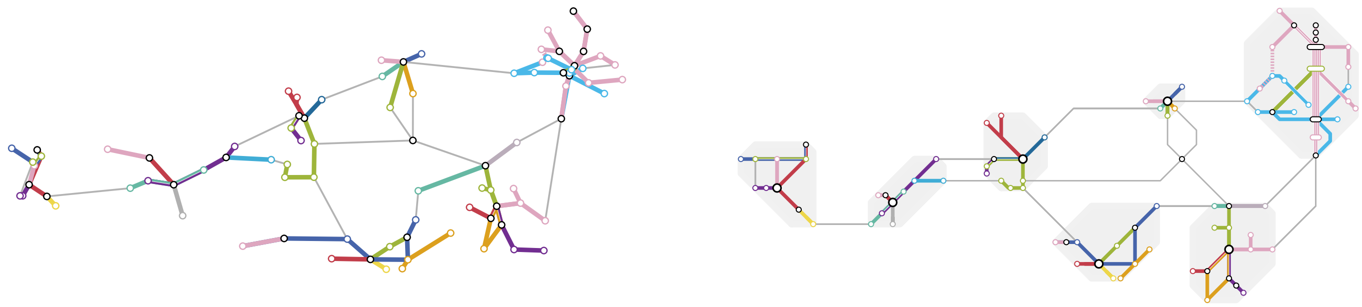
E. Post-processing of the Final Layout

Lastly, the created schematic layout $\mathcal{L}'$ requires post-processing. First, we replace the individual layouts $\mathcal{L}_i^s$ with their unaugmented layouts $\mathcal{L}_i$. Finally, all metro lines are restored where previously stations of degree-2 were merged.

V. Case Study

The presented example is based on a hand-crafted map, see Fig. 1b, of the commuter train network of Austria. In this map commuter train lines around state capitals and their respective inter-city connections are shown. As no raw dataset of the map exists we had to create an input dataset for our algorithm first, as seen in Fig. 2a. We extracted the geographic positions of the depicted stations and connected them according to the presented topology. Lastly, we removed open-ended connections to outside of Austria, as well as multi edges that exist in the hand-crafted version.

Applying our approach to the Austria data set yielded the map in Fig. 2b. The wall clock time of the computation is



(a) Input Instance

(b) Schematic Map

Fig. 2: Case study of input dataset of the Austrian commuter train network with stations at their geographic locations (a) and schematized by our approach (b). Widening of stations, parallel routing and dotted stylization of lines and subnetwork labels were replicated manually. The boundaries of the schematization of each subnetwork is shown as a gray backdrop.

approximately four minutes with Gurobi 9.5 on an 2.9Ghz Intel Core i7 with 16GB of memory. Our schematic map clearly separates subnetworks while keeping their relative positions. Similarly, the relative positions are preserved in the individual subnetworks as well. However, in the hand-crafted map (Fig. 1b) distances are more geographically accurate, because our model currently optimizes for R3-R5 which do not model distortion. Overall, the quality of the layout is a good draft for a map designer to create a final version.

VI. Conclusion

Schematic maps of multilevel transit networks provide a holistic view on public transport networks that goes beyond existing results. They could provide an interesting addition to the current state of the art. Additionally, we think that the quality of the case study demonstrates that the approach can be a valuable input for map designers.

Limitations. Even though the MILP model worked well in our case study, it is computationally complex. For more complex input it might not always find layouts of subnetworks in reasonable time or at all. Also, we assume that the imposed hierarchy creates non-overlapping subnetworks.

Open Questions. While our approach uses a MILP model to compute layouts, there are other approaches that provide layouts of similar quality. Especially the grid graph model [1], which introduced a fast, heuristic-based approach, could be interesting. Possibly, other sets of design criteria could yield faster or visually better results. Furthermore, the multi-stage pipeline approach can be reduced to a single MILP model that encodes different design criteria depending on which level a station or edge is used in. Mainly, the multi-stage pipeline is necessary to reduce computation time of the MILP. Another crucial question is how to integrate labeling, such that stations, lines and areas have meaningful, non-overlapping names. Also, it could be of interest to investigate the proposed schematic map from a user perspective and scientifically re-evaluate the proposed design criteria. Finally, the question arises if sufficient information can be communicated on a static map or if interaction is necessary.

References

[1] H. Bast, P. Brosi, and S. Storandt. Metro Maps on Octilinear Grid Graphs. *Computer Graphics Forum*, 39(3):357–367, 2020.

[2] H. Bast, P. Brosi, and S. Storandt. Metro Maps on Flexible Base Grids. In *Symp. Spatial and Temporal Databases, SSTD*, pages 12–22. ACM, 2021.

[3] S. Bilko. Liniennetzplan der S-Bahn Salzburg, 2006. [Online; accessed December 23, 2021].

[4] D. Chivers and P. Rodgers. Octilinear force-directed layout with mental map preservation for schematic diagrams. In *Diagrammatic Representation and Inference (DIAGRAMS'14)*, volume 8578 of *LNCS*, pages 1–8. Springer, 2014.

[5] M. Fink, H. J. Haverkort, M. Nöllenburg, M. J. Roberts, J. Schuhmann, and A. Wolff. Drawing Metro Maps Using Bézier Curves. In *Graph Drawing, GD*, volume 7704 of *LNCS*, pages 463–474. Springer, 2012.

[6] R. Munroe. XKCD 1196 – Subways of North America, 2013. [Online; accessed December 23, 2021].

[7] S. Nickel and M. Nöllenburg. Towards Data-Driven Multilinear Metro Maps. In *Diagrammatic Representation and Inference, Diagrams*, volume 12169 of *LNCS*, pages 153–161. Springer, 2020.

[8] M. Nöllenburg. A Survey on Automated Metro Map Layout Methods. In *1st Schematic Mapping Workshop*, Essex, UK, 2014.

[9] M. Nöllenburg and A. Wolff. Drawing and Labeling High-Quality Metro Maps by Mixed-Integer Programming. *IEEE Trans. Visualization and Computer Graphics*, 17(5):626–641, 2011.

[10] O. Oke and S. Siddiqui. Efficient Automated Schematic Map Drawing using Multiobjective Mixed Integer Programming. *Comput. Oper. Res.*, 61:1–17, 2015.

[11] C. Smere. S-Bahn Netzwerke in Österreich 2021/22, 2021. [Online; accessed December 23, 2021].

[12] J. Stott, P. Rodgers, J. C. Martínez-Ovando, and S. G. Walker. Automatic Metro Map Layout Using Multicriteria Optimization. *IEEE Trans. Visualization and Computer Graphics*, 17(1):101–114, 2011.

[13] T. C. van Dijk and D. Lutz. Realtime linear cartograms and metro maps. In *Advances in Geographic Information Systems, SIGSPATIAL*, pages 488–491. ACM, 2018.

[14] Y.-S. Wang and M.-T. Chi. Focus+context metro maps. *IEEE Trans. Visualization and Computer Graphics*, 17(12):2528–2535, 2011.

[15] J. M. Ware and N. Richards. An ant colony system algorithm for automatically schematizing transport network data sets. In *Evolutionary Computation, CEC*, pages 1892–1900. IEEE, 2013.

[16] H.-Y. Wu, B. Niedermann, S. Takahashi, and M. Nöllenburg. A survey on computing schematic network maps: The challenge to interactivity. In *Proceeding of the 2nd Schematic Mapping Workshop*, 2019.

[17] H.-Y. Wu, B. Niedermann, S. Takahashi, M. J. Roberts, and M. Nöllenburg. A Survey on Transit Map Layout – from Design, Machine, and Human Perspectives. *Computer Graphics Forum*, 39(3):619–646, 2020.