A Survey on Computing Schematic Network Maps: The Challenge to Interactivity

Hsiang-Yun Wu
TU Wien, Austria
hsiang.yun.wu@acm.org

Benjamin Niedermann
University of Bonn, Germany
niedermann@igg.uni-bonn.de

Shigeo Takahashi
University of Aizu, Japan
takahashis@acm.org

Martin Nöllenburg
TU Wien, Austria
noellenburg@ac.tuwien.ac.at

Abstract—**Schematic maps are in daily use to show the connectivity of subway systems and to facilitate travellers to plan their journeys effectively. This study surveys up-to-date algorithmic approaches in order to give an overview of the state of the art in schematic network mapping. The study investigates the hypothesis that the choice of algorithmic approach is often guided by the requirements of the mapping application. For example, an algorithm that computes globally optimal solutions for schematic maps is capable of producing results for printing, while it is not suitable for computing instant layouts due to its long running time. Our analysis and discussion, therefore, focus on the computational complexity of the problem formulation and the running times of the schematic map algorithms, including algorithmic network layout techniques and station labeling techniques. The correlation between problem complexity and running time is then visually depicted using scatter plot diagrams. Moreover, since metro maps are common metaphors for data visualization, we also investigate online tools and application domains using metro map representations for analytics purposes, and finally summarize the potential future opportunities for schematic maps.**

Index Terms—**Metro Maps, Graph Drawing, Metaphors**

I. INTRODUCTION

A metro map is a schematic visual representation of an underlying transit network that depicts the connectivity between metro stations and lines of a public transportation system [30]. With well-designed metro maps, travellers can effectively identify their locations and find their way or perform routing planning on a complex transportation system in a big city, such as London, Paris, or Tokyo. These maps are especially helpful since travellers often look for a quick solution for the shortest or cheapest path from station A to B, how to transfer from A to B, and how many stations left to B [33]. To support these tasks, Henry Beck introduced the so-called Tube Map of London Underground, and has proposed several drawing criteria to achieve this [14], and many extended versions have been evaluated [34] [35].

Nonetheless, the need for automatic drawing algorithms is still increasing due to the high cost and limited adaptability of hand-drawn maps. This has been considered as a difficult problem because several subproblems, including layout schematization, line crossing minimisation, and map label placement have been studied and proved as complex problems [26]. Two state-of-the-art reports from 2007 and 2014 have surveyed similar approaches before [50] [26]. Hence we focus on relatively new approaches from the last 5

years after Nöllenburg's report [26] and discuss the correlation in terms of computational complexity and interactivity, as well as the potential techniques that can be used in different research domains.

A. Problem Definition

The initial idea for a metro map lies on simplifying the layout geometry to facilitate users' comprehensive understanding of connectivity between metro stations. This allows us to formulate the drawing problem as network visualization problems, which are often studied to untangle visual clutter of the layout to improve its readability [51].

Metro Map Problem: Let us formulate the metro map problem by introducing an undirected graph $G = (V, E)$ embedded in a plane $\mathbb{R}^2$. Each vertex $v \in V$ represents a metro station and an edge $e = (v_i, v_j) \in E$ indicates the physical connection between two metro stations. In addition, a line $l \in L$ is a metro line containing a set of metro stations and edges that are defined from the corresponding metro system. Note that l is a set cover, which implies each edge e belongs to at least one $l \in L$. We call $MG = (G, L)$ here a *Metro Graph* as previously defined by Nöllenburg [26]. The input of the proposed problem is the connectivity of a Metro Graph MG together with its geographically accurate embedding, and the solution aims to find a schematic layout of MG that maximally satisfies several user-defined aesthetic drawing criteria.

B. Our Taxonomy Design

In this study, we investigate the trade-off between computation times and layout quality as well as application requirements. This is motivated by our observation that a user who is trying to create a map would accept longer computation times for an exact solution if the generated map is expected to be printed, while the user will not be patient if the application is expected to be an interactive application. To investigate our assumption systematically, we analyze several factors in this survey (see Tables I and III) that influence the computational complexity and optimality requirements of a problem together with the corresponding running time of the algorithmic approach (Table III). Our two primary topics cover algorithmic network layout techniques and station labeling techniques. The correlation between problem complexity and time complexity is then given using a scatter plot diagram visually.

This is done by categorizing publications along two primary coordinates, problem complexity and running time. These include (1) the range from local to global optimality in terms of the incorporated aesthetic criteria and objective functions, and (2) the range of suitability from static to interactive visualizations based on the computation speed. The values on the first coordinate are computed using the scoring Tables I and III. For simplicity, we assume that the degree of interactivity, running time in other words, of a schematic map algorithm is potentially linearly-correlated to the degree of criteria selected by the proposed approach. In other words, we expect the developed techniques would find a good balance between the efficiency of the algorithms and the quality of the schematic maps generated by those approaches. For better instructing readers along this assumption, we will focus our discussion on these two aspects in the following sections.

C. Tasks and Design Rules

It is known that drawing criteria are often designed based on users' effectiveness to accomplish tasks on a map. These tasks are similar to tasks on graphs, such as *Topology-Based Tasks*, *Attribute-Based Tasks*, and *Browsing Tasks* since finding shortest or cheapest paths, identifying a station of a line, and navigating along a specific route are mostly performed by the travellers. We revisit Nöllenburg's list [26] of design principles and investigate which of these serve as dominant constraints, for generating a globally optimal map, in the coming sections. In this survey, we primarily focus on two directions in this field, including network layout techniques **(N)** and labeling techniques **(L)** as summarized in Table I and Table III, respectively. Note that the selected criteria are ordered in the sense that the criterion with higher scores influences global structures of the layout, while the one with lower scores affects the layout in a local fashion. This scoring scheme will then be used as an indicator for guiding readers to navigate the diagrams created in the coming sections. Based on the aforementioned scoring scheme, we derive our novel taxonomy of metro map techniques in this survey paper.

The remainder of this paper is structured as follows. In Section II, we summarize relevant schematic network layout algorithms, and in Section III, research approaches integrating map features, mainly on text and image labels. In Section IV, we then introduce several tools that are accessible online and demonstrate the usability of the collected techniques in different research domains. Finally in Section VI, we conclude this paper and list several future directions to this topic.

II. ALGORITHMIC NETWORK LAYOUT TECHNIQUES

In this section we classify the different network layout algorithms proposed in the literature over the past 15 years along three criteria that determine the degree of globality of the algorithmic techniques. These criteria comprise combinatorial properties **(N1)** of the underlying graph structure (e.g., topology preservation), geometric properties **(N2)** of the input network and the schematic layout (e.g., directional deviations and line straightness), and the degree of optimality **(N3)** of the
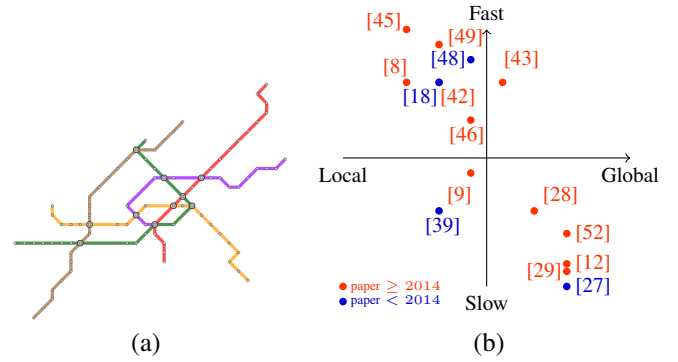

Fig. 1. Network map visualization, including an example of (1) Vienna metro map [27], and (b) layout techniques with respect to their globality and potential for interactiveness.

TABLE I
THE POINT SYSTEM FOR EVALUATING THE CRITERION EFFECTIVENESS.

ID	S	Description (x-coordinate in Figs. 1- 3)
(N1)		**Combinatorial property**
(N1.1)	4	Overall combinatorial optimization.
(N1.2)	3	Combinatorial criteria for sets of vertices or edges.
(N1.3)	2	Combinatorial criteria for pairs of vertices or edges.
(N1.4)	1	Combinatorial criteria for single vertices or edges.
(N2)		**Geometry property**
(N2.1)	4	Uniform geometric optimization.
(N2.2)	3	Geometric criteria for sets of vertices or edges.
(N2.3)	2	Geometric criteria for pairs of vertices or edges.
(N2.4)	1	Geometric criteria for single vertices or edges.
(N3)		**Approach optimality**
(N3.1)	4	Global optimality and exact global optimization.
(N3.2)	3	Global optimality, but local optimization.
(N3.3)	2	Local optimality, but global optimality for sub-problems.
(N3.4)	1	Local optimality, and local optimization.

computed solutions, i.e., whether an global optimum is computed or just a local optimum. Table I lists the three criteria, which are ranked by scores ranging from 1 (high locality) to 4 (high globality). Additionally, we assess each technique by the required computational resources and resulting degree of suitability for interactive applications (summarized in Table II).

Each of the papers discussed in the following paragraphs thus receives two scores that we use as coordinates in the two-dimensional scatter plot of Figure 1.

We start with four classic papers that serve as representatives of the main algorithmic techniques that were already discussed in the 2014 survey of Nöllenburg [26]. Since our focus is on new results from 2014 onward, we refer to the 2014 survey for a comprehensive discussion of the earlier literature. The force-based algorithm of Hong et al. [18] considers combinatorial and geometric parameters of individual and

TABLE II
THE POINT SYSTEM FOR EVALUATING THE TIME EFFECTIVENESS.

ID	S	Time complexity (y-coordinate in Figs. 1- 3)
T1	4	The layout is computed less than a second.
T2	3	The layout is computed in a few seconds.
T3	2	The layout is computed within a few minutes .
T4	1	The layout is computed in several hours or more.

pairs of edges and vertices in their objective function, e.g., the slope of edges, or the preservation of the local network topology at each vertex. The objective function considers the sum of several forces acting upon each vertex to push it toward a state of locally minimal energy. Layouts are computed within a few seconds.

Stott et al. [39] define an explicit global objective function for metro maps, which takes smaller values for maps with higher layout quality. This quality function measures, e.g., angles at each vertex or between neighboring edges, local edge length differences, or the octilinearity of edges, which corresponds to a score of 2 in these criteria. The optimization itself is by a hill climbing algorithm that moves vertices to the best position in a local neighborhood, but evaluates the global map quality. The reported computation times range between a few minutes and a few hours.

The mixed-integer linear programming (MIP) approach of Nöllenburg and Wolff [27] defines combinatorial and geometric constraints among pairs of edges and vertices to guarantee, e.g., the correct network topology, octilinear edges, or a bounded amount of angular distortion, as well as the minimization of bends along lines and subpaths or the total network length. Their solution model, however, guarantees a globally optimal solution. To achieve this highest degree of optimality, one needs to invest computation times ranging from several minutes for simple network maps to several hours for more complex instances.

Finally, Wang and Chi [48] define an energy function to represent the layout quality, with the goal of minimizing this energy function. As in the previous methods, this comprises aspects defined on pairs of vertices and edges such as edge lengths, angles, or octilinearity. Their optimization aims for global optimality, but using an iterative conjugate gradient method for least squares minimization to converge to a global minimum. Their algorithms typically run in less than a second.

Among the more recent papers, we mostly find approaches that improve and extend one of the classic techniques like MIP, local optimization heuristics, force-based iterative methods, or stroke-based incremental algorithms. Chivers and Rodgers [8] present a hybrid force-based method that defines both a set of classic spring embedder forces [13] as well as a magnetic force field that models octilinearity [41]. Their hybrid idea is to first let the spring forces find a well distributed layout and then let the magnetic forces become dominant to achieve octilinear edges. By the nature of their method they aim for locally optimal stable layouts, but the convergence is quite fast and typical metro maps can be computed in less than a second. A second and very fast approach with similar properties is given by van Dijk and Lutz [45]. They propose a method based on linearized least-squares minimization as an approximation of a non-linear model for computing linear cartograms, which are network drawings with prescribed edge lengths. This idea is then applied to metro map layout as one problem instance, where the length and slope of edges are soft constraints to be optimized. While several typical constraints of metro maps are not considered in their algorithm, it is an extremely fast tech-

nique with running times in the range of a few milliseconds. Wang and Peng [49] provide another system using a global energy function whose minima correspond to locally optimal metro layouts in their model. The optimization technique is least-squares minimization. It comprises both octilinear and curvilinear layouts and is designed for interactive editing by a user who can modify the positions of a few *handle* vertices. It is very fast and only requires a few milliseconds to compute medium-sized metro maps.

Three papers take the idea of decomposing the network into paths (also called *strokes*), computing schematic representations of the strokes and then composing them into a single network again, see e.g. [20]. Ti and Li [42] and Ti et al. [43] both propose an approach that first detects and enlarges areas in the geographically accurate representation of the metro network that have high vertex or edge density and then apply multifocal fisheye transformations to get a more uniform spatial distribution of the underlying network. In a second step, strokes are identified and locally schematized in an octilinear fashion. While no running times are reported in the papers, these methods are typically quite fast and run within at most a few seconds. On the other hand, the schematization only minimizes local distances to the input geometry and no explicit global objective function is optimized. The algorithm by van Dijk et al. [46] proposes schematic maps using strokes that are represented as circular arcs, and not as octilinear paths. They first simplify sequences of edges of the same metro line into longer paths, even across junctions, to reduce overall complexity of the layout and then find locally optimal representations of these paths/strokes by circular arcs guided by the Fréchet distance between paths and circular arcs.

Chivers and Rodgers [9] extend the search space of the hill climbing local search technique [39] by parameterizing grid spacing, local neighborhood size for node movements, and the number of iterations. The constraints and objectives are the same as those by Stott et al. [39]. With their improvements they gain a speed-up factor between 5 and 8 compared to the earlier multi-criteria hill climbing technique [39] and obtain the final (labeled) layouts within 5 to 60 minutes.

Finally, the MIP model has been improved and accelerated by several authors. Oke and Siddiqui [28] relax some of the integrality constraints of Nöllenburg and Wolff [27] and drop one term in the objective function. Their model improves the running time by up to one order of magnitude to computation times of few minutes. Onda et al. [29] also modify the existing MIP model [27]. They split the computation into two phases in order to speed up computation of large instances such as the Tokyo metro map. In the first phase they generate a rough layout that satisfies all hard constraints in an incremental face-by-face manner. The second phase optimizes the directions of short subpaths while keeping the directions of the remaining layout fixed. A layout of the Tokyo network was computed within 5 hours. Certain topological structures can be highlighted by Wu et al., who allow for straightening a user-specified path [55], or deforming a cyclic path into a circle [52] in order to emphasize meaningful structures. This

has been achieved by introducing additional constraints to the standard MIP model, which increases the time for finding the corresponding solutions by a small amount. Fink et al. [12] use the MIP technique to model layout of concentric metro maps, where the network consists of polylines composed of radial or circular segments. The optimization considers among others bend minimization, minimization of distinct radial slopes, or uniformity of edge lengths. The reported running times are in the range of several minutes for small input maps like Vienna and Montreal.
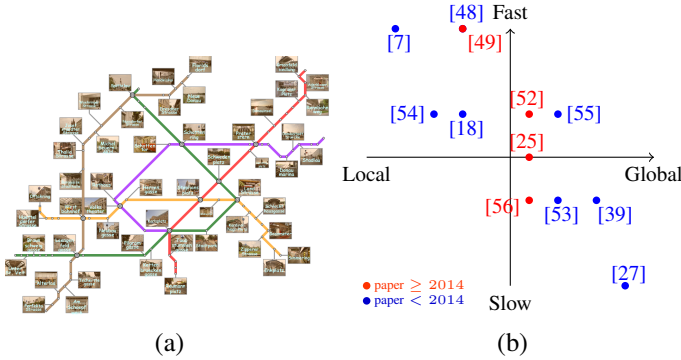
III. LABELING TECHNIQUES

Fig. 2. Labeled network maps, including an example of (1) Vienna metro map with image labels [53], and (b) labeling techniques with respect to their globality and interactiveness.

TABLE III
THE POINT SYSTEM FOR EVALUATING THE CRITERION EFFECTIVENESS.

ID	S	Description (x-coordinate in Figs. 2)
(L1)		**Adjustment of metro maps**
(L1.1)	4	The layout and label placement is done simultaneously.
(L1.2)	3	The layout is uniformly scaled to fit in labels.
(L1.3)	2	The directions of the edges are preserved.
(L1.4)	1	The layout is not changed when labeling is created.
(L2)		**Placement of labels**
(L2.1)	4	Uniform placement.
(L2.2)	3	Criteria for sets of labels.
(L2.3)	2	Criteria for pairs labels.
(L2.4)	1	Criteria for single labels only.
(L2)		**Optimality**
(L3.1)	4	Global optimality and exact global optimization.
(L3.2)	3	Global optimality, but local optimization.
(L3.3)	2	Local optimality, but global optimality for sub-problems.
(L3.4)	1	Local optimality, and local optimization.

In this section we discuss the current state on algorithmic labeling techniques for metro maps. Similarly to Section II, we classify the approaches with respect to their degree of globality and interactivity; see also Figure 2. We used the following three criteria to decide on the locality of a labeling approach, including adjustment of metro map **(L1)**, placement of labels **(L2)**, and optimality **(L3)** (see also Table III).

For each criterion and each labeling approach we determined the highest score that suits the approach. The overall degree of globality of a labeling approach is then the average over its three assigned scores. In order to decide on the degree of interactivity for each labeling approach, we applied a similar approach as we did in Section II using Table II. We assigned each approach to the lowest category that suits the approach best. Altogether, we obtain a measure for both the degree of globality and interactivity of a labeling approach, which we depict in Figure 2. In the following we discuss the single results in increasing order with respect to their degree of interactivity.

Nöllenburg and Wolff [27] present an approach that creates the layout of the metro map and the label placement simultaneously. To that end, they present a mixed integer linear programming formulation that optimally creates a metro map with respect to a global cost-function. They particularly require that labels in between two crossings lie on the same side of the metro line. However, their approach comes with a high running time, e.g., for the metro system of Sydney they report a running time up to 12 hours. Stott et al. [39] present an iterative approach that creates the layout and label placement in an integrated way using hill climbing based on a global multi-criteria function. Among other criteria they prefer labels on the same side of the metro line. They report running times between 2 and 120 minutes.

In contrast, Wu et al. [53] assume when placing labels that the layout is given. Still, they scale the map to increase free spaces, which they use for label placement based on a MIP approach. By considering the layout and labeling step independently, they achieve running times up to minutes. Yoshida et al. [56] scale single edges of the map while preserving the directions of the edges. The main idea is to create free spaces in dense parts of the map, while leaving other parts of the maps unchanged. Similarly to Wu et al. [53], Niedermann and Haunert [25] systematically scale the map layout to fit in the labeling. For the labeling procedure they propose a dynamic programming approach that labels a single metro line optimally. They use this method as a subroutine of their heuristic, which labels the map within few minutes.

Wu et al. [55] label the stations with photos instead of text labels. They place the photos alongside two boundaries of the rectangular map and associate them with their stations via connecting curves, also called *leaders*. They aim at crossing-free solutions that minimize the length of the leaders. Wu et al. [52] generalize this approach to all four sides of the map. Wu et al. [54] separate the labels from the metro map less clearly. They present a genetic algorithm that places overlap-free text labels and photos within the free spaces of the metro map connecting the labels with their stations using straight-line leaders. At the core of the genetic algorithm, which systematically explores different orders of placing labels, the actual placement is done by a simple and fast greedy procedure.

Hong et al. [18] separate the placement of the labels from the computation of the layout. They describe the labeling problem as a conflict graph based on label candidates and their occlusions. The label placement corresponds to an independent set of vertices within the graph, which they determine using simulated annealing. Although they maximize the number of labels, they cannot guarantee that each station gets a label.

Wang and Chi [48] also separate the placement of labels from the computation of the metro map layout. They describe the labeling problem as a multi-criteria energy function, which they locally optimize. Their approach particularly avoids occlusions between labels and prefers labels on the same side of the metro line. They report running times below one second, which makes their approach applicable for interactive real-time scenarios. Wang and Peng [49] use the same labeling approach for their interactive editing system.

Chivers and Rodgers [7] present an approach that after fixing the metro map layout selects for each station a label out of eight candidates using a simple greedy approach. They penalize label occlusions and prefer similarly placed labels of adjacent stations.

Altogether, we observe that with increasing degree of interactivity the globality of the approaches decreases. In particular, those approaches that are applicable for interactive real-world scenarios, typically assume a fixed metro map layout. While this provides the possibility of computing the label placement much faster than approaches integrating the process of creating the layout, the given metro map layout may not host all labels. This is either resolved by allowing occlusions [48] [49] or by labeling only a (possibly large) subset of stations [18]. Hence, those approaches are far away from achieving the same quality with respect to the labeling as approaches with a smaller degree of interactivity. We therefore deem the development of labeling algorithms with the potential for a high degree of interactivity as a still open and important research problem.

IV. APPLICATIONS AND ON-LINE TOOLS

Schematic network maps have widely been employed for representing abstract relationships between subjects. In such cases, the schematic maps serve as visual metaphors that effectively systematize the underlying structures hidden behind the target information space. In this section, we first focus on techniques for visualizing abstract information space using schematic map metaphors and then raise representative on-line tools for designing such maps.
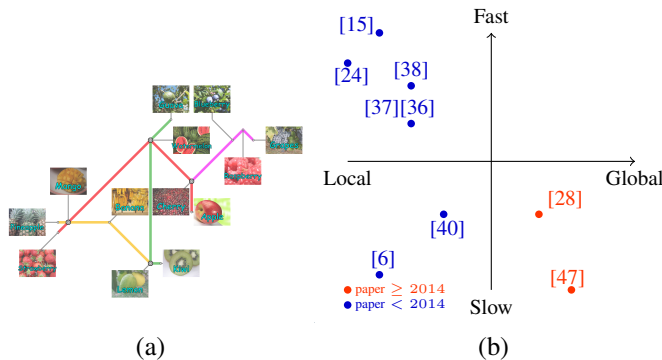

(a)   (b)

Fig. 3. Visualization using schematic maps as metaphors, including (1) an example of fruit maps, and (b) Applications with respect to their globality and interactiveness.

A. Metro Maps as Visual Metaphors

Effects of using schematic maps as visual metaphors have been investigated, though several case studies are at an early stage. Nesbitt [23] conducted preliminary studies to evaluate the capabilities of metro map metaphors in understanding abstract data and showed that they can provide a better mental model of the target data as a first impression. Pontikakis and Twaroch [32] also examined the potential use of metro maps to represent the topological connections between points of interest especially when the underlying topographic maps are unavailable.

Metro map metaphors have often been employed to explore the underlying structure of an information space. In this case, each line corresponds to a chain of subjects such as events, topics, plans, articles, etc. Stott et al. [40] visualized the time-lines of project plans and their dependency as a metro map by employing the early version of their hill-climbing algorithm [39]. Aguiregoitia et al. [6] applied such an idea to software project visualization where a different set of tasks is aligned with each metro line according to the software development process.

A set of documents is another target for which we can effectively employ a schematic map as a visual metaphor. Shahaf et al. [37] proposed an algorithm for constructing a metro map of documents in such a way that the topology of a document network reflects the desired sparseness inherent in ordinary railway networks. This technique has been devised to visualize chains of scientific papers to clarify the evolution of research fields [36], and then further extended to the scalable scheme that enables the the level of detail control in metro map metaphors [38].

As an application of schematic network metaphors to scientific data, Wahabzada et al. [47] successfully visualized plant disease dynamics by tracking spatiotemporal disease patterns with imaging techniques and composed metro map metaphors by taking advantage of [27]. Cancer pathways [16] were also visualized as schematic metro maps in the context of multiobjective optimization [28].

Schematic network metaphors also allows us to visualize the structure of low-dimensional space obtained through dimensionality reduction techniques. Neumayer et al. [24] employed a metro map metaphor to describe the configuration of segmented characteristic regions over the 2-dimensional plane obtained through self-organizing maps. Gorban et al. [15] applied this idea in the context of dimensionality reduction based on elastic energy minimization to schematize the distribution of data samples with a metro map metaphor.

Several manually designed metaphors of metro maps are available on-line. Example includes a Galaxy map that details relationships among planets [11], a science map composed of scientists and their fields of sciences [31], a human anatomy diagram consisting of organ systems as metro lines [17], a logistic map of supply chains [1], a map of leading names and domains on the Internet [5], and a map of digital workspace and marketing technology vendors [2]. The most timely ex-

ample is the Brexit tube maps [10], which model Brexit issues and sectors as stations and lines, respectively, to elucidate how they relate to one another.

B. Accessible On-line Tools

On-line tools often allows us to customize travel routes within the precomputed schematic network maps or even to manually design such schematic maps themselves. Mapway [4] and ExploreMetro [3] are mobile apps for finding a route to the destination on metro maps in major cities. Metro Map Maker [44] provides a web-based tool for designing schematic network maps, in which it is possible to manually place stations and draw metro lines by referring to a regular grid on a screen canvas. Several software libraries for designing interactive schematic maps are also available, including Unfolding Map Library [22] for Processing and Java, a library for interactive SVG metro maps with JointJS [21], and a jQuery plug-in for subway map visualization [19].

V. DISCUSSION AND GUIDANCE FOR FOLLOWERS

Our hypothesis about the correlation between problem complexity and running time has been visually supported since our scatter plots approximately show a top-left to bottom-right diagonal distribution of algorithmic approaches. This strongly suggests that researchers are looking for a compromise between the selected constraints for their problem formulation and the techniques to be used, in order to find the balance between the global scope of the solution as opposed to the running time of the algorithm. Doubtlessly, the final goal of schematic mapping algorithms is to develop an ultimate technique that pinpoints us in the top-right corner of the plots, representing high-quality maps computed instantaneously. As shown in Figure 1, the blue markers represent approaches investigated before 2014, and red markers depict the techniques developed after 2014. A tendency from left to right and from bottom to top can be observed indicating that researchers are heading towards the desired goal. While some techniques [29] [28] tend to accelerate the conventional approach by Nöllenburg and Wolff [27], which has been considered as the most global optimization method in this paper, the running times are still not very suitable for interactive applications. This also confirms that the schematic map problem is still a fundamentally difficult one, so that a certain quality loss needs to be tolerated in order to achieve fast performance [48]. More sophisticated algorithms are expected to be developed to open further opportunities. As summarized in Section IV, several researchers are beginning to use the maps generated by schematization techniques to visualize their research results, especially when the data is associated with entity connectivity and spatial relationship between entities. The previous plots can also serve as a visual guidance to instruct those researchers to select an appropriate technique for their research purposes.

VI. FUTURE RESEARCH DIRECTION

In this study, we have compiled the state of the art on schematic network visualization and have given a compre-

hensive overview of automatic network layout algorithms and map labeling algorithms, as well as tools and applications for use in multidisciplinary domains. Our taxonomy demonstrates that most methods carefully design their solutions in compliance with the problem space in order to provide effective approaches with reasonable quality. Nonetheless until now, no benchmark is provided to enable standard testing on schematic map production, although the Sydney metro map has been studied by most of the conventional map layout algorithms. Developing such a repository would help promoting the usage of algorithms not only in cartography but also in other scientific domains. Another challenge for future work in this field is to target advanced methodology to smartly handle global constraints to improve the scalability of the algorithms. This can be integrated with the interaction techniques together with appropriate user scenarios. For this purpose, we plan to further investigate visual representations and interaction schemes in accordance with the task taxonomy for map users.

REFERENCES

[1] 3pl subway map. https://www.supplychainmedia.eu/print/visuals/subway-maps/3pl-subway-map/.
[2] Digital workplace & marketing technology vendor map. https://www.realstorygroup.com/vendormap/.
[3] Exploremetro. https://www.exploremetro.com/.
[4] Mapway. https://www.mapway.com/.
[5] Web trend map 4: Coolest gift for geeks. https://ia.net/topics/wtm4.
[6] A. Aguirregoitia, J. J. D. Cosín, and C. Presedo. Software project visualization using task oriented metaphors. *Journal of Software Engineering and Applications*, 3(11):1015–1026, 2010.
[7] D. Chivers and P. Rodgers. Gesture-based input for drawing schematics on a mobile device. In *Information Visualisation (IV'11)*, pages 127–134, 2011.
[8] D. Chivers and P. Rodgers. Octilinear force-directed layout with mental map preservation for schematic diagrams. In T. Dwyer, H. Purchase, and A. Delaney, editors, *Diagrammatic Representation and Inference (Diagrams'14)*, pages 1–8. Springer Berlin Heidelberg, 2014.
[9] D. Chivers and P. Rodgers. Improving search-based schematic layout by parameter manipulation. *Int. J. Software Engineering and Knowledge Engineering*, 25(6):961–991, 2015.
[10] S. de Groot and M. J. Roberts. Brexit mapping. http://www.brexitmapping.com.
[11] J. Diaz. We are just a tiny station in the milky way subway map. https://gizmodo.com/we-are-just-a-tiny-station-in-the-milky-way-subway-map-5454587.
[12] M. Fink, M. Lechner, and A. Wolff. Concentric metro maps. In *Proceedings of the Schematic Mapping Workshop (SMW'14)*, 2014. Poster.
[13] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164, 1991.
[14] K. Garland. *Mr. Beck's Underground Map*. Capital Transport Publishing, 1994.
[15] A. N. Gorban, N. R. Sumner, and A. Y. Zinovyev. Beyond the concept of manifolds: Principal trees, metro maps, and elastic cubic complexes. In *Principal Manifolds for Data Visualization and Dimension Reduction*, pages 219–237. Springer, 2008.
[16] W. C. Hahn and R. A. Weinburg. A subway map of cancer pathways, 2002. https://www.nature.com/nrc/poster/subpathways/index.html [Subway map designed by C. Bentley.].

[17] B. P. C. Ho. A doctor created a human anatomy diagram in the style of a subway map and it's friggin' gorgeous. http://digg.com/2018/human-anatomy-subway-map.

[18] S.-H. Hong, D. Merrick, and H. A. D. do Nascimento. Automatic visualisation of metro maps. *Journal of Visual Languages and Computing*, 17(3):203–224, 2006.

[19] N. Kalyani. Subway map visualization jquery plugin. https://github.com/techbubble/subwayMap.

[20] Z. Li and W. Dong. A stroke-based method for automated generation of schematic network maps. *Int. J. Geographical Information Science*, 24(11):1631–1647, 2010.

[21] P. Maszczynski. Creating an interactive svg metro map with jointjs. https://www.netvlies.nl/tips-updates/applicaties/cases/creating-an-interactive-svg-metro-map-with-jointjs/.

[22] T. Nagel. Unfolding map library. http://unfoldingmaps.org/.

[23] K. V. Nesbitt. Getting to more abstract places using the metro map metaphor. In *Information Visualisation*, IV '04, pages 488–493, 2004.

[24] R. Neumayer, R. Mayer, G. Polzlbauer, and A. Rauber. The metro visualisation of component planes for self-organising maps. In *Int. Joint Conference on Neural Networks*, pages 2788–2793, 2007.

[25] B. Niedermann and J. Haunert. An algorithmic framework for labeling network maps. *Algorithmica*, 80(5):1493–1533, 2018.

[26] M. Nöllenburg. A survey on automated metro map layout methods. In *Schematic Mapping Workshop*, Essex, UK, 2014. https://i11www.iti.kit.edu/extra/publications/n-asamm-14.pdf.

[27] M. Nöllenburg and A. Wolff. Drawing and labeling high-quality metro maps by mixed-integer programming. *IEEE Transactions Visualization and Computer Graphics*, 17(5):626–641, 2011.

[28] O. Oke and S. Siddiqui. Efficient automated schematic map drawing using multiobjective mixed integer programming. *Computers & Operations Research*, 61:1–17, 2015.

[29] M. Onda, M. Moriguchi, and K. Imai. Automatic drawing for tokyo metro map. In *European Workshop on Computational Geometry (EuroCG'18)*, pages 62:1–62:6, 2018.

[30] M. Ovenden. *Metro Maps of the World*. Capital Transport Publishing, 2003.

[31] P. Plait. Metrocontextual science map. http://blogs.discovermagazine.com/badastronomy/2010/08/31/metrocontextual-science-map/\#.XFEYW88zbfB.

[32] E. Pontikakis and F. Twaroch. Schematic maps as an alternative to point coverages when topographic maps are not available. In *Information Visualisation*, IV '06, pages 297–303, 2006.

[33] M. J. Roberts. *Underground Maps Unravelled: Explorations in Information Design*. Self-published, 2012.

[34] M. J. Roberts, H. Gray, and J. Lesnik. Preference versus performance: Investigating the dissociation between objective measures and subjective ratings of usability for schematic metro maps and intuitive theories of design. *International Journal of Human-Computer Studies*, 98:109 – 128, 2017.

[35] M. J. Roberts, E. J. Newton, F. D. Lagattolla, S. Hughes, and M. C. Hasler. Objective versus subjective measures of paris metro map usability: Investigating traditional octolinear versus all-curves schematics. *International Journal of Human-Computer Studies*, 71(3):363 – 386, 2013.

[36] D. Shahaf, C. Guestrin, and E. Horvitz. Metro maps of science. In *Knowledge Discovery and Data Mining*, KDD '12, pages 1122–1130, 2012.

[37] D. Shahaf, C. Guestrin, and E. Horvitz. Trains of thought: Generating information maps. In *World Wide Web*, WWW '12, pages 899–908, 2012.

[38] D. Shahaf, J. Yang, C. Suen, J. Jacobs, H. Wang, and J. Leskovec. Information cartography: Creating zoomable, large-scale maps of information. In *Knowledge Discovery and Data Mining*, KDD '13, pages 1097–1105, 2013.

[39] J. Stott, P. Rodgers, J. C. Martínez-Ovando, and S. G. Walker. Automatic metro map layout using multicriteria optimization. 17(1):101–114, 2011.

[40] J. M. Stott, P. Rodgers, R. A. Burkhard, M. Meier, and M. T. J. Smis. Automatic layout of project plans using a metro map metaphor. In *Information Visualisation*, IV '05, pages 203–206, 2005.

[41] K. Sugiyama and K. Misue. Graph drawing by the magnetic spring model. *Journal of Visual Languages & Computing*, 6(3):217–231, 1995.

[42] P. Ti and Z. Li. Generation of schematic network maps with automated detection and enlargement of congested areas. *International Journal of Geographical Information Science*, 28(3):521–540, 2014.

[43] P. Ti, Z. Li, and Z. Xu. Automated generation of schematic network maps adaptive to display sizes. *The Cartographic Journal*, 52(2):168–176, 2015.

[44] S. Turner. Metro map maker. https://metromapmaker.com/.

[45] T. C. van Dijk and D. Lutz. Realtime linear cartograms and metro maps. In *Advances in Geographic Information Systems (SIGSPATIAL'18)*, pages 488–491. ACM, 2018.

[46] T. C. van Dijk, A. van Goethem, J.-H. Haunert, W. Meulemans, and B. Speckmann. Map schematization with circular arcs. In M. Duckham, E. Pebesma, K. Stewart, and A. U. Frank, editors, *Geographic Information Science (GIScience'14)*, pages 1–17. Springer International Publishing, 2014.

[47] M. Wahabzada, A.-K. Mahlein, C. Bauckhage, U. Steiner, E.-C. Oerke, and K. Kersting. Metro maps of plant disease dynamics-automated mining of differences using hyperspectral images. *PLOS ONE*, 10(1):1–20, 2015.

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

[49] Y.-S. Wang and W.-Y. Peng. Interactive metro map editing. *IEEE Transactions on Visualization and Computer Graphics*, 22(2):1115–1126, 2016.

[50] A. Wolff. Drawing subway maps: A survey. *Informatik – Forschung und Entwicklung*, 22(1):23–44, 2007.

[51] A. Wolff. Graph drawing and cartography. In R. Tamassia, editor, *Handbook of Graph Drawing and Visualization*, chapter 23, pages 697–736. CRC Press, 2013.

[52] H.-Y. Wu, S.-H. Poon, S. Takahashi, M. Arikawa, C.-C. Lin, and H.-C. Yen. Designing and annotating metro maps with loop lines. In *Information Visualization*, IV '15, pages 9–14. IEEE, 2015.

[53] H.-Y. Wu, S. Takahashi, D. Hirono, M. Arikawa, C.-C. Lin, and H.-C. Yen. Spatially efficient design of annotated metro maps. *Computer Graphics Forum*, 32(3):261–270, 2013.

[54] H.-Y. Wu, S. Takahashi, C.-C. Lin, and H.-C. Yen. A zone-based approach for placing annotation labels on metro maps. In *Smart Graphics (SG'11)*, volume 6815 of *LNCS*, pages 91–102. Springer-Verlag, 2011.

[55] H.-Y. Wu, S. Takahashi, C.-C. Lin, and H.-C. Yen. Travel-route-centered metro map layout and annotation. *Computer Graphics Forum*, 31(3):925–934, 2012.

[56] Y. Yoshida, K. Maruyama, T. Kawagoe, H.-Y. Wu, M. Arikawa, and S. Takahashi. Progressive annotation of schematic railway maps. In *Information Visualisation*, IV '18, pages 373–378. IEEE, 2018.