

# Drawing $k$ -linear Metro Maps

Soeren Nickel  
Algorithms and Complexity Group  
TU Wien  
Vienna, Austria  
soeren.nickel@tuwien.ac.at

Martin Nöllenburg  
Algorithms and Complexity Group  
TU Wien  
Vienna, Austria  
noellenburg@ac.tuwien.ac.at

**Abstract**—Schematic metro maps in practice as well as metro map layout algorithms usually adhere to an octilinear layout style with all paths composed of horizontal, vertical, and  $45^\circ$ -diagonal edges. Despite growing interest in non-octilinear metro maps, generic algorithms to draw metro maps based on a system of  $k \geq 2$  equidistant slopes have not been investigated thoroughly. In this paper we present and implement an adaptation of the octilinear mixed-integer linear programming approach of Nöllenburg and Wolff (2011) that can draw  $k$ -linear metro maps.

## I. INTRODUCTION

Algorithms for automated layout of metro maps have received substantial interest in the graph drawing and network visualization communities as well as in cartography and geovisualization over the last 15 years [8]. The vast majority of metro map layout algorithms focus on so-called *octilinear* metro maps, which are limited to Henry Beck’s classical and since then widely adopted  $45^\circ$ -angular grid of line orientations [4]. However, not all metro maps found in reality are octilinear. There is empirical evidence that the best set of line orientations used for drawing a metro map depends on different aspects of the respective transit network, and it may not always be an octilinear one [11], [12].

In this paper we present a first algorithmic approach using global optimization for computing unlabeled metro maps in the more flexible  *$k$ -linearity* setting, where each edge in the drawing must be parallel to one of  $k \geq 2$  equidistant orientations whose pairwise angles are multiples of  $360^\circ/2k$ . There exist a number of metro map layout algorithms (see [8] for a comprehensive survey) that would technically permit an adaptation to a different underlying angular grid, yet most previous papers optimize layouts in the well-known octilinear setting only and do not discuss extensions to  $k$ -linearity. A few algorithms for generic  $k$ -linear layouts exist [1], [2], [5], [7], but they are aimed at paths or polygons rather than entire metro maps. In the field of graph drawing many algorithms for planar orthogonal network layouts with  $k = 2$  as well as for polyline drawings with unrestricted slopes are known [3], but they do not generalize to  $k$ -linearity.

We decided to adapt the octilinear (i.e.,  $k = 4$ ) mixed-integer linear programming (ILP) model of Nöllenburg and Wolff [9] by generalizing their mathematical layout constraints to  $k$ -linearity. The main benefit of this model in comparison to other approaches is that it defines sets of hard and soft

constraints and guarantees that the computed layout satisfies all the hard constraints, while the soft constraints are optimized. The trade-off for providing such quality guarantees from a global optimization technique is that computation time is typically higher compared to other methods. By modeling fundamental metro map properties such as strict adherence to the  $k$ -linearity and topological correctness as hard constraints, we obtain layouts that satisfy these layout requirements. The soft constraints optimize for line straightness, compactness, and topographicity [10], i.e., low topographical distortion. Our modifications yield a flexible ILP model whose complexity measured by the number of variables and constraints grows linearly with  $k$ . Section II gives an overview of the ILP model and introduces the additional constraints. We demonstrate the effect of changing  $k$  with sample layouts of two metro networks for small values of  $k = 3, 4, 5$  in Section III.

## II. MODEL

### A. Preliminaries

We reuse the notation of Nöllenburg and Wolff [9]. The input is an embedded planar<sup>1</sup> metro graph  $G = (V, E)$  with  $n$  vertices and  $m$  edges. Each vertex  $v \in V$  represents a metro station with  $x$ - and  $y$ -coordinates and each edge  $e = (u, v) \in E$  is a curve linking vertices  $u$  and  $v$  that represents a physical rail connection between them. Let  $\mathcal{L}$  be a *line cover* of  $G$ , i.e., a set of paths of  $G$  such that each edge  $e \in E$  belongs to at least one path  $L \in \mathcal{L}$ . An element  $L \in \mathcal{L}$  is called a *line* and corresponds to a metro line in the underlying transport network. Finally,  $k \geq 2$  is an input parameter that defines the number of available edge orientations. Since every orientation can be used in two directions this yields  $2k$  available drawing directions. Let  $\mathcal{K}$  be the set of all directions. We note that since every edge is assigned exclusively to one outgoing direction of its incident vertex, this implies that the maximum degree of  $G$  can be at most  $2k$ .

The algorithmic problem is to find a  *$k$ -linear schematic layout* of  $(G, \mathcal{L})$ , i.e., a graph layout that preserves the input topology, uses only the edge directions from  $\mathcal{K}$ , and optimizes a weighted layout quality function composed of line straightness, topographicity, and compactness.

<sup>1</sup>For non-planar metro graphs we temporarily introduce a dummy vertex for each edge crossing, which preserves the crossing in the output layout.

## B. Hard constraints

The hard constraints of the ILP model comprise four aspects: a  $k$ -linear coordinate system, assignment of edge directions, combinatorial embedding, and planarity.

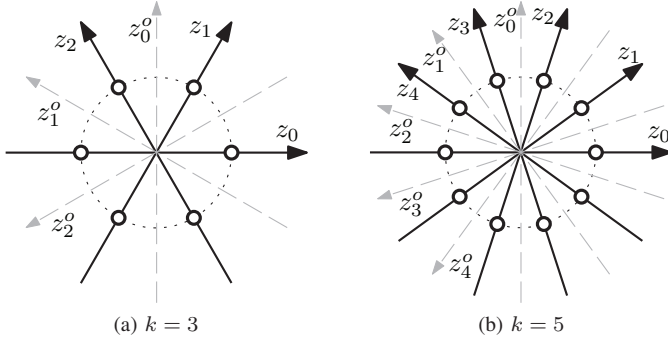


Fig. 1. Coordinate axes for  $k$ -linear orientation systems

1) *Coordinate system*: Every vertex  $u$  of  $G$  has two Cartesian coordinates in the plane  $\mathbb{R}^2$ , specified as  $x(u)$  and  $y(u)$ . In order to address vertex coordinates for a flexible number  $k$  of orientations, we define a redundant system of  $k$  coordinates  $z_0, \dots, z_{k-1}$ , which are all real-valued variables in the ILP model. The first coordinate  $z_0$  is defined as  $z_0(u) = x(u)$ . The remaining orientations are obtained by rotating<sup>2</sup> the x-axis counterclockwise by multiples of  $\theta = \pi/k$ .<sup>3</sup> We define the coordinate  $z_i(u)$  using  $x(u)$  and  $y(u)$  as

$$z_i(u) = \cos(i \cdot \theta) \cdot x(u) + \sin(i \cdot \theta) \cdot y(u). \quad (1)$$

In order to be able to express later that two vertices  $u, v$  are collinear on a line with slope in  $\mathcal{K}$ , we need the orthogonal orientation  $z_i^o$  for each coordinate  $z_i$ . For an even number  $k$ ,  $z_i^o$  is one of the  $k$  coordinates, but for odd  $k$  we need an additional coordinate  $z_i^o$  orthogonal to each  $z_i$ , see Figures 1a and 1b. Using a rotation by  $\pi/2$  we obtain

$$z_i^o(u) = -\sin(i \cdot \theta) \cdot x(u) + \cos(i \cdot \theta) \cdot y(u) \quad (2)$$

2) *Edge directions and minimum length*: Every edge  $(u, v) \in E$  has an original direction in the input layout of  $G$ , defined as the direction from  $u$  to  $v$ . Our  $k$ -linear coordinate system splits the plane into exactly  $2k$  sectors numbered from 0 to  $2k-1$  for each vertex  $u \in V$ , see Figure 2. We store the sector in which an edge  $(u, v)$  lies in the input drawing as a constant  $\text{sec}_u(v)$  that we call the *original sector* of  $(u, v)$ .

Next we define an integer variable  $\text{dir}(u, v)$  to encode the selected direction of the edge  $(u, v)$  in a  $k$ -linear solution. The range for  $\text{dir}(u, v)$  includes the original sector  $\text{sec}_u(v)$  and  $s \geq 1$  admissible neighboring sectors in both directions. The ILP model of Nöllenburg and Wolff [9] uses  $s = 1$ , which results in a range of three admissible edge directions for each edge.

For each edge  $(u, v)$  we define the set  $\mathcal{S}(u, v)$  of admissible directions<sup>4</sup> as  $\mathcal{S}(u, v) = \{i \mid \text{sec}_u(v) - s \leq i \leq \text{sec}_u(v) + s\}$ .

<sup>2</sup>All angles in this section are expressed in radian.

<sup>3</sup>Technically, any set of  $k$  angles  $\{\theta_1, \dots, \theta_k\} \subset [0, \pi)$  can be chosen.

<sup>4</sup>All index calculations are modulo  $2k$ .

For each  $i \in \mathcal{S}(u, v)$  we define its corresponding direction number as  $\text{sec}_u^i(v)$  and define a binary variable  $\alpha_i(u, v)$  of which only one can be true at any given time (3). These are then used to assign the correct value of  $\text{dir}(u, v)$  (4).

$$\sum_{i \in \mathcal{S}(u, v)} \alpha_i(u, v) = 1 \quad (3)$$

$$\text{dir}(u, v) = \sum_{i \in \mathcal{S}(u, v)} \text{sec}_u^i(v) \cdot \alpha_i(u, v) \quad (4)$$

We further define  $\text{dir}(v, u) = \text{dir}(u, v) + k$  for the opposite edge  $(v, u)$ .

To guarantee that the output layout draws the edge  $(u, v)$  in the selected direction  $\text{dir}(u, v)$  we need to ensure that the variables of  $u$  and  $v$  for the orthogonal coordinate axis  $z_i^o$  are equal, i.e.,  $z_{\text{dir}(u, v)}^o(u) = z_{\text{dir}(u, v)}^o(v)$  (5a) and that the coordinates  $z_{\text{dir}(u, v)}(u)$  and  $z_{\text{dir}(u, v)}(v)$  differ by at least the minimum edge length  $L_{\min}$ , i.e.,  $z_{\text{dir}(u, v)}(v) - z_{\text{dir}(u, v)}(u) \geq L_{\min}$  (5b).

$$\begin{aligned} z_{i'}^o(u) - z_{i'}^o(v) &\leq M(1 - \alpha_i(u, v)) \\ -z_{i'}^o(u) + z_{i'}^o(v) &\leq M(1 - \alpha_i(u, v)) \end{aligned} \quad (5a)$$

$$\begin{aligned} z_{i'}(v) - z_{i'}(u) &\geq -M(1 - \alpha_i(u, v)) + L_{\min} \quad \text{if } i < k \\ z_{i'}(u) - z_{i'}(v) &\geq -M(1 - \alpha_i(u, v)) + L_{\min} \quad \text{if } i \geq k \end{aligned} \quad (5b)$$

Note four things. First, the constraints are created for every  $i \in \mathcal{S}(u, v)$ . Second, we use  $i' = i \bmod k$ , since we only have  $k$  coordinates, but  $2k$  possible directions. Third, we need to distinguish whether the direction  $i$  is smaller than the number  $k$  of orientations, in which case  $u$  must have a smaller value than  $v$  in coordinate  $z_i$ , or otherwise if  $i \geq k$  then  $v$  must have the smaller coordinate and we need to invert the difference in (5b). And fourth, every triple of constraints for which  $\alpha_i(u, v) = 0$  is trivially satisfied by using a sufficiently big constant  $M$  in the constraints. Due to (3),  $\alpha_i(u, v) = 1$  for exactly one index  $i$  and only for that index  $i$  the constraints have an actual effect on the coordinates.

### 3) Combinatorial embedding:

We want to keep the combinatorial embedding, i.e., the topology of the input layout, which translates into preserving the cyclic order of the neighbors of each vertex. This can be expressed by requiring that the edge direction values strictly increase when visiting the incident edges in counterclockwise input order. There is exactly one exception, namely when going from the last used sector to the first one.

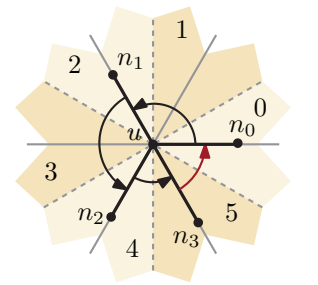


Fig. 2. For edges  $(u, n_3), (u, n_0)$  the direction value decreases from 5 to 0.

Figure 2 illustrates this situation, where the crossover point lies between the neighbors  $n_3$  and  $n_0$ , marked in red. Here we can add an offset of  $2k$  instead to make the condition hold. Since this must happen exactly once, we can use binary

variables  $\beta_1(v), \beta_2(v), \dots, \beta_{\deg(v)}(v)$  to select the respective edge pair (6), (7).

$$\sum_{i=1}^{\deg(v)} \beta_i = 1 \quad (6)$$

$$\begin{aligned} \text{dir}(v, u_1) + 1 &\leq \text{dir}(v, u_2) + 2k \cdot \beta_1(v) \\ \text{dir}(v, u_2) + 1 &\leq \text{dir}(v, u_3) + 2k \cdot \beta_2(v) \\ &\vdots \\ \text{dir}(v, u_{\deg(v)}) + 1 &\leq \text{dir}(v, u_1) + 2k \cdot \beta_{\deg(v)}(v) \end{aligned} \quad (7)$$

4) *Planarity*: For every pair of non-adjacent edges  $e = (u, v)$  and  $e' = (u', v')$  we need to find (at least) one separation line between  $e$  and  $e'$  in a direction of  $\mathcal{K}$  to guarantee that  $e, e'$  do not intersect. We define a set of  $2k$  binary variables  $\gamma_i(e, e')$  for which we require that at least one of them is set to true.

$$\sum_{i \in \mathcal{K}} \gamma_i(e, e') \geq 1 \quad (8)$$

Now we ensure that every pair of edges  $e, e'$  has a minimum distance  $d_{\min}$  in the selected directions, i.e., both endpoints of  $e$  have a distance of at least  $d_{\min}$  to both endpoints of  $e'$ .

$$\begin{aligned} z_{i'}(u') - z_{i'}(u) &\geq -M(1 - \gamma_i(e, e')) + d_{\min} \\ z_{i'}(u') - z_{i'}(v) &\geq -M(1 - \gamma_i(e, e')) + d_{\min} \\ z_{i'}(v') - z_{i'}(u) &\geq -M(1 - \gamma_i(e, e')) + d_{\min} \\ z_{i'}(v') - z_{i'}(v) &\geq -M(1 - \gamma_i(e, e')) + d_{\min} \end{aligned} \quad (9)$$

Note that the constraints are created for every  $0 \leq i < 2k$ , that we use  $i' = i \bmod k$  and that the first  $k$  sets of these equation look like (9), while the rest needs to invert the differences, e.g.,  $-z_{i'}(u') + z_{i'}(u)$ , since they change sides with respect to direction  $z_{i'}$ .

### C. Soft constraints

The soft constraints model the aesthetic quality criteria that should be optimized in the layout. We adapt the three criteria of Nöllenburg and Wolff [9] to  $k$ -linearity: line straightness, topographicity, and compactness. Each requires a set of linear constraints together with a corresponding linear term in the objective function.

1) *Line straightness*: We optimize for line straightness by minimizing the number and angles of bends along the metro lines in  $\mathcal{L}$ . First we create a variable  $\theta(u_1, u_2, u_3)$  for all pairs of consecutive edges  $e_1 = (u_1, u_2), e_2 = (u_2, u_3)$  along some path  $L \in \mathcal{L}$  that represents the cost of a potential bend between  $e_1$  and  $e_2$  on the metro line  $L$ . To assign  $\theta(u_1, u_2, u_3)$  we subtract the direction of  $e_2$  from the direction of  $e_1$ . If the edges do not have the same direction, the difference  $\text{dir}(u_1, u_2) - \text{dir}(u_2, u_3)$ , which we will call  $\Delta \text{dir}_{u_1, u_2, u_3}$ , will either be positive or negative and  $\Delta \text{dir}_{u_1, u_2, u_3} \in [-2k + 1, 2k - 1]$ . From [9] we know that  $\theta(u_1, u_2, u_3) = \min\{|\Delta \text{dir}_{u_1, u_2, u_3}|, 2k - |\Delta \text{dir}_{u_1, u_2, u_3}|\}$ , i.e.,  $\theta \in [-k + 1, k - 1]$ . By using two binary correction variables  $\delta_1$  and  $\delta_2$  we can ensure that  $\theta$  takes the desired

minimal value (10), which then lets us define the bend cost function (11).

$$\begin{aligned} -\theta(u_1, u_2, u_3) &\leq \Delta \text{dir}_{u_1, u_2, u_3} - 2k \cdot \delta_1 + 2k \cdot \delta_2 \\ \theta(u_1, u_2, u_3) &\geq \Delta \text{dir}_{u_1, u_2, u_3} - 2k \cdot \delta_1 + 2k \cdot \delta_2 \end{aligned} \quad (10)$$

$$\text{cost}_{\text{bends}} = \sum_{L \in \mathcal{L}} \sum_{(u_1, u_2), (u_2, u_3) \in L} \theta(u_1, u_2, u_3) \quad (11)$$

2) *Topographicity*: In order to support the mental map [6] of the user, we want the shape of the output drawing to resemble the input drawing as closely as possible. For this we try to preserve the input directions of the edges. Formally we want to minimize the difference between the input direction and the output direction, i.e.,  $\sum_{(u, v) \in E} |\text{dir}(u, v) - \text{sec}_u(v)|$ . In order to minimize the absolute value we define a new variable  $\xi(u, v) = |\text{dir}(u, v) - \text{sec}_u(v)|$  by imposing (12) and minimizing  $\xi(u, v)$  in the cost function (13). The topographicity cost function is simply the sum over all  $\xi$ -variables (13).

$$\begin{aligned} \text{dir}(u, v) - \text{sec}_u(v) &\leq \xi(u, v) \\ -\text{dir}(u, v) + \text{sec}_u(v) &\leq \xi(u, v) \end{aligned} \quad (12)$$

$$\text{cost}_{\text{topo}} = \sum_{(u, v) \in E} \xi(u, v) \quad (13)$$

3) *Compactness*: To ensure a compact layout we minimize the total edge length of the output drawing. Here we note that the Euclidean length of an edge  $e = (u, v)$  in a  $k$ -linear layout is defined by the maximum absolute value  $|z_i(u) - z_i(v)|$  in all  $k$  coordinates (the projections in all other directions are shorter), which we model by a variable  $\lambda(u, v)$ . The compactness cost function is the sum of all edge lengths.

$$\begin{aligned} z_i(u) - z_i(v) &\leq \lambda(u, v) \\ -z_i(u) + z_i(v) &\leq \lambda(u, v) \end{aligned} \quad (14)$$

$$\text{cost}_{\text{length}} = \sum_{(u, v) \in E} \lambda(u, v) \quad (15)$$

4) *Objective function*: The objective function, which we want to minimize, is put together from the three different terms  $\text{cost}_{\text{bends}}, \text{cost}_{\text{topo}}$  and  $\text{cost}_{\text{length}}$  defined above. Each term can be weighted with factors  $f_1, f_2, f_3$  depending on their relative importance.

$$\text{minimize } f_1 \cdot \text{cost}_{\text{bends}} + f_2 \cdot \text{cost}_{\text{topo}} + f_3 \cdot \text{cost}_{\text{length}} \quad (16)$$

### D. Model size and improvements

Our first observation from generalizing the model of Nöllenburg and Wolff [9] is that their numbers of constraints and variables now scale linearly with the number  $k$  of orientations. So as long as  $k$  is a (small) constant the asymptotics with respect to the graph size parameters  $n$  and  $m$  remain the same. Yet, in practice, doubling the size of the model may yield a significant slow-down in the actual solution time.

Further, Nöllenburg and Wolff [9] devised several practical improvements to accelerate their method. For instance, the

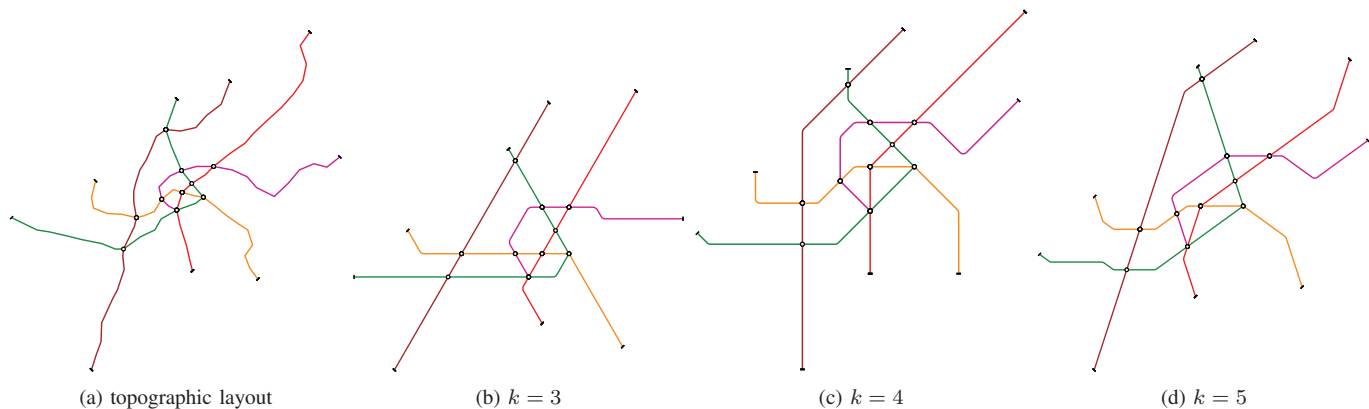


Fig. 3. Vienna metro map in different linearities

number of planarity constraints (Sect. II-B4) grows quadratically with the number of edges, but most of them are never critical as any reasonable layout satisfies them trivially. So they suggested ways of reducing the number of necessary constraints and to add them only on demand, which immediately carries over to our generalized model. Similarly, their proposed techniques for reducing the graph size by temporarily removing degree-2 vertices from the graph can be adapted to the  $k$ -linear setting. This is not yet included in our algorithm.

### III. EXAMPLES

To showcase the different types of metro maps that can be generated with our method we generated schematic layouts of the metro networks of Vienna ( $n = 90$ ,  $m = 96$ ) and Washington DC ( $n = 97$ ,  $m = 101$ ) for  $k = 3, 4, 5$ , see Figures 3 and 4. For these layouts we added planarity constraints on demand and concentrated on the layout geometry and interchanges without drawing individual stops along the lines. The weights in the objective function were set to the default values of  $f_1 = 3$ ,  $f_2 = 2$ ,  $f_3 = 1$  and we used  $s = 1$  admissible neighboring sectors for each original edge direction (recall Section II-B2).

The layouts were computed by IBM CPLEX 12.8.0.0 on an Intel Core i7-2600 CPU @ 3.4GHz running an Ubuntu 18.04.1 LTS system with 8GB RAM. Table I provides model sizes and measured running times. While the model size grows only moderately with  $k$ , the effect on the observed running times is much higher. The additional layout flexibility obtained from introducing more edge directions results in a strong increase in the running times for computing optimal solutions. While for  $k = 3$  optimal layouts can be computed in less than 20 seconds and for  $k = 4$  within a few minutes, the layouts for  $k = 5$  took already up to three hours to compute. Yet near-optimal solutions with a remaining optimality gap of less than 5% could be computed in under two minutes even for  $k = 5$ .

In terms of layout appearance, larger values of  $k$  typically improve the topographicity of the map since the angular distance between input slopes and the available directions in  $\mathcal{K}$  decreases, yet strictly speaking this does not need to be

TABLE I  
MODEL PARAMETERS AND RUNNING TIMES (WALL CLOCK TIME)

	Vienna			Washington DC		
	$k = 3$	$k = 4$	$k = 5$	$k = 3$	$k = 4$	$k = 5$
# var	21463	28205	34947	29379	38725	48071
# constr	6306	6675	7047	7895	8293	8691
time (sec)	8	254	10043	19	20	2401

true unless one value of  $k$  is a multiple of the other. On the other hand, one often sees fewer parallel lines and less symmetries for larger  $k$ , which means that the layouts can get a less structured and less uniform appearance.

### IV. CONCLUSIONS

We presented an algorithm to generate  $k$ -linear metro maps for arbitrary values of  $k$  as an extension of an existing ILP model for octilinear metro maps [9]. Our initial experiments show that this approach can generate metro maps of medium-sized benchmark instances in different linearity systems. It may thus serve as a useful tool for a map designer to examine the effect of different values of  $k$  on the potential general appearance of a map. Finding the linearity value that works best for a given input map is currently left to the map designer, although one could also try to define suitable quantitative measures and then optimize these over different linearity values. As future work, we want to include station labeling and investigate additional improvements of the practical performance, as well as integrating it into a human-in-the-loop tool for schematic map design.

### REFERENCES

- [1] K. Buchin, W. Meulemans, A. van Renssen, and B. Speckmann. Area-preserving simplification and schematization of polygonal subdivisions. *ACM Trans. Spatial Algorithms and Systems*, 2(1):2:1–2:36, 2016.
- [2] D. Delling, A. Gemsa, M. Nöllenburg, T. Pajor, and I. Rutter. On  $d$ -regular schematization of embedded paths. *Comput. Geom. Theory Appl.*, 47(3A):381–406, 2014.
- [3] C. A. Duncan and M. T. Goodrich. Planar orthogonal and polyline drawing algorithms. In R. Tamassia, editor, *Handbook of Graph Drawing and Visualization*, chapter 7, pages 223–246. CRC Press, 2013.

- [4] K. Garland. *Mr Beck's Underground Map*. Capital Transport Publishing, 1994.
- [5] D. Merrick and J. Gudmundsson. Path simplification for metro map layout. In M. Kaufmann and D. Wagner, editors, *Graph Drawing (GD'06)*, volume 4372 of *LNCS*, pages 258–269. Springer, 2007.
- [6] K. Misue, P. Eades, W. Lai, and K. Sugiyama. Layout adjustment and the mental map. *J. Visual Languages and Computing*, 6(2):183–210, 1995.
- [7] G. Neyer. Line simplification with restricted orientations. In F. K. Dehne, A. Gupta, J.-R. Sack, and R. Tamassia, editors, *Algorithms and Data Structures (WADS'99)*, volume 1663 of *LNCS*, pages 13–24. Springer, 1999.
- [8] M. Nöllenburg. A survey on automated metro map layout methods. In *Schematic Mapping Workshop*, Essex, UK, Apr. 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] M. J. Roberts. What's your theory of effective schematic map design? In *Schematic Mapping Workshop*, Essex, UK, 2014.
- [11] 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. *Int. J. Human-Computer Studies*, 2016.
- [12] 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. *Int. J. Human-Computer Studies*, 71:363–386, 2013.

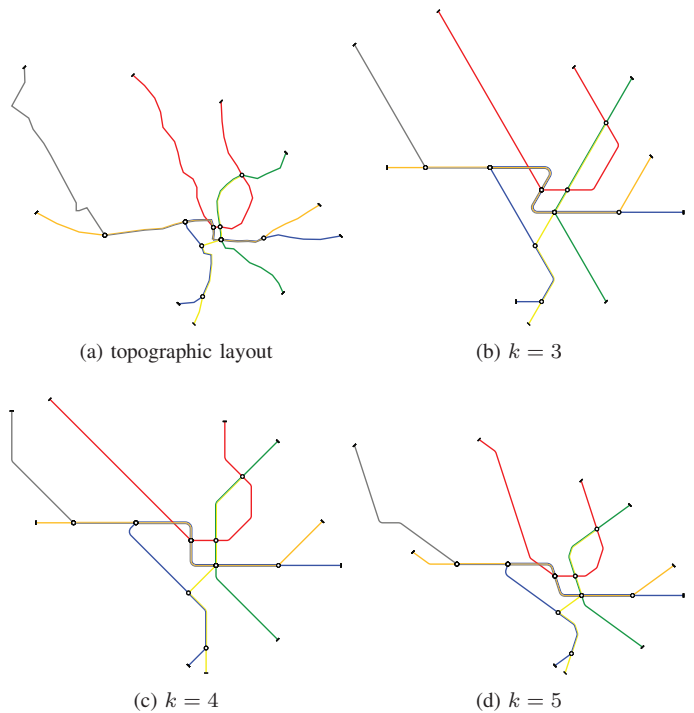


Fig. 4. Washington DC metro map in different linearities