Efficient automated schematic map drawing using multiobjective mixed integer programming

CrossMark

Olufolajimi Oke [a,b,*], Sauleh Siddiqui [a,b,c]

[a] *Department of Civil Engineering, The Johns Hopkins University, Baltimore, MD 21218, United States*
[b] *Systems Institute, The Johns Hopkins University, Baltimore, MD 21218, United States*
[c] *Department of Applied Mathematics and Statistics, The Johns Hopkins University, Baltimore, MD 21218, United States*

ARTICLE INFO

ABSTRACT

We present an efficient multiobjective mixed binary linear program that automates schematic mapping for network visualization and navigation. Schematic mapping has broad applications in representing transit networks, circuits, disease pathways, project tasks, organograms, and taxonomies. Good schematic maps employ distortion while preserving topology to facilitate access to physical or virtual networks. Automation is critical to saving time and costs, while encouraging adoption. We build upon previous work, particularly that of Nöllenburg and Wolff, improving upon the computational efficiency of their model by relaxing integrality constraints and reducing the number of objectives from three to two. We also employ an efficient augmented ϵ-constraint method to assist in obtaining all Pareto optimal solutions, both supported and unsupported, for a given network. Through the Vienna Underground network and a cancer pathway, along with three numerical examples, we demonstrate the applications of our methods. Finally, we discuss future directions for research in this area.

1. Introduction

A schematic map is a linear cartogram of a given network. Symbolic representations of pathways in schematic diagrams simplify complexity and facilitate orientation in a network. Thus, schematization increases the visual impact, making it easier to digest information. Distortion is regularly employed to make this possible. Fig. 1 illustrates the influence of schematization on a network user. A well-designed diagram is readily accessible and, through regular interaction, a user is ultimately able to recreate the image of the network (a process known as "mental mapping") to hasten the processes of orientation and navigation.

Schematic map drawing dates back centuries. A classic example is the *Tabula Peutingeriana* from the fifth century, a symbolic and distorted representation of the Roman road network across three continents [1]. In modern times, the power of schematization has been most visible in the representation of electric circuits and in the mapping of modern urban transit networks. Harry Beck's work in schematizing the London Underground map over a 30-year

period beginning in the 1930s set standards that are still relevant today [2,3]. Beck's representation did away with the curves of previous maps and replaced these with piecewise straight lines oriented in multiples of 45° (a scheme now termed *octilinear*). Beck also significantly enlarged the central portion of the map to make it more readable [2] (see Fig. 2). Subsequent official revisions to the Underground map by Beck and others rarely departed from his original ideal. The majority of the metro maps created after Beck's effort also followed his design cues [3], and a comparative study of these maps reveals a consistent set of principles now known as "metro map rules." Beyond metro networks and transportation networks in general, schematic maps have been found to be effective in other areas. They have been successfully used to represent cancer pathways [4], as well as organizational structures and project plans [5]. Schematic maps have broad applications, as they are effective in visualizing pathways in networks of all kinds.

Producing a good schematic map is an expensive and time-consuming process [6]. The best diagrams are usually created by professional designers with the aid of computer graphics software. Any represented network may undergo changes during its lifespan, and the needs and perceptions of its users will also vary with time. These usually require modifications to its schematic map. A procedure that can therefore automate the schematization process to give consistent results subject to specified requirements in real

---

* Corresponding author at: Department of Civil Engineering, The Johns Hopkins University, 3400 N Charles St, Baltimore, MD 21218, United States.
Tel. +1 413 329 6183.
E-mail addresses: ooke1@jhu.edu (O. Oke), siddiqui@jhu.edu (S. Siddiqui).

time would be ideal and efficient. Efforts in this area of research are detailed in Section 2. Of interest is Nöllenburg and Wolff's multiobjective mixed integer program implementation [7].

This paper describes our improvements on Nöllenburg and Wolff's method—a reduction in the number of objective functions and a relaxation of integer constraints—for better performance. Essentially, we reduced a mixed integer program to a mixed binary linear program and implemented a method of finding the entire set of non-dominated (Pareto optimal) solutions, which is critical as not all Pareto optimal solutions can be obtained by the weighting method approach employed by Nöllenburg and Wolff. We also establish a framework for analysis and decision-making in the Pareto space of equally efficient solutions. Finally, we show through some examples how schematic maps can be automated for a variety of applications.

2. Background and related work

Over the past 60 years, important insights have been gained in the automation of schematic maps. Purchase et al. [8] conducted a study that underscored the importance of aesthetics in schematization. In particular, their work proved that minimal line crossings and bends make maps easier to use and remember. In their review, Avelar and Hurni [6] motivate the importance of good design in schematizing transportation networks, and they propose a standardization in representation.
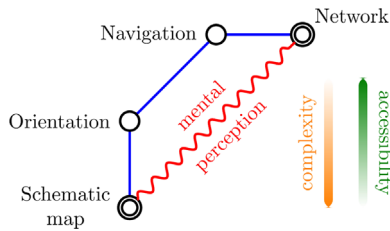

Fig. 1. Impact of schematization.

Tamassia [9] developed an algorithm that utilized the network properties of graphs to solve the problem of embedding a graph in a rectilinear grid. The problem was modeled in terms of minimum cost flow and the solution preserved topology while minimizing the bend count. Tamassia's work [9] did much to characterize important graphical elements of schematic maps, and also generalized the implementation for k-gonal graphs. Avelar and Müller [10] proposed an iterative algorithm for schematic map generation that also preserved input topology. They showed that topological accuracy could be maintained using relatively simple geometric analyses, as opposed to elaborate techniques. Their formulation also emphasized the role of the designer in setting constraints for aesthetics and legibility. Cabello et al. [11] describe a generalized procedure for schematizing transport maps using a path-endpoint framework. Their approach, which runs in linearithmic time, preserves the original layout but terminates unsuccessfully if no correct solution can be found.

Metro maps are a subset of schematic maps. Much of the work done in automating metro map drawings is generally transferrable to schematic maps, which tend to follow similar rules. Hong et al. [12] were the first to completely automate metro map graph layouts, and they demonstrated this through the use of various force-directed algorithms. Hong et al.'s implementation neither preserved the topology of the input embedding nor produced octilinear solutions. Ribeiro et al. [13] also use a force-directed implementation to automatically draw schematic maps. Their results are efficiently produced and aesthetically pleasing, but they focus on using spider maps in representing transit networks. Stott et al. [14] later introduced a hill-climbing algorithm within a multiobjective optimization framework to solve the metro map automation problem. Their optimizer gave improved results, but octilinearity was not enforced and optimality could not be guaranteed. Nöllenburg and Wolff [15,7] then developed a multiobjective mixed integer program to solve the metro map drawing problem. Like Stott et al. [14], Nöllenburg and Wolff [7] used the weighting method to find optimal octilinear results.

While Nöllenburg and Wolff's multiobjective formulation is a significant improvement compared to previous efforts and is able to
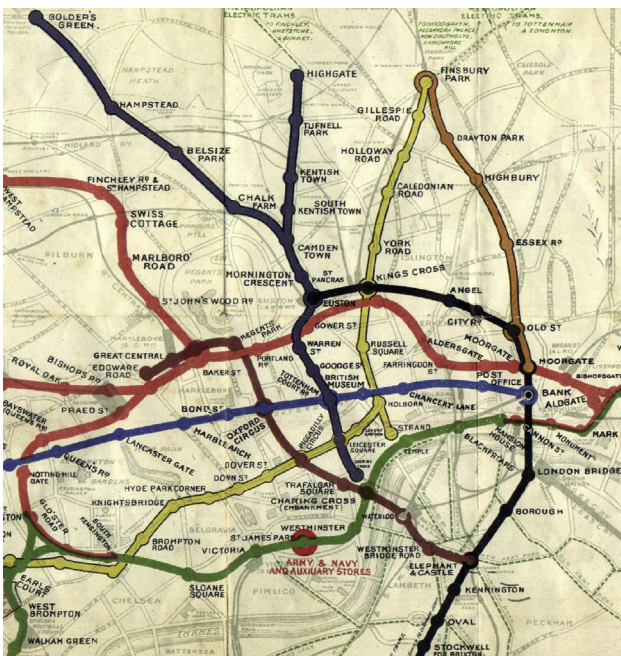

Fig. 2. Central sections of the 1908 edition of the London Underground map and Beck's first schematic version published in 1933 [2].

produce good unlabeled maps, there is room for further development. Nöllenburg and Wolff used the weighting method, which may ignore potentially desirable solutions [16]. Their tri-objective model, while elegant, does not lend itself easily to analysis of the solution space. Finally, the coordinate integrality requirements unnecessarily slowed down their method. We sought to address these issues by simplifying their model, relaxing coordinate integer constraints, and implementing an augmented ϵ-constraint method developed by Mavrotas and Florios [17] to determine the set of efficient solutions and analyze the objective space.

Multiobjective mixed integer programs (MOMIPs) have been successfully used to model numerous real-life problems. For example, Scappara and Church developed a two-level approach to a mixed-integer program for infrastructure management [18]. Much important work has been done in finding non-dominated solutions to optimization problems, and several algorithms have been tailored to their specific applications [19,20]. In the area of visual representation, Cano et al. [21] introduced a novel meta-heuristic approach to solving the problem of proportionally symbolic cartograms. Marti and Estruch [22] also proposed a heuristic method for solving the edge-crossing minimization problem.

Various other research problems related to automated schematic map drawing have also been tackled, such as those dealing with bend minimization and path simplification [23,24]. In a departure from convention, Fink et al. [25] have proposed a "curvilinear" drawing approach which has produced metro maps using a force-directed algorithm. Hurter et al. [26] have also created a schematic automation method specifically for air traffic control routes. They use simulated annealing to solve a cost minimization model that penalizes crossings, density and distortion.

2.1. Why automation is important

It has been amply demonstrated that distortion, when correctly applied, can greatly enhance the mental perception of a network [8,14,6]. The best diagrams still require significant manual effort to produce and usually at great expense [6]. For example, the London Underground network map took over three decades to attain the widely accepted form that has not changed much since the 1950s [2]. This is why computer automated drawings are crucial for modern applications. With the emergence of modern mobile technology, demand is ripe for dynamic maps that can toggle on or off layered information. We focus on static maps in this paper, but we keep in mind that advances in computing technology will mean that complex problems will be increasingly solvable in real time. The computer-automated process in schematic diagramming can save significant amounts of time and money in map making, with a potential for impact in transit network design [27], understanding disease progressions [4], and visualizing complex plans [5] and pathways.

Guo [27] has shown that schematic maps influence how passengers perceive and utilize a transit network. Using the London Underground network as a case study, it can be argued that passengers' favorable disposition to a network is affected by how it is represented, a phenomenon Guo coins as the "map effect." In the case of bicycle networks, numerous efforts, both personal and official, have been made to schematize their maps in the hopes of boosting ridership. An automated solution to this can significantly increase the production of schematic maps. Bicycle networks are of particular interest [28], as urban sustainability through reduced automobile congestion and increased public health benefits [29] are critical outcomes of positive network perceptions.

2.2. This paper's contributions and improvements

Our work is significant in that it shows we can generate schematic solutions much quicker by simplifying, reformulating, and tightening the multiobjective model presented by Nöllenburg and Wolff [7]. In particular, we relax the integrality constraints on the coordinate variables to achieve this efficiency. We still obtain proven optimal solutions, and we show this through three numerical examples and two applications.

An important characteristic of an automatic schematization tool is to provide a useful starting point for a designer to produce a final representation, as computers have limited decision-making ability. Beyond manually selecting weights to obtain tailored solutions, we show how the ϵ-constraint method of solving multiobjective programs can be used to efficiently generate a Pareto frontier of non-dominated solutions. Specifically, we adapt Mavrotas and Florios' augmented ϵ-constraint implementation [17]. Our results show how a framework can be developed for evaluating desirable schematic diagrams from a set of efficient (Pareto optimal) solutions. This is the first time this method has been used in solving the schematic map drawing problem, and we believe this is a critical development in that all Pareto points—which may not always be found by varying weights—can be obtained by the user or the designer.

Automatic schematic map drawing has largely been presented within the sphere of metro map drawing. This is understandable, as much of the interest in this subject arose from efforts to automate metro maps. In this paper, we implement our method on the Vienna Underground as a case study. However, we go further to illustrate how the automatic schematic method can also be applied to disease pathway representation. We hope that our efforts will spur further work in broadening the applications of automatic schematic mapping.

3. The schematic drawing problem

The schematic map drawing problem is formally defined as follows:

Find the best drawing Γ of the graph $(G, \mathcal{L})$ subject to schematic rules, where $G(V, E)$ is a planar input embedding $\Pi(v)$ of a set V of vertices and a set E of edges, with $\mathcal{L}$ the set of lines or line cover of G.

The vertices v_i and edges e_i, elements of V and E, respectively, constitute the topology of G. Together, these elements divide the graph into bounded zones known as faces, where each face f is an element of F, the set of all faces. Trailing or circumferential edges collectively make up the unbounded external face [9]. A graph can thus be further described by a list of the edges, in circular order, occurring in each face—a planar representation [9]. The sets of vertices, edges and faces in a planar graph are related by Euler's formula:

$$|E| + 2 = |V| + |F| \tag{3.1}$$

We summarize these definitions below:

$G(V, E)$	planar input graph, as a function of V and E
$(G, \mathcal{L})$	possible metro graph
Γ	drawing operator
$\Pi(v)$	geographic location of each vertex v in the plane
$\mathcal{L}$	set of lowest number of paths linking all vertices in G (line cover)
L	metro line (element in $\mathcal{L}$)
E	set of edges e in graph
V	set of vertices v in graph G
F	set of faces f in graph
l	number of lines in G
m	number of edges of G
n	number of vertices of G
r	number of faces of G

The nature of the schematic rules enforced depend on the type of map being drawn. For the purposes of this paper, we restrict ourselves to the Beck-inspired diagram whose aesthetics are encapsulated within the following major guidelines:

(A) *Octilinearity*: All edges must originate at angles that are multiples of $45°$. This necessitates that the output be on an octilinear grid with four axes: x ($0°$), y ($90°$), z_1 ($45°$) and z_2 ($-45°$).

(B) *Bend avoidance*: Along each line, bends must avoided as much as possible. Obtuse bends are favored over acute or right ones.

(C) *Topological correctness*: The relative positions of vertices and their connectivities must remain the same, for the sake of readability and mental mapping. In other words, the face structure of the solution is identical to that of the input.

(D) *Edge uniformity and spacing*: Edge lengths should be kept as even as possible. Nonadjacent edges must also keep a minimum distance to ensure clarity.

(E) *Edge shifting*: Imposing an optimal octilinear output necessitates linear distortion. Edges are allowed to be shifted in keeping with this geometry as long their relative positions to other adjacent edges are maintained.

A few of these guidelines are illustrated in Fig. 3.

We do not address the labeling problem in this paper. If labels were to be included, however, they would ideally not overlap and would be grouped into blocks with the same orientations. Line colors should also be optimally distinguishable, but this component is not included in our implementation. In the mixed integer formulation that will be discussed in the following section, guidelines (A), (C) and (D) are modeled as hard constraints since they are strict requirements. Guidelines (B) and (E) are however modeled as objective cost functions or soft constraints, as they describe aesthetic preferences.

4. Mixed integer program formulation

The mixed integer program presented in this paper develops the model introduced by Nöllenburg and Wolff [7] into a multi-objective mixed binary linear program. Unlike Nöllenburg and Wolff's formulation, which has three objective cost functions, ours utilizes only two, namely: shift and bend. We eliminate their "total edge length" cost function, replacing this with an upper bound on edge lengths ℓ_{max}. This can have the same value for all edges, or take on different values based on the edge or face properties (as we demonstrate in the Vienna Metro and cancer pathway examples). The conversion to a biobjective problem facilitates analyses, especially regarding tradeoffs on a two-dimensional Pareto frontier. Notably, we also find that it is not necessary to impose integer constraints on the vertex coordinates. The subsequent relaxation greatly speeds up solve times, as we will demonstrate. Numerical evidence indicates that Pareto optimal solutions will remain on the octilinear grid and we obtain them faster with integrality relaxation in place.

The bend and shift costs describe the objective functions that address the aesthetic considerations of schematic maps in the Beck style [2]. The objectives are subject to three sets of hard constraints that deal with the following properties: octilinearity and edge length, circular ordering and nonadjacent edge spacing. We describe each set of constraints in detail. Before this, we introduce the geometric and combinatorial framework for the input problem layout.

4.1. Coordinate space for octilinear grid

To facilitate constraint formulation, Nöllenburg and Wolff proposed two new axes in the $\pm 45°$ directions, namely z_1 and z_2. Each vertex v can thus be referenced by four coordinates: $x(v), y(v), z_1(v), z_2(v)$, although only the first two are sufficient to locate the vertex. The new axes are related to the conventional ones in the following manner:

$$z_1(v) = \frac{x(v)+y(v)}{2}$$
$$z_2(v) = \frac{x(v)-y(v)}{2} \qquad (4.1)$$

This definition reinforces the L^∞ metric used as the basis for the grid. Thus, all vertices at the corner of a unit square are equidistant from the origin as are those centered on each side of the square. The coordinate system is shown in Fig. 4.

It is important to note that integer coordinates (on all four axes) will always guarantee grid placement. Our numerical results indicate that we can obtain Pareto optimal grid-based points without specifying integer constraints on the coordinates x, y, z_1 and z_2, in a departure from Nöllenburg and Wolff's implementation [7]. By relaxing the coordinate integer constraints

$$0 \le x(v), y(v), z_1(v), z_2(v) \in \mathbb{R} \qquad (4.2)$$

we gain efficiency. There are still binary variables in the model that remain unchanged. In a few examples in Section 4, we compare our relaxation results against those of Nöllenburg and Wolff, highlighting significant improvements in execution time.

We hold that optimal solutions to the schematic drawing problem will always be found on the octilinear grid. We demonstrate this idea through a few examples. While some planar representations can have off-grid optimal solutions, as we will show, it appears that on-grid optimal solutions are the cheaper to compute. It is on this basis that we motivate the relaxation of the coordinate integer constraints Nöllenburg and Wolff imposed in their model. Our observation is that we still obtain integer solutions but in a shorter period of time, as fewer nodes have to be traversed during the branch-and-bound search.

Consider the graph in Fig. 5a, which is the solution to a given input embedding, where $x(1)=0$, $x(2)=2$ and $x(3)=4$. If we fixed $x(3)=4.5$ before solving, then the solution would result in a shift
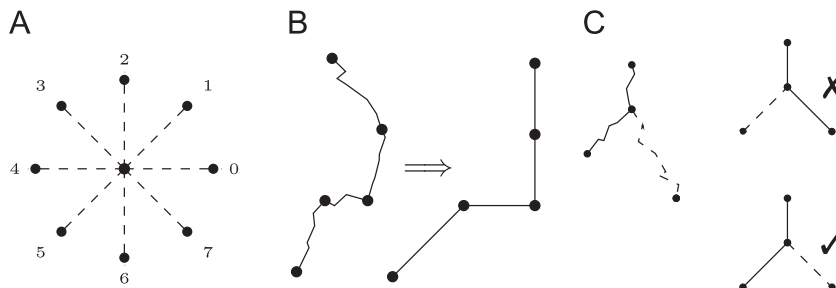

Fig. 3. Illustration of three guidelines under consideration for schematization. (a) Edge octilinearization, (b) line bend minimization, (c) topology preservation. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

of $x(1)$ and $x(2)$, as well (Fig. 5b). This is essentially a translation of the grid in order to maintain octilinearity. If we decided, however, to keep $x(2)$ at 2, while $x(3)$ was also fixed at 4.5, the solution would result in a shift of $y(3)$ to maintain octilinearity (Fig. 5c).

Also, moving any one of the y coordinates would result in a solution that translates the entire graph by the same amount. If two y coordinates are forced to be in a position where the resulting edge cannot be octilinear, then the problem becomes infeasible.

Non-integer edge lengths can always be forced to appear, as long as octilinearity is not violated. Sometimes, this would result in scaling or translating the grid. In such cases, the solution would still be considered as on a grid. As will be seen in the next several subsections, the minimum integer edge lengths we set also encourage integer solutions. Also, solver architecture ensures we never encounter non-integer solutions. An optimal integer result is always favored, and even more so since we set integer edge lengths as starting points.

To bring this point further home, we consider a second example (Fig. 6a). This graph has one bounded face. A possible outcome of fixing $y(1)$ at $v_{1'}$ is a translation of the entire graph (Fig. 6b). The solution is still essentially on a grid. While y coordinates have non-integer values, the edges maintain integer lengths. Forcing both $x(1)$ and $y(1)$ to remain at $v_{1''}$ might result in a scalar expansion of the graph, a solution which still lies on a grid (Fig. 6c). Scaling the grid accordingly essentially means the edge lengths have integer values on the new scale.

The examples in Figs. 5 and 6 are simple but atypical for most real-world applications. They both have only one unbounded face. For a graph with a more complicated structure (two or more faces), it becomes even more difficult to find an off-grid Pareto optimal point. We consider a four-face planar graph (Fig. 7). A non-integer edge length may be impossible to have if adjacent edges cannot be adjusted accordingly to retain the face structure and

maintain octilinearity. If edge $e(5, 6)$ (in the z_2 direction), for example, were to have a non-integer edge length that was not proportional to that of $e(1, 6)$ and $e(6, 3)$, an octilinear structure would not be possible.

We can further motivate the existence of a drawing with integer edge lengths for every octilinear graph of a given input embedding by assuming that all edge lengths in such a graph are rational numbers. It is then trivial to show that there is always a number that can be multiplied by each edge length to produce an integer solution.

In summary, integer Pareto optimal points can be obtained without enforcing integer constraints. The major difference in our model, compared to Nöllenburg and Wolff's [7], is the reduction of a mixed integer program to a mixed binary linear program, which is demonstrably easier to solve, and the application of the augmented ϵ-constraint method to find efficient solutions.

4.2. Sector assignments

The key building block of this formulation, which Nöllenburg and Wolff introduced, is the sectorialization of edges. Based on the eight-direction octilinear grid, each edge $e(u, v)$ in the input embedding is assigned to a sector $sec_0(u, v)$ based on its angular distance ($\pm 22.5°$) from it. Thus,

$$sec_0(u, v) = \begin{cases} 0 & 337.5° \leq \angle uv < 22.5° \\ 1 & 22.5° \leq \angle uv < 67.5° \\ 2 & 67.5° \leq \angle uv < 112.5° \\ \vdots & \\ 6 & 247.5° \leq \angle uv < 292.5° \\ 7 & 292.5° \leq \angle uv < 337.5° \end{cases} \qquad (4.3)$$

In pursuit of an optimal solution, each edge is allowed to either remain in its original sector sec_0 or move a sector forward sec_{+1} or backward sec_{-1}:

$$sec_{+1} = sec_0 + 1 \bmod 8 \qquad (4.4)$$

$$sec_{-1} = sec_0 - 1 \bmod 8 \qquad (4.5)$$

The variable $dir(u, v)$ holds the value of the sector each edge in the solution eventually takes. This disjunction sets up a combinatorial problem which necessitates the use of binary variables α_0, α_{+1} and α_{-1} for each edge, which can only take one of three possible sectors. These variables must satisfy the relation

$$\sum_{i \in \{-1,0,1\}} \alpha_i(u, v) = 1 \qquad (4.6)$$

The directional variables are thus described as follows:

$$dir(u, v) = \sum_{i \in (-1,0,1)} sec_i(u, v) \cdot \alpha_i(u, v) \qquad (4.7)$$

$$dir(v, u) = \sum_{i \in (-1,0,1)} sec_i(v, u) \cdot \alpha_i(u, v) \qquad (4.8)$$

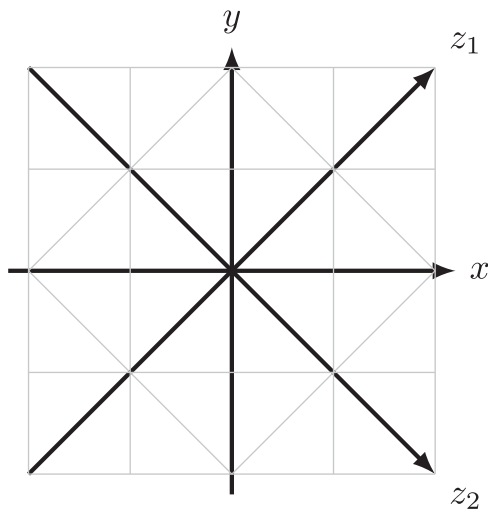The direction and sector variables are directional, and all movements are represented in modulo 8. Thus,

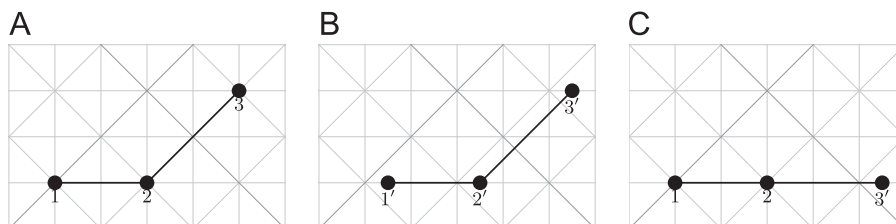**Fig. 4.** Quadraxial coordinate system and octilinear grid.

Fig. 5. Non-integer and off-grid solutions are possible in a simple face structure such as this one, but they are typically more costly to find.
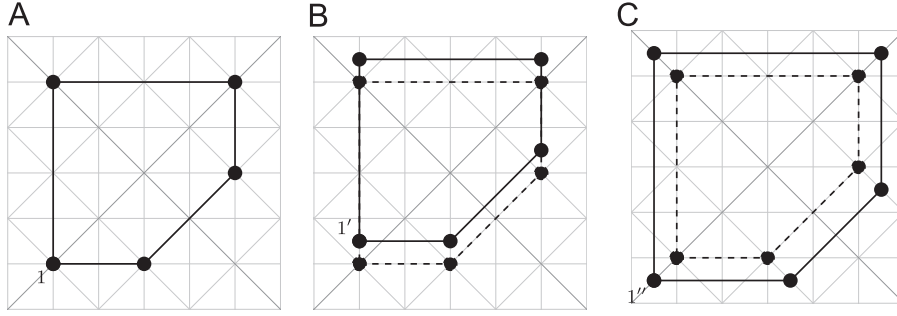
Fig. 6. Fixing non-integer coordinates creates solutions which retain integer edge lengths on a translated or scaled grid.
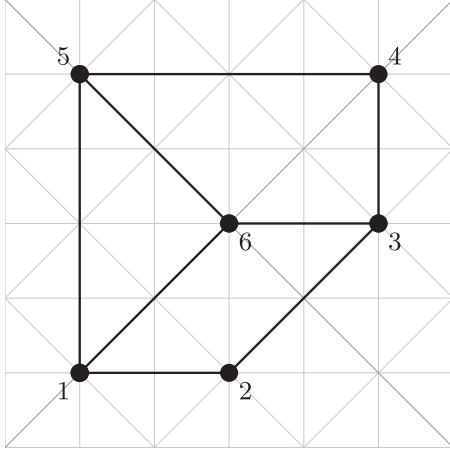

Fig. 7. A more complex face structure makes it difficult or impossible to find completely off-grid solutions.

$$\mathrm{sec}_i(u,v) = \mathrm{sec}_i(v,u) + 4 \bmod 8 \qquad (4.9)$$

$$\mathrm{dir}(u,v) = \mathrm{dir}(v,u) + 4 \bmod 8 \qquad (4.10)$$

4.3. Octilinearity and edge length constraints

Further disjunctive constraints are necessary for ensuring octilinearity. For each of the eight possible directions, they require the ordinates of the edge be equal, while the abscissae must differ by a minimum edge length $\ell_{\min}(u,v)$. The constraints are shown below for sectors 0 and 1. Similarly constructed constraints hold for the six other sectors. They are disjunctive constraints [30], represented using a conjunction of linear constraints in what is termed the "big-M formulation":

$$\mathrm{sec}_i(u,v) = 0 : \begin{cases} y(u) - y(v) \le M(1 - \alpha_i(u,v)) \\ -y(u) + y(v) \le M(1 - \alpha_i(u,v)) \\ -x(u) + x(v) \ge -M(1 - \alpha_i(u,v)) + \ell_{\min}(u,v) \end{cases} \qquad (4.11)$$

$$\mathrm{sec}_i(u,v) = 1 : \begin{cases} z_2(u) - z_2(v) \le M(1 - \alpha_i(u,v)) \\ -z_2(u) + z_2(v) \le M(1 - \alpha_i(u,v)) \\ -z_1(u) + z_1(v) \ge -M(1 - \alpha_i(u,v)) + \ell_{\min}(u,v) \end{cases}$$

$$(4.12)$$

The coordinate values are bounded above by M:

$$x(v), y(v) \le M \quad \forall v \in V \qquad (4.13)$$

If M is too large, the model might become unstable or computation time may be lengthened. If M is not large enough, some otherwise optimal solutions may be rendered infeasible. In our implementation, we have chosen M as

$$M = \sum_{e \in E} e(u,v) \cdot \ell_{\max}(u,v) \qquad (4.14)$$

which gives an upper bound with the assumption that all the edges are sequentially connected, thus spanning the map in any direction in L^∞. This value has proved large enough to accommodate all Pareto optimal points, while keeping the model stable.

In another departure from Nöllenburg and Wolff's implementation, we assume edge length maxima are user-defined. While Nöllenburg and Wolff minimize the sum of all edge lengths, we argue for setting a hard constraint on these:

$$\begin{aligned} x(u) - x(v) &\le \ell_{\max}(u,v) \\ -x(u) + x(v) &\le \ell_{\max}(u,v) \\ y(u) - y(v) &\le \ell_{\max}(u,v) \\ -y(u) + y(v) &\le \ell_{\max}(u,v) \end{aligned} \qquad (4.15)$$

We not only gain time in computation, but more critically, our formulation is now biobjective, making it easier to analyze (two-dimensional Pareto frontier) and quicker to solve (fewer variables).

4.4. Circular order constraints

An important set of constraints that preserve topology (input face structure) and guarantee readability are the circular order constraints. They ensure that spanning edges from a vertex retain their relative position and do not overlap. This set of constraints is expressed as

$$\begin{aligned} \mathrm{dir}(u,v_j) &\le \mathrm{dir}(u,v_{j++1}) - 1 + 8\beta(u,v_j), \\ &\forall u : \deg(u) \ge 2, j = \{1, 2, \ldots, \deg(u)\} \end{aligned} \qquad (4.16)$$

where

$$\sum_{j=1}^{\deg(v)} \beta(u,v_j) = 1 \quad \forall u : \deg(u) \ge 2 \qquad (4.17)$$

The order constraints only hold for vertices whose degree (number of spanning edges) is greater than or equal to 2. We can further control the number of variables required by restricting the application of these constraints to vertices of degree 3 or greater and only to degree-2 vertices for which the edges are 2 or fewer sectors apart.

4.5. Edge spacing constraints

Edge spacing constraints ensure a specified minimum distance $d_{\min}$ between nonadjacent edges on the same face. The binary variable γ_i, indexed over the 8 cardinal (or axial) directions, ensures that this minimum distance is kept in at least one direction (or the relevant direction) for each nonadjacent same-face edge pairing. Hence,

$$\sum_{i \in \{N, NW, \ldots, E, NE\}} \gamma(e_1(u,v), e_2(u,v), f) \ge 1 \qquad (4.18)$$

The big-M formulation is again employed to model disjunction. The two sets for the north (N) and the northwest (NW) directions

are given below.

$$y(u_2)-y(u_1) \leq M(1-\gamma_N(e_1,e_2,f))-d_{min}$$
$$y(u_2)-y(v_1) \leq M(1-\gamma_N(e_1,e_2,f))-d_{min}$$
$$y(v_2)-y(u_1) \leq M(1-\gamma_N(e_1,e_2,f))-d_{min}$$
$$y(v_2)-y(v_1) \leq M(1-\gamma_N(e_1,e_2,f))-d_{min} \qquad (4.19)$$

$$-z2(u_2)+z2(u_1) \leq M(1-\gamma_{NW}(e_1,e_2,f))-d_{min}$$
$$-z2(u_2)+z2(v_1) \leq M(1-\gamma_{NW}(e_1,e_2,f))-d_{min}$$
$$-z2(v_2)+z2(u_1) \leq M(1-\gamma_{NW}(e_1,e_2,f))-d_{min}$$
$$-z2(v_2)+z2(v_1) \leq M(1-\gamma_{NW}(e_1,e_2,f))-d_{min} \qquad (4.20)$$

The minimum spacing requirement also ensures that edge crossings are avoided. (See the Appendix for a full list of the inequalities.)

4.6. Cost functions and soft constraints

The optimal solution must maximize the obtuseness of all bends in each line. The bend cost, to thus be minimized, is defined as

$$C_{bend} = \sum_{L \in \mathcal{L}} \sum_{uv,vw \in L} bend(u,v,w) \qquad (4.21)$$

This cost can be described as the sum of all the bend costs for adjacent edge pairings in each line. Constraints are designed to assign costs of 1, 2, and 3 to bends of $135°, 90°$ and $45°$, respectively [7]:

$$\Delta dir(u,v,w) = dir(u,v) - dir(v,w) \qquad (4.22)$$

$$bend(u,v,w) = \min\{|\Delta dir(u,v,w)|, 8-|\Delta dir(u,v,w)|\} \qquad (4.23)$$

We recall that $dir(u,v) \in \{0,1,...,7\}$ denotes the sector assigned to the edge $e(u,v)$ in the solution. The difference $\Delta dir(u,v,w)$ between the directional variables of adjacent edges $e(u,v)$ and $e(v,w)$ gives a measure of the bend angle. Since there is a wrap-around at the modulus 8, we find the minimum of the absolute value of Δdir and its difference from 8 (Eq. (4.24)). Linearizing this equation requires the use of two more binary variables, $\delta_1(u,v,w)$ and $\delta_2(u,v,w)$, thus:

$$-bend(u,v,w) \leq \Delta dir(u,v,w) - 8\delta_1(u,v,w) + 8\delta_2(u,v,w)$$
$$bend(u,v,w) \geq \Delta dir(u,v,w) - 8\delta_1(u,v,w) + 8\delta_2(u,v,w) \qquad (4.24)$$

Eqs. (4.24) simply represent the inequality

$$bend(u,v,w) \geq |\Delta dir(u,v,w) - 8\delta_1(u,v,w) + 8\delta_2(u,v,w)| \qquad (4.25)$$

As the bend cost is minimized, δ_1, δ_2 or both are switched on or off in order to correctly calculate the bend cost for each edge, which will have a value of 0, 1, 2, or 3. For instance, if two adjacent edges have $dir(u,v) = 1$ and $dir(v,w) = 7$, respectively, then $\Delta dir(u,v,w) = 1-7 = -6$. The cost of this bend would be $bend(u,v,w) = 2$, where $\delta_1(u,v,w) = 0$ and $\delta_2(u,v,w) = 1$.

Each edge also incurs a cost of 1 for shifting forward or backward from its original sector. The total shift cost of the octilinear drawing is given by

$$C_{shift} = \sum_{uv \in E} shift(u,v), \qquad (4.26)$$

where

$$-M \cdot shift(u,v) \leq dir(u,v) - sec_{orig}(u,v) \leq M \cdot shift(u,v) \qquad (4.27)$$

For each edge $e(u,v)$, $shift(u,v)$ is thus a binary variable. For example, consider $sec_0(u,v) = 3$ for a given edge in an input embedding. If $dir(u,v) = sec_0(u,v) = 3$, then $shift(u,v) = 0$. However, if $dir(u,v) = sec_{-1}(u,v) = 2$ or $dir(u,v) = sec_{+1}(u,v) = 4$, then $shift(u,v) = 1$. The total shift cost is the sum of all $shift(u,v)$ for every edge $e(u,v)$ in the graph.

4.7. Implementing the augmented ϵ-constraint method

An important question in schematic map automation is how the decision on the final solution is made. For any given input embedding, a number of acceptable solutions may exist. In previous multiobjective formulations, particularly the one developed by Nöllenburg and Wolff [15,7], the weighting method has been used to find these points. In our modified biobjective case, this would be written as

$$\min: \quad \lambda_{bend}C_{bend} + \lambda_{shift}C_{shift} \qquad (4.28)$$

$$\text{s.t.} \quad \text{octilinearity and edge length constraints} \qquad (4.29)$$

$$\text{circular order constraints} \qquad (4.30)$$

$$\text{edge spacing constraints} \qquad (4.31)$$

where λ_{bend} and λ_{shift} are appropriately chosen weights for bend and shift, respectively. This method is useful for generating a single result based on the desired weighting combination. However, the weighting method may fail to generate the complete set of Pareto optimal solutions, regardless of the choice of weighting combinations [16]. These supported solutions are obtained by optimizing a convex combination of the objectives. However, unsupported Pareto optimal solutions may also exist. The ϵ-constraint method is therefore necessary in filling any such gaps in finding efficient solutions. Scaling can also be an issue in the weighting method, and one might need to adjust the objective functions to reduce the effects of uneven matching. This is a non-issue in the ϵ-constraint method [31].

We thus implement an efficient version of the ϵ-constraint method to counter the influence of weight scaling, relieve the user of the burden of deciding on weighting combinations and produce both nonextreme and, importantly, unsupported Pareto optimal solutions. Our algorithm of choice is the augmented ϵ-constraint method (AUGMECON2) developed by Mavrotas and Florios [17] and implemented in GAMS. (Earlier, Mavrotas [31] presented the AUGMECON method, which has since been superseded by AUGMECON2 in terms of performance.)

The basic augmented ϵ-constraint method for a multiobjective problem optimizes the first objective and successively uses the others as constraints bounded by the values generated in a payoff table partitioned into grid intervals. AUGMECON2 is preferred over the basic ϵ-constraint method for a number of reasons. First, it employs lexicographic optimization to calculate the payoff table, thus avoiding dominated solutions. Second, it ensures that only efficient (and not weakly efficient) solutions are produced. This is implemented by replacing the inequalities for the constrained objective functions with equalities and corresponding slack variables; the main objective is modified accordingly. Finally, AUGMECON2 [17] uses a "jump" procedure, in which bypass coefficients are calculated from the innermost slack variables to skip over grid locations that do not produce Pareto optimal solutions. This bypass feature improves performance, as does the "early exit loop" feature (first introduced in AUGMECON [31]) that skips to the next row of grid points once an infeasibility is detected. Overall, these features make Mavrotas and Florios' augmented ϵ-constraint method attractive for solving multiobjective mixed integer programs such as ours.

Our formulation is given below. The second term in Eq. (4.32) is the "augmentation."

$$\min: \quad C_{bend} - \epsilon \frac{s_{shift}}{r_{shift}} \qquad (4.32)$$

$$\text{s.t.} \quad C_{shift} + s_{shift} = e_{shift} \qquad (4.33)$$

$$e_{\text{shift}} = \text{ub}_{\text{shift}} - \frac{i_{\text{shift}} r_{\text{shift}}}{g_{\text{shift}}} \qquad (4.34)$$

where $\epsilon = 10^{-3}$ (a small scalar quantity), s_{shift} is the slack (or surplus) variable for the shift function, r_{shift} is the range of the payoff table for the shift cost function (used as a scaling factor), and e_{shift} is the RHS of the constrained shift function, whose value depends on the grid location in the payoff table. ub_{shift} is the upper bound of the shift function C_{shift}, and $i_{\text{shift}} \in \{0, 1, \ldots, g_{\text{shift}}\}$, where g_{shift} is the number of grid intervals determined by the user. The complete Pareto frontier is obtained when $g_{\text{shift}} := r_{\text{shift}}$. Fewer grid points would result in shorter computation times but would give a sparser Pareto frontier [31]. The user, therefore, may want to carefully consider the selection of grid intervals, especially when solving large problems.

Using AUGMECON2, we obtain a Pareto frontier that enables us to visually and quantitatively evaluate the tradeoff between shift and bend across all the noninferior solutions.

5. Preliminary evaluation

We present three working examples to evaluate the performance of our implementation, compared to the earlier work of Nöllenburg and Wolff. The three examples are introduced in order of increasing complexity.

5.1. Minimal example

This first minimal example consists of four vertices and two lines (1–2–3 and 2–4). Fig. 8 shows the input embedding and the original sectors to which each edge belongs.

For ease of reference, we have assigned numerical labels to the vertices. From Fig. 8b, we see that edge $e(2, 3)$ is originally positioned at an angle slightly greater than $22.5°$ to the horizontal. It is thus assigned to sector 1, that is $\sec_0(2, 3) = 1$. The circled numbers in Fig. 8b denote the sector numbers from 0 to 7. Similarly, edge $e(2, 4)$ lies within $\pm 22.5°$ of $90°$, hence it is assigned to sector 2: $\sec_0(2, 4) = 2$. Table 1 shows the sector assignments for each edge. In each case, $\sec_{-1}$ and $\sec_{+1}$ are also shown. In the solution(s), each edge can be assigned to any of the three sectors through the directional variable $\text{dir}(u, v)$, as long as none of the hard constraints are violated.

Vertex 2 has three spanning edges, thus:

$$\deg(2) = 3 \qquad (5.1)$$

It is clear that in this model, edges $e(1, 2)$ and $e(2, 3)$ will not overlap under any circumstances (see Eq. (4.7)). The circular order constraints (Eqs. (4.16) and (4.17)), however, ensure that other edge pairs $e(2, 3)$ and $e(2, 4)$ or $e(1, 2)$ and $e(2, 4)$ never overlap. The edge spacing constraints are not under consideration here, as there are no nonadjacent edge pairs.

The optimal solutions to this input embedding are trivial (Fig. 9). In the first solution (Fig. 9a), $\{C_{\text{bend}}, C_{\text{shift}}\} = \{0, 1\}$, while in the second (Fig. 9b), $\{C_{\text{bend}}, C_{\text{shift}}\} = \{1, 0\}$, since all three edges retain their original sector positions.

These two figures show that while mathematical models can produce Pareto optimal solutions, humans remain the best equipped to decide which solutions are desirable. Pareto point (A) has zero line bends but it sacrifices relative accuracy in the positioning of edge $e(2, 3)$. For the decision-maker who does not consider a $135°$ bend a problem but prioritizes accuracy in representation, then the preferred solution would be (B).

5.2. Dual-line network

We present a second example that still has only 2 lines (1–2–4–5–6 and 2–3–5–7) but with a more complex structure. This dual-line network has 2 faces, 7 edges and 7 vertices (Fig. 10). The internal face is bounded by the edges $(2, 3), (3, 5), (2, 4)$ and $(4, 5)$. There are several nonadjacent same-face edge pairings in this example, thus making the edge spacing constraints (Eqs. (4.18) and (4.19)) relevant.

Table 2 shows the sector assignments for all seven edges. We recall that the angle between two vertices u and v is measured counterclockwise from the horizontal. From Fig. 10, it is clear that $\sec_0(3, 5) = 0$. For this and the other edges, Eq. (4.3) is used to determine the sector assignments.

Our augmented ϵ-constraint implementation generates three Pareto optimal solutions (Fig. 11). Solution (A) has the lowest bend cost but it also has the highest shift cost, illustrating why solutions with smoother lines may appear increasing dissimilar to the original map. Solution (C) has no shifted edges, but it has the costliest bends. A designer, for instance, might be dissatisfied with the amount of geometric distortion in Solution (B), especially those of edges $e(1, 2)$ and $e(5, 7)$, in which case they may decide to increase their lengths. Other factors, such as the placement of geographic features or label positioning, can influence the choice of a solution.

The graph of the Pareto frontier in Fig. 12 visualizes the tradeoff between shift and bend for the three solutions. A graphic such as this one would be useful in evaluating the merits of each representation relative to the others.

Since this example is still relatively simple, the solutions may seem obvious, not necessarily requiring automated assistance to generate. The Pareto frontier, however, enables human involvement in the decision-making process.

Table 1
Sector parameters for each edge.

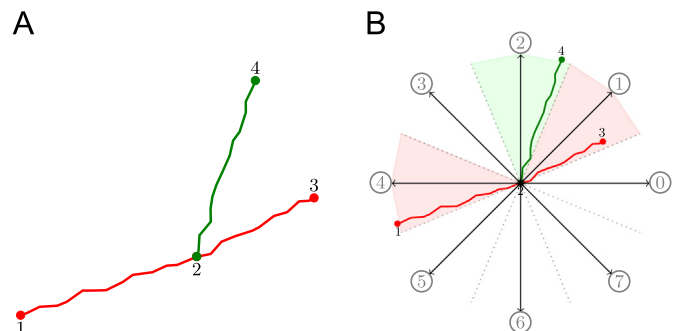| Edge uv	$\sec_{-1}(u, v)$	$\sec_0(u, v)$	$\sec_{+1}(u, v)$
1–2	7	0	1
2–3	0	1	2
2–4	1	2	3

Fig. 8. Minimal example: 2 lines (1–2–3 and 2–4), 3 edges, 4 vertices, 1 face; $\{l, m, n, f\} = \{2, 3, 4, 1\}$. (a) Input embedding, (b) initial sectors of the edges.
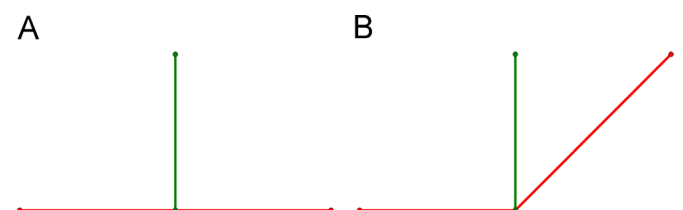

Fig. 9. Pareto optimal solutions to minimal example. (a) $\{C_{\text{bend}}, C_{\text{shift}}\} = \{0, 1\}$, (b) $\{C_{\text{bend}}, C_{\text{shift}}\} = \{1, 0\}$.

5.3. Four-line network

This four-line example is more robust with a structure likely to be found in an actual network. It has 13 edges, 11 vertices and 4 faces (Fig. 13).

Five Pareto optimal points exist (Fig. 14), and our method enables us to visually compare them alongside one another (Fig. 15).

Solution (E) has zero shift, and while it may be more appealing due to its similarity to the input embedding, the other solutions are equally valid and worthy of consideration. For instance, solution (A) has the lowest bend objective value and may be regarded by some as easiest to navigate. We checked with the weighting method and found that of the five Pareto points, solution (D) is unsupported, and may not have been found without the augmented ϵ-constraint method.

5.4. Preliminary evaluation

We solved the examples described above on a 2.3 GHz Intel i7 machine (16 GB RAM). The examples were programmed in GAMS and Python and solved with the CPLEX 12.6 solver. Our tests show that relaxing the integer constraints on the coordinates and

simplifying the model to a biobjective one produces identical optimal results to Nöllenburg and Wolff's method [7] but in much less computational time. Using the AUGMECON2 [17] method also has the added benefit of generating multiple efficient solutions, including those unsupported, in a single run. In the augmented ϵ-constraint formulation, we chose unit-spaced grid intervals spanning the range of the shift objective. For the minimal example, we used only 2 grid points. In the dual-line and four-line networks, we used 3 and 5 grid points, respectively. The performance benefit of using fewer grid points is not readily apparent for these examples, since they are small. However, using the maximum possible number of grid points (equal to the range of the shift objective plus 1) will always produce the complete Pareto set for a given problem in our formulation.

In order to highlight our improvements, we compare the performance of three methods (Table 3). The first, N&W–WM₃, is Nöllenburg and Wolff's model—tri-objective with coordinate integer constraints—implemented using the weighting method. The second, O&S–WM₂, is our improved model—biobjective with integer coordinates relaxed—also implemented using the weighting method only for the sake of direct comparison to Nöllenburg and

Fig. 10. Dual-line network: 2 lines (1–2–4–5–6 and 2–3–5–7), 7 edges, 7 vertices, 2 faces; $\{l, m, n, r\} = \{2, 7, 7, 2\}$.
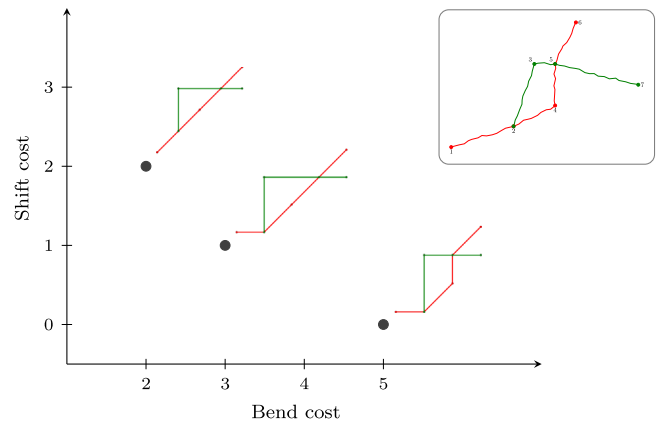

Fig. 12. Pareto frontier for dual-line network solutions.

Table 2
Sector parameters for each edge in the dual-line network.

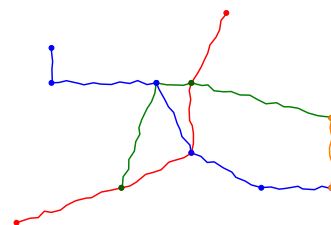| Edge uv	$sec_{-1}(u,v)$	$sec_0(u,v)$	$sec_{+1}(u,v)$
1–2	7	0	1
2–3	1	2	3
2–4	0	1	2
3–5	7	0	1
4–5	1	2	3
5–6	0	1	2
5–7	7	0	1

Fig. 13. Four-line network: 4 lines, 13 edges, 11 vertices, 4 faces; $\{l, m, n, r\} = \{4, 13, 11, 4\}$.
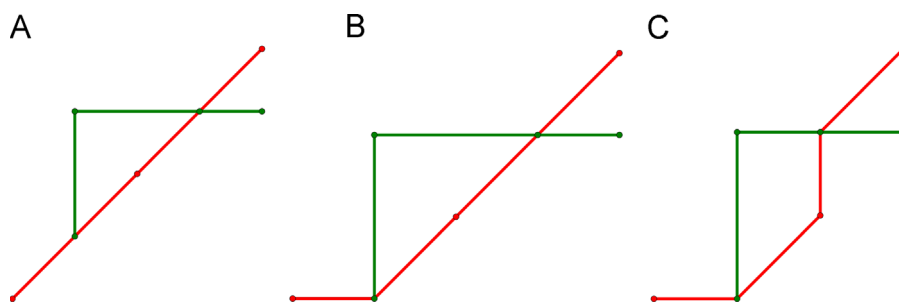

Fig. 11. Solutions to dual-line network. (a) $\{C_{bend}, C_{shift}\} = \{2, 2\}$, (b) $\{C_{bend}, C_{shift}\} = \{3, 1\}$, (c) $\{C_{bend}, C_{shift}\} = \{5, 0\}$.
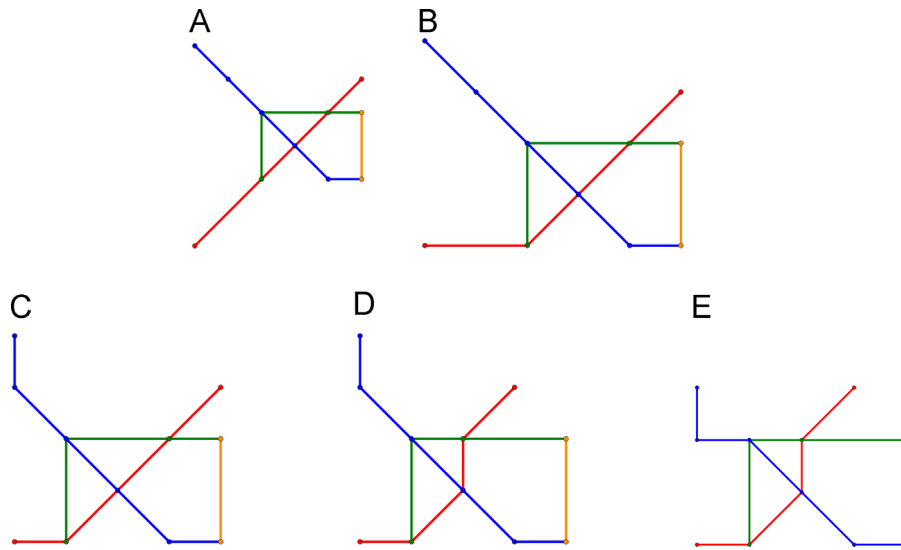
Fig. 14. Pareto optimal solutions to four-line network. (a) $\{C_{bend}, C_{shift}\} = \{3, 4\}$, (b) $\{C_{bend}, C_{shift}\} = \{4, 3\}$, (c) $\{C_{bend}, C_{shift}\} = \{5, 2\}$, (d) $\{C_{bend}, C_{shift}\} = \{7, 1\}$, (e) $\{C_{bend}, C_{shift}\} = \{9, 0\}$.
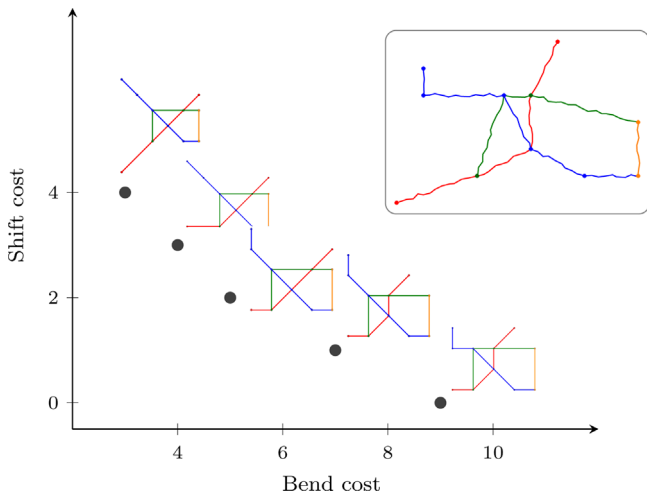

Fig. 15. Pareto frontier for four-line network.

Table 3
Description of methods compared for performance (N&W – Nöllenburg and Wolff; O&S – Oke and Siddiqui).

Method	Objective functions	Integer constraints	Optimization approach
N&W–WM$_3$	3	Yes	Weighting method
O&S–WM$_2$	2	No	Weighting method
O&S–AUGMECON2	2	No	Augmented ϵ-constraint method

Wolff's method. The third, O&S–AUGMECON2, is the augmented ϵ-constraint implementation of our improved model. We note again that the integer relaxation only affects the coordinate variables. The binary variables ($\alpha, \beta, \delta_1, \delta_2, \gamma$, shift) remain intact, as they are essential in formulating the disjunctive constraints. Relaxing the integrality constraints reduces the number of nodes visited in the branch-and-bound algorithm used by the CPLEX MIP solver.

Generally, both our implementations demonstrate better performance than Nöllenburg and Wolff's method, as shown in Table 4. Comparing the weighting approaches, ours (O&S–WM$_2$) solves to optimality in 15% less time for both the minimal and dual-line network examples. For the four-line network, there is a

30% decrease in execution time between N&W–WM$_3$ and O&S–WM$_2$. With O&S–AUGMECON2, we obtain multiple unique solutions in each single run. Considering the average execution time per solution, we observe a decrease in performance for the minimal example. This is not the case for the dual- and four-line networks, where we record a performance gain of 24% and 61%, respectively, in comparison to N&W–WM$_3$. The efficiency of AUGMECON2 increases substantially as the problem grows in size, and this is more telling for large networks, as our case studies will show.

6. Evaluation on existing networks: two case studies

Our goal for improving the performance of this model is to increase its accessibility and scalability, such that it can be effectively applied over a broad range of situations. We therefore apply our method to the Vienna Underground network in Vienna, Austria, and to a cancer pathway map [4].

6.1. Vienna Metro

The Vienna Metro network has 5 lines, 90 edges, 84 vertices and 8 faces. Fig. 16 shows the geographic layout and the official schematic version of the metro.

First, we used the weighting method based on our improvements (biobjective mixed binary) to find a single solution to the network. We compared our results to those obtained via Nöllenburg and Wolff's tri-objective mixed integer model, also implemented within our framework. One set of constraints that generates the most variables in this model is the edge spacing constraint group. Especially in metro networks, these constraints are mostly relevant when there are pendant (trailing) edges on the external face of a graph. They are also important for preventing interior faces from being too small in a final solution. In this example, we are able to solve the network using the weighting method without even calling upon the edge spacing constraints. In this case, Nöllenburg and Wolff's model execution time is 1.7 times as long as ours. When we enforce the edge spacing constraints, however, Nöllenburg and Wolff's model takes 14 times as long as ours to solve to optimality (Table 5).

For these tests, we used the weighting ratio $\lambda_{bend} : \lambda_{shift} = 7 : 3$. In every case, the weights add up to 1. Thus, in O&S–WM$_2$, $\lambda_{bend} = 0.7$ and $\lambda_{shift} = 0.3$. We point out that in the N&W–WM$_3$

Table 4
Average execution time per unique solution for three numerical examples, using three implementations; N&W – Nöllenburg and Wolff (3 objectives); O&S – Oke and Siddiqui (2 objectives).

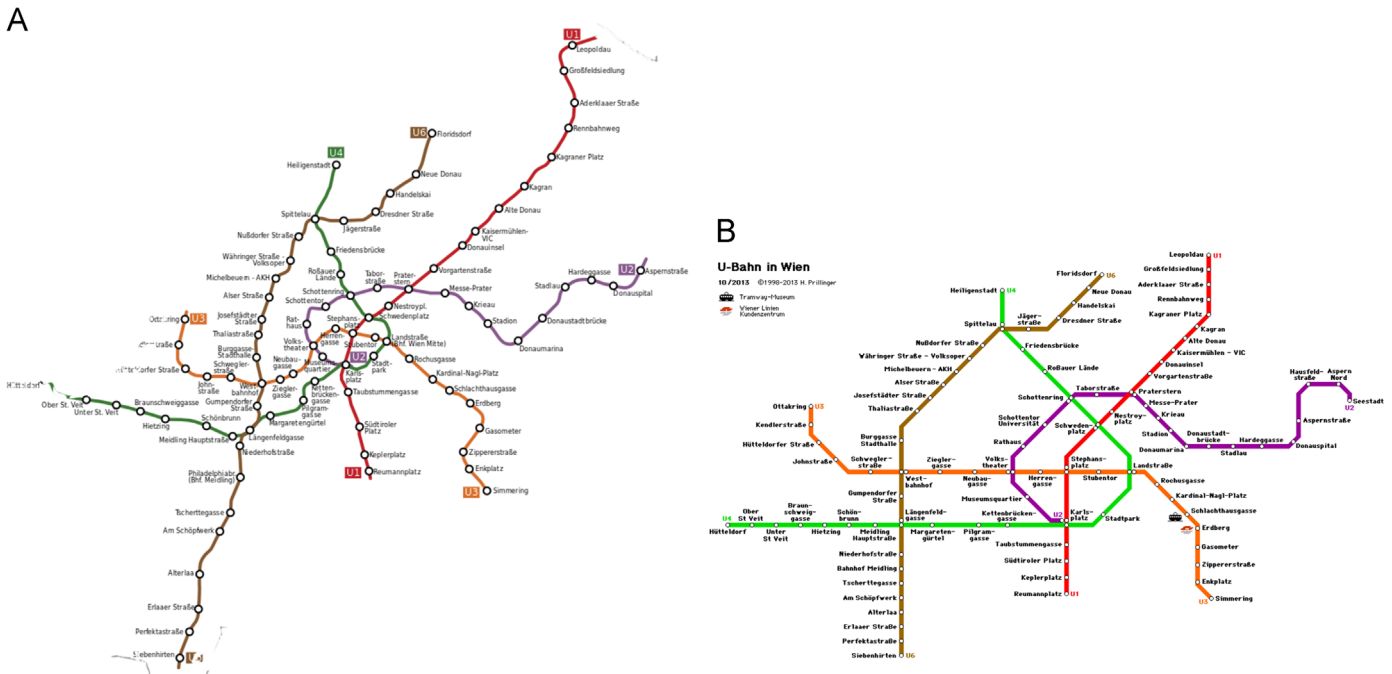| Method	Minimal example		Dual-line network		Four-line Network	
	Solutions generated per run	Execution time per solution (s)	Solutions generated per run	Execution time per solution (s)	Solutions generated per run	Execution time per solution (s)
N&W–$_{WM_3}$	1	0.778	1	1.069	1	1.767
O&S–$_{WM_2}$	1	0.661	1	0.913	1	1.188
O&S–$_{AUGMECON}$2	2	1.145	3	0.810	5	0.682

Fig. 16. Geographic and official layouts of the Vienna Metro network. (a) Vienna Metro (geographic layout), (b) Vienna Metro (official layout) [32].

Table 5
Execution times for the Vienna Metro problem, with $\lambda_{bend} : \lambda_{shift} = 7 : 3$; N&W – Nöllenburg and Wolff (3 objectives); O&S – Oke and Siddiqui (2 objectives).

Method	Vienna (no edge spacing constraints)		Vienna (with edge spacing constraints)	
	Objective cost $\{C_{bend}, C_{shift}\}$	Execution time per solution (s)	Objective cost $\{C_{bend}, C_{shift}\}$	Execution time per solution (m:s)
N&W–$_{WM_3}$	{16, 25}	10.7	{16, 25}	28:41
O&S–$_{WM_2}$	{13, 28}	6.2	{13, 28}	2:39

case, we have an additional objective function—total edge length—to consider. So, in order to maintain the same bend-shift weighting ratio for the sake of comparison, we use $\lambda_{length} = 0.2$, while $\lambda_{bend} = 0.56$ and $\lambda_{shift} = 0.24$. The solutions obtained via the implementations O&S–$_{WM_2}$ and N&W–$_{WM_3}$ are fairly identical, but that of O&S–$_{WM_2}$ (Fig. 17a) appears to be more spatially optimized, since we do not minimize the sum of edge lengths. The tri-objective Nöllenburg and Wolff implementation, N&W–$_{WM_3}$, generates solutions that lie on a three-dimensional Pareto frontier, which explains why its solution {16, 25} is not optimal in O&S–$_{WM_2}$ or O&S–$_{AUGMECON}$2.

With this particular weighting ratio, dispensing with the edge spacing constraints does not affect the solutions for both O&S–$_{WM_2}$ and N&W–$_{WM_3}$ (Fig. 17). This implies that edge spacing constraints may not always be required. However, if they are not enforced, certain solutions may feature crossing violations that alter the face structure of the graph. We observe this for a bend-shift weighting

ratio of 1 : 1 in O&S–$_{WM_2}$. There is a clear edge violation between pendant edges on the green and orange lines in the solution generated (Fig. 18). The edge spacing constraints must therefore be consistently enforced, especially for O&S–$_{AUGMECON}$2, as we cannot always predict where they will be redundant. For the Vienna network, however, the edge spacing constraints can be restricted to pairs of nonadjacent pendant edges on the external face.

We then solve the Vienna network via O&S–$_{AUGMECON}$2. The Pareto frontier obtained is shown in Fig. 20. To generate the complete Pareto set, we use integer-spaced grid points spanning the range of the shift objective. In this case, the number of grid points is given by

$$n_g = \max\{C_{shift}\} - \min\{C_{shift}\} + 1 = 38 - 1 + 1 = 38 \qquad (6.1)$$

There are 27 Pareto points in the complete set, five of which are displayed in Fig. 19. A single run took 70 min to execute (an average of 2:36 min per solution), using four threads in CPLEX. Eight of these
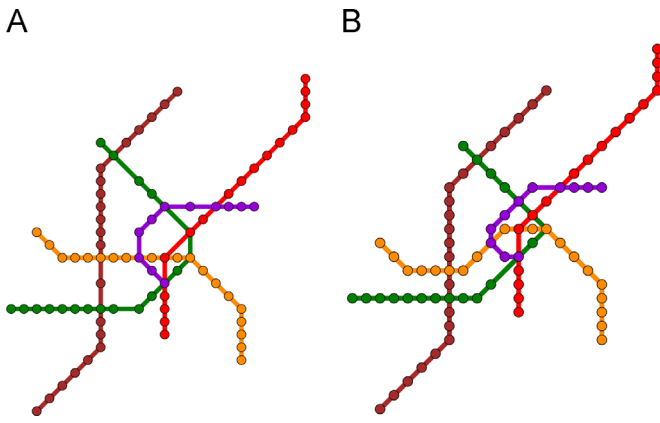
Fig. 17. Vienna solutions computed via the weighting approach; $\lambda_{bend} : \lambda_{shift} = 7 : 3$. N&W – Nöllenburg and Wolff (3 objectives); O&S – Oke and Siddiqui (2 objectives). (a) O&S–wm_2; $\{C_{bend}, C_{shift}\} = \{13, 28\}$, (b) N&W–$wm_3$; $\{C_{bend}, C_{shift}, C_{length}\} = \{16, 25, 186\}$.

solutions were found to be supported, that is, they could be found using the weighting method. We checked this first by manually varying the weights in steps of 0.1. We then used the Gather-Update-Solve-Scatter (GUSS) extension in GAMS [33] to solve 39 scenarios generated from sequential weighting combinations. The augmented ϵ-constraint method was therefore indispensable in obtaining the other 19 unsupported Pareto optimal solutions.

In this case of the Vienna network, the decision-maker might determine that any existing pool of efficient solutions intermediate between (D) and (F) are likely to be very similar, but if such solutions (high number of bends but similar to original layout) are of interest, then more grid points could then be chosen. Furthermore, the
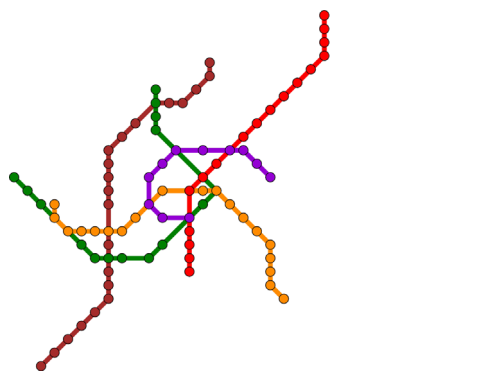

Fig. 18. Edge crossing violation (O&S–wm_2). $\lambda_{bend} : \lambda_{shift} = 1 : 1$ and $\{C_{bend}, C_{shift}\} = \{24, 13\}$. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)
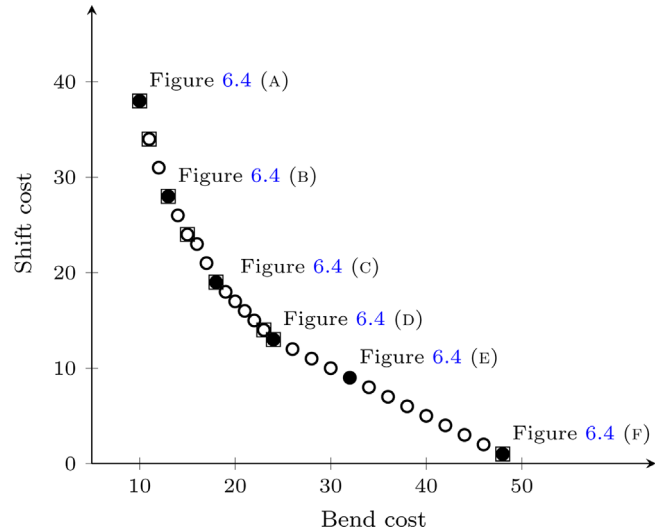

Fig. 20. Pareto frontier for Vienna Metro network (obtained via O&S–augmecon2). The points circumscribed by squares represent the supported solutions. The filled points (labeled) are shown in Fig. 19.
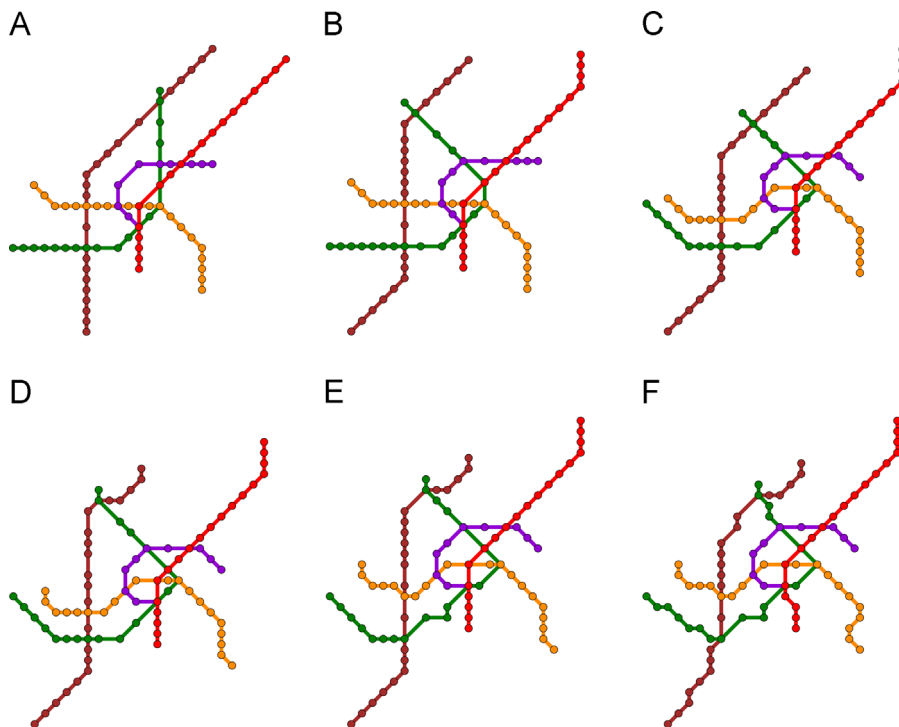

Fig. 19. Pareto optimal points for the Vienna underground network. (a) $\{C_{bend}, C_{shift}\} = \{10, 38\}$, (b) $\{C_{bend}, C_{shift}\} = \{13, 28\}$, (c) $\{C_{bend}, C_{shift}\} = \{18, 19\}$, (d) $\{C_{bend}, C_{shift}\} = \{24, 13\}$, (e) $\{C_{bend}, C_{shift}\} = \{32, 9\}$, (f) $\{C_{bend}, C_{shift}\} = \{48, 1\}$.

AUGMECON2 algorithm can be modified to have unequal grid intervals, so that more solutions can be generated within the area of interest. The initial layout of the Pareto set and the corresponding solutions enable the decision-maker to be more acquainted with the solution space, and further steps can be taken from there regarding the choice or discovery of a final satisfactory solution.

An important innovation Harry Beck introduced in his creation of the London Underground map [2] was the magnification of the central portion of a transit map, which tended to be denser with a higher concentration of tracks and lines. This central clustering is clearly visible in the geographically accurate map of the Vienna system (Fig. 16a). Beck not only evened out distances between stations on his map but also increased the relative distances of central stations, creating the iconic "vacuum flask" shape that is a highlight of the London map (Fig. 2). This type of distortion is perhaps more critical for
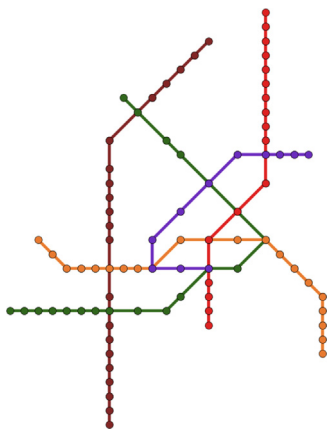
transit networks, where interchanges tend to cluster at one or more central locations. We can correct for this in the Vienna implementation by doubling the minimum edge length $\ell_{min}(u,v)$ for those edges found in the smaller interior faces. At the same time, we decrease the maximum edge length $\ell_{max}(u,v)$ constraint for the pendant (trailing) edges on the exterior face. These refinements ensure that the solution is spatially balanced with no visual overcrowding. They can be consistently applied in solving similar transit networks. The effects of this in our solution (Fig. 19) may not be too evident, but we could always increase the bounds of these constraints for more desirable results, thereby creating other classes of solutions (Pareto frontiers). Fig. 21 illustrates a solution we obtained by properly addressing this issue. The edges in the four smallest faces have greater edge lengths than those on the external face.

6.2. Cancer pathway

Our second example is the "subway map of cancer pathways" developed by Hahn and Weinberg, and designed by Bentley [4] (Fig. 22). While it is already a schematic map, we apply our method here to see if we can obtain comparable or better Pareto optimal points within our framework of requirements. Our implementation has 9 lines, 51 vertices, 56 edges and 6 faces.

Pathways, such as this one, may be challenging to model, as they tend to have more vertices with spanning edges than typical metro networks. The edge spacing constraints are essential here, as these also prevent edge crossings. In solving the Vienna metro network, we restricted the edge spacing constraints to the pendant edges on the external face. In this example, however, there are pendant edges on the internal faces, as well. Thus, we extend the application of the constraints to non-incident pairs of pendant and non-pendant edges in the same faces.

Using the AUGMECON2 implementation, we obtain the complete Pareto set in 165 min. A few selected solutions are shown in Fig. 23.
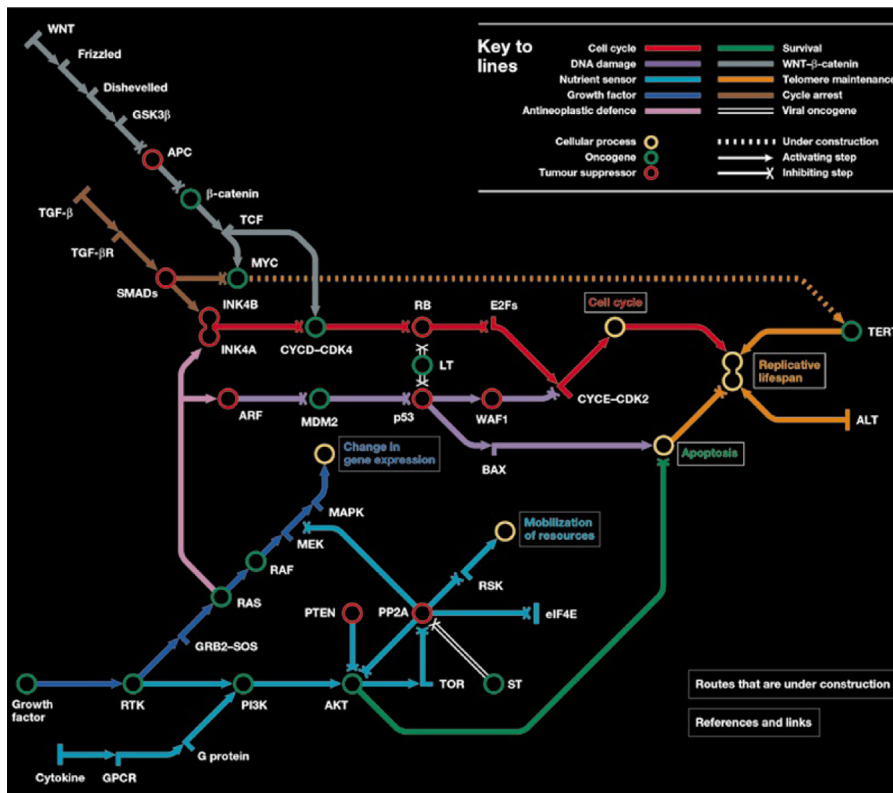

Fig. 21. Vienna Pareto optimal point with central magnification emphasized; $\{C_{bend}, C_{shift}\} = \{17, 37\}$.

Fig. 22. Subway map of cancer pathways [4].

Unsurprisingly, one of the Pareto points (solution (D) in Fig. 23) is nearly identical to the input embedding (Fig. 22), which indicates how well-designed the cancer map is. However, we now have a range of elegant solutions from which to choose. The complete Pareto frontier is shown in Fig. 24, and it consists of 15 points. We note that all the pendant edges have an upper bound of length 4, while all other edges have a maximum length of 8 imposed. Changing any of these bounds would produce a Pareto set with a different range of solutions, which could also be explored.

As we have seen from the examples of the Vienna Underground and the Hahn and Weinberg cancer pathway, metro maps do not always share similar properties with other pathway or network representations. Depending on the nature of the application, certain considerations (for instance, edge length or edge spacing restrictions) may be more relevant. To facilitate the versatile deployment of this model, a framework must be developed for standardizing constraints to accommodate all possible design specifications. We can fine-tune our implementation to eliminate or reduce programming inefficiencies and speed up execution times.

7. Conclusions and future work

We have shown in this paper that relaxing integrality constraints, simplifying the problem to two objective functions, and reformulating certain equations leads to obtaining more Pareto optimal solutions to the schematic map drawing problem than current methods. Moreover, the computational time to obtain solutions is greatly reduced. We built on Nöllenburg and Wolff's implementation and simplified their model to a biobjective mixed binary linear problem and obtained Pareto optimal solutions much quicker. We developed three hypothetical examples to demonstrate our results. We were also able to test our implementation on two real-world examples. A new development we have shown is the ability to compute a complete Pareto set for a schematic drawing problem, using Mavrotas and Florios' augmented ϵ-constraint method (AUGMECON2). This implementation allows for multiple supported and unsupported efficient solutions to be generated in a single run, enabling decision-makers to quickly evaluate potential candidates for schematic mapping solutions. We have also demonstrated that AUGMECON2 is preferred to the weighting method used by Nöllenburg and Wolff, as it guarantees finding all Pareto optimal solutions, including those that are nonextreme and unsupported.

Currently, we are working on a formal proof in support of our conjecture that integer optimal solutions can be found without enforcing coordinate integer constraints. Also, we would like to explore alternatives to the big-M formulation. Various methods have been developed to model disjunctivity, which features in the octilinearity and edge spacing constraints of this model. Care must be taken in determining the size of M, and in our implementation, it is equivalent to the maximum total edge length (Eq. (4.14)). Thus far, this has worked well, but we would like to implement and test the merits of the Reformulation Linearization Technique (RLT). Khurana et al. have demonstrated success in applying RLT to mixed
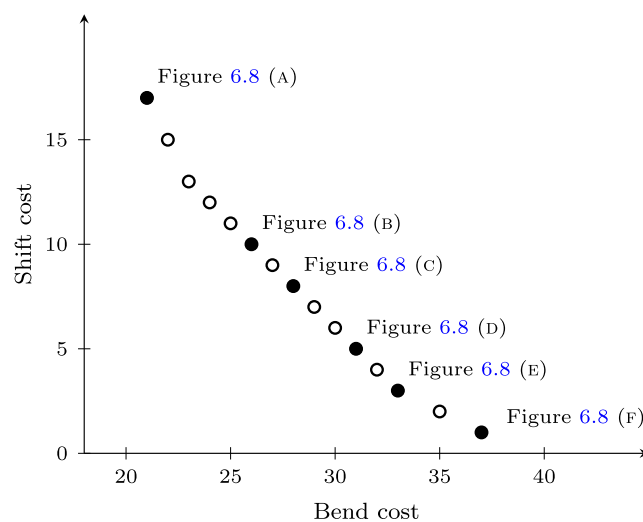

Fig. 24. Pareto frontier for Hahn and Weinberg cancer pathway (obtained via O&S–AUGMECON2). Solutions corresponding to the filled points (labeled) are shown in Fig. 23.
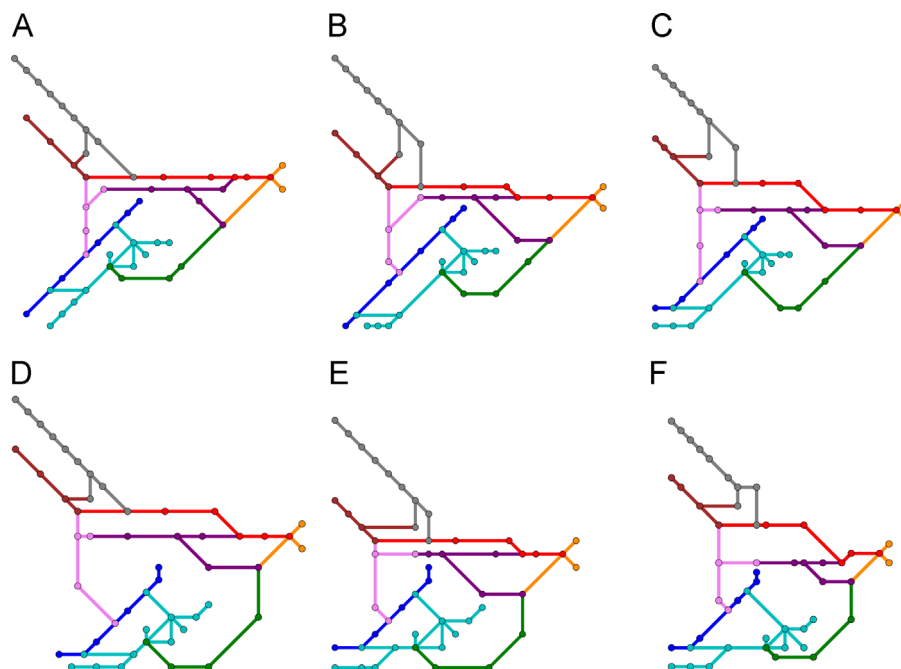

Fig. 23. Selected Pareto optimal points for the cancer pathway. (a) $\{C_{bend}, C_{shift}\} = \{21, 17\}$, (b) $\{C_{bend}, C_{shift}\} = \{26, 10\}$, (c) $\{C_{bend}, C_{shift}\} = \{28, 8\}$, (d) $\{C_{bend}, C_{shift}\} = \{31, 5\}$, (e) $\{C_{bend}, C_{shift}\} = \{33, 3\}$, (f) $\{C_{bend}, C_{shift}\} = \{37, 1\}$.

0-1 programs [34]. We expect that implementing this would significantly increase performance in our case. Also in consideration is an attempt to reformulate this problem as a mathematical program with equilibrium constraints (MPEC). Siddiqui and Gabriel

Appendix A

Here is a summary of the variables and equations used in this paper. For more background, refer to [7] and [17].

$x(u), y(u), z_1(u), z_2(u)$	coordinate variables
$\delta_k(u, v, w), \quad i \in \{1, 2\}$	binary variables for bend cost
$\gamma_c(e_1, e_2, f), \quad c \in \{N, NE, \dots, NW\}$	binary variables for edge spacing in each cardinal (axial)direction
$i \in \{-1, 0, 1\}$	positional index : original position or one sector back or forward
$\alpha_i(u, v)$	binary variable that switches position for sector and direction
$\sec_i(u, v) \quad i \in \{-1, 0, 1\}$	sector assignment for edge
$\dir(u, v)$	directional variable indicating sector chosen in solution
$\bend(u, v, w)$	bend cost for edges uv and vw
$\shift(u, v)$	shift cost per edge
M	large constant used in modeling disjunctivity
$\ell_{\max}(u, v), \ell_{\min}(u, v)$	upperandlowerboundsonlengthofedgeuv
$d_{\min}$	minimum spacing between vertices on nonadjacent edge pairs
$\beta(u, v_j), \quad j \in \{1, \dots, \deg(u)\}$	set of vertices of specified lower bound of degree
$\epsilon = 10^{-3}$	augmentation factor in AUGMECON2 ϵ-constraint method
$s_{\shift}$	slack variable for shift function in AUGMECON2
$r_{\shift}$	range of shift payoff table values in AUGMECON2
$e_{\shift}$	RHS of equality constraint computed from payoff table
$ub_{\shift}$	upper bound of shift function in payoff table
$i_{\shift} \in \{0, 1, \dots, g_{\shift}\}$	steps for grid points in AUGMECON2
$g_{\shift}$	number of desired grid intervals for AUGMECON2
$\lambda_{\bend}$	weighting factor for bend function
$\lambda_{\shift}$	weighting factor for shift function

[35] developed a method that employs Schur's decomposition using type-1 special-ordered set (SOS1) variables to provide global optima for MPECs. The challenge in this case, would be devising ways to remodel the schematic requirements as complementarity constraints. If we succeed in doing this, we will then compare the performance of Siddiqui and Gabriel's method with the numerical relaxation approach for MPECs formulated by Steffensen and Ulbrich [36]. Finally, Vincent et al. [37] have recently refined a branch-and-bound algorithm (initially developed by Mavrotas and Diakoulaki [38]) for solving our specific program class (mixed binary linear programs). We hope to adapt this to the map drawing problem as well.

We will continue efforts to improve the performance and expand the applications of automatic schematic mapping. A few new rules may have to be developed along the way depending on the network in question. For example, bicycle and bus networks have slightly different specifications compared to those of urban rail. Ultimately, we are interested in creating a dynamic user-friendly optimization tool that would be accessible for professional and public use in a wide variety of situations, able to produce solutions that address all considerations and facilitate the decision-making process.

Objective functions:

$$C_{\bend} = \sum_{L \in \mathcal{L}} \sum_{uv, vw \in L} \bend(u, v, w) \tag{A.1}$$

$$C_{\shift} = \sum_{uv \in E} \shift(u, v) \tag{A.2}$$

Augmented ϵ-constraint formulation (O&S–AUGMECON2):

$$\min : \quad C_{\bend} - \epsilon \frac{s_{\shift}}{r_{\shift}} \tag{A.3}$$

$$\text{s.t.} \quad C_{\shift} + s_{\shift} = e_{\shift} \tag{A.4}$$

$$e_{\shift} = ub_{\shift} - \frac{i_{\shift} r_{\shift}}{g_{\shift}} \tag{A.5}$$

General constraints $\qquad\qquad\qquad\qquad\qquad (*)$

Weighting method formulation (O&S–WM3)

$$\min : \quad \lambda_{\bend} C_{\bend} + \lambda_{\shift} C_{\shift} \tag{A.6}$$

$$\text{s.t.} \quad \text{General constraints} \qquad\qquad (*)$$

General constraints $(*)$:

$$\Delta\dir(u, v, w) = \dir(u, v) - \dir(v, w) \tag{A.7}$$

$$\bend(u, v, w) = \min\{|\Delta\dir(u, v, w)|, 8 - |\Delta\dir(u, v, w)|\} \tag{A.8}$$

$$-\bend(u, v, w) \leq \Delta\dir(u, v, w) - 8\delta_1(u, v, w) + 8\delta_2(u, v, w)$$
$$\bend(u, v, w) \geq \Delta\dir(u, v, w) - 8\delta_1(u, v, w) + 8\delta_2(u, v, w) \tag{A.9}$$

$$-M \cdot \shift(u, v) \leq \dir(u, v) - \sec_{\orig}(u, v) \leq M \cdot \shift(u, v) \tag{A.10}$$

$$M = \sum_{e \in E} e(u,v) \cdot \ell_{\max}(u,v) \tag{A.11}$$

$$\sec_0(u,v) = \begin{cases} 0 & 337.5° \leq \angle uv < 22.5° \\ 1 & 22.5° \leq \angle uv < 67.5° \\ 2 & 67.5° \leq \angle uv < 112.5° \\ 3 & 112.5° \leq \angle uv < 157.5° \\ 4 & 157.5° \leq \angle uv < 202.5° \\ 5 & 202.5° \leq \angle uv < 247.5° \\ 6 & 247.5° \leq \angle uv < 292.5° \\ 7 & 292.5° \leq \angle uv < 337.5° \end{cases} \tag{A.12}$$

$$\sec_{+1} = \sec_0 + 1 \bmod 8 \tag{A.13}$$

$$\sec_{-1} = \sec_0 - 1 \bmod 8 \tag{A.14}$$

$$\sec_i(u,v) = \sec_i(v,u) + 4 \bmod 8 \tag{A.15}$$

$$\sum_{i \in \{-1,0,1\}} \alpha_i(u,v) = 1 \tag{A.16}$$

Octilinearity and edge length constraints

$$\mathrm{dir}(u,v) = \sum_{i \in \{-1,0,1\}} \sec_i(u,v) \cdot \alpha_i(u,v) \tag{A.17}$$

$$\mathrm{dir}(v,u) = \sum_{i \in \{-1,0,1\}} \sec_i(v,u) \cdot \alpha_i(u,v) \tag{A.18}$$

$$\mathrm{dir}(u,v) = \mathrm{dir}(v,u) + 4 \bmod 8 \tag{A.19}$$

$$\sec_i(u,v) = 0 : \begin{cases} y(u) - y(v) \leq M(1 - \alpha_i(u,v)) \\ -y(u) + y(v) \leq M(1 - \alpha_i(u,v)) \\ -x(u) + x(v) \geq -M(1 - \alpha_i(u,v)) + \ell_{\min}(u,v) \end{cases} \tag{A.20}$$

$$\sec_i(u,v) = 1 : \begin{cases} z_2(u) - z_2(v) \leq M(1 - \alpha_i(u,v)) \\ -z_2(u) + z_2(v) \leq M(1 - \alpha_i(u,v)) \\ -z_1(u) + z_1(v) \geq -M(1 - \alpha_i(u,v)) + \ell_{\min}(u,v) \end{cases} \tag{A.21}$$

$$\sec_i(u,v) = 2 : \begin{cases} x(u) - x(v) \leq M(1 - \alpha_i(u,v)) \\ -x(u) + x(v) \leq M(1 - \alpha_i(u,v)) \\ -y(u) + y(v) \geq -M(1 - \alpha_i(u,v)) + \ell_{\min}(u,v) \end{cases} \tag{A.22}$$

$$\sec_i(u,v) = 3 : \begin{cases} z_1(u) - z_1(v) \leq M(1 - \alpha_i(u,v)) \\ -z_1(u) + z_1(v) \leq M(1 - \alpha_i(u,v)) \\ z_2(u) - z_2(v) \geq -M(1 - \alpha_i(u,v)) + \ell_{\min}(u,v) \end{cases} \tag{A.23}$$

$$\sec_i(u,v) = 4 : \begin{cases} y(u) - y(v) \leq M(1 - \alpha_i(u,v)) \\ -y(u) + y(v) \leq M(1 - \alpha_i(u,v)) \\ x(u) - x(v) \geq -M(1 - \alpha_i(u,v)) + \ell_{\min}(u,v) \end{cases} \tag{A.24}$$

$$\sec_i(u,v) = 5 : \begin{cases} z_2(u) - z_2(v) \leq M(1 - \alpha_i(u,v)) \\ -z_2(u) + z_2(v) \leq M(1 - \alpha_i(u,v)) \\ z_1(u) - z_1(v) \geq -M(1 - \alpha_i(u,v)) + \ell_{\min}(u,v) \end{cases} \tag{A.25}$$

$$\sec_i(u,v) = 6 : \begin{cases} x(u) - x(v) \leq M(1 - \alpha_i(u,v)) \\ -x(u) + x(v) \leq M(1 - \alpha_i(u,v)) \\ y(u) - y(v) \geq -M(1 - \alpha_i(u,v)) + \ell_{\min}(u,v) \end{cases} \tag{A.26}$$

$$\sec_i(u,v) = 7 : \begin{cases} z_1(u) - z_1(v) \leq M(1 - \alpha_i(u,v)) \\ -z_1(u) + z_1(v) \leq M(1 - \alpha_i(u,v)) \\ -z_2(u) + z_2(v) \geq -M(1 - \alpha_i(u,v)) + \ell_{\min}(u,v) \end{cases} \tag{A.27}$$

$$x(v), y(v) \leq M \quad \forall v \in V \tag{A.28}$$

$$\begin{aligned} x(u) - x(v) &\leq \ell_{\max}(u,v) \\ -x(u) + x(v) &\leq \ell_{\max}(u,v) \\ y(u) - y(v) &\leq \ell_{\max}(u,v) \\ -y(u) + y(v) &\leq \ell_{\max}(u,v) \end{aligned} \tag{A.29}$$

Circular order constraints

$$\mathrm{dir}(u,v_j) \leq \mathrm{dir}(u,v_{j+1}) - 1 + 8\beta(u,v_j), \quad j = \{1,2,\ldots,\deg(u)\} \tag{A.30}$$

$$\sum_{j=1}^{\deg(v)} \beta(u,v_j) = 1 \quad \forall u : \deg(u) \geq 2 \tag{A.31}$$

Edge spacing and crossing constraints

$$\sum_{c \in \{N,\ldots,NE\}} \gamma(e_1,e_2,f) \geq 1 \tag{A.32}$$

$$\begin{aligned} x(u_2) - x(u_1) &\leq M(1 - \gamma_E(e_1,e_2,f)) - d_{\min} \\ x(u_2) - x(v_1) &\leq M(1 - \gamma_E(e_1,e_2,f)) - d_{\min} \\ x(v_2) - x(u_1) &\leq M(1 - \gamma_E(e_1,e_2,f)) - d_{\min} \\ x(v_2) - x(v_1) &\leq M(1 - \gamma_E(e_1,e_2,f)) - d_{\min} \end{aligned} \tag{A.33}$$

$$\begin{aligned} z_1(u_2) - z_1(u_1) &\leq M(1 - \gamma_{NE}(e_1,e_2,f)) - d_{\min} \\ z_1(u_2) - z_1(v_1) &\leq M(1 - \gamma_{NE}(e_1,e_2,f)) - d_{\min} \\ z_1(v_2) - z_1(u_1) &\leq M(1 - \gamma_{NE}(e_1,e_2,f)) - d_{\min} \\ z_1(v_2) - z_1(v_1) &\leq M(1 - \gamma_{NE}(e_1,e_2,f)) - d_{\min} \end{aligned} \tag{A.34}$$

$$\begin{aligned} y(u_2) - y(u_1) &\leq M(1 - \gamma_N(e_1,e_2,f)) - d_{\min} \\ y(u_2) - y(v_1) &\leq M(1 - \gamma_N(e_1,e_2,f)) - d_{\min} \\ y(v_2) - y(u_1) &\leq M(1 - \gamma_N(e_1,e_2,f)) - d_{\min} \\ y(v_2) - y(v_1) &\leq M(1 - \gamma_N(e_1,e_2,f)) - d_{\min} \end{aligned} \tag{A.35}$$

$$\begin{aligned} -z_2(u_2) + z_2(u_1) &\leq M(1 - \gamma_{NW}(e_1,e_2,f)) - d_{\min} \\ -z_2(u_2) + z_2(v_1) &\leq M(1 - \gamma_{NW}(e_1,e_2,f)) - d_{\min} \\ -z_2(v_2) + z_2(u_1) &\leq M(1 - \gamma_{NW}(e_1,e_2,f)) - d_{\min} \\ -z_2(v_2) + z_2(v_1) &\leq M(1 - \gamma_{NW}(e_1,e_2,f)) - d_{\min} \end{aligned} \tag{A.36}$$

$$\begin{aligned} -x(u_2) + x(u_1) &\leq M(1 - \gamma_W(e_1,e_2,f)) - d_{\min} \\ -x(u_2) + x(v_1) &\leq M(1 - \gamma_W(e_1,e_2,f)) - d_{\min} \\ -x(v_2) + x(u_1) &\leq M(1 - \gamma_W(e_1,e_2,f)) - d_{\min} \\ -x(v_2) + x(v_1) &\leq M(1 - \gamma_W(e_1,e_2,f)) - d_{\min} \end{aligned} \tag{A.37}$$

$$\begin{aligned} -z_1(u_2) + z_1(u_1) &\leq M(1 - \gamma_{SW}(e_1,e_2,f)) - d_{\min} \\ -z_1(u_2) + z_1(v_1) &\leq M(1 - \gamma_{SW}(e_1,e_2,f)) - d_{\min} \\ -z_1(v_2) + z_1(u_1) &\leq M(1 - \gamma_{SW}(e_1,e_2,f)) - d_{\min} \\ -z_1(v_2) + z_1(v_1) &\leq M(1 - \gamma_{SW}(e_1,e_2,f)) - d_{\min} \end{aligned} \tag{A.38}$$

$$\begin{aligned} -y(u_2) + y(u_1) &\leq M(1 - \gamma_S(e_1,e_2,f)) - d_{\min} \\ -y(u_2) + y(v_1) &\leq M(1 - \gamma_S(e_1,e_2,f)) - d_{\min} \\ -y(v_2) + y(u_1) &\leq M(1 - \gamma_S(e_1,e_2,f)) - d_{\min} \\ -y(v_2) + y(v_1) &\leq M(1 - \gamma_S(e_1,e_2,f)) - d_{\min} \end{aligned} \tag{A.39}$$

$$\begin{aligned} z_2(u_2) - z_2(u_1) &\leq M(1 - \gamma_{SE}(e_1,e_2,f)) - d_{\min} \\ z_2(u_2) - z_2(v_1) &\leq M(1 - \gamma_{SE}(e_1,e_2,f)) - d_{\min} \\ z_2(v_2) - z_2(u_1) &\leq M(1 - \gamma_{SE}(e_1,e_2,f)) - d_{\min} \\ z_2(v_2) - z_2(v_1) &\leq M(1 - \gamma_{SE}(e_1,e_2,f)) - d_{\min} \end{aligned} \tag{A.40}$$

References

[1] Talbert RJA. Rome's world: the Peutinger map reconsidered. New York, NY, USA: Cambridge University Press; 2010.
[2] Garland K. Mr. Beck's underground map. Harrow, UK: Capital Transport Publishing; 1994.
[3] Ovenden M. Transit maps of the world. New York, NY: Penguin Books; 2007.

[4] Hahn WC, Weinberg RA. A subway map of cancer pathways, ⟨http://www.nature. com/nrc/poster/subpathways/index.html⟩; 2002 [Subway map designed by Claudia Bentley, Web design by Nick Allin].

[5] Stott J, Rodgers P, Burkhard R, Meier M, Smis M. Automatic layout of project plans using a metro map metaphor. In: Proceedings of the ninth international conference on information visualisation; 2005. p. 203–6.

[6] Avelar S, Hurni L. On the design of schematic transport maps. Cartographica 2006;41:217–28.

[7] Nöllenburg M, Wolff A. Drawing and labeling high-quality metro maps by mixed-integer programming. IEEE Trans Vis Comput Graph 2011;17.

[8] Purchase HC, Cohen RF, James M. Validating graph drawing aesthetics. In: Lecture notes in computer science, vol. 1027, 1996, p. 435–46.

[9] Tamassia R. On embedding a graph in the grid with the minimum number of bends. SIAM J Comput 1987;16:421–44.

[10] Avelar S, Müller, M. Generating topologically correct schematic maps. In: Proceedings of ninth international symposium on spatial data handling; 2000.

[11] Cabello S, de Berg M, van Kreveld M. Schematization of networks. Comput Geom: Theory Algorithms 2005;30:223–38.

[12] Hong S-H, Merrick D, do Nascimento HAD. The metro map layout problem. In: Conferences in research and practice in information technology, vol. 35; 2004. p. 91–100.

[13] Ribeiro JT, Rijo R, Leal A. Fast automatic schematics for public transport spider maps. Proc Technol 2012;5:659–69.

[14] Stott J, Rodgers P, Martinez-Ovando JC, Walker SG. Automatic metro map layout using multicriteria optimization. IEEE Trans Vis Comput Graph 2011;17.

[15] Wolff A. Drawing subway maps: a survey. Inf—Forsch Entwick 2007;22:23–44.

[16] Siddiqui S, Azarm S, Gabriel SA. On improving normal boundary intersection method for generation of pareto frontier. Struct Multidiscip Optim 2012;46:839–52.

[17] Mavrotas G, Florios K. An improved version of the augmented epsilon-constraint method (AUGMECON2) for finding the exact pareto set in multi-objective integer programming problems. Appl Math Comput 2013;219:9652–69.

[18] Scaparra MP, Church RL. A bilevel mixed-integer program for critical infra-structure protection planning. Comput Oper Res 2008;35:1905–23 Part Special Issue: {OR} Applications in the Military and in Counter-Terrorism.

[19] Eusébio A, Figuiera JR. Finding non-dominated solutions in bi-objective integer network flow problems. Comput Oper Res 2009;36:2554–64.

[20] Solanki R. Generating the noninferior set in mixed integer biobjective linear programs: an application to a location problem. Comput Oper Res 1991;18:1–15.

[21] Cano RG, Kunigami G, de Souza CC, de Rezende PJ. A hybrid GRASP heuristic to construct effective drawings of proportional symbol maps. Comput Oper Res 2013;40:1435–47.

[22] Martt R, Estruch V. Incremental bipartite drawing problem. Comput Oper Res 2001;28:1287–98.

[23] Dwyer T, Hurst N, Merrick D. A fast and simple heuristic for metro map path simplification. In: Advances in visual computing, Lecture notes in computer science, vol. 5359, Berlin, Heidelberg: Springer; 2008. p. 22–30. URL ⟨http://dx. doi.org/10.1007/978-3-540-89646-3_3⟩.

[24] Bekos MA, Kaufmann M, Potika K, Symvonis A. Line crossing minimization on metro maps. In: Hong S-H, Nishizeki T, Quan W, editors. Graph Drawing, Lecture notes in computer science, vol. 4875, Berlin, Heidelberg: Springer; 2008. p. 231–42. URL ⟨http://dx.doi.org/10.1007/978-3-540-77537-9_24⟩.

[25] Fink M, Haverkort H, Nöllenburg M, Roberts M, Schuhmann J, Wolff A. Drawing metro maps using bézier curves. In: Graph drawing, Lecture notes in computer science, vol. 7704, Berlin, Heidelberg: Springer; 2013. p. 463–74. URL ⟨http://dx.doi.org/10.1007/978-3-642-36763-2_41⟩.

[26] Hurter C, Serrurier M, Alonso R, Tabart G, Vinot J-L. An automatic generation of schematic maps to display flight routes for air traffic controllers: structure and color optimization. In: Proceedings of the international conference on advanced visual interfaces, AVI '10. New York, NY, USA: ACM Press; 2010. p. 233–40.

[27] Guo Z. Mind the map! The impact of transit maps on travel decisions in public transit. Transp Res Part A: Policy Pract 2011;45:625–39.

[28] Sherwin H, Melia S. Cycle mapping in the UK and the London cycle map. Project Report, University of the West of England, Bristol; 2012 [Ideas in Transit].

[29] U.S. Dept of Health and Human Services. Physical activity and health: a report of the Surgeon General, Atlanta, GA, DIANE Publishing; 1996. URL ⟨http:// books.google.com/books?id=WZZPc1FmL7QC⟩.

[30] Gabriel SA, Conejo AJ, Ruiz C, Siddiqui S. Solving discretely constrained, mixed linear complementarity problems with applications in energy. Comput Oper Res 2013;40:1339–50.

[31] Mavrotas G. Effective implementation of the ε-constraint method in multi-objective mathematical programming problems. Appl Math Comput 2009;213:455–65.

[32] Prillinger H. U-bahn in Wien, ⟨http://homepage.univie.ac.at/horst.prillinger/ metro/m/largemap.html⟩; 2013.

[33] Kallrath J, editor. Algebraic modeling systems, Applied optimization, vol. 104, Berlin, Heidelberg: Springer-Verlag; 2012, p. 35–56.

[34] Khurana A, Sundaramoorthy A, Karimi IA. Improving mixed integer linear programming formulations. In: Proceedings of AIChE; 2005.

[35] Siddiqui S, Gabriel SA. An SOS1-based approach for solving MPECs with a natural gas market application. Netw Spat Econ; 2013.

[36] Steffensen S, Ulbrich M. A new relaxation scheme for mathematical programs with equilibrium constraints. SIAM J Optim 2010;20:2504–39.

[37] Vincent T, Seipp F, Ruzika S, Przybylski A, Gandibleux X. Multiple objective branch and bound for mixed 0-1 linear programming: corrections and improvements for the biobjective case. Comput Oper Res 2013;40:498–509.

[38] Mavrotas S, Diakoulaki D. A branch and bound algorithm for mixed zero-one multiple objective linear programming. Eur J Oper Res 1998;107:530–41.