

REA-FM: automated generation of natural-looking flow maps through river extraction algorithm

Zhiwei Wei, Su Ding, Wenjia Xu, Jifei Fang, Chunbo Liu & Yang Wang

To cite this article: Zhiwei Wei, Su Ding, Wenjia Xu, Jifei Fang, Chunbo Liu & Yang Wang (04 Mar 2024): REA-FM: automated generation of natural-looking flow maps through river extraction algorithm, *Cartography and Geographic Information Science*, DOI: [10.1080/15230406.2024.2311259](https://doi.org/10.1080/15230406.2024.2311259)

To link to this article: <https://doi.org/10.1080/15230406.2024.2311259>

 View supplementary material 

 Published online: 04 Mar 2024.

 Submit your article to this journal 

 View related articles 

 View Crossmark data 
CrossMark



REA-FM: automated generation of natural-looking flow maps through river extraction algorithm

Zhiwei Wei ^{a,b,c}, Su Ding ^d, Wenjia Xu ^e, Jifei Fang ^{b,c}, Chunbo Liu ^{b,c} and Yang Wang ^{b,c}

^aGuangdong Laboratory of Artificial Intelligence and Digital Economy(SZ), Shenzhen, China; ^bKey Laboratory of Network Information System Technology (NIST), Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing, China; ^cThe Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing, China; ^dCollege of Environmental and Resource Science, Zhejiang A & F University, China; ^eSchool of Information and Communication Engineering, Beijing University of Posts and Telecommunications, China

ABSTRACT

A flow map is a type of thematic visualization that depicts the movement of objects across a geographical space using a tree layout resembling a natural river system. In this paper, we introduce an innovative and automated approach called REA-FM, which leverages the power of the maze-solving algorithm to extract rivers from digital elevation models (DEMs). This enables the creation of flow maps that originate from a single source and extend to multiple destinations. Initially, REA-FM represents the mapping space of a flow map using a DEM. Subsequently, a maze-solving algorithm is adapted to extract flow paths from the destinations to the origin within the DEM data, with constraints on search directions, direction weights, and search ranges based on quality criteria specific to flow maps. To obtain comprehensive flow maps, the maze-solving algorithm is employed iteratively, considering the importance of each flow path, as determined by their respective lengths. These obtained paths are finally rendered smoothly with varying widths using Bézier curves, thereby enhancing the visual aesthetics of the flow map. A comparative evaluation with existing approaches demonstrates that REA-FM can generate natural-looking flow maps with reduced total length and improved node distribution, eliminating node overlaps and edge crossings. Furthermore, the effectiveness of REA-FM is validated through three extension experiments involving heterogeneous mapping spaces and areas with obstacles. Parameter analysis confirms that REA-FM offers intuitive control over the layout of flow maps. Project website: <https://github.com/TrentonWei/FlowMap>

ARTICLE HISTORY

Received 27 July 2023
Accepted 23 January 2024

KEYWORDS

Flow map; automatic cartography; hydrology model; maze solving algorithm; digital elevation model

1. Introduction

Technological advancements in recent decades have led to a more open and networked global society, allowing frequent movements such as migration or trade flows. Consequently, mapping these flows across various regions has emerged as a critical and central task in graph visualization (Jenny et al., 2018). This study focuses explicitly on a widely prevalent category of flows known as one-to-many flows, which depict movements from a single source to multiple destinations.

To visualize one-to-many flows, the tree-like layout is a pivotal form due to its ability to reduce visual clutter and provide clear spatial distribution insights into moving phenomena (Viau et al., 2023). Previous approaches have attempted to bundle edges into tree-like layouts through methods like stub bundling, spiral tree, or force-directed techniques (Buchin et al., 2011; Debiasi et al., 2014; Nocaj & Brandes, 2013; Phan et al., 2005). However, these approaches often yield artificial-looking

results. Sun (2019) addressed this issue by simulating the dendritic drainage patterns of natural river systems to construct an approximate Steiner tree for generating one-to-many flow maps. Nevertheless, the proposed approach may result in a longer total length.

Despite these shortcomings, Sun's (2019) approach showed promise in generating natural-looking flow maps by mimicking natural river systems. As such, we have adopted his idea as a basis and aimed to produce natural-looking one-to-many flow maps while addressing its drawbacks. Unlike Sun's (2019) approach, we focused more on the river extraction process, which has been a central research task in hydrological applications. By taking the digital elevation model (DEM) as a basic data model, many excellent algorithms have been proposed in the past decades to extract a natural river system automatically. In this regard, if we model the mapping space of a one-to-many flow map as DEM data, then river extraction algorithms in hydrological applications can be adapted to automate the production

of the one-to-many flow maps such as the maze-solving algorithm (Zhang et al., 2017). Moreover, given that the DEM is a potent data model to represent geographic spaces, this approach can be used to represent various mapping spaces.

Motivated by the aforementioned thoughts, we propose a new approach called REA-FM to produce one-to-many flow maps automatically by adapting a river extraction algorithm. To achieve this, we model the mapping space of the flow map using the DEM first. Recognizing that the mapping space is typically assumed to be uniform, we represent it as a flat surface by defining the cell shape, resolution, and range of the DEM. Next, we adapt the maze-solving algorithm, a river extraction algorithm to calculate flow paths from the destination to the origin iteratively. The search directions, direction weights, and search ranges of the maze-solving algorithm are constrained according to the quality criteria of the one-to-many flow maps. These allow users to transform the quality criteria as properties of a DEM or parameters of the maze-solving algorithm, making the flow map production process more intuitive. Specifically, experiments with various parameters on three datasets with homogeneous, heterogeneous, or obstacle-ridden mapping spaces demonstrate that our approach produces flow maps more intuitively, with smaller total lengths, while achieving similar performance as Sun (2019).

2. Related works

2.1. The one-to-many flow-maps

The concept of one-to-many flow maps has a historical foundation dating back to Henry Drury Harness in 1837, as noted by Robinson (1982). Charles Joseph Minard further advanced this concept to represent economic topics such as the import and export of wine, cotton, and coal. However, these primitive flow maps were originally hand-drawn, and Tobler (1987) introduced the first automated system to generate such maps. This system was based on straight lines of varying widths connecting destinations to the origin, which caused significant visual clutters, subsequently becoming a notable concern in later approaches.

To mitigate the problem of visual congestion, Tobler (1987) implemented a filtering strategy that was later incorporated by van den Elzen and van Wijk (2014). Another technique used to reduce visual clutter in flow maps is edge bundling. Phan et al. (2005) introduced an algorithm for bundling edges based on hierarchical clustering. However, their approach may not always smooth all paths. Debiasi et al. (2014) proposed an

alternative approach that employed a supervised force-directed algorithm for edge bundling, but it may require manual intervention. Steiner tree has also been a widely utilized technique in flow maps, as they connect points with the minimum total length by incorporating additional points. Buchin et al. (2011) introduced a spiral tree by constraining the angle of a Steiner tree to produce smooth and crossing-free flow maps. Similarly, Nocaj and Brandes (2013) employed a stub bundling strategy based on a spiral tree. Although these approaches are useful, they may sometimes yield artificial-looking layouts. In contrast, Sun (2019) generated flow maps by simulating the motion of flows based on the formation of the dendritic drainage pattern of natural river systems, resulting in more natural layouts. For particular purposes, such as 2D and 3D flow maps in immersive environments, have been investigated, as in the work of Yang et al. (2019), while Viau et al. (2023) proposed map distortions to increase readability and overcome visual clutter when dealing with large volumes of data.

2.2. River extraction algorithms over flat surfaces in a digital elevation model

River extraction from DEM data is essential for various hydrological applications (Yu et al., 2014). However, DEM data often contain flat surfaces resulting from flat terrains, interpolation methods, or depression-filling operations (Liu et al., 2017). Consequently, extracting river information from DEM data in areas with flat surfaces is also a critical requirement in hydrological applications. The extraction process typically involves three steps: flow path calculation, flow accumulation, and flow tracking, with flow path calculation being particularly important as it facilitates the other two processes.

Numerous algorithms have been developed for calculating flow paths in areas with flat surfaces, which can be categorized into two main types: effectiveness-first and efficiency-first approaches. Effectiveness-first algorithms prioritize accuracy in flow path calculation. For example, Jenson and Domingue (1988) utilized an eight-neighbors technique to assign flow directions, but their approach could produce parallel flows. To address this issue, Garbrecht and Martz (1997) introduced a gradient from higher to lower terrain to assign flow directions. Because this approach can generate flow paths that match the actual terrain, subsequent research has improved upon it (Liu et al., 2017; Soille et al., 2003). Alternatively, flow paths can be obtained iteratively by their importance. Tribe (1992) extracted flow directions in flat areas by assuming that the main flow path was

a straight line from inlet to outlet. Other flow paths were then connected to these initial paths one by one. However, since not all paths are perfectly straight, Zhang et al. (2017) improved Tribe's (1992) approach by introducing a maze-solving algorithm to calculate flow directions and obtained realistic flow paths. Efficiency-first algorithms aim to reduce the computational cost or speed up the calculation. For example, Zhu et al. (2006) applied a neighbor-grouping scan loop strategy to reduce calculation, while Wang and Liu (2006) presented a priority-flood algorithm that processed depressions from the edge grid cells to the interior cells to reduce computation complexity. Barnes et al. (2014) applied different increments to flat cells in masked DEMs to avoid iterative calculation. Yu et al. (2014) used a “first-in, first-out” queue to process flat areas and a priority queue to process other areas, which can effectively reduce the computation complexity.

3. Definitions and quality criteria

3.1. Definitions

To achieve a better illustration of the proposed approach, we present the definitions and illustrate them in Figure 1.

Origin node: A node representing the origin, e.g. node n_1 .

Destination node: $n_4n_6n_8n_9n_{10}$

Flow-in node: An intersection node where two edges intersect, e.g. nodes n_2 , n_3 , n_5 , and n_7 .

Flow-in edge and flow-out edge: These edges are associated with a flow-in node. Since the flow map depicts the movement of objects originating from a single source and extending to multiple destinations (or vice versa), the flow-in edges and flow-out edges are determined based on the flow direction. For instance, when considering flow-in node n_3 , the edges n_3n_5 and n_3n_7 are flow-in edges, and edge n_2n_3 is the flow-out edge.

Hanging edge: An edge that connects a destination node with a flow-in node, e.g. edges n_2n_4 , n_5n_6 , n_5n_8 , n_7n_9 , and n_7n_{10} .

Non-hanging edge: An edge that is not hanging, e.g. edges n_1n_2 , n_2n_3 , n_3n_5 , and n_3n_7 .

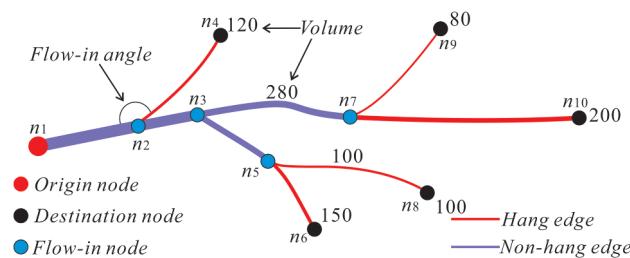


Figure 1. Definitions on the one-to-many flow maps.

Path: The route from one node to another, e.g. $n_1n_2n_3n_7n_{10}$ is a path from destination node n_1 to origin node n_{10} .

Volume: The amount of objects flowing along an edge, represented by values such as 100 for edge n_5n_8 and 280 for edge n_3n_7 .

Flow-in angle: The angle formed between a flow-in edge and its corresponding flow-out edge, e.g. angle $n_1n_2n_4$.

3.2. Quality criteria

The design requirements of a flow map are typically translated into quality criteria to guide its production, and different quality criteria have been proposed for various purposes (Wei et al., 2023). For instance, Debiasi et al. (2014) put forward five criteria, including crossing-free and overlap avoidance between nodes and edges, while Jenny et al. (2017) identified eight criteria for graph length and symmetry based on a user study. In this paper, we present a summary of the quality criteria as follows:

(1) Geometry

Curved edge preference (GC_1): Curved edges are preferred over straight edges, and an edge is more likely to be rendered as a curve wherever possible (Dong et al., 2018; Jenny et al., 2018).

Curve difference necessity (GC_2): The users would prefer curved edges for aesthetics, but prefer straight edges for clarity in graph visualization (Xu et al., 2012). Additionally, the main branches of a flow map need to be emphasized, which may require slightly curved ones for edges with high volumes, such as the non-hang edges (Debiasi et al., 2014).

Total length minimization (GC_3): A larger total length of a flow map means a larger visual burden, and needs to be minimized (Buchin et al., 2011; Sun, 2019).

(2) Relation

Acute flow-in angle avoidance (RC_1): Acute flow-in angles may have negative impacts on response time, and should be avoided (Huang et al., 2014).

Edge crossing avoidance (RC_2): Edge crossings may lead to confusion, and should be avoided (Buchin et al., 2011; Debiasi et al., 2014; Sun, 2019).

Overlap avoidance between nodes and edges (RC_3): Overlaps between destination nodes and edges can result in visual clutter, and should be avoided (Buchin et al., 2011; Debiasi et al., 2014; Sun, 2019).

Crossing avoidance between edges and important map objects (RC_4): Edges must avoid important map objects to aid in recognizability (Buchin et al., 2011; Sun, 2019).

Suitable distance between nodes and edges (RC_5): Destination nodes should be far away enough from the edges for clarity (Sun, 2019).

(3) Distribution

Tree layout (DC_1): The destination nodes should be organized hierarchically into levels and represented as a tree layout, where the origin is the root node and the destinations are leaf nodes; edge widths of the flow map are then drawn atop thick ones to the thin ones according to the hierarchy (Buchin et al., 2011; Sun, 2019).

4. Methodology

4.1. Framework

We propose REA-FM, which aims to produce natural-looking one-to-many flow maps via simulation of a river extraction process in DEM data. REA-FM consists of three key steps, which are outlined in Figure 2.

Step 1. Model the mapping space as DEM data. The mapping space of a flow map is represented as DEM data by determining its fundamental properties based on user demands. Destination and origin nodes are represented by corresponding cells where destination cells hold outflows that converge at the origin cell (Section 4.2)

Step 2. Flow path calculation. This step involves iteratively computing the flow paths from each destination cell to the origin cell using a river extraction algorithm. It consists of three sub-steps:

- **Step 2.1. Calculation of potential flow paths.** In our proposed approach, the generation of flow paths occurs iteratively. This means that a destination cell may flow into a preexisting flow path leading to the origin cell. In cases where destination cells lack a direct path to the origin cell, we employ a maze-solving algorithm tailored for river extraction. This algorithm is adapted to calculate all potential paths from each destination cell to the origin cell by determining the shortest path from the destination cells to each cell within a previously generated path. The maze-solving algorithm considers GC_3 , RC_2 , RC_3 , RC_4 , and RC_5 (Section 4.3.1).
- **Step 2.2. Identification of valid paths.** For destination cells that lack a path to the origin cell, the valid path from the destination cell to the origin cell is identified. A valid path refers to the shortest path among all potential paths, in which GC_3 and RC_1 are considered in path length calculation (Section 4.3.2).
- **Step 2.3. Select a valid path.** The valid path with the highest importance among all valid paths from the destination cells to the origin cell is selected as the calculated flow path. This path is added to the previously calculated paths, in which GC_3 is considered in path importance calculation (Section 4.3.3). If any remaining destination cells lack a path to the origin cell, Steps 2.1 to 2.3 are repeated.

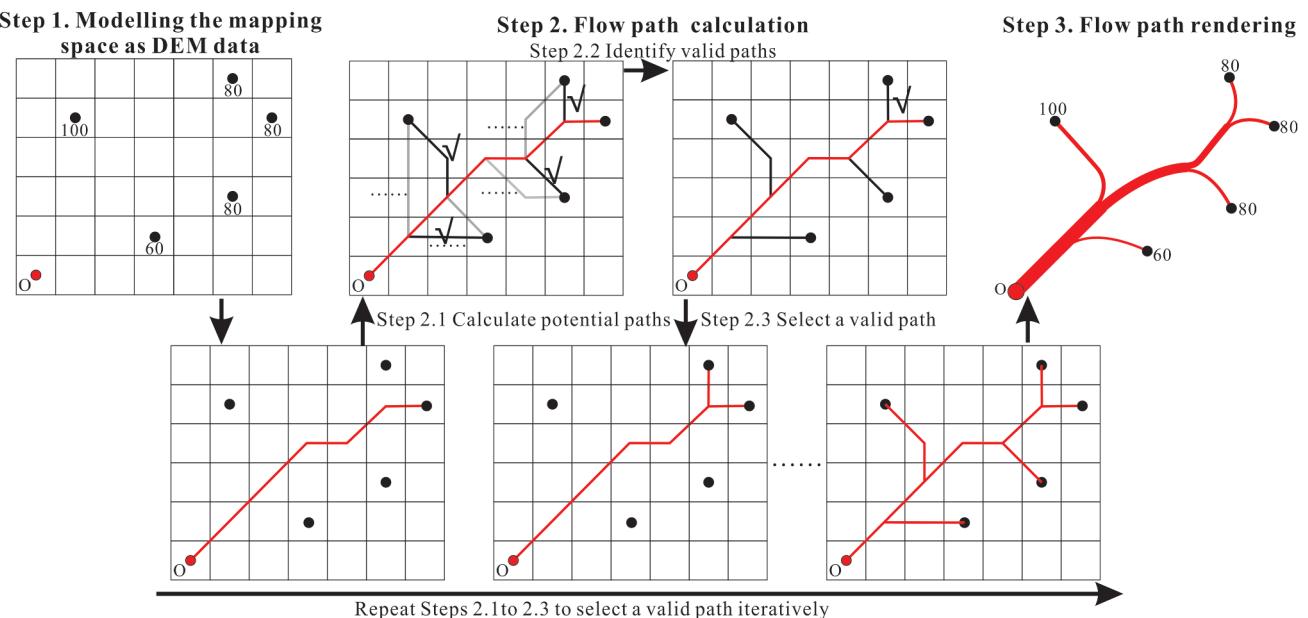


Figure 2. Framework.

Step 3. Flow path rendering. In this step, the flow volume per cell is calculated based on the obtained flow paths. The edges are then rendered smoothly using Bézier curves, with their widths varying depending on the flow volumes. GC_1 and GC_2 are considered in this step (Section 4.4).

4.2. Step 1. Model the mapping space as DEM data

DEM is a continuous surface representation of terrain elevation, where cell shape, resolution, and range are fundamental properties and need to be defined. Given origin node (n_{ori}) and N_{des} destination nodes (n_{des}^i) as $\{n_{ori}\} \cup \{n_{des}^i\}_{i=1}^{N_{des}}$, the base map with N_{reg} regions as $\{reg_i\}_{i=1}^{N_{reg}}$. The DEM used for mapping space modeling is defined as follows.

- (1) **Cell shape.** Square cells are adopted due to their frequent use and lower computational cost.
- (2) **Cell resolution.** To ensure that the two closest nodes in the DEM data do not fall into the same cell while avoiding higher computational costs with increasing cell resolutions. We determine the cell resolution R by considering the shortest distance between node pairs in $\{n_{ori}\} \cup \{n_{des}^i\}_{i=1}^{N_{des}}$, which is defined by Equation (1) (Hengl, 2006).

$$R = \frac{\overline{D}_{(5\%th)}}{4} \quad (1)$$

where $\overline{D}_{(5\%th)}$ is the average distance between the first 5% of closest node pairs in $\{n_{ori}\} \cup \{n_{des}^i\}_{i=1}^{N_{des}}$.

- (3) **Range.** The range can be set based on the envelope of either $\{n_{ori}\} \cup \{n_{des}^i\}_{i=1}^{N_{des}}$ or $\{reg_i\}_{i=1}^{N_{reg}}$, or by considering the boundary of $\{n_{ori}\} \cup \{n_{des}^i\}_{i=1}^{N_{des}}$ or $\{reg_i\}_{i=1}^{N_{reg}}$. In our approach, the boundary of $\{n_{ori}\} \cup \{n_{des}^i\}_{i=1}^{N_{des}}$ can be defined based on the convex hull or minimal alpha shape that encompasses $\{n_{ori}\} \cup \{n_{des}^i\}_{i=1}^{N_{des}}$ (Guo et al., 2017). Furthermore, when the range is defined based on the envelope of $\{n_{ori}\} \cup \{n_{des}^i\}_{i=1}^{N_{des}}$, the range needs to extend a half-cell outwards to prevent nodes from being on the boundary of the defined range. A detailed analysis of the range definition will be presented in Section 7.3. Figure 3 shows an example of how we define it based on $\{reg_i\}_{i=1}^{N_{reg}}$ envelope. Our approach allows for all of these

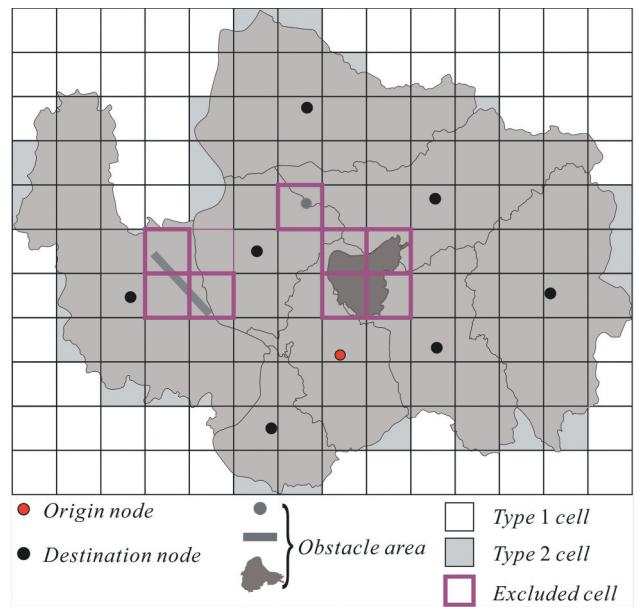


Figure 3. Modeling the mapping space of a flow map as DEM data.

definitions, and users can select the one that best suits their needs in practice.

- (4) **Obstacle areas/important objects consideration.** Certain areas such as those with adverse weather conditions may require avoidance when generating flow paths (Buchin et al., 2011; Sun, 2019). Therefore cells intersecting these areas need to be excluded from the existing range so calculated paths within the DEM data never flow across them. Based on a similar idea, our approach can represent point, linear, or planar obstacle areas or important objects by excluding their corresponding cells (Figure 3).

- (5) **Heterogeneous mapping space consideration.** Users may sometimes consider area differences, such as land having a higher travel cost than sea when mapping good movements (Buchin et al., 2011; Sun, 2019). Our approach offers two strategies: representing different areas as cells with various types or cell resolutions. Figure 3 shows how we differentiate non- $\{reg_i\}_{i=1}^{N_{reg}}$ and $\{reg_i\}_{i=1}^{N_{reg}}$ areas by assigning them Type 1 and 2 respectively.

With these definitions, the mapping space is then modeled as DEM data with N_{col} columns and N_{row} rows, the cells are denoted as $CellS = \{c_{ij}\}_{i=1,j=1}^{i \leq N_{col}, j \leq N_{row}}$, where i represents the column number, and j represents the row number. Because we set the cell resolution to ensure that the two closest nodes in $\{n_{ori}\} \cup \{n_{des}^i\}_{i=1}^{N_{des}}$ do not fall into the same cell, each node in $\{n_{ori}\} \cup \{n_{des}^i\}_{i=1}^{N_{des}}$ is represented by the cell it occupies, and n_{des}^i has

a property f_i , representing the flow volume from n_{ori} to n_{des}^i .

4.3. Step 2. Flow path calculation

The flow path calculation entails three crucial components: (1) The maze-solving algorithm, is applied to calculate potential flow paths from a destination cell to the origin cell; (2) The path length calculation, is used to identify the valid flow path for each destination cell toward the origin cell among all its potential flow paths; (3) The path importance calculation, is applied to select a valid flow path as a calculated one among all other valid paths and add it to previously calculated flow paths.

4.3.1. The maze-solving algorithm

As the flow paths are generated iteratively in our approach, a destination cell may flow in a previously generated flow path to the origin cell. Since shorter length often indicates a valid flow path (Liu et al., 2017), a potential flow path p from a destination cell to the origin cell is defined as follows: p should flow in a cell of the previously generated flow path with the shortest path.

Therefore, we apply the maze-solving algorithm to calculate a potential flow path from a destination cell to the origin cell since it aims to calculate the shortest path between two cells (Zhang et al., 2017). Three key components are involved in the maze-solving algorithm, (1) search directions, (2) direction weights, and (3) search ranges which are defined in the following sections.

4.3.1.1. Search direction definition. Search directions determine the available search directions from a cell to its neighboring cells within the maze-solving algorithm. In DEM data, a cell's outflow direction toward its neighbors can encompass eight directions (8D) denoted as $8D = \{0,1,2,3,4,5,6,7\}$, as shown in Figure 4a) (Callaghan & Mark, 1984). However exploring all eight directions each time is time-consuming, hence it is more efficient only to search in the directions from start to end cells. For instance, when searching from cell c_{22} to cell c_{61} in Figure 4a), it may be more viable to use search directions as $\{0,1,2\}$, and while searching from cell c_{46} to cell c_{43} , search directions $\{0,1,7\}$ may be more appropriate.

Given a start cell, c_{ij} , that searches toward an end cell, c_{mn} , the angle between the line that connects the two cells and the horizontal direction is denoted as θ_{ij-mn} . The search directions (D_{ij-mn}) from c_{ij} towards c_{mn} in the maze-solving algorithm are defined as Eq. (2).

$$D_{ij-mn} = \{x | x = y \bmod 8, y \in \{z - 1, z, z + 1\}, \\ z = \lfloor \theta_{ij-mn}/45 \rfloor\} \quad (2)$$

4.3.1.2. Direction weight definition. Direction weights determine which direction from a cell to its neighboring cells should be prioritized for exploration within the maze-solving algorithm. For river extraction over flat surfaces using DEMs, it has been observed that cells with higher flow accumulation are more likely to be part of the mainstream (Zhang et al., 2017). Thus, when exploring from a given cell (c_{ij}) to its neighboring cell (c_{mn}), directions leading toward cells with larger differences in their potential flow accumulation values are prioritized. To this end, the direction weight DW_{ij-mn} for the direction from c_{ij} to c_{mn} is defined as Equation (3).

$$DW_{ij-mn} = \text{Sum}f_{ij} - \text{Sum}f_{mn} + f_T \quad (3)$$

where f_T is a large constant ensuring $DW_{ij-mn} > 0$; $\text{Sum}f_{ij}$ and $\text{Sum}f_{mn}$ represent potential flow accumulation of c_{ij} and c_{mn} respectively, determined by calculating the total volume of the cell in k -order surrounding as per Equation (4) (Garbrecht & Martz, 1997).

$$\text{Sum}f_{ij}^k = \sum_{x=i-k, y=i-k}^{x \leq i+k, y \leq i+k} f_{xy} \quad (4)$$

where f_{xy} is the flow volume of cell c_{xy} , Figure 4b) shows an example of k -order surroundings.

The definition of direction weights can implicitly aid in establishing a suitable tree layout (DC_1) while minimizing total length (GC_3). As depicted in Figure 4b), this results in obtaining the path $c_{27}c_{26}c_{35}c_{34}c_{61}$ as the final path from destination cell c_{27} to origin cell c_{61} , instead of potential paths $c_{27}c_{25}c_{61}$ and $c_{27}c_{45}c_{43}c_{61}$, which are of equal length. But more destination cells are likely to be connected to $c_{27}c_{26}c_{35}c_{34}c_{61}$ with shorter paths, as compared to connecting them to $c_{27}c_{25}c_{61}$ and $c_{27}c_{45}c_{43}c_{61}$, thereby resulting in a more suitable tree layout with a smaller total length.

4.3.1.3. Search range definition. The maze-solving algorithm utilizes predefined search directions and direction weights to navigate a search range to identify the shortest path between two cells. Therefore, the exclusion of certain cells from the search range ensures that the generated paths will not flow through these cells, namely, no overlaps (RC_3) or crossings (RC_2 and RC_4) will be made on these cells.

Given a start cell, c_{ij} , that searches toward an end cell, c_{mn} . The search range is defined as excluding destination nodes' t -order surrounding and previously

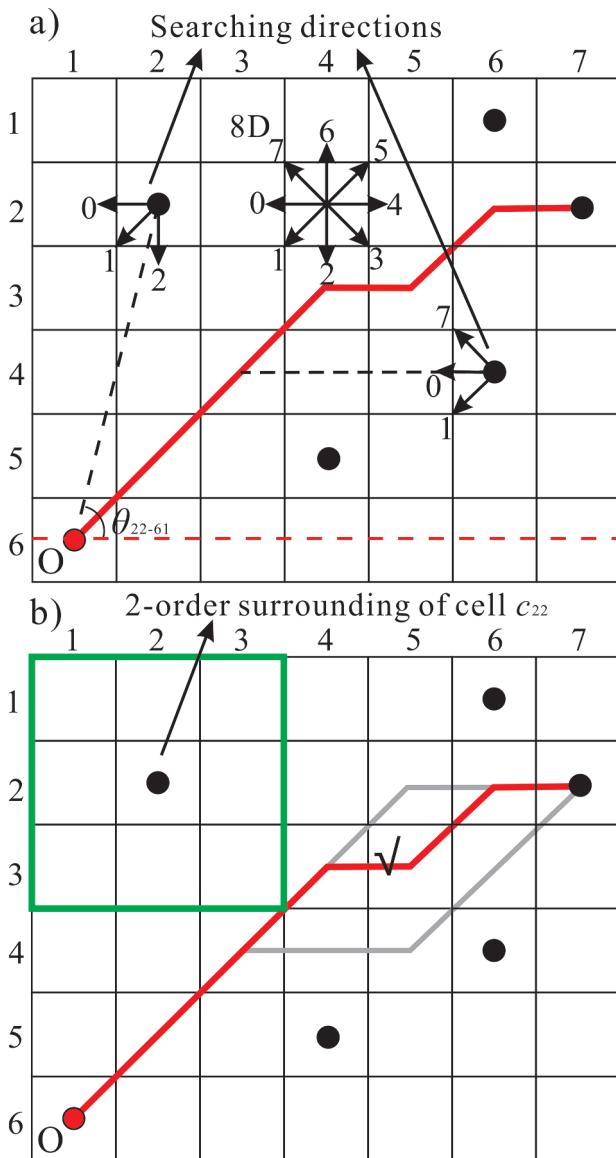


Figure 4. a) Search direction definition, where only directions from the start cell toward the end cell are explored. b) Direction weight definition, where exploration prioritizes direction toward a neighborhood cell with higher potential flow accumulation.

generated flow path cells except for cells c_{ij} and c_{mn} . The remaining cells are then considered within the search range. By adjusting the value of t , we can ensure that the distance between nodes and edges is suitable, consequently satisfying RC_5 . In Figure 5a), for instance, the cells representing the destination nodes and previously generated paths (except the cells c_{46} and c_{52}) are excluded from the search range when obtaining a path from cell c_{46} to c_{52} . This guarantees that no overlaps or crossings occur with nodes and edges, resulting in an obtained path as $c_{46}c_{45}c_{44}c_{53}c_{52}$.

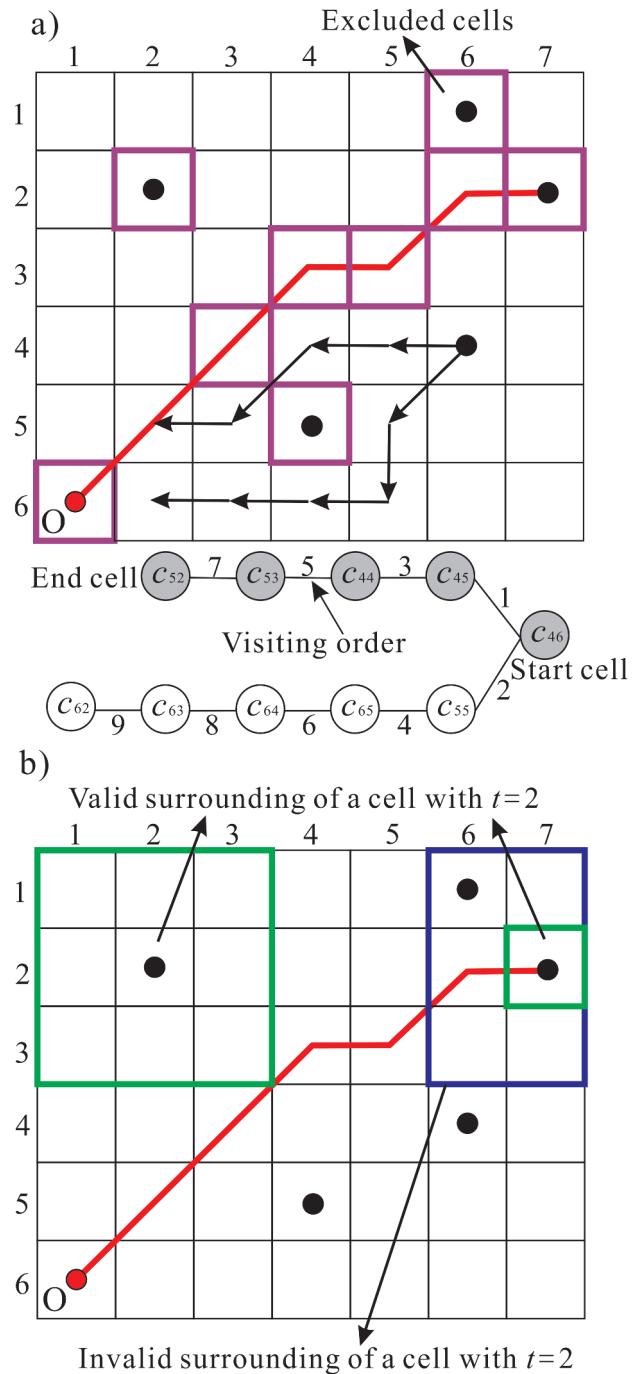


Figure 5. a) An example of the shortest path calculation with the maze-solving algorithm from cell c_{46} to cell c_{52} . b) adjustment for close destination nodes: t -order surrounding is adaptively set from $t = 2$ to $t = 1$ for cell c_{27} because cell c_{16} is in its 2-order surrounding.

It is worth noting that in situations where the destination nodes are in the t -order surrounding of other cells, they will be excluded from the search range as per our definition, which in turn produces no available paths. To address this, we adaptively adjust the value

of t to a smaller one for these cells. As seen in Figure 5b), cell c_{16} is in a 2-order surrounding of cell c_{27} . When attempting to obtain a path with cell c_{27} as a start cell by using the maze-solving algorithm with $t = 2$, cell c_{27} would be excluded from the search range. Thus, we set $t = 1$ for cell c_{16} in this context.

4.3.2. Path length calculation

Once all potential flow paths from the destination cells to the origin cell are obtained using the maze-solving algorithm, we then identify the shortest one for each destination cell as a valid path among its all potential flow paths. Any potential path that fails to meet quality criteria for flow maps would be generally longer and be less likely to be considered a valid path, thereby contributing to the generation of a flow map that better adheres to the quality criteria. In this regard, we primarily consider three factors, namely, the total length minimization (GC_3), avoidance of acute flow-in angles (RC_1), and curved edge preference (GC_1), while computing the path length.

Suppose a potential path from a destination cell to the origin cell is p , it can be further divided into sub-paths $sub-p_1$ and $sub-p_2$, where $sub-p_2$ is a sub-path of a previously generated path while $sub-p_1$ is not. The flow-in angle of p is denoted as θ_{flow} . The path length $L(p)$ is computed as follows.

First, GC_3 dictates that the overall path length needs to be minimized. Hence, we prioritize the length of $sub-p_1$ while calculating $L(p)$, as it is newly generated and a shorter $sub-p_1$ would help minimize the total graph length. For instance, in Figure 6, two different paths $c_{46}c_{35}c_{34}c_{61}$ and $c_{46}c_{43}c_{61}$ are both potential paths from destination cell c_{46} to origin cell c_{61} . However, $c_{46}c_{35}c_{34}c_{61}$ is more likely to be the shortest one according to GC_3 , as sub-paths $c_{35}c_{34}c_{61}$ and $c_{43}c_{61}$ are both parts of a previously generated path ($c_{27}c_{26}c_{35}c_{34}c_{61}$), and sub-path $c_{46}c_{35}$ is shorter than sub-path $c_{46}c_{43}$.

Second, RC_1 mandates that acute flow-in angles are not allowed. To meet RC_1 , we deploy a penalty strategy to refine $L(p)$, and a path with acute flow-in angles is then less likely to be the shortest one. For example, in Figure 6, path $c_{54}c_{43}c_{61}$ violates RC_1 as it has an acute flow-in angle at cell c_{43} , and as such, path $c_{54}c_{52}c_{61}$ is more likely to be selected as the shortest path from destination cell c_{54} to origin cell c_{61} .

Third, GC_1 dictates a preference for curved edges and stipulates that edges should be rendered as curved lines. Short non-hanging edges may result in uneven turns after the rendering process and should be avoided. To align with GC_1 , we also employ a penalty strategy to refine $L(p)$, making paths that include short non-hanging edges less likely to be considered the shortest. Specifically, we define

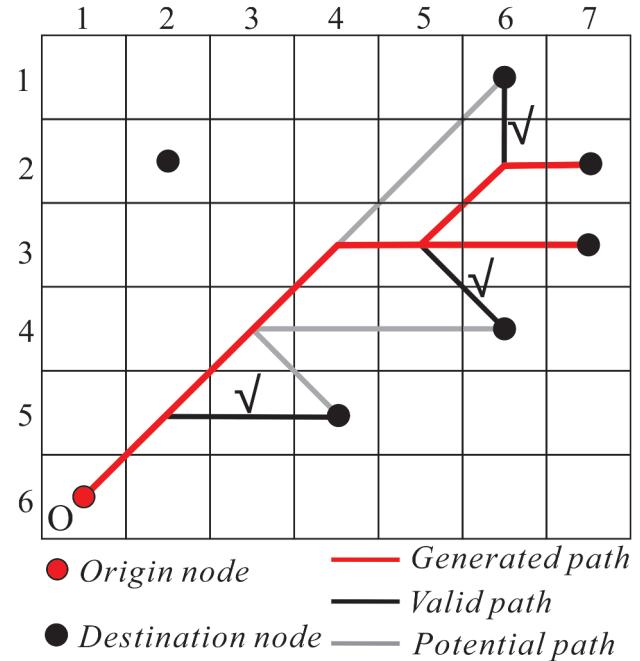


Figure 6. Path length definition.

a non-hanging edge as short if its length is less than R (cell resolution defined in Equation (1)). For example, in Figure 6, path $c_{16}c_{35}c_{34}c_{61}$ violates GC_1 as it has a short non-hanging edge $c_{35}c_{34}$, and as such, path $c_{16}c_{26}c_{35}c_{34}c_{61}$ is more likely to be selected as the shortest path from destination cell c_{16} to origin cell c_{61} .

Based on the above three considerations, $L(p)$ is defined as Equation (5).

$$L(p) = \begin{cases} L(sub-p_1) + L(sub-p_2) \times \omega & (\theta_{flow} < \theta_T \wedge \text{Min}(L_{\text{non-e}}) < L_T) \\ L(sub-p_1) + L(sub-p_2) \times \omega + L_{\text{pen}} & (\text{else}) \end{cases} \quad (5)$$

where $L(sub-p_1)$ and $L(sub-p_2)$ represents the lengths of $sub-p_1$ and $sub-p_2$, respectively. ω is a weight and $\omega \leq 1$, indicating that $L(sub-p_1)$ is more important than $L(sub-p_2)$. L_{pen} is a large constant and serves as a penalty value if the path violates RC_1 by having a flow-in angle (θ_{flow}) greater than a threshold value (θ_T) or violates GC_1 by having a non-hanging edge with its length ($L_{\text{non-e}}$) less than a threshold value (L_T).

4.3.3. Path importance calculation

After obtaining the valid paths for each destination cell to the origin cell, the valid path with the largest importance is selected as a calculated flow path and added to previously generated paths. Because the flow map needs to minimize the total graph length (GC_3), we establish the importance $I(p)$ of a path p from a destination cell to the origin cell based on its length $L(p)$. p can be categorized as either a Type I or Type II, based on whether its

sub-path is part of a previously generated path, where the Type I path directly connects the destination cell to the origin cell. Such paths are more likely to be part of the mainstream in the river system extraction process (Zhang et al., 2017). Consequently, Type I paths are considered more important than Type II paths. Therefore, $I(p)$ is defined as Equation (6).

$$I(p) = \begin{cases} L(p) + I_{\text{pen}} & (p \text{ is Type I path}) \\ L(p) & (p \text{ is Type II path}) \end{cases} \quad (6)$$

where I_{pen} is a large constant to make sure that a type I path is more important than a type II path.

4.3.4. The iterative process for flow path calculation

The flow paths from all destination cells to the origin cell are calculated iteratively as Algorithm 1.

Algorithm 1: Flow path calculation

```

Require: DEM data as  $Cells = \{c_{ij}\}_{i=1}^{N_{\text{des}}} \times \{j\}_{j=1}^{N_{\text{row}}}$ 
The cell represents the origin node as  $c_{\text{ori}}$ 
Cells represent the destination nodes as  $\{c^i_{\text{des}}\}_{i=1}^{N_{\text{des}}}$ 
Calculated flow paths from  $\{c^i_{\text{des}}\}_{i=1}^{N_{\text{des}}}$  to  $c_{\text{ori}}$  as  $PS_{\text{cal}}$ 
Cells represent  $PS_{\text{cal}}$  as  $Cells_{\text{path}}$ 
Potential flow paths as  $PS_{\text{pot}}$ 
Valid flow paths as  $PS_{\text{valid}}$ 
Initialize  $PS_{\text{cal}} \leftarrow \emptyset$ ,  $Cells_{\text{path}} \leftarrow \{c_{\text{ori}}\}$ 
While  $\{c^i_{\text{des}}\}_{i=1}^{N_{\text{des}}} \text{ Not } \emptyset$  Do
    Initialize  $PS_{\text{valid}} \leftarrow \emptyset$ 
    For  $c^i_{\text{des}} \in \{c^i_{\text{des}}\}_{i=1}^{N_{\text{des}}}$  Do
        Initialize  $PS_{\text{pot}} \leftarrow \emptyset$ 
        For  $c_j \in Cells_{\text{path}}$  Do
            Calculate the potential path ( $p_{\text{pot}}$ ) from  $c^i_{\text{des}}$  to  $c_{\text{ori}}$  which
            flows in  $PS_{\text{cal}}$  at cell  $c_j$  with the maze-solving algorithm
            If  $p_{\text{pot}} \text{ Not } \emptyset$  Do Add  $p_{\text{pot}}$  to  $PS_{\text{pot}}$ 
            Get the shortest path  $p_{\text{short}}$  in  $PS_{\text{pot}}$  and add  $p_{\text{short}}$  to  $PS_{\text{valid}}$ 
            Get the most important path  $p_{\text{imp}}$  in  $PS_{\text{valid}}$  and add  $p_{\text{imp}}$  to  $PS_{\text{cal}}$ 
            Update  $Cells_{\text{path}}$ 
            Remove the corresponding destination cell of  $p_{\text{imp}}$  from  $\{c^i_{\text{des}}\}_{i=1}^{N_{\text{des}}}$ 
Return  $PS_{\text{cal}}$ 
```

4.4. Step 3. Flow path rendering

After establishing flow paths connecting all destination cells to the origin cell, we elegantly visualize these paths with dynamically adjusted widths, employing Bézier curves as outlined by Sun (2016) for visually captivating outcomes. For a comprehensive understanding of this process, further details can be found in our appendix.

5. Experiment

5.1. Dataset

(1) Dataset for benchmark

For benchmarking, we selected the population migrations from Texas to other states in the 2000 US Census as our experiment data. This dataset has been widely used as a benchmark for flow map production

(Buchin et al., 2011; Debiasi et al., 2014; Nocaj & Brandes, 2013; Phan et al., 2005; Sun, 2019).

(2) Datasets for extencsion

To evaluate the extension of our approach, we employed three additional datasets. Firstly, we considered the population migration from California to other states in the 2000 US Census. In this dataset, we focused on mapping spaces with obstacle areas, such as the Great Salt Lake and a part of the Mississippi River. Secondly, we explored the transportation of goods from Russia to other European countries in 2019. This dataset allowed us to examine the mapping of spaces where sea areas needed to be avoided to minimize transportation expenses. Thirdly, we integrated the gridded population data of the world (v4) with a spatial resolution of 30 arc-seconds (approximately 1 km). This dataset was utilized to showcase our approach's capability to harmonize with diverse datasets.

5.2. Evaluation metric

We employed seven general metrics to evaluate the quality of flow maps, following the quality criteria outlined in Section 3.2 (Sun, 2019; Ware et al., 2002). These metrics are as follows:

- (1) Mean value of visual smoothness index (*MSI*): This metric was used to assess the preference for curved edges (**GC**₁) and the necessity of curve differences (**GC**₂), which utilized a visual smoothness index (*SI*) proposed by Sun (2019). *SI* assigns higher values to edges with small splitting angles while penalizing unnecessary bends and large curvatures. Higher values of *MSI* indicate visually smooth and natural-looking.
- (2) Total length of the flow map (*TL*): *TL* was utilized to evaluate the minimization of total length (**GC**₃).
- (3) Number of acute flow-in angles (*N_{aa}*): *N_{aa}* was utilized to evaluate avoidance toward acute flow-in angle (**RC**₁). An acute flow-in angle is defined using a threshold(θ_T), where angles smaller than θ_T are designated as such. In this approach we set $\theta_T=120$.
- (4) Number of edge crossings (*N_{ec}*): *N_{ec}* was utilized to evaluate avoidance toward edge cross (**RC**₂).
- (5) Number of overlaps between nodes and edges (*N_{ov}*): *N_{ov}* was utilized to evaluate overlap avoidance between nodes and edges (**RC**₃).
- (6) Number of crossings between edges and important objects (*N_{oc}*): *N_{oc}* was utilized to evaluate

- crossing avoidance between edges and important map objects (RC_4).
- (7) Minimum distance between nodes and nearby edges (MD): MD provided insights into suitable distance maintenance between nodes and nearby edges(RC_5).

5.3. Result

5.3.1. The visualization result

Our proposed approach draws inspiration from Sun's (2019) work and utilizes a dendritic drainage pattern simulation to generate flow maps that resemble natural river systems. To evaluate our approach, we compared it with Sun's (2019) approach and other non-river simulation approaches. As flow map outcomes may be influenced by parameter settings, we generated three distinct flow maps using different parameters. A detailed discussion of the impact of these parameter settings on flow maps is presented in Section 7.2. To ensure a fair comparison, we constrained the flow lines within the experimental area. This was achieved by defining the mapping space range as follows: Initially, we generated DEM data based on the envelope of the node set; Subsequently, we retained only the cells that intersected with the experimental area as the defined mapping space range. Notably, we set T_L (as defined in Section 4.3.2) to be $2R$, and T_{offset} (as described in Section 4.4.1) to be $\sqrt{2}/2R$. Table 1 provides further detailed information for each flow map, and Figures 7 and 8 display visualizations of each corresponding flow map.

Table 1. The detailed information for each flow map.

Name	Descriptions
FD	The flow map was generated using a force-directed approach (Debiasi et al., 2014)
ST	The flow map was generated using a spiral tree approach (Buchin et al., 2011)
SB	The flow map was generated using a stub bundling approach (Nocaj & Brandes, 2013)
TNSS ₁	The flow map was generated using Sun's (2019) approach with $\gamma = 0.65$
TNSS ₂	The flow map was generated using Sun's (2019) approach with $\gamma = 0.50$
TNSS ₃	The flow map was generated using Sun's (2019) approach with $\gamma = 0.30$
REA-FM ₁	The flow map was generated using the proposed approach with $w = 0.65, k = 4, t = 1$
REA-FM ₂	The flow map was generated using the proposed approach with $w = 0.30, k = 4, t = 1$
REA-FM ₃	The flow map was generated using the proposed approach with $w = 0.65, k = 4, t = 2$

5.3.2. Statistical analysis

Table 2 presents the quantitative results of the generated flow maps with the existing approaches. The following comparisons were made:

(1) Comparison with the non-river simulation methods.

As shown in Table 2, we can observe that REA-FM₁, REA-FM₂, and REA-FM₃ outperform FD, ST, and SB in terms of MSI values. This demonstrates that our proposed approach produces more natural-looking flow maps than these non-river simulation approaches which is why we opted for river simulation as our main methodology. Moreover, REA-FM₁, REA-FM₂, and REA-FM₃ have larger MD_{\min} values compared to FD and SB, and fewer nodes are located within $100 \times 10^3 \text{m}$ and $70 \times 10^3 \text{m}$ to their nearby edges compared to FD, ST, and SB. These indicate our proposed approach achieves better maintenance of suitable distance between nodes and edges. In terms of TL values, ST has the smallest value, but REA-FM₂ also comes at a close second exhibiting similar small TL values (an increase from 22.01×10^6 to $22.30 \times 10^6 \text{m}$). These indicate that our proposed approach is competitive in minimizing the total graph length when compared to the non-river simulation methods. Furthermore, ST, SB, REA-FM₁, REA-FM₂, and REA-FM₃ all avoid acute flow-in angles whilst also preventing any edge crossings or node overlaps from occurring during processing thereby demonstrating that our proposed approach successfully avoids such issues as observed in existing processes.

(2) Comparison with the river simulation methods.

As shown in Table 2, we can observe that REA-FM₁ and REA-FM₂ outperform TNSS₁, TNSS₂, and TNSS₃ in terms of TL values. These indicate that our proposed approach better minimizes total graph length when compared to Sun's (2019) approach. In terms of MD_{\min} values, REA-FM₃ has fewer nodes located within $100 \times 10^3 \text{m}$ and $70 \times 10^3 \text{m}$ to their nearby edges, and the other layouts exhibit comparable results. These indicate that the proposed approach can achieve competitive if not superior results when compared to Sun's (2019) approach. In terms of MSI values, REA-FM₁, REA-FM₂, REA-FM₃, TNSS₁, TNSS₂, and TNSS₃ exhibit very close MSI values where the largest deviation is less than 0.048. These indicate that the river simulation methods including our proposed approach and Sun's (2019) approach both are capable of achieving natural-looking flow maps.

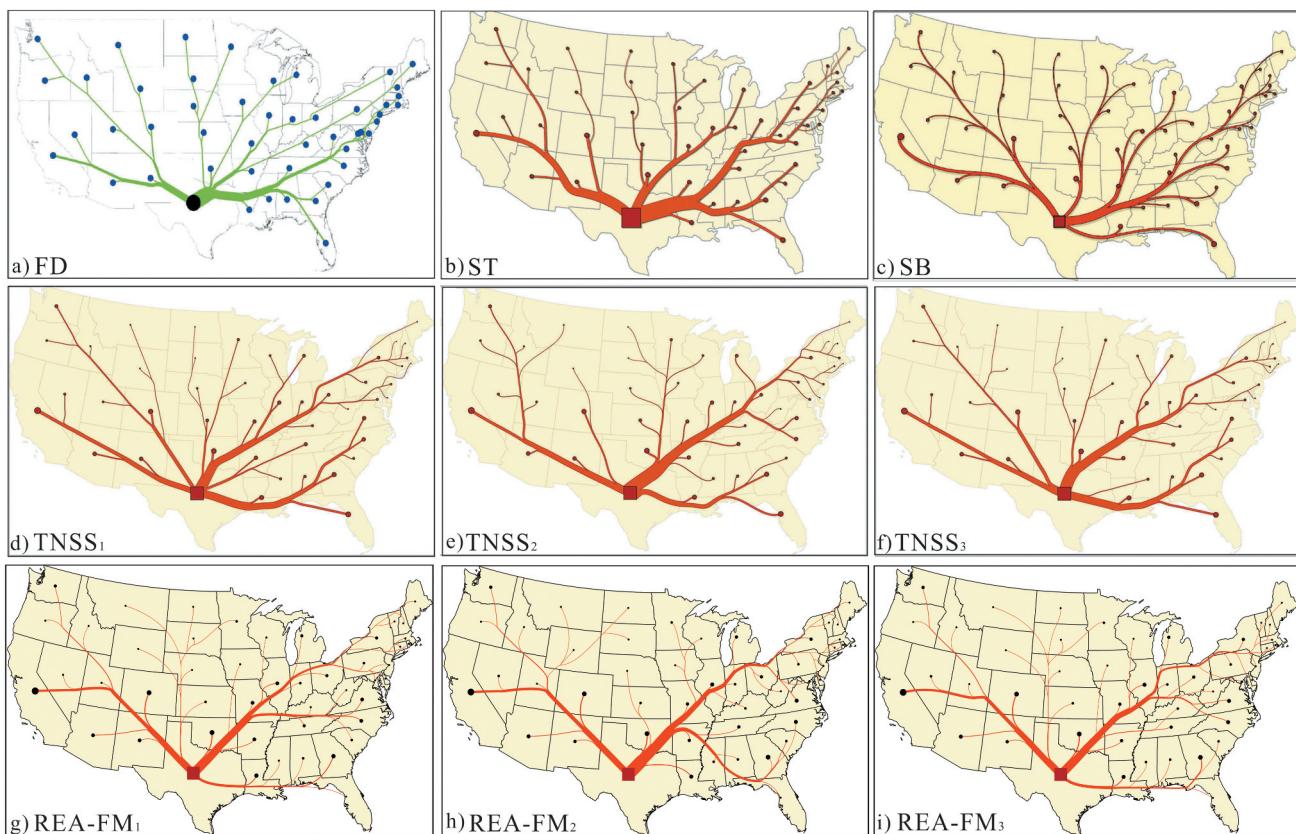


Figure 7. The flow maps that were generated using a) FD, b) ST, c) SB, d) TNSS₁, e) TNSS₂, f) TNSS₃, g) REA-FM₁, h) REA-FM₂, i) REA-FM₃, respectively.



Figure 8. The enlarged view of REA-FM₁.

Table 2. Quantitative assessment of flow map quality.

Measures	FD	ST	SB	TNSS ₁	TNSS ₂	TNSS ₃	REA-FM ₁	REA-FM ₂	REA-FM ₃
$TL (10^6 \text{m}) \downarrow$	23.61	22.01	24.46	27.67	25.59	25.09	24.93	22.30	25.58
$MSI \uparrow$	0.625	0.621	0.698	0.745	0.726	0.698	0.717	0.726	0.746
$MD (10^3 \text{ m})$									
$MD_{\min} \uparrow$	7.8	49.7	13.0	27.0	37.3	47.6	22.3	22.3	28.4
$N(MD < 100) \downarrow$	29	27	28	16	15	15	13	18	8
$N(MD < 70) \downarrow$	27	14	17	5	2	8	6	10	5
$N(MD < 40) \downarrow$	5	0	10	3	0	0	1	2	2
$N(MD < 20) \downarrow$	1	0	1	0	0	0	0	0	0
$N_{aa} (< 120^\circ) \downarrow$	1	0	0	0	0	0	0	0	0
$N_{ec} \downarrow$	0	0	0	0	0	0	0	0	0
$N_{ov} \downarrow$	1	0	0	0	0	0	0	0	0

The statistics about ST, SB, FD, TNSS₁, TNSS₂, and TNSS₃ in Table 2 are from Sun (2019).

Furthermore, REA-FM₁, REA-FM₂, REA-FM₃, TNSS₁, TNSS₂, and TNSS₃ all avoid acute flow-in angles whilst also preventing any edge crossings or node overlaps from occurring during processing thereby demonstrating that our proposed approach successfully avoids such issues as Sun' (2019) approach.

6. Extensions

6.1. Case 1. Mapping space with obstacle areas

To demonstrate how our proposed approach can be applied when dealing with obstacle areas, we considered population migration from California to other states using data from the US Census Bureau in 2000 where the Great Salt Lake and part of the Mississippi River were taken as obstacles. In this case, cells intersecting the obstacle areas were excluded while modeling the DEM data for mapping space. The parameter settings were the same as REA-FM₁, and the results are shown in Figure 9 and Table 3. To measure whether the flow map can avoid obstacle areas, we employed N_{oc} to measure the number of crossings between edges and obstacle areas.

From the results, we can observe that our approach successfully produces a flow map with no crossings between edges and obstacle areas while slightly increasing total length by $0.59 \times 10^6 \text{m}$ (2.5%). Conversely, if these obstacle areas had not been considered then 4 crossings between edges and obstacles would have occurred which can be observed in Areas A and B of Figure 9a). However, such issues could both well avoid node overlaps or edge crossings thereby meeting quality criteria for flow maps.

6.2. Case 2. Heterogeneous mapping space

To further demonstrate how our proposed approach handles heterogeneous mapping spaces, we took good exports from Russia (>0.1%) to European countries as experiment data where sea areas needed avoidance as much as possible to save transportation expenses. To avoid crossings between edges and sea areas we set smaller cell resolution for cells within sea areas as $1/3R$, R is the cell resolution of the DEM data. Because the mapping space is searched cell by cell in the maze-solving algorithm, smaller cell resolution means more steps will cost by searching across sea area leading avoidance of crossings between edges and sea areas.

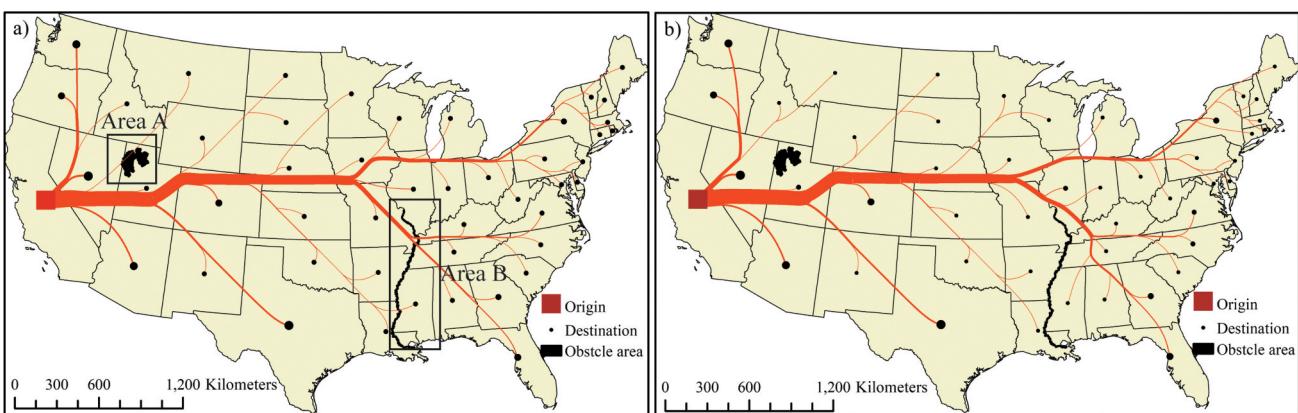


Figure 9. Flow maps that were generated a) without consideration of obstacle areas, b) with such areas taken into account.

Table 3. Quantitative assessment of flow map quality.

Measures	Without obstacle areas	With obstacle areas
$TL (10^6 \text{ m}) \downarrow$	23.09	23.68
$MSI \uparrow$	0.728	0.719
$MD (10^3 \text{ m})$		
$MD_{\min} \uparrow$	24.9	43.5
$N(MD < 100) \downarrow$	14	16
$N(MD < 70) \downarrow$	5	4
$N(MD < 40) \downarrow$	1	0
$N(MD < 20) \downarrow$	0	0
$N_{aa} (< 120^\circ) \downarrow$	0	0
$N_{ec} \downarrow$	0	0
$N_{ov} \downarrow$	0	0
$N_{oc} \downarrow$	4	0

Parameter settings remained the same as REA-FM₁, and the results are shown in Figure 10 and Table 4. To measure whether the flow map can avoid sea areas, we employed N_{oc} to measure the number of crossings between edges and sea areas.

From the results, we can observe that a flow map with fewer crossings between edges and sea areas was successfully produced while the total length of the flow map slightly increased by $1.15 \times 10^6 \text{ m}$ (3.7%). However, four crossings between edges and sea areas observed in Figure 10b) are inevitable due to these involved areas being separate from the main lands. Conversely, if the sea areas had not been considered then five more crossings between edges and sea areas would have occurred,

as shown in Areas A to E in Figure 10a). Nonetheless, both generated flow maps with our approach can avoid node overlaps or edge crossings thereby meeting the quality criteria for flow maps.

6.3. Case 3. Integration with other data

To further exemplify the integration capability of our proposed approach with external datasets, we experimented using the gridded population data of the world (v4) with a resolution of 30 arc seconds, roughly equivalent to 1 kilometer. In this experiment, the cells of the dataset were directly utilized as our modeling DEM data, with each cell assigned a population density value. Notably,

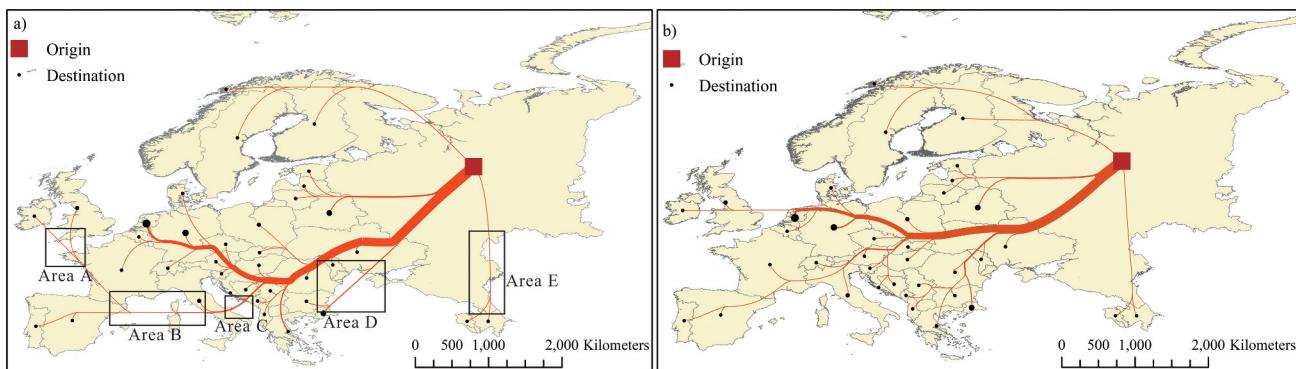


Figure 10. Flow maps that were generated a) without consideration of sea areas, b) with avoidance of such areas as much as possible.

Table 4. Quantitative assessment of flow map quality.

Measures	Homogeneous mapping space	Heterogeneous mapping space
$TL (10^6 \text{ m}) \downarrow$	30.92	32.07
$MSI \uparrow$	0.761	0.749
$MD (10^3 \text{ m})$		
$MD_{\min} \uparrow$	68.4	78.3
$N(MD < 100) \downarrow$	5	3
$N(MD < 70) \downarrow$	1	0
$N(MD < 40) \downarrow$	0	0
$N(MD < 20) \downarrow$	0	0
$N_{aa} (< 120^\circ) \downarrow$	0	0
$N_{ec} \downarrow$	0	0
$N_{ov} \downarrow$	0	0
$N_{oc} \downarrow$	4	9

our approach prioritizes cells with higher population density during the path search, resulting in flow paths more likely to traverse areas with denser populations. The parameter settings were kept consistent with *REA-FM*₁, and the outcomes are presented in Figure 11 and Table 5. Furthermore, to assess the population density of the areas through which the flow paths traverse, we employed the average population density (*AvePD*) as a metric. *AvePD* was computed as the average of all population density values within the grids through which the flow paths passed. It is important to note that, given the projection of the experiment data is WGS-84, we opted not to convert it to the Albert projection as used in other USA-based datasets.

From the results, we can observe that the integration of gridded population data successfully led to the creation of a flow map with a higher average population density (2.79↑), while the total length of the flow map slightly increased by 0.11×10^6 m (0.4%). Notably, when examining specific regions, such as Areas A, B, and C in Figure 11, it becomes evident that the integration of the gridded population data played a pivotal role. Without this integration, the flow map would have traversed areas characterized by lower population density. Nonetheless, both generated flow maps with our approach can avoid node overlaps or edge crossings thereby meeting the quality criteria for flow maps.

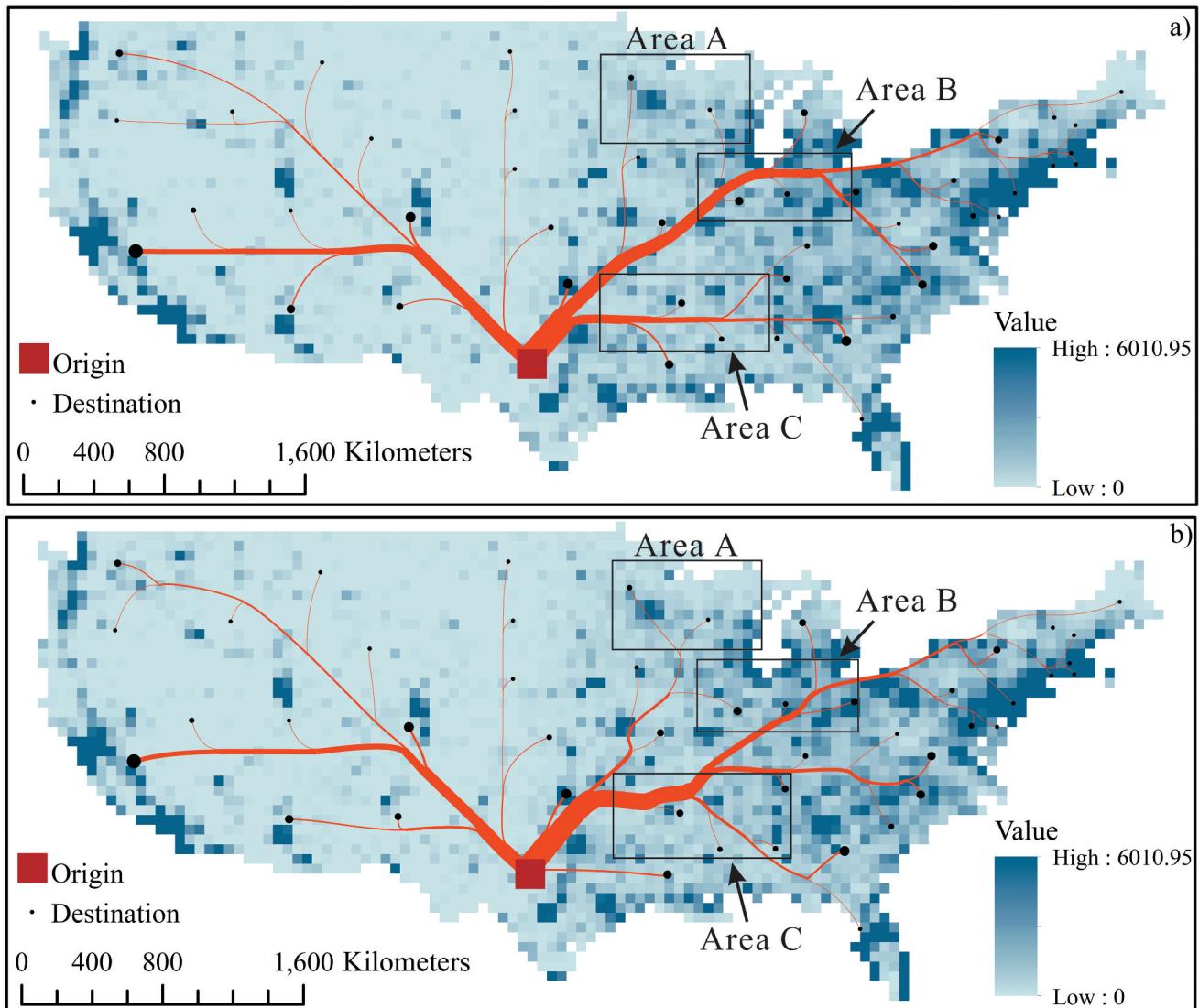


Figure 11. Flow map that were generated a) without integration of population density data, b) with integration of population density data.

Table 5. Quantitative assessment of flow map quality.

Measures	Without data integration	With data integration
$TL (10^6 \text{ m}) \downarrow$	25.98	26.09
$MSI \uparrow$	0.780	0.779
$AvePD$	58.73	61.52
$MD (10^3 \text{ m})$		
$MD_{\min} \uparrow$	11.2	11.2
$N(MD < 100) \downarrow$	19	19
$N(MD < 70) \downarrow$	12	11
$N(MD < 40) \downarrow$	4	2
$N(MD < 20) \downarrow$	1	1
$N_{aa} (<120^\circ) \downarrow$	0	0
$N_{ec} \downarrow$	0	0
$N_{ov} \downarrow$	0	0

7. Discussions

7.1. Strategy effectiveness analysis

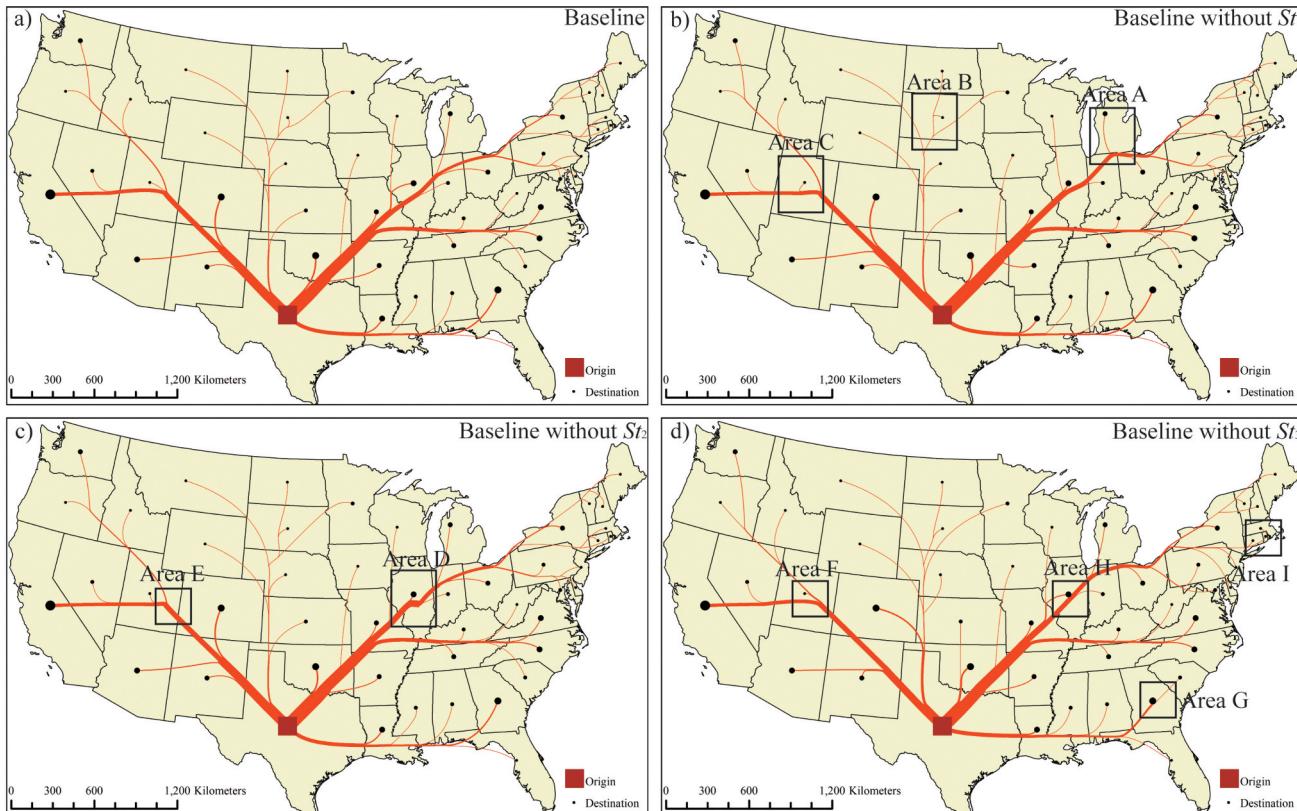
To generate a satisfactory flow map, our approach employed various strategies which were summarized in [Table 6](#). To validate their feasibility, ablation experiments were conducted with REA-FM₁ as the baseline. Three additional flow maps were produced without

implementing the strategies outlined in [Table 6](#), as illustrated in [Figure 12](#).

- (1) St_1 aims at avoiding acute flow-in angles. As shown in [Figure 12b](#)), if St_1 is not applied, three acute flow-in angles will be present. However, if St_1 is applied ([Figure 12a](#)), such angles can be effectively avoided although longer branches may

Table 6. Strategies applied in our proposed approach.

Name	Descriptions	Location
St_1	Path length calculation by considering acute flow-in angle avoidance.	Section 4.3.2
St_2	Path length calculation by considering curved edge preference	Section 4.3.2
St_3	Search range definition by considering overlap avoidance and suitable distance maintenance between nodes and edges	Section 4.3.1

**Figure 12.** Flow maps that were produced without b) St_1 , c) St_2 , and d) St_3 , a) is the baseline.

be required leading to an increase of total length by 0.95×10^6 m (3.9%). Nonetheless, both generated flow maps with or without St_1 can avoid node overlaps or edge crossings thereby meeting quality criteria for flow maps except for avoidance of acute flow-in angles.

- (2) St_2 aims at avoiding unnatural turns in flow maps by avoiding short non-hanging edges. In contrast to when it was not implemented and two unnatural turns would be made (Areas D and E in Figure 12c), successful avoidance of these issues occurred when this strategy was employed while generating the resulting map shown in Figure 12a). However, longer edges may be required leading to an increase of total length by approximately 0.11×10^6 m (0.4%) due to the avoidance of short non-hanging edges.
- (3) St_3 aims at avoiding overlaps and maintaining suitable distances between nodes and edges. In contrast to when it was not implemented and seven overlaps between nodes and edges occurred along with six nodes being less than 40×10^3 m away from nearby edges (Figure 12d), successful avoidance of these issues occurred when this strategy was employed while generating the resulting map shown in Figure 12a). However, longer edges may

be required leading to an increase of total length by approximately 0.27×10^6 m (1.1%) due to its ability to maintain a suitable distance between nodes and edges while preventing overlaps.

Overall, these ablation experiments demonstrate that all three strategies are effective in generating a satisfactory flow map by avoiding acute flow-in angles, unnatural turns, preventing overlap, and maintaining suitable distance between nodes and edges as well as satisfying other quality criteria for such maps.

7.2. Parameter sensitivity analysis

In this section, we analyze how parameter settings influence flow map layouts and show how these parameters can intuitively control the layouts.

(1) ω in Equation. (3)

The parameter ω is used to define path length by considering total length minimization. Smaller values of ω lead to smaller total lengths (TLs). We generated three different layouts with varying ω values (0.30, 0.65, and 1.0) while other parameters remained the same as REA-FM₁, and the results are shown in Figure 13.

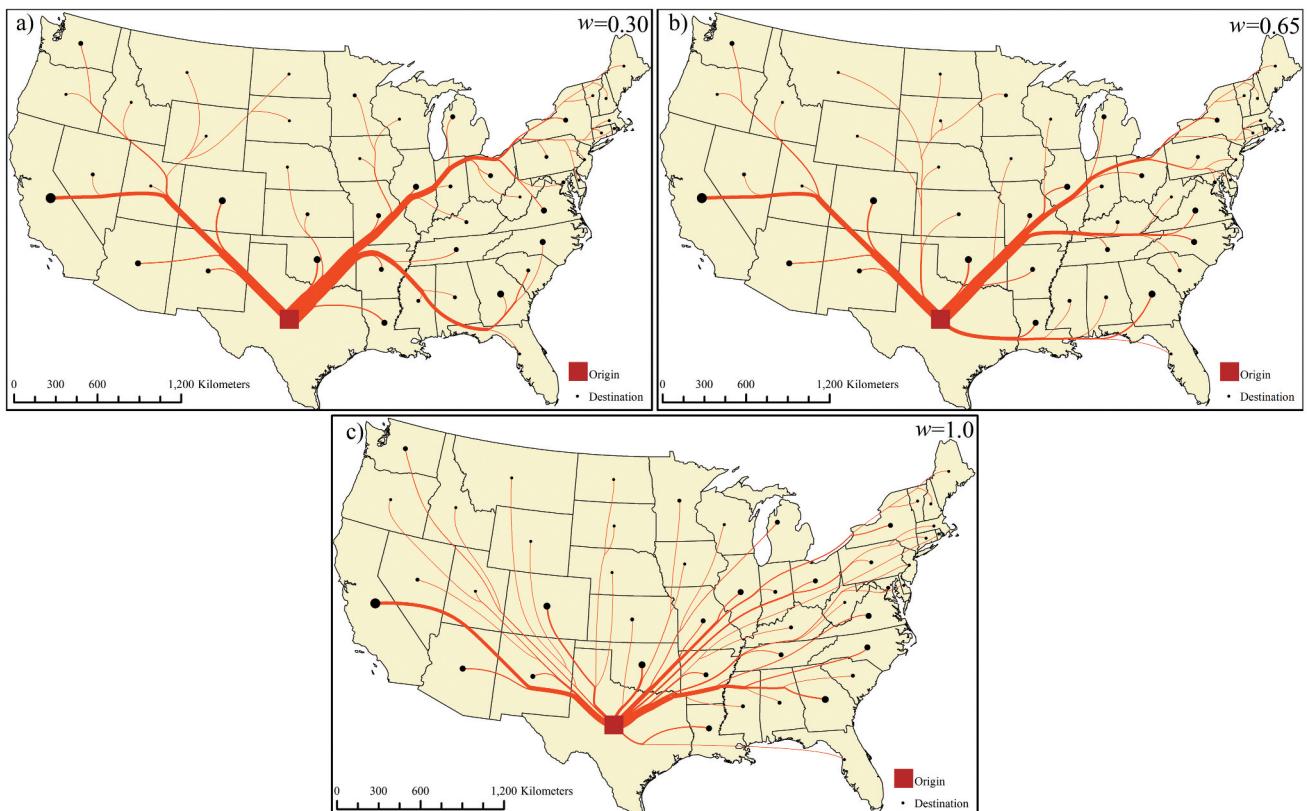


Figure 13. Flow maps that were generated with a) $\omega = 0.30$, b) $\omega = 0.65$, c) $\omega = 1.0$.

From Figure 13, we observe that paths from destination nodes tend to flow in nearby paths when a smaller value of ω is set ($\omega = 0.30$) as shown in Figure 13a). Conversely, if a larger value of ω is set ($\omega = 1.0$), paths from destination nodes tend to converge closer toward the origin node as seen in Figure 13c). These observations indicate longer branches would be generated with increasing ω -values. Similarly, higher TL results from increased ω -values: TL increases from 22.30×10^6 m at $\omega = 0.30$ to 24.93×10^6 m at $\omega = 0.65$ and 41.87×10^6 m at $\omega = 1.0$. However, nodes located within 100×10^3 m to nearby edges increase whether users opt for smaller or larger values of ω : $N(MD < 100)$ increases from 13 at $\omega = 0.65$ to 18 at $\omega = 0.30$ and 24 at $\omega = 1.0$. Consequently, a moderate value of ω is recommended since it better maintains suitable distances between edge and nodes such as when using $\omega = 0.65$. If users prefer a smaller total length, a smaller ω -value is recommended; if users prefer longer branches, a larger ω -value is suggested. Nevertheless, the generated layouts using different ω -values meet quality criteria for flow maps without node overlaps or edge crossings.

(2) k in Equation (6)

The parameter k defines the k -order surrounding of a cell which is used for potential flow

accumulation ($Sumf$) computation. The direction to a cell with a higher $Sumf$ has priority in the maze-solving algorithm, thereby minimizing the total length (TL). We generated three different layouts with varying k values (0, 4, and 8) while other parameters remained the same as REA-FM₁, and the results are shown in Figure 14.

From Figure 14, we observe that setting either smaller or larger values of k increases TL : TL increases from 24.93×10^6 m at $k = 4$ to 24.98×10^6 m at $k = 0$ and 25.14×10^6 m at $k = 8$. Thus, a suitable k needs to be set in practice and $k = 4$ is recommended. Nevertheless, the generated layouts using different values of k meet quality criteria for flow maps without node overlaps or edge crossings.

(3) t for search range definition

The parameter t is used to ensure overlap avoidance while maintaining suitable distances between nodes and edges. When $t = 0$, it means that overlap avoidance is not considered, whereas larger values of t lead to a greater distance between nodes and edges. We generated three different layouts with varying t values (0, 1, and 2) while other parameters remained the same as REA-FM₁, and the results are shown in Figure 15.

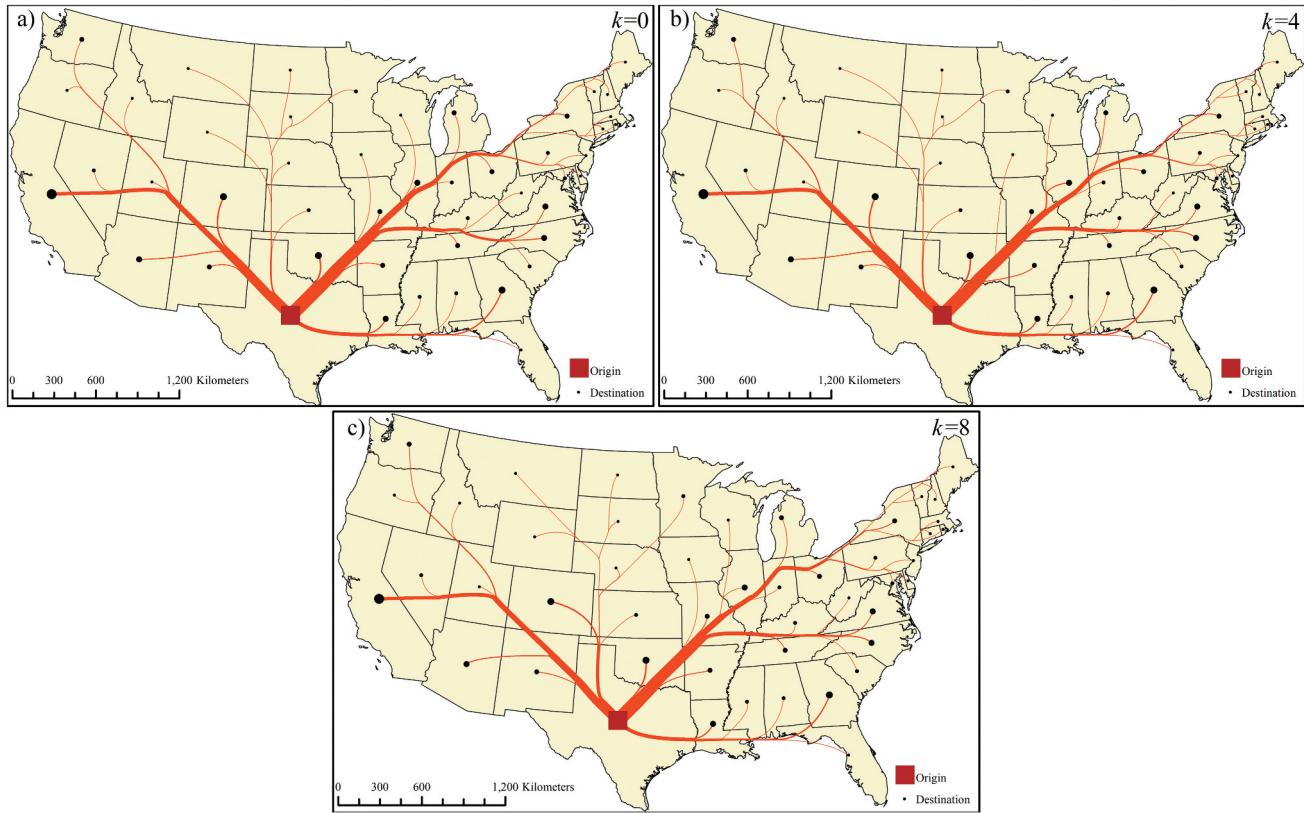


Figure 14. Flow maps that were generated with a) $k = 0$, b) $k = 4$, c) $k = 8$.

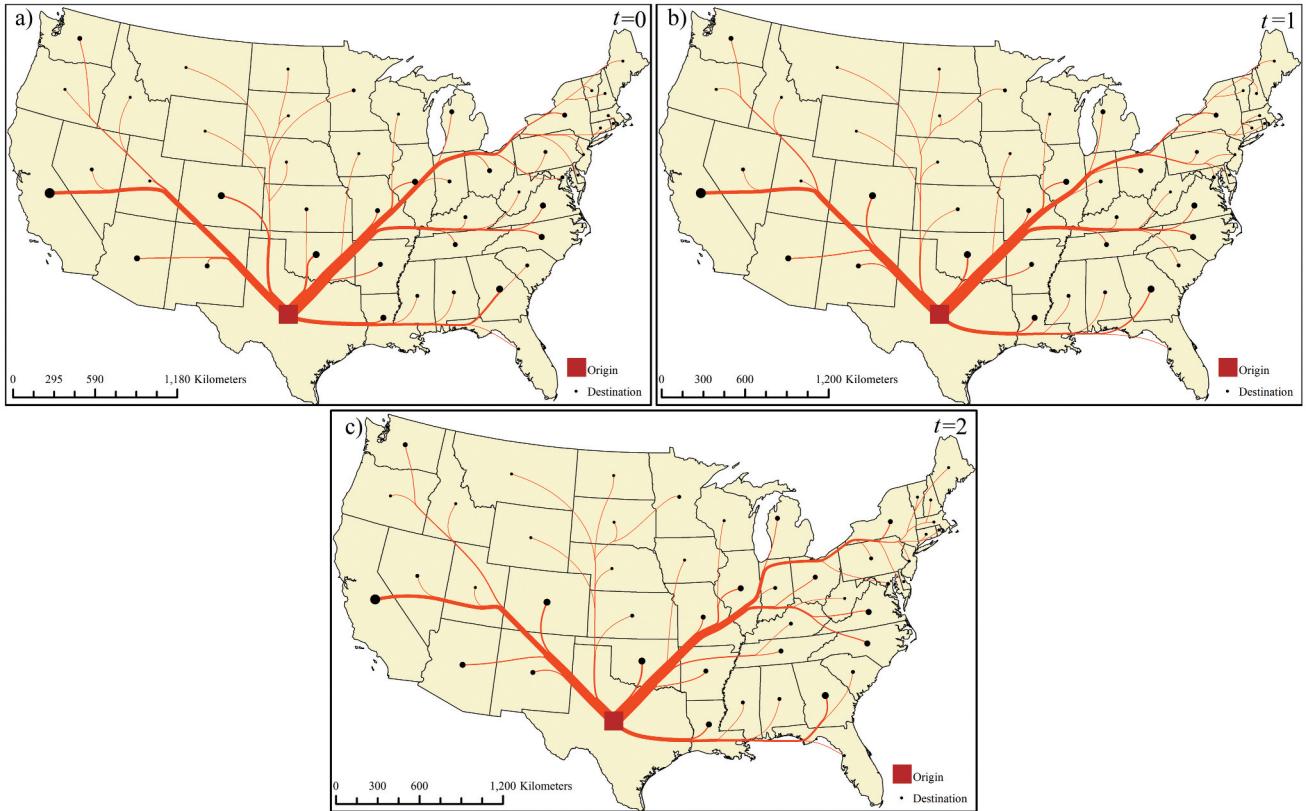


Figure 15. Flow maps that were generated with a) $t=0$, b) $t=1$, c) $t=2$.

From Figure 15, we observe that setting $t=0$ leads to three overlaps occurring between nodes and edges along with six nodes that are within a distance of 40×10^3 m from nearby edges. These observations indicate that $t=0$ cannot avoid node overlaps or edge crossings. If a larger value of t is set ($t=2$), then the number of nodes located within small distances ($<100 \times 10^3$ m) reduces from 13 at $t=1$ to 8 at $t=2$. However, increasing t values also results in higher total length, from 24.93×10^6 m at $t=1$ to 25.58×10^6 m at $t=2$. Furthermore, a larger t may also lead to more turns to keep a larger distance to nearby destination nodes, as shown in Figure 15c). Nonetheless, if users set $t>0$, no overlaps between nodes and edges will occur.

Overall, a suitable value for t (i.e. $t>0$) in practice can well avoid node overlaps or edge crossings. If users prefer larger distances between nodes and edges, choosing a larger value for t is recommended but it may also lead to an increase in total length.

7.3. Apply different ranges for the mapping space

In our proposed approach, it is essential to establish a defined range for the mapping space where the

flow map will be generated. In this section, we delve into an analysis of how various ranges can impact the outcomes. Specifically, we employed four distinct ranges, as outlined in Table 7, to create the flow maps, with the results presented in Figure 16. The statistical findings are detailed in Table 8. We utilized *CCount* to represent the number of cells encompassed within these ranges and *Time* for efficiency analysis.

As outlined in Table 8, when we do not take the experimental area into account while defining the mapping space range (as seen in *Range*₂'s absence), a larger range results in a longer total length (25.76×10^6 m for *Range*₁, 25.31×10^6 m for *Range*₃, and 21.80×10^6 m for *Range*₄), with fewer nodes found in proximity to their respective edges ($N(MD < 100)$ is 15 for *Range*₁, 16 for *Range*₃, and 18 for *Range*₄). However, this approach also consumes more time

Table 7. Different ranges for the mapping space.

Name	Descriptions
<i>Range</i> ₁	The envelope of the node-set.
<i>Range</i> ₂	The range in which only the cells that intersect with the experimental area are retained (the range applied in our baseline map)
<i>Range</i> ₃	The convex hull of the node set
<i>Range</i> ₄	The minimal alpha shape that contains the node-set

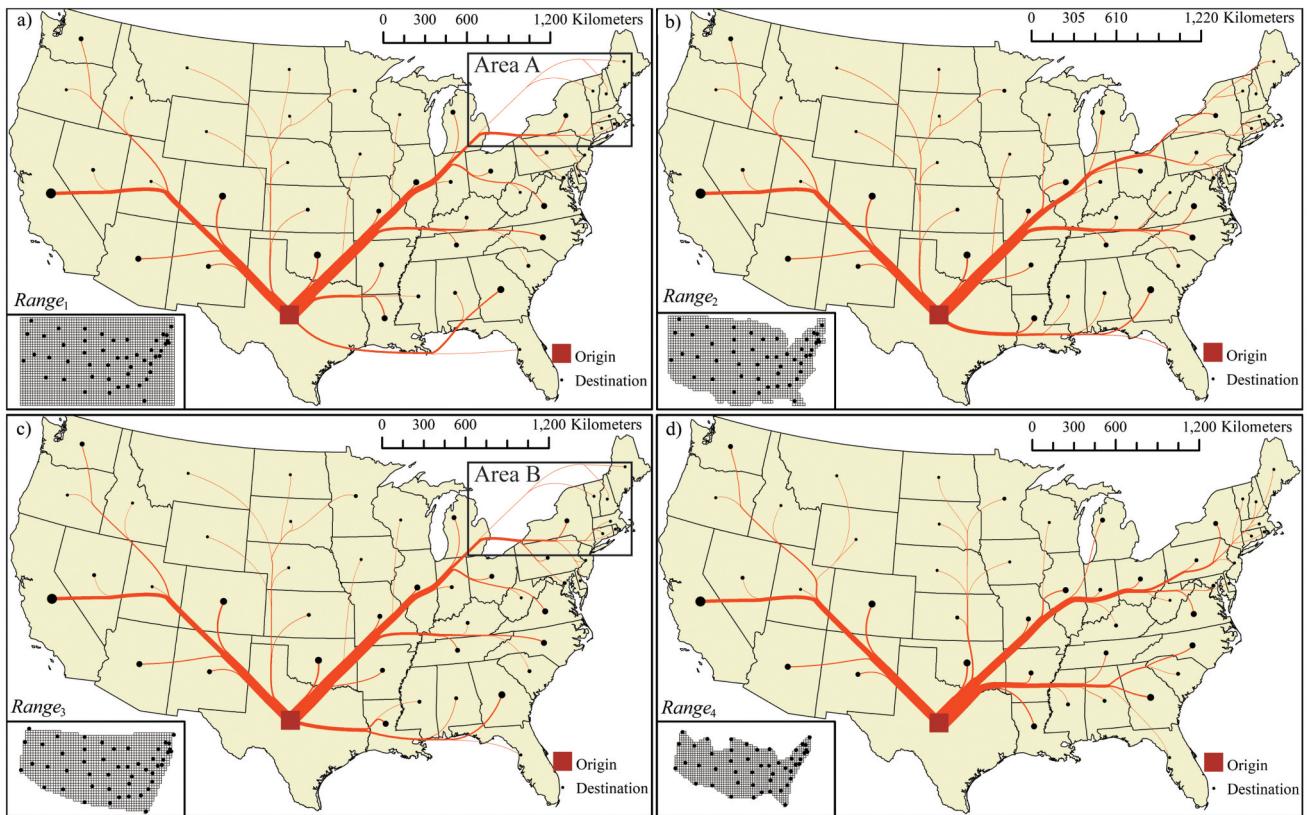


Figure 16. Flow maps that were produced with a) $Range_1$, b) $Range_2$, c) $Range_3$, and d) $Range_4$. $Range_2$ is a range that we applied in our baseline map.

Table 8. Quantitative assessment of flow map quality.

Measures	$Range_1$	$Range_2$	$Range_3$	$Range_4$
$TL (10^6 \text{ m}) \downarrow$	25.76	24.93	25.31	21.80
$MSI \uparrow$	0.698	0.717	0.703	0.687
$MD(10^3 \text{ m})$				
$MD_{\min} \uparrow$	22.3	22.3	22.3	22.3
$N(MD < 100) \downarrow$	15	13	16	18
$N(MD < 70) \downarrow$	9	6	10	12
$N(MD < 40) \downarrow$	1	1	2	3
$N(MD < 20) \downarrow$	0	0	0	0
CellCount	2268	1725	1505	1150
Time(s)	48.2	42.3	39.4	36.7

with a larger range (48.2s for $Range_1$, 42.3s for $Range_3$, and 36.7s for $Range_4$). Notably, due to the absence of the experimental area consideration, some flow paths generated under $Range_1$ and $Range_3$ extend beyond the experimental area boundaries, as evidenced in Areas A, and B in Figure 16a) and c).

To address this issue, we incorporate $Range_2$ into our approach, which ensures that all flow paths remain within the experimental area, as depicted in Figure 16b). However, it's essential to recognize that even with $Range_2$ in consideration, a larger range still consumes more time, although it does not necessarily increase the total length or the number of nodes near

edges. Specifically, for $Range_2$, the total length is $24.93 \times 10^6 \text{ m}$, and for $Range_3$, it is $25.31 \times 10^6 \text{ m}$, $N(MD < 100)$ is 13 for $Range_2$ and 16 for $Range_3$, while CCount is 1725 for $Range_2$ and 1505 for $Range_3$.

In summary, users seeking a faster flow map generation should opt for a smaller range. Those who prioritize greater spatial separation between nodes and edges may prefer a larger range, though this choice may lead to an increase in the total length. Furthermore, users who desire flow maps with paths restricted to the experimental area should incorporate the experimental area's boundaries in their range definition.

7.4. Limitation analysis

While our proposed approach can generate satisfactory flow maps in various mapping spaces, it also has some limitations that need to be addressed.

- (1) As the flow paths are calculated through an iterative process using a maze-solving algorithm, each result obtained meets quality criteria as much as possible but may not be optimal. To improve results, other strategies such as backtracking can be introduced in the future if unexpected outcomes occur.
- (2) Although our approach considers heterogeneous mapping spaces, nodes within the mapping data may distribute heterogeneously leading to different layouts when adapting the k -order surrounding definition for potential flow accumulation to different densities. An adaptive surrounding definition may be necessary for future works.
- (3) Generating a flow map for experiment data with one origin and 46 destinations takes approximately 42.3s. More efficient strategies should be developed going forward since several have been suggested previously for processing digital elevation model data that could potentially apply here too.

8. Conclusions

To generate flow maps that possess a natural appearance, originating from a single source to multiple destinations, we propose a novel approach called REA-FM. It models the mapping space using a DEM and employs an adapted maze-solving algorithm for river extraction to calculate flow paths within the flow maps. Experimental results indicate that the flow maps generated by our approach exhibit superior quality in terms of node placement, with nodes being effectively positioned away from the edges. Additionally, these maps demonstrate a reduced total length and avoidance of node overlaps or edge crossings. Moreover, the experimental findings establish the applicability of our approach to mapping spaces characterized by heterogeneity and obstacle areas. Furthermore, the proposed approach allows for intuitive control of quality criteria through user-defined parameter settings.

Future works will focus on (1) developing an adaptive algorithm capable of accommodating different distributions of mapping data, (2) enabling the visualization of flow maps with multiple origins and dynamic data, and (3) enhancing the efficiency of the algorithm to improve computational performance.

Acknowledgments

The authors wish to thank Dr. Tingzhong Huang for his help in data collection and Professor Yalong Yang for sharing his JavaScript code.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

The work was supported by the National Natural Science Foundation of China [62301063]; National Key Laboratory of Microwave Imaging Technology Grant [No number].

ORCID

Zhiwei Wei  <http://orcid.org/0000-0002-3494-3686>
Chunbo Liu  <http://orcid.org/0009-0001-0584-0856>

Data and code availability statement and data deposition

The data and code that support the findings of this study are all openly available, website is: <https://github.com/TrentonWei/FlowMap>.

References

- Barnes, R., Lehman, C., & Mulla, D. (2014). An efficient assignment of drainage direction over flat surfaces in raster digital elevation models. *Computers & Geosciences*, 62, 128–135. <https://doi.org/10.1016/j.cageo.2013.01.009>
- Buchin, K., Speckmann, B., & Verbeek, K. (2011). Flow map layout via spiral trees. *IEEE Transactions on Visualization and Computer Graphics*, 17(12), 2536–2544. <https://doi.org/10.1109/TVCG.2011.202>
- Callaghan, J. F., & Mark, D. M. (1984). The extraction of drainage networks from digital elevation data. *Computer Vision, Graphics and Image Processing*, 28(3), 323–344. [https://doi.org/10.1016/S0734-189X\(84\)80011-0](https://doi.org/10.1016/S0734-189X(84)80011-0)
- Debiasi, A., Simões, B., & De Amicis, R. (2014). Supervised force directed algorithm for the generation of flow maps. In V. Skala (Ed.), *WSCG 2014-Full Papers Proceedings* (pp. 193–202). Union Agency. http://wscg.zcu.cz/WSCG2013/_2013-WSCG-Full-proceedings.pdf
- Dong, W., Wang, S., Chen, Y., & Meng, L. (2018). Using eye tracking to evaluate the usability of flow maps. *ISPRS International Journal of Geo-Information*, 7(7), 281. <https://doi.org/10.3390/ijgi7070281>
- Garbrecht, J., & Martz, L. W. (1997). The assignment of drainage direction over flat surfaces in raster digital elevation models. *Canadian Journal of Fisheries and Aquatic Sciences*, 193(1–4), 204–213. [https://doi.org/10.1016/S0022-1694\(96\)03138-1](https://doi.org/10.1016/S0022-1694(96)03138-1)
- Guo, Q., Wei, Z., Wang, Y., & Lin, W. A. N. G. (2017). The method of extracting spatial distribution characteristics of buildings combined with feature classification and

- proximity graph. *Acta Geodaetica et Cartographica Sinica*, 46(5), 631–638. <https://doi.org/10.11947/j.AGCS.2017.20160374>
- Hengl, T. (2006). Finding the right pixel size. *Computers & Geosciences*, 32(9), 1283–1298. <https://doi.org/10.1016/j.cageo.2005.11.008>
- Huang, W., Eades, P., & Hong, S. H. (2014). Larger crossing angles make graphs easier to read. *Journal of Visual Languages & Computing*, 25(4), 452–465. <https://doi.org/10.1016/j.jvlc.2014.03.001>
- Jenny, B., Stephen, D. M., Muehlenhaus, I., Marston, B. E., Sharma, R., Zhang, E., & Jenny, H. (2017). Force-directed layout of origin-destination flow maps. *International Journal of Geographical Information Science*, 31(8), 1521–1540. <https://doi.org/10.1080/13658816.2017.1307378>
- Jenny, B., Stephen, D. M., Muehlenhaus, I., Marston, B. E., Sharma, R., Zhang, E., & Jenny, H. (2018). Design principles for origin-destination flow maps. *Cartography and Geographic Information Science*, 45(1), 62–75. <https://doi.org/10.1080/15230406.2016.1262280>
- Jenson, S. K., & Domingue, J. O. (1988). Extracting topographic structure from digital elevation data for geographic information system analysis. *Photogrammetric Engineering & Remote Sensing*, 54, 1593–1600. <https://pubs.usgs.gov/publication/70142175>
- Liu, X., Wang, N., Shao, J., & Chu, X. (2017). An automated processing algorithm for flat areas resulting from DEM filling and interpolation. *ISPRS International Journal of Geo-Information*, 6(5), 302. <https://doi.org/10.3390/ijgi6110376>
- Nocaj, A., & Brandes, U. (2013). Stub bundling and confluent spirals for geographic networks. In *International symposium on graph drawing* (pp. 388–399). Springer. https://doi.org/10.1007/978-3-319-03841-4_34
- Phan, D., Xiao, L., Yeh, R., Hanrahan, P., & Winograd, T. (2005). Flow map layout. In *IEEE Symposium on Information Visualization (InfoVis 2005)* (pp. 219–224). IEEE. <https://doi.org/10.1109/infvis.2005.1532150>
- Robinson, A. H. (1982). *Early thematic mapping in the history of cartography*. University of Chicago Press.
- Soille, P., Vogt, J., & Colombo, R. (2003). Carving and adaptive drainage enforcement of grid digital elevation models. *Water Resources Research*, 39(12), 1366. <https://doi.org/10.1029/2002WR001879>
- Sun, S. (2016). An automated spatial flow layout algorithm using triangulation, approximate Steiner tree, and path smoothing. In *AutoCarto. The 21st International Research Symposium on Computerbased Cartography and GIScience*. <http://www.cartogis.org/docs/proceedings/2016/Sun.pdf>.
- Sun, S. (2019). A spatial one-to-many flow layout algorithm using triangulation, approximate steiner trees, and path smoothing. *Cartography and Geographic Information Science*, 46(3), 243–259. <https://doi.org/10.1080/15230406.2018.1437359>
- Tobler, W. R. (1987). Experiments in migration mapping by computer. *Cartography and Geographic Information Science*, 14(2), 155–163. <https://doi.org/10.1559/152304087783875273>
- Tribe, A. (1992). Automated recognition of valley lines and drainage networks from grid digital elevation models: A review and a new method. *Canadian Journal of Fisheries and Aquatic Sciences*, 139(1–4), 263–293. [https://doi.org/10.1016/0022-1694\(92\)90206-B](https://doi.org/10.1016/0022-1694(92)90206-B)
- van den Elzen, S., & van Wijk, J. J. (2014). Multivariate network exploration and presentation: From detail to overview via selections and aggregations. *IEEE Transactions on Visualization and Computer Graphics*, 20(12), 2310–2319. <https://doi.org/10.1109/TVCG.2014.2346441>
- Viau, L., Sallaberry, A., Rodriguez, N., Girres, J.-F., & Poncelet, P. (2023). Polygon vector map distortion for increasing the readability of one-to-many flow maps. *International Journal of Geographical Information Science*, 37(6), 1288–1314. <https://doi.org/10.1080/13658816.2023.2190374>
- Wang, L., & Liu, H. (2006). An efficient method for identifying and filling surface depressions in digital elevation models for hydrologic analysis and modeling. *International Journal of Geographical Information Science*, 20(2), 193–213. <https://doi.org/10.1080/13658810500433453>
- Ware, C., Purchase, H., Colpoys, L., & McGill, M. (2002). Cognitive measurements of graph aesthetics. *Information Visualization*, 1(2), 103–110. <https://doi.org/10.1057/palgrave.ivs.9500013>
- Wei, Z., Ding, S., Xu, W., & Yan, F. (2023). Elastic beam algorithm for generating circular cartograms. *Cartography and Geographic Information Science*, 45(6), 539–555. <https://doi.org/10.1080/15230406.2023.2196732>
- Xu, K., Rooney, C., Passmore, P., Ham, D.-H., & Nguyen, P. H. (2012). A user study on curved edges in graph visualization. *Transactions on Visualization & Computer Graphics*, 18(12), 2449–2456. <https://doi.org/10.1109/TVCG.2012.189>
- Yang, Y., Dwyer, T., Jenny, B., Marriott, K., Cordeil, M., & Chen, H. (2019). Origin-destination flow maps in immersive environments. *Transactions on Visualization & Computer Graphics*, 25(1), 693–703. <https://doi.org/10.1109/TVCG.2018.2865192>
- Yu, W., Su, C., Yu, C., Wang, X. Z., Feng, C. J., & Zhang, X. C. (2014). An efficient algorithm for depression filling and flat-surface processing in raster DEMs. *IEEE Geoscience & Remote Sensing Letters*, 12, 424–428. <https://doi.org/10.1109/LGRS.2014.2324616>
- Zhang, H., Yao, Z., Yang, Q., Li, S., Baartman, J. E. M., Gai, L., Yao, M., Yang, X., Ritsema, C. J., & Geissen, V. (2017). An integrated algorithm to evaluate flow direction and flow accumulation in flat regions of hydrologically corrected DEMs. *Catena*, 151, 174–181. <https://doi.org/10.1016/j.catena.2016.12.009>
- Zhu, Q., Tian, Y., & Zhao, J. (2006). An efficient depression processing algorithm for hydrologic analysis. *Computers & Geosciences*, 32(5), 615–623. <https://doi.org/10.1016/j.cageo.2005.09.001>