Similarity Search of Spatiotemporal Scenario Data for Strategic Air Traffic Management

Junfei Xie *, Akhil Reddy Kothapally †
Texas A&M University at Corpus Christi, Corpus Christi, TX, 78412
Yan Wan ‡, Chenyuan He §
University of Texas at Arlington, Arlington, TX, 76019
Christine Taylor ¶, Craig Wanke ‖
The MITRE Corporation, McLean, VA, 22102
Matthias Steiner **
National Center for Atmospheric Research, Boulder, Colorado, 80307

Designing strategic air traffic management (ATM) solutions at a large spatiotemporal scale in real time is challenging, considering the range of uncertainties at the strategic time frame. Big data techniques have drawn increasing attentions to develop optimal ATM solutions to address these challenges. ATM data, such as convective weather spread and congestion propagation, represent a new data type called spatiotemporal scenario data, which has not been systematically studied. This new data type differs from the traditional spatiotemporal pointwise data in its unique spatiotemporally correlated spread patterns. As a step towards closing the loop of big data and real-time decision-making for ATM,[1] this paper introduces an effective similarity search algorithm for this new data type. This similarity search algorithm utilizes a multiresolution distance measure, which captures the difference between spatiotemporal scenarios. Unique properties of this distance measure are exploited to significantly reduce the computational cost associated with accessing and processing scenarios in a large database. Using real weather forecast datasets as the case study, we investigate feasibility of the proposed similarity search algorithm. Systematic parameter impact analysis is conducted through simulation studies, which provide guidelines for parameter selection. Comparative simulation studies validate the effectiveness and efficiency of the proposed similarity search algorithm for spatiotemporal scenario data.

I. Introduction

Strategic air traffic management (ATM), which plans traffic flows at a long look-ahead time (2-15 hours in advance), reallocates limited airspace resources in advance to enhance airspace safety.[2–4] Managing air traffic at this strategic time scale is challenging due to the existence of multifarious uncertainties,[5–7] such as convective weather, the main cause of traffic delays.[8,9] It is non-trivial to design effective ATM solutions that are robust to the wide range of uncertainties in real time, considering the large state-, decision-, and uncertainty- spaces. The new advance of big data techniques brings promising solutions to address these challenges. The rich information in the large historical ATM datasets, if exploited, can significantly benefit real-time decision-making for strategic ATM.[10,11]

*Assistant professor, Department of Computing Sciences. AIAA Member. Email: Junfei.Xie@tamucc.edu
†Master Student, Department of Computing Sciences.
‡Associate professor, Department of Electrical Engineering. AIAA Senior Member.
§Ph.D. Student, Department of Electrical Engineering.
¶Principle Simulation Modeling Engineer, M/S 450. AIAA Senior Member. MST Committee Member.
‖Senior Principal Aviation Systems Engineer, M/S 450. AIAA Associate Fellow.
**Director, Aviation Applications Program. AIAA Member. ASE Committee Member.

ATM data, such as convective weather spread and congestion propagation, represent a new data type called spatiotemporal scenario data.[12,13] Such type of data, typically generated from physical processes of spatiotemporal evolving dynamics, has not been systematically investigated. Unlike the spatiotemporal pointwise data[13–15] that describe point objects with static or varying spatial locations over time, the spatiotemporal scenario data are featured by spatiotemporally correlated spread patterns of changing shape, size, location and intensity. For instance, convective weather can appear/disappear, change in size and intensity at any time and any location. Such data are very commonly observed in natural and engineered systems. Other than weather, epidemic spread dynamics, cascading failures in power and other networked systems, can all be represented using this data type. With the growing interest of using big-data techniques to facilitate the management of modern large-scale dynamical systems,[16] innovative data analytics and processing tools designed specifically for this new data type are urgently needed. In this paper, we seek an efficient similarity search algorithm that allows quick retrieval of similar spatiotemporal scenarios from a database. Similarity search is a critical component of the end-to-end spatiotemporal scenario data-driven decision-making framework, which is based on the principle that similar scenarios will lead to similar management solutions. In particular, stored solutions tagged with similar spatiotemporal scenario data and designed through offline optimization approaches can be leveraged to significantly speed up the online optimal management design.

Similarity search relies on a distance/similarity measure to find objects that are most similar to the query objects. Studies on spatiotemporal scenario data are very limited in the literature. Recently, several papers[12,13,17–19] proposed distance measures to cluster spatiotemporal weather-impact scenarios to support strategic ATM. Paper[17] introduced three distance measures which aggregate values along the spatial or/and temporal dimension. An adjacency weighted distance measure was introduced in papers[18,19] which also aggregates values along the spatial or temporal dimension, with the consideration of additional neighboring information during the aggregation process. In these methods, the spatial and temporal dimensions are considered separately. The loss of the correlation information across spatial and temporal dimensions may lead to misleading similarity results.[13] To address this problem, we developed a novel multiresolution distance measure that can capture concurrent spatiotemporal correlations.[12,13] This method adopts the moving window concept,[20,21] and uses a 3-dimensional (3-D) spatiotemporal window of varying size to capture the spatiotemporally correlated spread patterns of spatiotemporal scenarios. The accuracy of this distance measure was validated through systematic analyses and comprehensive comparison studies. Guidelines of parameter selection were also provided through simulation studies on the impact of parameters in the distance generation algorithm. In this paper, we base on this multiresolution distance measure to explore effective similarity search methods for spatiotemporal scenario data.

To speed up searches, many search structures have been developed for pointwise data. The commonly used indexing structures include the B-tree,[22] R-tree,[23] KD tree,[24] cover tree[25] and their variants.[26,27] These structures utilize numerical constraints, such as the triangle inequality and bounding surfaces, to prune and select objects. Although they can find the objects that are most similar to the query object, the similarity search process can be computationally infeasible, when the database is large or the computation of similarity/distance is costly. To improve efficiency, various approximate indexing structures have been developed for pointwise data.[28–31] Examples include the BD-tree,[28] Locality Sensitive Hashing,[29,30] and spatial approximation sample hierarchy.[31] However, these indexing techniques cannot be simply combined with the multiresolution distance measure to perform similarity search for the spatiotemporal scenario data, due to the inherent non-linearity and complexity of the distance measure.

In this study, we develop a novel similarity search algorithm for spatiotemporal scenario data, based on the newly developed multiresolution distance measure. Through exploiting properties of the distance measure, this approach iteratively prunes the search space using bounds of the distances to significantly reduce the computational cost. Data access strategies are also developed to further enhance efficiency of the proposed similarity search algorithm for databases of large size. To validate and illustrate the performance of the proposed approaches, extensive numerical and simulation studies are conducted using real weather forecast data as the case study. Systematic analysis of parameters' impact on the query results is also performed, which provides guidelines for parameter selection.

In the rest of the paper, we first briefly describe the spatiotemporal scenario data and review the multiresolution distance measure for such type of data in Section II. We then formulate the similarity search problem for spatiotemporal scenario data, and describe the similarity search algorithm and data access strategies in Section III. Simulation studies, parameter selection guidelines, and performance analysis are presented in Section IV. Section V concludes the paper and discusses future works.

II. Review of the Multiresolution Distance Measure for Spatiotemporal Scenario Data

In this section, we first describe the spatiotemporal scenario data drawn from the dynamics of physical processes and its unique features. We then briefly introduce the multiresolution distance measure developed in our previous studies,[12,13] which is the first in the literature that truthfully quantifies the similarities of spatiotemporal scenario data.

II.A. Spatiotemporal Scenario Data

The spatiotemporal scenario data[12,13] is a new data type with spatiotemporally correlated spread patterns of changing shapes, sizes, locations and intensities over time. Figure 1 shows an example spatiotemporal scenario, which has 9 spatial cells with varying intensities over 3 time points. The intensity value of each spatial cell at each time point is visualized using colors with darker color indicating higher intensity. As we can see from the figure, the colored area changes shape, size, location and intensity over time.

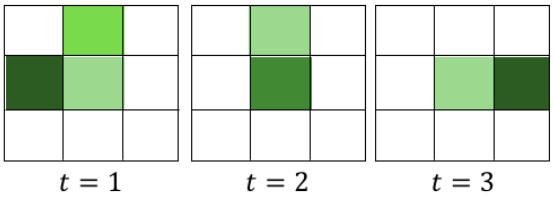


Figure 1. An example spatiotemporal scenario.

The spread dynamics of spatiotemporal scenario data make it significantly different from the traditional spatiotemporal pointwise data that have been widely studied in the literature, including 1) *events* (e.g., crimes) that have static spatial locations over time; 2) *georeferenced data* (e.g., sensor data) that have changing values but static spatial locations over time; and 3) *moving data* (e.g., trajectories of moving objects) that have changing spatial locations but static sizes and shapes over time. We may consider the spatiotemporal scenarios as georeferenced data if viewing the snapshots of spatial maps at different time points as independent images (see Figure 1), but such view loses the spatial spread information.

To describe a spatiotemporal scenario s_i, we let $g_z \in G$ be a spatial cell, $t_l \in T$ be a time point, and $I_{i,z,l} \geq 0$ be the intensity of scenario s_i at spatial cell g_z and time point t_l, where G and T represent the full set of spatial cells and time points in this scenario, respectively. Note that the full set of spatial cells G forms a map defining the spatial structure of a scenario, and a scenario is described by a set of snapshots of the spatial map G captured at a set of continuous time points T.

II.B. Multiresolution Distance Measure

The multiresolution distance measure developed in our previous studies[12,13] has several promising features that enable capturing the unique aspects of spatiotemporal scenario data. In particular, it captures the spread patterns of spatiotemporal scenarios through scanning scenarios at different resolutions using a 3-D spatiotemporal moving window of increasing size. The simultaneous scans along spatial and temporal dimensions retains the spatiotemporal correlations of physical processes. A variety of other features of this distance measure include the allowance of irregularly shaped spatial cells, heterogeneous contributions of spatial cells and time points, and boundary effects removal. Systematic analyses and a series of comparison studies[12,13] validate the accuracy of this distance measure.

Let us now describe the basics of the multiresolution distance measure. Consider two spatiotemporal scenarios s_i and s_j, each of which is composed of the same number of spatial cells $g_z \in G$ and time points $t_l \in T$. To capture the similarity of the two scenarios, a 3-D spatiotemporal moving window of increasing size is used to scan the two scenarios and calculate their distance. Figure 2 illustrates the scanning process, with spatiotemporal windows marked in red. The definition of the spatiotemporal window considers arbitrary shaped spatial cells. In particular, along the spatial dimension, a window $\phi_{z,w}$ of size w centered at cell g_z contains all cells within $(w-1)$ hops to g_z. Along the temporal dimension, a window $\phi_{l,h}$ of size h starting from time point t_l includes h consecutive time points. Larger windows indicate coarser resolutions. Based

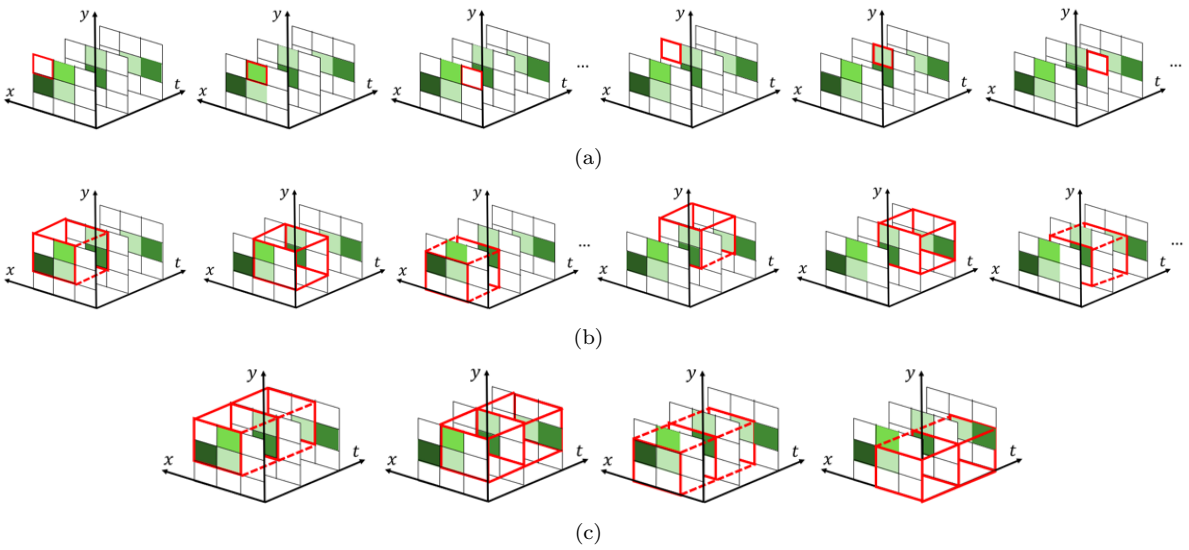


Figure 2. Illustration of the scanning process with window sizes equal to a) $w = 1$ and $h = 1$, b) $w = 2$ and $h = 2$, and c) $w = 2$ and $h = 3$.

on this definition, a distance $d_{i,j,w,h}$ between two scenarios, s_i and s_j, is computed after each scan using the spatiotemporal window of size w and h, by comparing the aggregated intensities within the windows, i.e.,

$$d_{i,j,w,h} = \sum_{\phi_{z,w} \in \Phi_w} \sum_{\phi_{l,h} \in \Phi_h} \frac{1}{|\phi_{z,w}||\phi_{l,h}||\Phi_h|} \left| \sum_{g_n \in \phi_{z,w}} \sum_{t_m \in \phi_{l,h}} \frac{\hat{I}_{i,n,m}}{\lambda_{n,w}\tau_{m,h}} - \sum_{g_n \in \phi_{z,w}} \sum_{t_m \in \phi_{l,h}} \frac{\hat{I}_{j,n,m}}{\lambda_{n,w}\tau_{m,h}} \right| \tag{1}$$

where $\hat{I}_{i,z,l} = \beta_{z,l} I_{i,z,l}$ is the weighted intensity of scenario s_i at cell $g_z \in G$ and time point $t_l \in T$, where $\beta_{z,l} > 0$ is a constant that weights important cells and time points.[13] In this study, we set $\beta_{z,l} = 1$, $\forall g_z \in G, t_l \in T$. Φ_w represents the full set of spatial windows of size w. Φ_h is the full set of temporal windows of size h. $\lambda_{z,w}$ and $\tau_{l,h}$ are the spatial contribution factor of cell g_z and the temporal contribution factor of time point t_l, respectively. These contribution factors are used to correct the boundary effect so that each spatial cell or time point has the same contribution to the distance calculation. Specifically, $\lambda_{z,w}$ and $\tau_{l,h}$ are computed using following equations:

$$\begin{aligned} \lambda_{z,w} &= \sum_{\phi_{n,w} \in \{\phi_{n,w} | g_z \in \phi_{n,w}\}} \frac{1}{|\phi_{n,w}|} \\ \tau_{l,h} &= \sum_{\phi_{m,h} \in \{\phi_{m,h} | t_l \in \phi_{m,h}\}} \frac{|T|}{|\phi_{m,h}||\Phi_h|} \end{aligned}$$

The overall distance $D_{i,j}$ between scenarios s_i and s_j is the weighted sum of $d_{i,j,w,h}$ obtained at different spatiotemporal resolutions. In particular,

$$D_{i,j} = \sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} d_{i,j,w,h} \frac{\delta_w \alpha_h}{\sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} \delta_w \alpha_h} \tag{2}$$

where w_{max} and h_{max} represent the coarsest spatial and temporal resolutions to evaluate, respectively. $\delta_w > 0$ and $\alpha_h > 0$ are weighting factors. They typically decrease with the increase of window sizes w and h, indicating that less contributions are associated with coarser resolutions. In this study, we set $\delta_w = e^{-\sigma(w-1)}$ and $\alpha_h = e^{-\rho(h-1)}$, where σ, $\rho \geq 0$.

Algorithm 1 summarizes the key steps to calculate the distance $D_{i,j}$, given $\beta_{z,l} = 1$, $\forall g_z \in G, t_l \in T$, and pre-calculated windows ($\phi_{z,w}$, $\phi_{l,h}$) and contribution factors ($\lambda_{z,w}$ and $\tau_{l,h}$). The computation cost of this distance measure is in the range of $O(|G||T|w_{max}h_{max}^2)$ and $O(|G|^2|T|w_{max}h_{max}^2)$.[13]

Algorithm 1: Multiresolution Distance Generation Algorithm

Input: Scenarios s_i and s_j
Output: Distance $D_{i,j}$
foreach *pair of spatial resolution* $w = 1 : w_{max}$ *and temporal resolution* $h = 1 : h_{max}$ **do**
  foreach *pair of spatial cell* $g_z \in G$ *and time point* $t_l \in T$ **do**
    Calculate the distance for windows $\phi_{z,w}$ and $\phi_{l,h}$ as $\left|\sum_{g_n \in \phi_{z,w}} \sum_{t_m \in \phi_{l,h}} \frac{\hat{I}_{i,n,m}}{\lambda_{n,w}\tau_{m,h}} - \sum_{g_n \in \phi_{z,w}} \sum_{t_m \in \phi_{l,h}} \frac{\hat{I}_{j,n,m}}{\lambda_{n,w}\tau_{m,h}}\right|$;
  end
  Calculate the distance $d_{i,j,w,h}$ for resolutions w and h using Equation (1);
end
Calculate the overall distance $D_{i,j}$ using Equation (2);

III. Similarity Search for Spatiotemopral Scenario Data

In this section, we describe the similarity search algorithm for spatiotemporal scenario data. The algorithm exploits bounds of the multiresolution distance measure to prune the search space and reduce the computational cost. We first formulate the problem and discuss the motivation underlying the proposed approach. The bounds of the distance measure are then explored, followed by the introduction of two data access strategies to improve the search speed in large databases. We then describe the similarity search algorithm and provide the pseudocode.

III.A. Problem Formulation and Motivation

Given a database S and a query scenario s_q, the similarity search query over S can be performed in the following two formats:

- *K-nearest neighbor query (K-NN):* find a set $S_c \subseteq S$ with size $|S_c| = \mathtt{K}$ that satisfies $D_{i,q} \leq D_{j,q}$ for all $s_i \in S_c$ and $s_j \in S \setminus S_c$, where $|A|$ is the cardinality of set A and $\setminus$ is the set difference operator.
- *Rank query:* find a set $S_c \subseteq S$ that satisfies $D_{i,q} \leq r$ for all $s_i \in S_c$, given a constant real value $r \geq 0$.

In this study, we focus on the K-NN query. We note that the proposed approach can be easily extended to solve the rank query problem.

The simplest way to retrieve similar scenarios from the database is to perform an exhaustive search, sequentially scanning the whole database, computing the distance $D_{i,q}$ for each scenario $s_i \in S$, and then picking scenarios with the smallest distance values.[32] This approach is computationally expensive, especially when the database is large and/or the calculation of the distance measure is costly. Utilizing a proper search structure can help reduce the number of scenarios to examine. However, as the multiresolution distance measure for spatiotemporal scenario data does not obey the triangle inequality, traditional indexing techniques cannot be directly applied. In addition, the computation of the multiresolution distance measure requires multiple rounds of scenario comparison at different resolutions, which further challenges the query processing and construction of efficient search structures.

To address the aforementioned challenges, our idea is to prune the search space after each resolution run of a scenario comparison to reduce the number of scenarios to examine, instead of waiting for all the resolutions to be examined. This is achieved by finding the upper and lower bounds of the distance measure, and using these bounds to filter out dissimilar scenarios. In particular, suppose $\underline{D}_{i,j}$ and $\overline{D}_{i,j}$ are the lower and upper bounds of $D_{i,j}$ respectively. Given a query scenario s_q, let $A_{\mathtt{K}} = \{s_i\} \subseteq S$, where $A_{\mathtt{K}}$ is the set of K scenarios with the smallest upper bound values, i.e., $\overline{D}_{i,q} \leq \overline{D}_{j,q}$ for all $s_i \in A_{\mathtt{K}}$ and $s_j \in S \setminus A_{\mathtt{K}}$. Let $M_{\mathtt{K}}$ be the largest upper bound value in $A_{\mathtt{K}}$ (corresponding to the K-th smallest upper bound value in S), i.e., $M_{\mathtt{K}} = \max \overline{D}_{i,q}$, $s_i \in A_{\mathtt{K}}$. We can then safely discard all scenarios $s_i \in S$ that satisfy the following condition:

$$\underline{D}_{i,q} > M_{\mathtt{K}}, \tag{3}$$

which excludes scenarios that are unlikely to be the query results, while retaining all the top K most similar scenarios. Similar procedures have been used in the No Random Access (NRA) algorithm[33] to perform top-K

queries for objects of sorted feature values. In the following section, we explore the lower bound $\underline{D}_{i,j}$ and upper bound $\overline{D}_{i,j}$ of the multiresolution distance measure for spatiotemporal scenario data.

III.B. Lower and Upper Bounds of the Multiresolution Distance Measure

The overall pairwise distance $D_{i,j}$ in Equation (2) is a weighted sum of distances $d_{i,j,w,h}$ computed at each spatial resolution w and temporal resolution h, where $1 \leq w \leq w_{max}$ and $1 \leq h \leq h_{max}$. Therefore, the bounds of $D_{i,j}$ are determined by the bounds of $d_{i,j,w,h}$. In our previous study,[13] we have proved that finer resolutions (smaller window sizes) always lead to larger $d_{i,j,w,h}$. In particular, $d_{i,j,w,h}$ satisfies the following inequalities:

$$\begin{aligned} d_{i,j,w,h} &\leq d_{i,j,w,1} \leq d_{i,j,1,1} \\ d_{i,j,w,h} &\leq d_{i,j,1,h} \leq d_{i,j,1,1} \end{aligned} \tag{4}$$

where $d_{i,j,1,1}$ is the distance calculated at the finest resolution. It is computed by comparing the intensity of each spatial cell at each time point using the following equation:

$$d_{i,j,1,1} = \sum_{g_z \in G} \sum_{t_l \in T} \frac{1}{|T|} \left| \hat{I}_{i,z,l} - \hat{I}_{j,z,l} \right| \tag{5}$$

Therefore, we have

$$\begin{aligned} D_{i,j} &= \sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} d_{i,j,w,h} \frac{\delta_w \alpha_h}{\sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} \delta_w \alpha_h} \\ &\leq \sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} d_{i,j,1,1} \frac{\delta_w \alpha_h}{\sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} \delta_w \alpha_h} \\ &= d_{i,j,1,1} \sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} \frac{\delta_w \alpha_h}{\sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} \delta_w \alpha_h} = d_{i,j,1,1} \end{aligned} \tag{6}$$

The upper bound of $D_{i,j}$ can be reduced when distances $d_{i,j,w,h}$ are calculated at coarser resolutions. In particular, suppose a spatiotemporal window of size $w = a$ and $h = b$ is used to scan the scenarios at the k-th iteration, where $1 \leq a \leq w_{max}$, $1 \leq b \leq h_{max}$, and $k \in \{1, 2, \ldots, w_{max} h_{max}\}$. We can gradually tighten the upper bound of $D_{i,j}$ using the following equation:

$$\overline{D}_{i,j}[k] = \begin{cases} d_{i,j,1,1} & \text{if } \ k = 1 \\ \overline{D}_{i,j}[k-1] + \frac{\delta_a \alpha_b}{\sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} \delta_w \alpha_h} (d_{i,j,a,b} - d_{i,j,1,1}) & \text{if } \ k \in \{2, 3, \ldots, w_{max} h_{max}\} \end{cases} \tag{7}$$

where $\overline{D}_{i,j}[k]$ represents the upper bound of $D_{i,j}$ computed at the k-th iteration. $\overline{D}_{i,j}[k] \leq \overline{D}_{i,j}[k-1]$ as $d_{i,j,a,b} \leq d_{i,j,1,1}$ according to Equation (4). The deduction of Equation (7) is provided in the Appendix in Section VI.B.

Similarly, we can prove that coarser resolutions (larger window sizes) always lead to smaller $d_{i,j,w,h}$, i.e.,

$$\begin{aligned} d_{i,j,w,h} &\geq d_{i,j,w,h^*} \geq d_{i,j,w^*,h^*} \\ d_{i,j,w,h} &\geq d_{i,j,w^*,h} \geq d_{i,j,w^*,h^*} \end{aligned} \tag{8}$$

where w^* and h^* represent the sizes of the largest spatial and temporal windows that cover the whole spatial and temporal spaces, respectively. d_{i,j,w^*,h^*} can be computed using the following equation

$$d_{i,j,w^*,h^*} = \frac{1}{|T|} \left| \sum_{g_z \in G} \sum_{t_l \in T} \hat{I}_{i,z,l} - \sum_{g_z \in G} \sum_{t_l \in T} \hat{I}_{j,z,l} \right| \tag{9}$$

which is the weighted difference of two scenarios' total intensities. The proof of Equation (8) can be found in the Appendix in Section VI.A. The overall pairwise distance $D_{i,j}$ then satisfies

$$\begin{aligned} D_{i,j} &= \sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} d_{i,j,w,h} \frac{\delta_w \alpha_h}{\sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} \delta_w \alpha_h} \\ &\geq \sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} d_{i,j,w^*,h^*} \frac{\delta_w \alpha_h}{\sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} \delta_w \alpha_h} \\ &= d_{i,j,w^*,h^*} \sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} \frac{\delta_w \alpha_h}{\sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} \delta_w \alpha_h} = d_{i,j,w^*,h^*} \end{aligned} \tag{10}$$

The lower bound d_{i,j,w^*,h^*} of $D_{i,j}$ can also be gradually tightened with distances $d_{i,j,w,h}$ at coarser resolutions. In particular,

$$\underline{D}_{i,j}[k] = \begin{cases} d_{i,j,w^*,h^*} + \frac{\delta_1 \alpha_1}{\sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} \delta_w \alpha_h} (d_{i,j,1,1} - d_{i,j,w^*,h^*}) & \text{if } k = 1 \\ \underline{D}_{i,j}[k-1] + \frac{\delta_a \alpha_b}{\sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} \delta_w \alpha_h} (d_{i,j,a,b} - d_{i,j,w^*,h^*}) & \text{if } k \in \{2, 3, \ldots, w_{max} h_{max}\} \end{cases} \tag{11}$$

where $w = a$ and $h = b$ at the k-th iteration. The deduction of Equation (11) is provided in the Appendix in Section VI.B.

III.C. Data Access Strategies

In this section, we introduce two data access strategies to expedite the similarity search in large databases based on the bounds of the multiresolution distance measure.

1. Indexing and Filter-Restart

Note that each spatiotemporal scenario s_i is characterized by $|G| \times |T|$ intensity values $I_{i,z,l}$, where $I_{i,z,l}$ is the intensity at the spatial cell $g_z \in G$ and time point $t_l \in T$. For a large database S, it will be very costly to access all scenarios and process all $|S| \times |G| \times |T|$ records. In this section, we design an efficient search structure to reduce the number of scenarios to examine.

Traditional indexing techniques cannot directly be applied for the spatiotemporal scenario data, as the multiresolution distance measure for spatiotemporal scenario data does not obey the triangle inequality. To address this challenge, one possible solution is to construct a hierarchical tree using the clustering algorithm.[12,13] The search is then performed within clusters that are closer to the query scenario. However, the construction of this indexing structure for a large database will be very costly, considering the complexity of the multiresolution distance measure for spatiotemporal scenario data.

Here we propose a search structure that is simpler and requires very low construction cost. The key idea is to pre-calculate and store the weighted total intensity $I_i = \sum_{g_z \in G} \sum_{t_l \in T} \hat{I}_{i,z,l}$ of each scenario $s_i \in S$ as an independent table denoted as I, and create indices based on I_i using traditional approaches such as B-tree. As the overall pairwise distance $D_{i,j}$ is lower bounded by d_{i,j,w^*,h^*}, which can be directly computed using table I, i.e., $d_{i,j,w^*,h^*} = |I_i - I_j|/|T|$ according to Equation (1), we can perform an initial search using this lower bound by only accessing table I and a small set of scenarios. Specifically, given a query scenario s_q, the search starts by first calculating its weighted total intensity I_q. We then adopt the Filter-Restart concept[32] described in detail below to retrieve an initial candidate set $S_c \subseteq S$, which contains all the top K most similar scenarios.

The selection of an initial candidate set S_c in the traditional *filtering* step is achieved using the range query. Specifically, S_c includes all scenarios that satisfy $|I_i - I_q| \leq I_{thrd}$, where I_{thrd} is the cutoff threshold that limits the number of scenarios to retrieve. The *restart* step then expands S_c by increasing the threshold I_{thrd}, if the initial set is not sufficient to answer the K-NN query. Whether S_c needs to be expanded is determined by comparing the lower and upper bounds of $D_{i,q}$ for all $s_i \in S_c$, which are computed using Equations (7) and (11) with $k = 1$. In particular, if scenario $s_i \in S_c$ that satisfies $\underline{D}_{i,q} > M_{\mathtt{K}}$ exits, which indicates that for all $s_j \in S \setminus S_c$, $\underline{D}_{j,q} > M_{\mathtt{K}}$, then the top-K most similar scenarios must be in set S_c. Otherwise, restart is performed to expand set S_c. Algorithm 2 summarizes the procedures to obtain set S_c using this approach.

Algorithm 2: Filter-Restart for Dense Spatiotemporal Scenario Databases

Input: Query s_q, database S, threshold I_{thrd}, and query coefficient K
Output: An initial candidate set $S_c \subseteq S$

```
Calculate I_q;
S_c ← {s_i}, where s_i ∈ S and |I_i − I_q| ≤ I_thrd;
foreach s_i ∈ S_c do
    Calculate D̄_{i,q} and D̲_{i,q} using Equations (7) and (11), where k = 1;
end
Determine the value of M_K;
while D̲_{i,q} ≤ M_K, ∀s_i ∈ S_c do
    Increase the value of I_thrd;
    Perform Steps 2-6.
end
S_c ← S_c \ {s_i}, where D̲_{i,q} > M_K and s_i ∈ S_c;
```

The threshold I_{thrd} can be determined by analyzing the statistics of data in the database.[32] For sparse databases including many scenarios that are relatively different from all other scenarios, it is not easy to find a proper threshold. In general, an improper threshold will lead to frequent restarts or a large set S_c being returned, and hence significantly degrade the query efficiency. To address this issue, we further propose a K-NN based Filter-Restart procedure that imposes restrictions on the number of scenarios to return. In particular, the initial search starts by first locating the scenario s_m with I_m closest to I_q, i.e., $|I_m - I_q| = \min |I_i - I_q|$, where $s_m, s_i \in S$. We then retrieve up to $f|S|$ scenarios that satisfy $I_i - I_q \geq I_m - I_q$, which are saved to set S_{cu}, and up to $f|S|$ scenarios that satisfy $I_i - I_q < I_m - I_q$, which are saved to set S_{cl}. Hence, $S_c = S_{cu} \cup S_{cl}$. f is a threshold that controls the size of S_c. To ensure $|S_c| \geq$ K, we let $f|S|$ larger than or equal to K. To determine whether a restart is needed, we calculate the bounds of $D_{i,q}$ for all $s_i \in S_c$ and derive the value of $M_{\mathtt{K}}$. After that, we perform a slightly different procedure as the I_{thrd}-based approach discussed above. In particular, we examine S_{cu} and S_{cl} respectively. If there exists a scenario s_i in S_{cu} (or S_{cl}) that satisfies $\underline{D}_{i,q} > M_{\mathtt{K}}$, we do not need to perform the restart, otherwise, S_{cu} (or S_{cl}) is expanded by increasing the threshold f. Algorithm 3 summarizes the procedures of this approach. Notice that both Algorithms 2 and 3 naturally support the approximate similarity search by removing the *restart* step.

2. Prioritizing Window Sizes

In this section, we introduce the other data access strategy, motivated by the Stream-Combine algorithm[33] developed for the top-K query, to further reduce the query processing time. Note that the number of additional scenarios that can be discarded at each iteration k is determined by how much the bounds of $D_{i,j}$ can be tightened. The faster the bounds are tightened, the quicker the search space is reduced and thus the earlier is the termination. Since the increase (decrease) of each lower (upper) bound at each iteration is directly affected by the weighting factor $\delta_w \alpha_h$ according to Equations (7) and (11), we prioritize window sizes w and h with large weights $\delta_w \alpha_h$. This is achieved by sorting pair (w, h) based on associated weights $\delta_w \alpha_h$. We denote the resulting sorted list of window sizes as $\mathcal{W} = \{(w[k], h[k])\}_{k=1}^{w_{max} h_{max}}$, where $w[k]$ and $h[k]$ represent the spatial and temporal window sizes at the k-th iteration respectively, and $\delta_{w[i]} \alpha_{h[i]} \geq \delta_{w[j]} \alpha_{h[j]}$, $\forall i < j,\ i, j \in \{1, 2, \ldots, w_{max} h_{max}\}$. Typically, larger window sizes, indicating coarser resolutions, have smaller weights and thus less contributions to the distance calculation. Here we always have $w[1] = 1$ and $h[1] = 1$.

3. Discussion

The query processing speed can be further increased by allocating more storage resources. Notice that the aggregated total intensity of each scenario s_i at a particular spatiotemporal resolution w and h, i.e., $\sum_{g_z \in G} \sum_{t_l \in T} \frac{\hat{I}_{i,z,l}}{\lambda_{z,w} \tau_{l,h}}$, can be pre-calculated and stored into the database. The distance $d_{i,j,w,h}$ can thus be easily computed, which is the weighted difference of the aggregated total intensities (see Equation (1)). Although this strategy will help reduce the time for computing $d_{i,j,w,h}$ online, it requires large amount of

Algorithm 3: Filter-Restart for Sparse Spatiotemporal Scenario Databases

Input: Query s_q, database S, threshold f, and query coefficient $\mathtt{K}$
Output: An initial candidate set $S_c \subseteq S$

1 Calculate I_q;
2 Find $s_m \in S$ with $|I_m - I_q| = \min |I_i - I_q|$, where $s_i \in S$;
3 $S_{cu} \leftarrow \{s_i\}$, where $s_i \in S$, $I_i - I_q \geq I_m - I_q$ and $|S_{cu}| \leq \max\{f|S|, \mathtt{K}\}$;
4 $S_{cl} \leftarrow \{s_i\}$, where $s_i \in S$, $I_i - I_q < I_m - I_q$ and $|S_{cl}| \leq \max\{f|S|, \mathtt{K}\}$;
5 $S_c \leftarrow S_{cu} \cup S_{cl}$;
6 **foreach** $s_i \in S_c$ **do**
7 | Calculate $\overline{D}_{i,q}$ and $\underline{D}_{i,q}$ using Equations (7) and (11), where $k = 1$;
8 **end**
9 Determine the value of $M_{\mathtt{K}}$;
10 **while** $\underline{D}_{i,q} \leq M_K$, $\forall s_i \in S_{cu}$ *or* $\underline{D}_{i,q} \leq M_K$, $\forall s_i \in S_{cl}$ **do**
11 | Increase the value of f;
12 | **if** $\underline{D}_{i,q} \leq M_K$, $\forall s_i \in S_{cu}$ **then**
13 | | Perform Step 3;
14 | **end**
15 | **if** $\underline{D}_{i,q} \leq M_K$, $\forall s_i \in S_{cl}$ **then**
16 | | Perform Step 4;
17 | **end**
18 | Perform Steps 5-9;
19 **end**
20 $S_c \leftarrow S_c \setminus \{s_i\}$, where $\underline{D}_{i,q} > M_{\mathtt{K}}$ and $s_i \in S_c$;

storage space. In particular, if a single scenario requires an amount of storage resources equal to B, storing the aggregated total intensities for all scenarios at all resolution levels requires around $B|S|(w_{max}h_{max} - 1)$ additional storage space, which can be very expansive. In this study, we do not consider this strategy.

III.D. Algorithm Description

In this section, we describe the multiresolution distance based similarity search algorithm that incorporates the aforementioned features for spatiotemporal scenario data. The key idea of this algorithm is to use the bounds of the distance measure tightened at each iteration to progressively prune the search space, so as to quickly find the top $\mathtt{K}$ most similar scenarios. The pseudocode is provided in Algorithm 4. In particular, the algorithm starts by performing the Filter-Restart procedures described in Section III.C to obtain an initial candidate set S_c (Line 1). Whether to use Algorithm 2 or Algorithm 3 depends on the sparsity of the database. With an initial candidate set S_c obtained, this set is then evaluated at multiple spatiotemporal resolutions in a sorted order specified by $\mathcal{W}$ to find the top $\mathtt{K}$ most similar scenarios. Specifically, at each spatiotemporal resolution, the bounds of $D_{i,q}$ for all $s_i \in S_c$ are first updated (Lines 3-7), which are then used to reduce the size of the candidate set S_c (Lines 9-11). The iteration terminates when the top $\mathtt{K}$ most similar scenarios are found or all resolutions are evaluated.

IV. Simulation Studies

In this section, we conduct a series of simulation studies using real weather forecast datasets to illustrate the use and performance of the multiresolution distance based similarity search algorithm for spatiotemporal scenario data. This algorithm is first prototyped using Matlab, with data directly imported from local files, to analyze the parameters' impact on the query results and evaluate the performance of proposed approaches. This algorithm is then implemented using Java with data stored in a relational database. The Dell Precision Tower 7810 Workstation with Xeon® CPU of 2.4GHz and 32GB memory is used to run all simulations.

Algorithm 4: Multiresolution Distance-based Similarity Search Algorithm

Input: Query s_q, Database S, and K
Output: A set of scenarios $S_c \subseteq S$ of size K that are most similar to query s_q

Apply Algorithm 2 or Algorithm 3 to find an initial candidate set S_c;
for $k = 2$ *to* $w_{max}h_{max}$ **do**
  foreach $s_i \in S_c$ **do**
    Calculate $d_{i,q,w[k],h[k]}$ using Equation (1), where $(w[k], h[k]) \in \mathcal{W}$;
    Calculate $\overline{D}_{i,q}[k]$ using Equation (7);
    Calculate $\underline{D}_{i,q}[k]$ using Equation (11);
  end
  if $|S_c| > K$ **then**
    Determine the value of M_K;
    Remove all scenarios s_i that satisfy $\underline{D}_{i,q}[k] > M_K$ from the candidate set S_c;
  else
    Exit from the **for** loop;
  end
end
if $|S_c| > K$ **then**
  $S_c \leftarrow$ K scenarios selected from S_c that have the smallest upper bound values $\overline{D}_{i,q}[k]$;
end

IV.A. Dataset Description

In the simulation studies, we use a precipitation dataset generated from an ensemble weather forecasting tool called short-range ensemble forecast (SREF) system, which produces 21 ensemble forecasts per hour to capture the uncertainty of weather forecast and has been used for air traffic decision support.[34–36] This dataset, which was also used in our previous studies,[12,13] is reported at the 40×40 km^2 resolution and processed to be sector-specific. It consists of 62 days of weather forecast with each day characterized by 21 SREF ensemble scenario members, leading to $21 \times 62 = 1302$ weather scenarios. Each weather scenario consists of $|G| = 151$ spatial cells covering 3 airspace centers: Cleveland, Chicago and Indianapolis, and $|T| = 12$ time points corresponding to 12 hours (from 10:00Z to 21:00Z). Therefore, each scenario s_i is described by 151×12 records $\{g_z, t_l, I_{i,z,l}\}$, where $I_{i,z,l}$ is the precipitation intensity of scenario s_i at spatial cell g_z and time point t_l. Figure 3 visualizes one example weather scenario, with darker colors indicating higher precipitations.

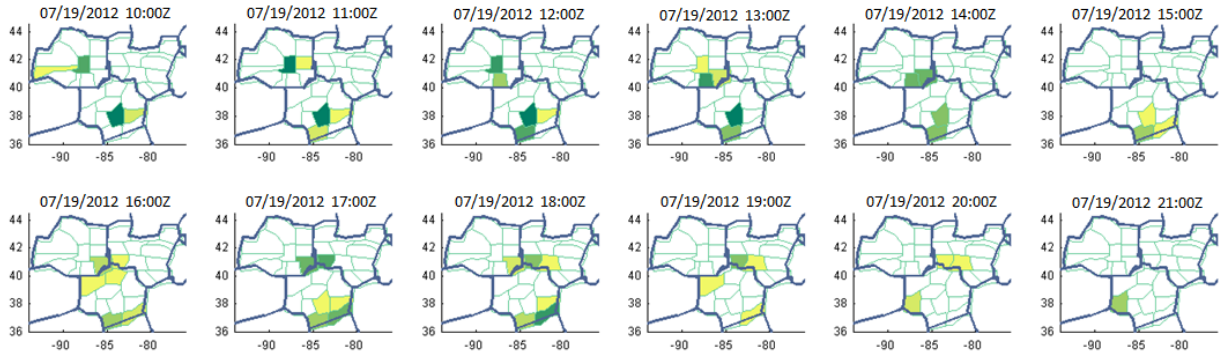


Figure 3. Visualization of an example spatiotemporal weather scenario.

IV.B. Parameter Impact Analysis

In this study, we use a small dataset S of $|S| = 124$ scenarios to analyze the impact of parameters in the proposed similarity search algorithm. This dataset is sampled from the larger dataset described in Section

IV.A, by randomly picking 2 out of 21 SREF ensemble scenarios for each day. The proposed similarity search algorithm has the following parameters: 1) the maximum spatial and temporal window sizes to evaluate, w_{max} and h_{max}, 2) spatial and temporal window size weighting factors, δ_w and α_h, 3) query coefficient K, and 4) the threshold I_{thrd} in Algorithm 2 or f in Algorithm 3 for determining the size of initial candidate set S_c. Due to the lack of real weather scenario data, we implement Algorithm 3 in all simulation studies, with f increasing by f_0 at Step 11, where f_0 is the initial value of the threshold. In the rest of this section, we analyze the impact of w_{max}, h_{max}, δ_w, α_h, and K. The threshold f, which impacts the query efficiency, is investigated in Section IV.C on performance evaluation.

1. Impact of maximum spatial and temporal window sizes, w_{max} and h_{max}

The parameters w_{max} and h_{max} have a direct impact on the accuracy of the multiresolution distance measure and thus the query results. Larger w_{max} and h_{max} lead to a more powerful distance measure that can capture patterns at more spatiotemporal resolutions and thus generate query results of higher confidence. However, larger w_{max} and h_{max} also lead to more scenario comparisons at more resolutions and thus increase the computational cost. Therefore, the selection of w_{max} and h_{max} needs to balance between the efficiency and accuracy of the query results. To achieve this tradeoff, we conduct the following studies to understand the impact of w_{max} and h_{max} on the distance measure and the query results.

We first conduct a local sensitivity analysis to illustrate the impact of a small change of w_{max} and h_{max} on the change of the overall pairwise scenario distance $D_{i,j}$. In particular, the sensitivity of $D_{i,j}$ to the change of w_{max} evaluated at a particular value $w_{max} = w_0$ is measured by

$$\left.\frac{\partial D_{i,j}}{\partial w_{max}}\right|_{w_0} = \frac{D_{i,j}|_{w_0} - D_{i,j}|_{w_0-1}}{w_0 - (w_0 - 1)}, \tag{12}$$

with values of the other parameters being fixed. $A|_{w_0}$ represents that function A is evaluated at w_0. The sensitivity of $D_{i,j}$ to the change of h_{max} evaluated at a particular value $h_{max} = h_0$ is measured in a similar way. Figure 4(a) shows the sensitivity of $D_{i,j}$ with respect to w_{max} (with $h_{max} = 1$) and with respect to h_{max} (with $w_{max} = 1$). Each sensitivity value is obtained by averaging the results for all possible pairs of scenarios. The weighting factors are set to $\delta_w = e^{-0.8(w-1)}$ and $\alpha_h = e^{-0.8(h-1)}$. As we can see from the figure, with the increase of w_{max} or h_{max}, $D_{i,j}$ becomes less sensitive to the changes of their values. This indicates that relatively small window sizes are sufficient to capture the difference between two scenarios.

To understand the impact of w_{max} (or h_{max}) on the query results, we evaluate the changes of query results with the increase of w_{max} (or h_{max}), by comparing the results with those obtained at the upper bound of w_{max} (or h_{max}), which generate the most trustworthy query results. In particular, w_{max} is upper bounded by 13, as a spatial window of size $w = 13$ is needed to cover the complete spatial map G. h_{max} is upper bounded by 12, as a temporal window of size $h = 12$ is needed to cover the entire time sequence T. To measure the difference of the query results obtained at an arbitrary w_{max} (or h_{max}) and those obtained at its upper bound, denoted as S_{c1} and S_{c2} respectively, we adopt the *agreement* metric described by the following equation

$$\text{Agreement} = \frac{\text{Number of scenarios common to both } S_{c1} \text{ and } S_{c2}}{\texttt{K}} \tag{13}$$

A low agreement value represents that the query results obtained at a particular $w_{max} < 13$ (or $h_{max} < 12$) are very different from those obtained at $w_{max} = 13$ (or $h_{max} = 12$), indicating a low level of confidence. The blue solid line in Figure 4(b) shows the change of agreement as w_{max} grows with $h_{max} = 1$, which quickly converges. This reflects that a small w_{max} is sufficient to retrieve similar scenarios with high confidence. A similar conclusion for the selection of h_{max} can be reached by observing the red dashed line in Figure 4(b), which shows the change of agreement as h_{max} increases with $w_{max} = 1$. In both experiments, $\texttt{K} = 10$, $\delta_w = e^{-0.8(w-1)}$ and $\alpha_h = e^{-0.8(h-1)}$. Each agreement value is obtained by averaging the results from $|S| = 124$ queries, where the top K scenarios from set $S \setminus s_i$ that are most similar to scenario $s_i \in S$ are retrieved in the i-th query, $i \in \{1, 2, \ldots, |S|\}$.

The comparison between the blue solid lines and the red dashed lines in Figures 4(a)-4(b) indicates that the overall pairwise distance $D_{i,j}$ and the query results are less sensitive to temporal resolutions than spatial resolutions. This can be more clearly observed in Figure 4(c) that visualizes the change of agreement with the concurrent increase of w_{max} and h_{max}, where the agreement is evaluated by comparing the query

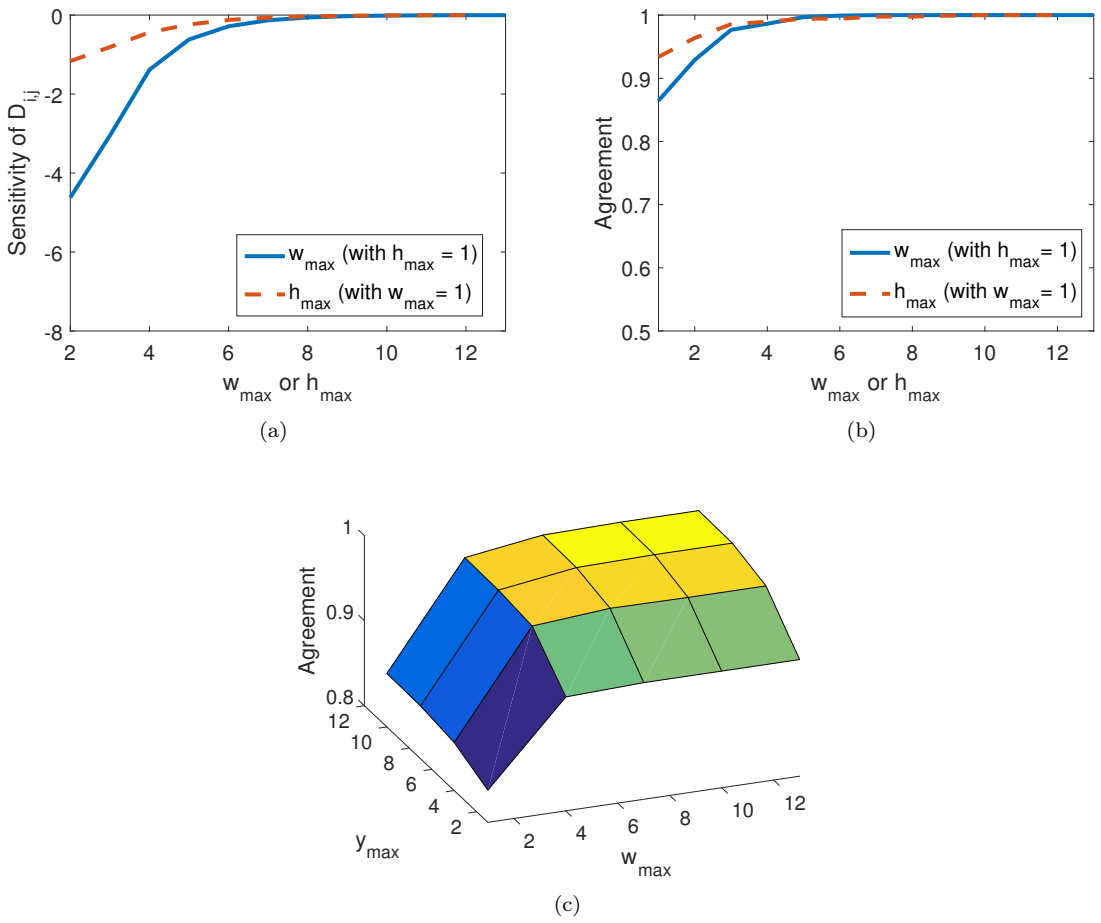


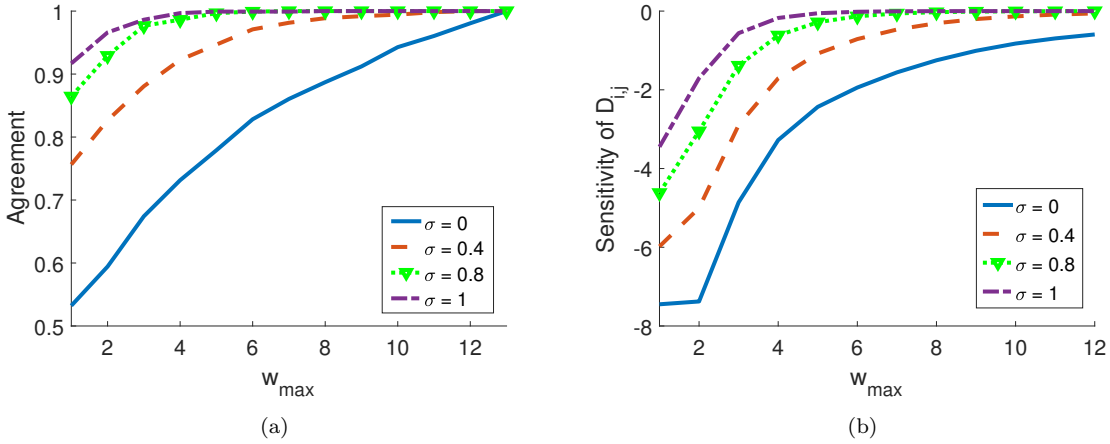
Figure 4. a) Sensitivity of $D_{i,j}$ to the changes of w_{max} (or h_{max}). Agreement of query results at increasing values of b) w_{max} (or h_{max}) and c) w_{max} and h_{max} simultaneously.

results obtained at each pair of (w_{max}, h_{max}) and those obtained at $(w_{max} = 13, h_{max} = 12)$. Therefore, we typically select $w_{max} > h_{max}$.

2. Impact of spatial and temporal window size weighting factors, δ_w and α_h

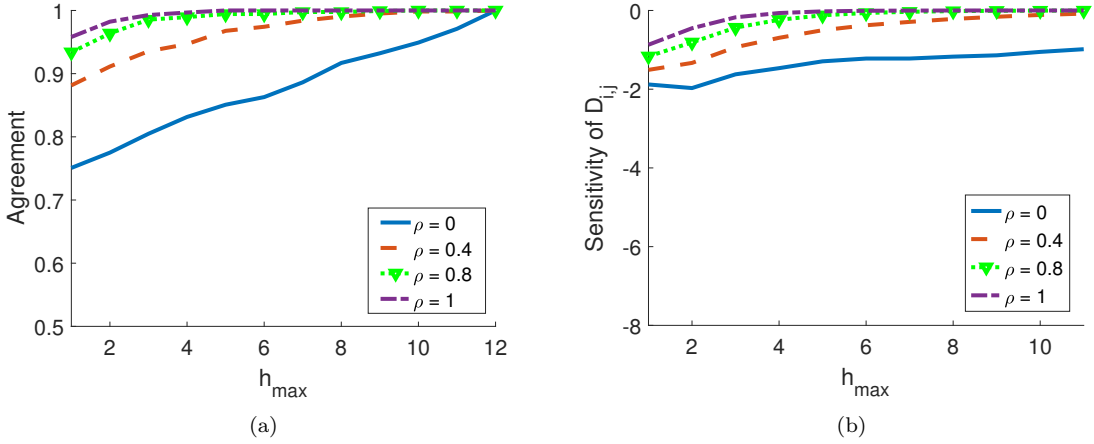
The window size weighting factors, $\delta_w = e^{-\sigma(w-1)}$ and $\alpha_h = e^{-\rho(h-1)}$, determine the contribution of distance $d_{i,j,w,h}$ for resolutions w and h to the overall pairwise distance $D_{i,j}$. Coarser resolutions (larger window sizes) usually have less contributions (smaller weights). To analyze the impact of the spatial window size weighting factor δ_w on the query results, we evaluate the changes of the query results with the increase of σ. Figure 5(a) shows the trajectories of the agreement between query results obtained at increasing values of w_{max} and those obtained at $w_{max} = 13$ for different values of σ, with $h_{max} = 1$, $\rho = 0$ and $\mathtt{K} = 10$. As shown in this figure, the time required to reach an agreement of 1 descreases with the increase of σ. This is because the contributions of $d_{i,j,w,h}$ at coarse resolutions (large w) are smaller for larger σ, leading to quicker convergence of $D_{i,j}$, as illustrated in Figure 5(b). In the special case when $\sigma = 0$, both the query results (Figure 5(a)) and the overall pairwise scenario distance $D_{i,j}$ (Figure 5(b)) fail to converge, as $d_{i,j,w,h}$ evaluated at different spatial resolutions w contribute equally to $D_{i,j}$. This study suggests that the selection of w_{max} needs to take the spatial weighting factors into the consideration. Specifically, if a small σ is used, a relatively large w_{max} is required to capture the difference between two scenarios and to achieve higher query accuracy.

Similar experiments have been conducted to analyze the impact of the temporal window size weighting factor α_h on the query performance. The results are shown in Figure 6 with $w_{max} = 1$, $\sigma = 0$, and $\mathtt{K} = 10$,



(a) (b)

Figure 5. a) Agreement of query results at increasing values of w_{max} and σ. b) Sensitivity of $D_{i,j}$ to the changes of w_{max} at different values of σ.



(a) (b)

Figure 6. a) Agreement of query results at increasing values of h_{max} and ρ. b) Sensitivity of $D_{i,j}$ to the changes of h_{max} at different values of ρ.

from which similar observations and conclusions can be drawn. The lines in Figure 6 are flatter than those in Figure 5 as temporal resolutions have less impact than spatial resolutions on the overall pairwise distance and thus the query results. In the following studies, we choose $\sigma = \rho = 0.8$ and thus $\delta_w = e^{-0.8(w-1)}$ and $\alpha_h = e^{-0.8(h-1)}$.

3. Impact of query coefficient K

The query coefficient K determines the number of similar scenarios to retrieve from the database. To understand whether the selection of w_{max} and h_{max} needs to take K into the consideration, we evaluate the changes of query results with the increase of w_{max} and h_{max} at different values of K. As shown in Figure 7, given w_{max} and h_{max}, the query results are relatively robust to the changes of K, indicating that approximately the same query accuracy can be achieved at different values of K using the same w_{max} and h_{max}. An exception happens at $K = 1$, where the query results converge quickly with the increase of w_{max} or h_{max}. This suggests that small w_{max} and h_{max} are sufficient to identify the most similar scenario.

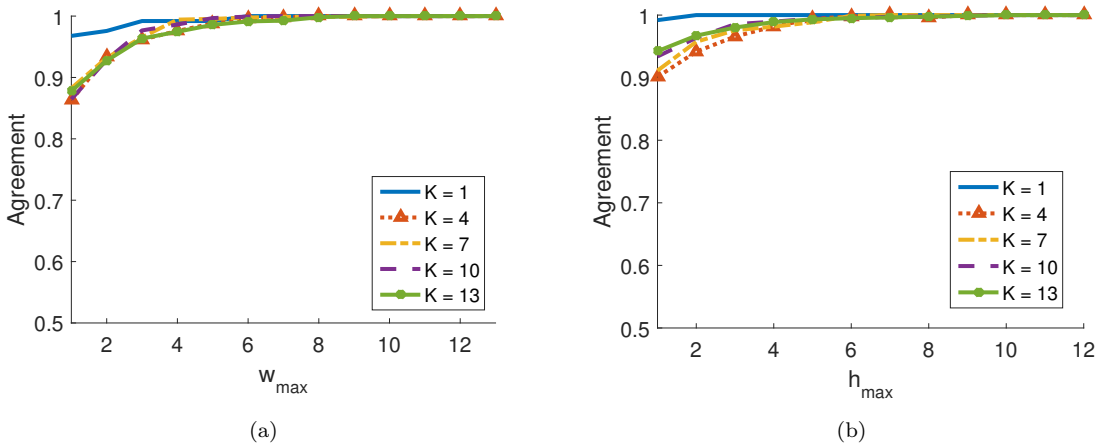


Figure 7. Agreement of query results at increasing values of a) w_{max} and b) h_{max} for different values of K.

IV.C. Comparative Performance Studies

In this section, we conduct various comparative studies to evaluate and illustrate the performance of the proposed similarity search algorithm.

1. *Effectiveness of the multiresolution distance based similarity search algorithm*

The effectiveness of a similarity search algorithm relies on the accuracy of the underlying distance measure. As we have justified in our previous work,[13] the multiresolution distance measure quantifies the similarity between spatiotemporal scenarios with high accuracy. Based on this distance measure, the proposed similarity search algorithm always finds the exact top K most similar scenarios, since it only eliminates scenarios that are unlikely to be the query results at each iteration. The bounds of $D_{i,j}$ used to prune the search space cover all top K most similar scenarios. They are tightened at each iteration and finally converge to the exact distance value $D_{i,j}$, as illustrated in Figure 8(a).

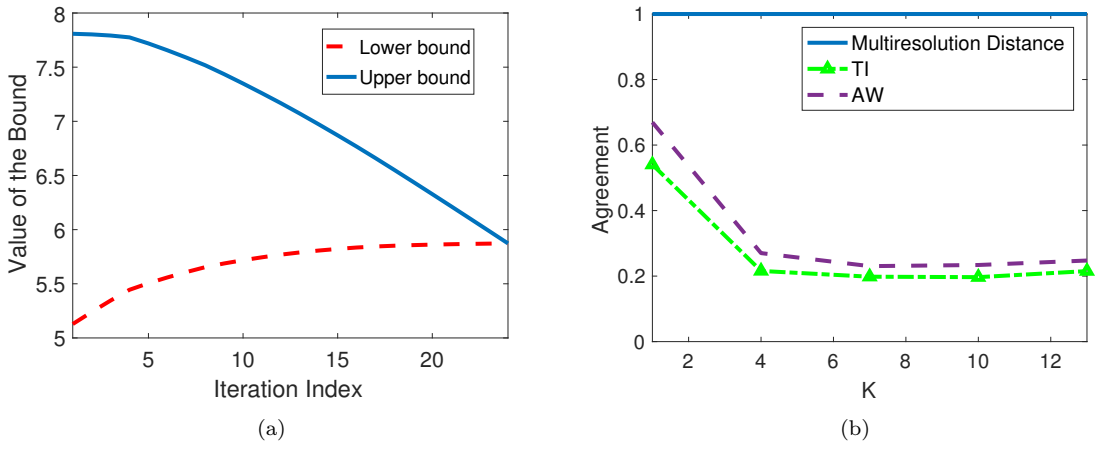


Figure 8. a) Trajectories of upper and lower bounds for the distance between two randomly selected scenarios. b) Agreement of query results obtained using different distance measures.

To illustrate the effectiveness of the proposed similarity search algorithm, we compare our algorithm with two alternative exhaustive search-based approaches that use 1) the total intensity-based (TI) distance measure that aggregates all intensity values,[17] and 2) the adjacency weighted (AW) distance measure[18,19] respectively. Both exhaustive search-based approaches compare the query scenario with each scenario in

the database, and pick the top K scenarios with the smallest distance values. As this is the first work that investigates the similarity search for spatiotemporal scenario data, we are unable to find existing similarity search methods that can be directly applied for such data type. The small dataset of $|S| = 124$ scenarios described in Section IV.B is then used to conduct experiments, with the parameters of our similarity search algorithm set to $w_{max} = 4$, $h_{max} = 3$, $\delta_w = e^{-0.8(w-1)}$, $\alpha_h = e^{-0.8(h-1)}$, and $f = 0.1$.

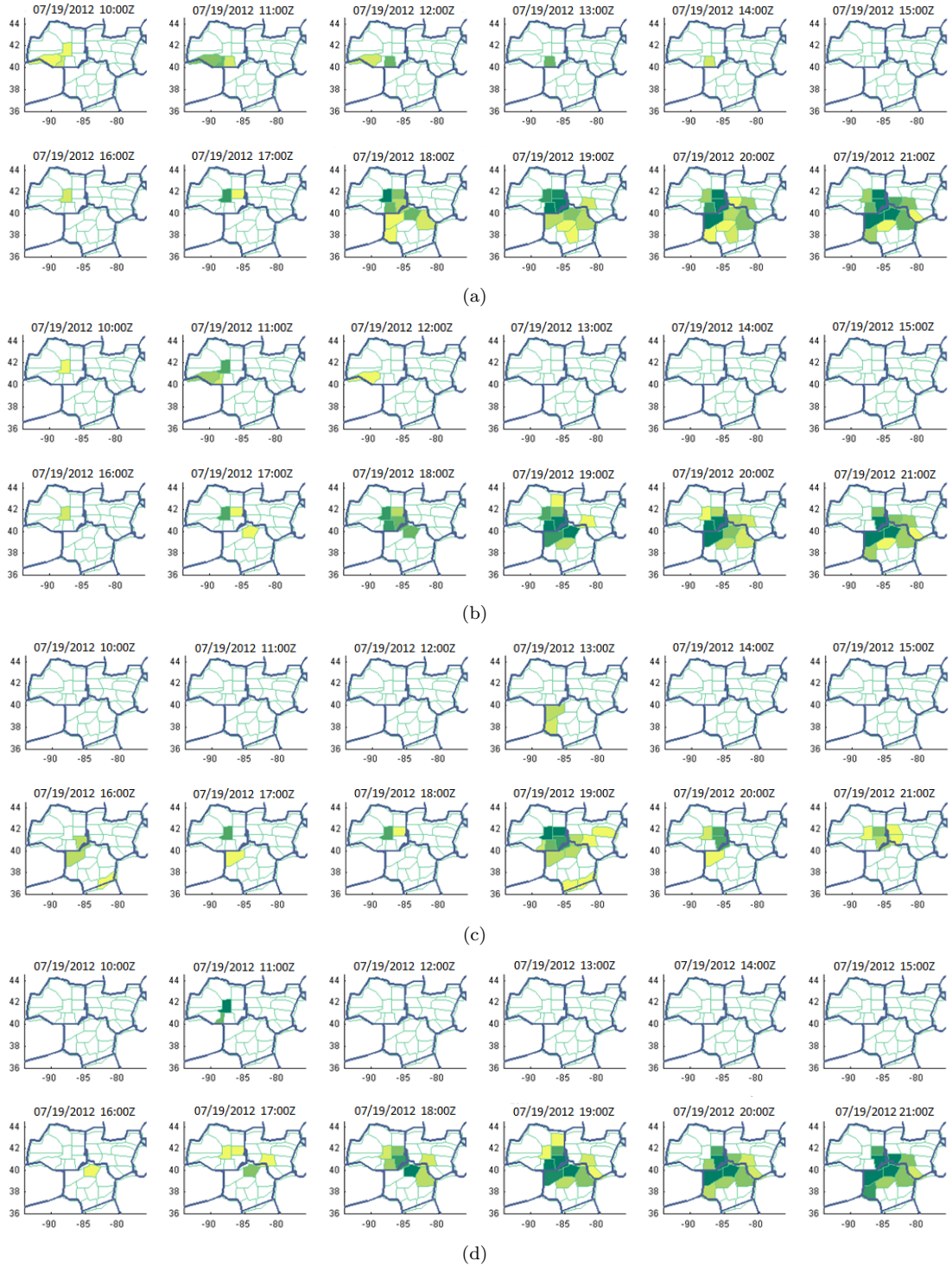


Figure 9. Snapshots of scenarios a) 12, b) 14, c) 10, and d) 15.

In the first experiment, we run the three algorithms to find the scenario that is most similar (i.e., $\mathtt{K} = 1$) to scenario 12. The scenarios retrieved by our algorithm, TI based approach, and AW based approach are 14, 10, and 15, respectively. Figure 9 visualizes these scenarios. As we can see from the figure, the query scenario 12 (Figure 9(a)) is very similar to scenario 14 (Figure 9(b)) found by our algorithm, both of which demonstrate a developing precipitation over the 12h span. However, the intensity in scenario 10 (Figure 9(c)) found by the TI based approach is much lower than that in scenarios 12 and 14. Scenario 15 (Figure 9(d)) found by the AW based approach is similar to the query scenario 12 in general, but shows relatively different temporal patterns in the first six hours.

To show the impact of inaccurate distance measures on the query results, we plot in Figure 8(b) the agreement between the query results obtained using the multiresolution distance measure of high accuracy, and those obtained using the inaccurate TI and AW based distance measures at different values of K. This figure demonstrates the importance of using the multiresolution distance measure in the similarity search tasks.

2. *Efficiency of the multiresolution distance based similarity search algorithm*

To evaluate the computational efficiency of these approaches, we compare our algorithm with the exhaustive search-based approach that uses the multiresolution distance measure and the TI and AW based approaches. We use the small dataset of $|S| = 124$ scenarios and measure the execution time of these approaches to perform the K-NN query at different values of K. The comparison results are shown in Figure 10, with $w_{max} = 4$, $h_{max} = 3$, $\delta_w = e^{-0.8(w-1)}$, $\alpha_h = e^{-0.8(h-1)}$, and $f = 0.1$. Each value is obtained by averaging the time spent to process $|S|$ queries with $s_i \in S$ being the query scenario and $S \setminus s_i$ being the database at the i-th query. This figure shows that the TI and AW based approaches are the most efficient, while the multiresolution distance based exhaustive search method is the least efficient. The TI and AW based approaches are more efficient than our approach as they compare scenarios at a single resolution. However, as we have demonstrated in the previous section, both TI and AW based approaches often fail to find the most similar scenarios.

Figure 10 also shows a significant improvement of computational efficiency for our approach compared with the multiresolution distance based exhaustive search method. This improvement illustrates the effectiveness of the iterative search space pruning procedures in reducing the computational cost. Of interest, the efficiency of the exhaustive search-based approaches is independent from K, while the efficiency of our approach degrades with the increase of K. This is because exhaustive search-based approaches perform the same number of scenario comparisons regardless of K. However, in our approach, a larger K typically leads to more scenarios to evaluate at more resolutions.

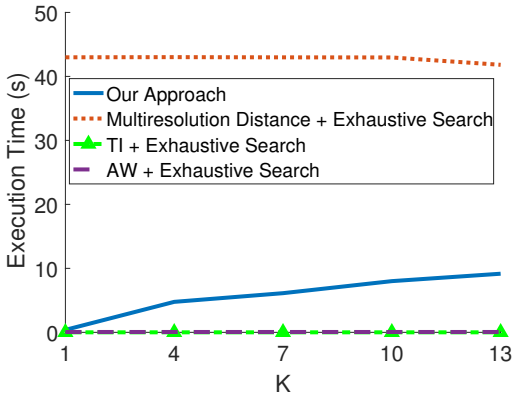


Figure 10. Comparison of the execution time of different similarity search algorithms.

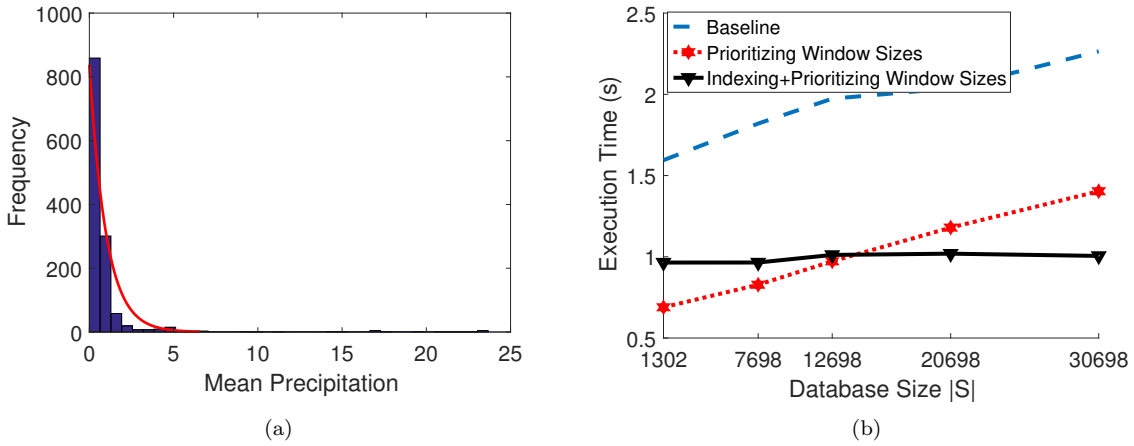


Figure 11. Illustration of a) distribution of $\bar{I}_i$; and b) efficiency of the proposed similarity search algorithm before and after implementing the data accessing strategies.

3. Efficiency of the data access strategies

We here investigate the effectiveness of the data access strategies described in Section III.C in improving the query efficiency. To evaluate their performance, we compare the efficiency of the proposed similarity search algorithm before and after implementing the data access strategies, and vary the size of the database to study the capability of these strategies in handling large databases.

As we only have 1302 real weather scenarios, databases with size larger than 1302 are created by inserting random scenarios, where the precipitation $I_{i,z,l}$ in each random scenario follows the same distribution as the mean precipitation of the real weather scenarios. In particular, the mean precipitation $\bar{I}_i = \frac{\sum_{g_z \in G, t_l \in T} I_{i,z,l}}{|G||T|}$ of the real weather scenario approximately follows an exponential distribution with mean 0.9932, as shown in Figure 11(a). The comparison results are shown in Figure 11(b), with $w_{max} = 4$, $h_{max} = 3$, $\delta_w = e^{-0.8(w-1)}$, $\alpha_h = e^{-0.8(h-1)}$, $\mathtt{K} = 2$, and $f = 0.1$. Each value is obtained by averaging the time spent to perform 21 query requests. The baseline approach does not use any data access strategies, i.e., it examines all scenarios with randomly sorted window sizes. As shown in Figure 11(b), the baseline approach has the worst performance in terms of execution time. Prioritizing window sizes significantly improves the efficiency. Implementing the indexing and the Filter-Restart schemes further reduces the computational cost for large databases, but degrades the performance when the database is relatively small. This is because implementing these schemes introduces additional computational cost, which exceeds the cost to access all scenarios in the databases of small sizes.

4. Impact of threshold f

In this study, we analyze the impact of the threshold f on the query efficiency, which determines the number of scenarios to examine. Figure 12(a) shows the execution time of the proposed similarity search algorithm with f taking different values and $\mathtt{K} = 2$. As we can see from the figure, the highest efficiency is achieved at $f = 0.01$, while smaller or larger values lead to degraded performance. This is because a small f would lead to frequent restarts, and a large f would lead to a large number of unnecessary scenarios being retrieved. We also notice that the optimal value of f may change for different $\mathtt{K}$, as shown in Figure 12(b) with $|S| = 1302$. This is easy to understand as larger $\mathtt{K}$ typically leads to more scenarios to examine and thus a larger f may achieve higher efficiency. For instance, when $\mathtt{K} = 13$, the highest efficiency is achieved at $f = 0.3$.

IV.D. Java-based Implementation

We further implement the similarity search algorithm using Java and store weather scenarios into a relational database using MySQL, where the scenario table has $|S|$ rows and four columns including the

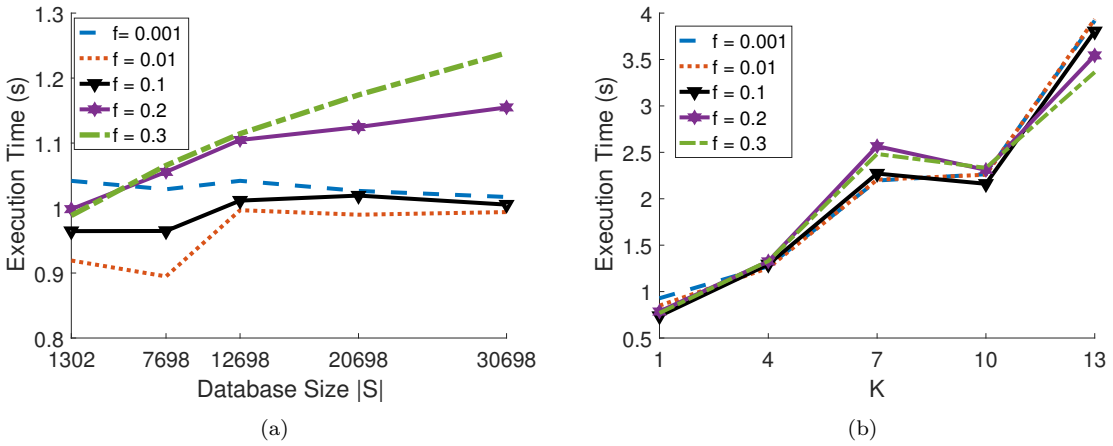


Figure 12. Execution time versus a) the size of the database and b) the query coefficient K at different values of the threshold f.

scenario index i, spatial cell g_z, time point t_l, and the associated intensity value $I_{i,z,l}$. Other information stored in the database include the weighted total intensity table I, spatial windows $\phi_{z,w}$ centered at each cell g_z with different window sizes w, contribution factors $\lambda_{z,w}$ and $\tau_{l,h}$, and spatial and temporal weights δ_w, α_h. User interfaces are also designed using Java SWINGS to improve the usability of the proposed similarity search algorithm, which allow the users to configure parameters, upload the query scenario, analyze and download the query results, and update the database.

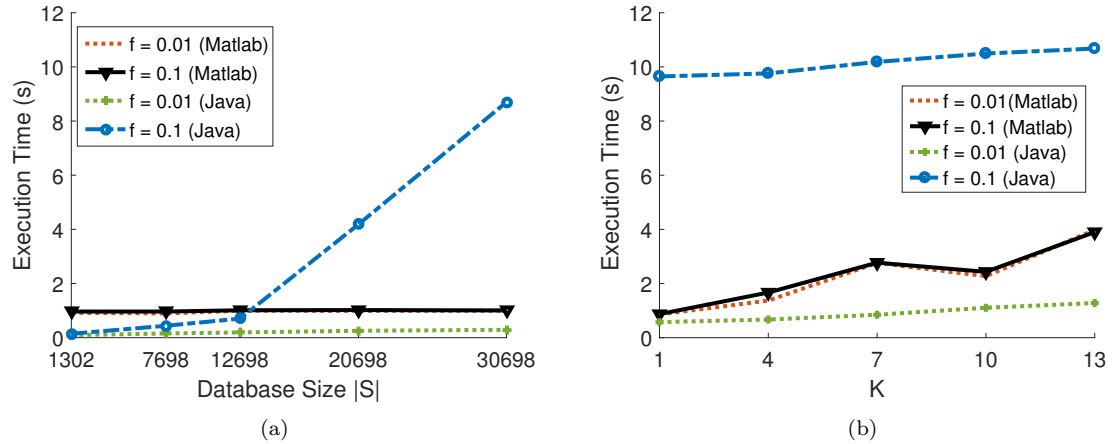


Figure 13. Comparison of the efficiency of Matlab-based prototype and Java-based implementation with increasing a) database size and c) query coefficient K at different values of f.

To evaluate the performance of the Java-based implementation, we conduct two experiments similar as the ones described in Section IV.C.3. In the first experiment, we vary the database size and compare the efficiency of the Matlab-based prototype and the Java-based implementation. Note that the Matlab-based prototype directly reads weather scenarios from local MAT files. Although Matlab can also connect to MySQL databases, accessing databases from Matlab is very time consuming and thus is not evaluated here. In this experiment, the parameter values are set to $\mathtt{K} = 2$, $w_{max} = 4$, $h_{max} = 3$, $\delta_w = e^{-0.8(w-1)}$, $\alpha_h = w^{-0.8(h-1)}$, and $f \in \{0.01, 0.1\}$. As shown in Figure 13(a), the Java-based implementation is the most efficient when the database size or the threshold f is small, even if the Matlab-based prototype accesses data directly from local files. However, the performance of the Java-based implementation degrades significantly when the database size or the threshold is large. This is because a larger database size or threshold f leads

to more scenarios to examine and thus more pairwise comparisons, which are realized by costly loops in Java but efficient vectorized operations in Matlab. In the second experiment, we vary the query coefficient `K` and connect to the largest database with $|S| = 30698$. Figure 13(b) shows the comparison results, which further illustrates the high efficiency of the Java-based implementation when a proper threshold f is selected.

V. Conclusion

In this paper, an innovative similarity search algorithm for a new data type, spatiotemporal scenario data, was developed to achieve quick retrieval of similar scenarios from the database. The algorithm allows the similarity search for spatiotemporal scenario data to be handled by commodity computing hardware for real-time decisions. The correctness of this similarity search algorithm is guaranteed by a novel multiresolution distance measure, which quantifies the differences between spatiotemporal scenarios. To address the computational complexity of this distance measure that scans scenarios at multiple resolutions, we developed an iterative procedure that uses gradually tightened lower and upper bounds of the distance measure to prune the search space. An indexing and Filter-Restart scheme were further developed to reduce the number of scenarios to examine for large databases. We further prioritized window sizes to achieve earlier termination of the iterations. To illustrate the use and performance of the proposed approaches, extensive simulation studies using real weather forecast data have been conducted. In particular, the impact of the parameters in the similarity search algorithm was analyzed, providing insights on the guidelines for parameter selection. Comprehensive comparative studies were then conducted, which demonstrate the effectiveness of the proposed approaches. This similarity search algorithm is a critical component of the spatiotemporal scenario-data driven decision-making framework[1] that closes the loop between big data and control to face the challenges of real-time management for large-scale dynamical systems. Building on the similarity search algorithm developed in this paper, we will develop the scenario-driven decision-making framework[1] for strategic ATM in our future work. We will also investigate the similarity search for spatiotemporal scenarios described by multiple metrics.

VI. Appendix

VI.A. Lower Bound of $d_{i,j,w,h}$

In this section, we show the proof of Equation (8) in Section III.B by following similar procedures of the proof for Equation (4) given in the Appendix in reference.[13]

First, let us prove $d_{i,j,w,h} \geq d_{i,j,w^*,h}$. Note that when $w = w^*$, we can easily derive $\phi_{z,w^*} = G$, $|\phi_{z,w^*}| = |G|$, $\Phi_{w^*} = \{\underbrace{G, G, \ldots, G}_{|G|}\}$, and $\lambda_{z,w^*} = 1$. Therefore, according to Equation (1),

$$
\begin{aligned}
d_{i,j,w^*,h} &= \sum_{\phi_{l,h}\in\Phi_h} \sum_{\phi_{z,w^*}\in\Phi_{w^*}} \frac{1}{|\phi_{z,w^*}||\phi_{l,h}||\Phi_h|} \left| \sum_{g_n\in\phi_{z,w^*}} \sum_{t_m\in\phi_{l,h}} \frac{\hat{I}_{i,n,m}}{\lambda_{n,w^*}\tau_{m,h}} - \sum_{g_n\in\phi_{z,w^*}} \sum_{t_m\in\phi_{l,h}} \frac{\hat{I}_{j,n,m}}{\lambda_{n,w^*}\tau_{m,h}} \right| \\
&= \sum_{\phi_{l,h}\in\Phi_h} \frac{|G|}{|G||\phi_{l,h}||\Phi_h|} \left| \sum_{g_n\in G} \sum_{t_m\in\phi_{l,h}} \frac{\hat{I}_{i,n,m}}{\tau_{m,h}} - \sum_{g_n\in G} \sum_{t_m\in\phi_{l,h}} \frac{\hat{I}_{j,n,m}}{\tau_{m,h}} \right| \\
&= \sum_{\phi_{l,h}\in\Phi_h} \frac{1}{|\phi_{l,h}||\Phi_h|} \left| \sum_{g_n\in G} \left(\sum_{t_m\in\phi_{l,h}} \frac{\hat{I}_{i,n,m}}{\tau_{m,h}} - \sum_{t_m\in\phi_{l,h}} \frac{\hat{I}_{j,n,m}}{\tau_{m,h}} \right) \right|
\end{aligned}
$$

We now compare $d_{i,j,w,h}$ and $d_{i,j,w^*,h}$. According to Equation (1) and the triangular inequality rule,

$$\begin{aligned} d_{i,j,w,h} &\geq \sum_{\phi_{l,h}\in\Phi_h} \frac{1}{|\phi_{l,h}||\Phi_h|} \left| \sum_{\phi_{z,w}\in\Phi_w} \frac{1}{|\phi_{z,w}|} \left(\sum_{g_n\in\phi_{z,w}} \sum_{t_m\in\phi_{l,h}} \frac{\hat{I}_{i,n,m}}{\lambda_{n,w}\tau_{m,h}} - \sum_{g_n\in\phi_{z,w}} \sum_{t_m\in\phi_{l,h}} \frac{\hat{I}_{j,n,m}}{\lambda_{n,w}\tau_{m,h}} \right) \right| \\ &= \sum_{\phi_{l,h}\in\Phi_h} \frac{1}{|\phi_{l,h}||\Phi_h|} \left| \sum_{\phi_{z,w}\in\Phi_w} \sum_{g_n\in\phi_{z,w}} \frac{1}{|\phi_{z,w}|\lambda_{n,w}} \left(\sum_{t_m\in\phi_{l,h}} \frac{\hat{I}_{i,n,m}}{\tau_{m,h}} - \sum_{t_m\in\phi_{l,h}} \frac{\hat{I}_{j,n,m}}{\tau_{m,h}} \right) \right| \\ &= \sum_{\phi_{l,h}\in\Phi_h} \frac{1}{|\phi_{l,h}||\Phi_h|} \left| \sum_{g_z\in G} \frac{1}{\lambda_{z,w}} \left(\sum_{\phi_{n,w}\in\{\phi_{n,w}|g_z\in\phi_{n,w}\}} \frac{1}{|\phi_{n,w}|} \right) \left(\sum_{t_m\in\phi_{l,h}} \frac{\hat{I}_{i,z,m}}{\tau_{m,h}} - \sum_{t_m\in\phi_{l,h}} \frac{\hat{I}_{j,z,m}}{\tau_{m,h}} \right) \right| \end{aligned}$$

As $\lambda_{z,w} = \sum_{\phi_{n,w}\in\{\phi_{n,w}|g_z\in\phi_{n,w}\}} \frac{1}{|\phi_{n,w}|}$, we then have

$$\begin{aligned} d_{i,j,w,h} &\geq \sum_{\phi_{l,h}\in\Phi_h} \frac{1}{|\phi_{l,h}||\Phi_h|} \left| \sum_{g_z\in G} \left(\sum_{t_m\in\phi_{l,h}} \frac{\hat{I}_{i,z,m}}{\tau_{m,h}} - \sum_{t_m\in\phi_{l,h}} \frac{\hat{I}_{j,z,m}}{\tau_{m,h}} \right) \right| \\ &= d_{i,j,w^*,h} \end{aligned}$$

Next, let us prove $d_{i,j,w,h} \geq d_{i,j,w,h^*}$ by following a similar procedure. Note that when $h = h^*$, we find $\phi_{l,h^*} = T$, $|\phi_{l,h^*}| = |T|$, $\Phi_{h^*} = T$, $|\Phi_{h^*}| = 1$, and $\tau_{l,h^*} = 1$. Therefore,

$$\begin{aligned} d_{i,j,w^*,h} &= \sum_{\phi_{l,h^*}\in\Phi_{h^*}} \sum_{\phi_{z,w}\in\Phi_w} \frac{1}{|\phi_{z,w}||\phi_{l,h^*}||\Phi_h^*|} \left| \sum_{g_n\in\phi_{z,w}} \sum_{t_m\in\phi_{l,h^*}} \frac{\hat{I}_{i,n,m}}{\lambda_{n,w}\tau_{m,h^*}} - \sum_{g_n\in\phi_{z,w}} \sum_{t_m\in\phi_{l,h^*}} \frac{\hat{I}_{j,n,m}}{\lambda_{n,w}\tau_{m,h^*}} \right| \\ &= \sum_{\phi_{z,w}\in\Phi_w} \frac{1}{|\phi_{z,w}||T|} \left| \sum_{t_m\in T} \left(\sum_{g_n\in\phi_{z,w}} \frac{\hat{I}_{i,n,m}}{\lambda_{n,w}} - \sum_{g_n\in\phi_{z,w}} \frac{\hat{I}_{j,n,m}}{\lambda_{n,w}} \right) \right| \end{aligned}$$

Again, according to Equation (1) and the triangular inequality rule,

$$\begin{aligned} d_{i,j,w,h} &\geq \sum_{\phi_{z,w}\in\Phi_w} \frac{1}{|\phi_{z,w}|} \left| \sum_{\phi_{l,h}\in\Phi_h} \frac{1}{|\phi_{l,h}||\Phi_h|} \left(\sum_{g_n\in\phi_{z,w}} \sum_{t_m\in\phi_{l,h}} \frac{\hat{I}_{i,n,m}}{\lambda_{n,w}\tau_{m,h}} - \sum_{g_n\in\phi_{z,w}} \sum_{t_m\in\phi_{l,h}} \frac{\hat{I}_{j,n,m}}{\lambda_{n,w}\tau_{m,h}} \right) \right| \\ &= \sum_{\phi_{z,w}\in\Phi_w} \frac{1}{|\phi_{z,w}|} \left| \sum_{\phi_{l,h}\in\Phi_h} \sum_{t_m\in\phi_{l,h}} \frac{1}{|\phi_{l,h}||\Phi_h|\tau_{m,h}} \left(\sum_{g_n\in\phi_{z,w}} \frac{\hat{I}_{i,n,m}}{\lambda_{n,w}} - \sum_{g_n\in\phi_{z,w}} \frac{\hat{I}_{j,n,m}}{\lambda_{n,w}} \right) \right| \\ &= \sum_{\phi_{z,w}\in\Phi_w} \frac{1}{|\phi_{z,w}|} \left| \sum_{t_l\in T} \frac{1}{\tau_{l,h}} \left(\sum_{\phi_{m,h}\in\{\phi_{m,h}|t_l\in\phi_{m,h}\}} \frac{1}{|\phi_{m,h}||\Phi_h|} \right) \left(\sum_{g_n\in\phi_{z,w}} \frac{\hat{I}_{i,n,l}}{\lambda_{n,w}} - \sum_{g_n\in\phi_{z,w}} \frac{\hat{I}_{j,n,l}}{\lambda_{n,w}} \right) \right| \end{aligned}$$

As $\tau_{l,h} = \sum_{\phi_{m,h}\in\{\phi_{m,h}|t_l\in\phi_{m,h}\}} \frac{|T|}{|\phi_{m,h}||\Phi_h|}$, we have

$$\begin{aligned} d_{i,j,w,h} &\geq \sum_{\phi_{z,w}\in\Phi_w} \frac{1}{|\phi_{z,w}||T|} \left| \sum_{t_l\in T} \left(\sum_{g_n\in\phi_{z,w}} \frac{\hat{I}_{i,n,l}}{\lambda_{n,w}} - \sum_{g_n\in\phi_{z,w}} \frac{\hat{I}_{j,n,l}}{\lambda_{n,w}} \right) \right| \\ &= d_{i,j,w,h^*} \end{aligned}$$

Now have proved that $d_{i,j,w,h} \geq d_{i,j,w^*,h}$ and $d_{i,j,w,h} \geq d_{i,j,w,h^*}$, which naturally leads to Equations (8).

VI.B. Upper and Lower Bounds of $D_{i,j}$

In this section, we prove Equations (7) and (11) in Section III.B.

First, let us prove Equation (7). Denote $W_k \subset W$ and $H_k \subset H$ as the set of spatial and temporal window sizes that have been evaluated so far after k iterations, where $W = \{1, 2, \ldots, w_{max}\}$, $H = \{1, 2, \ldots, h_{max}\}$

and $k \in \{1, 2, \ldots, w_{max}h_{max} - 1\}$. We can then rewrite the formula of $D_{i,j}$ as follows

$$\begin{aligned} D_{i,j} &= \sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} d_{i,j,w,h} \frac{\delta_w \alpha_h}{\sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} \delta_w \alpha_h} \\ &= \frac{1}{\sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} \delta_w \alpha_h} \left(\sum_{h \in H_k} \sum_{w \in W_k} d_{i,j,w,h} \delta_w \alpha_h + \sum_{h \in H \setminus H_k} \sum_{w \in W \setminus W_k} d_{i,j,w,h} \delta_w \alpha_h \right) \end{aligned}$$

where $w \in W \setminus H_k$ and $h \in H \setminus H_k$ are the window sizes that haven't been evaluated at the k-th iteration. By approximating $d_{i,j,w,h}$ by $d_{i,j,1,1}$ for all $w \in W \setminus H_k$ and $h \in H \setminus H_k$, where $d_{i,j,w,h} \leq d_{i,j,1,1}$ according to Equation (6), we derive the upper bound $\overline{D}_{i,j}[k]$ at the k-th iteration:

$$\overline{D}_{i,j}[k] = \frac{1}{\sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} \delta_w \alpha_h} \left(\sum_{h \in H_k} \sum_{w \in W_k} d_{i,j,w,h} \delta_w \alpha_h + \sum_{h \in H \setminus H_k} \sum_{w \in W \setminus W_k} d_{i,j,1,1} \delta_w \alpha_h \right)$$

In the special case when $k = 1$, we have $\overline{D}_{i,j}[1] = d_{i,j,1,1}$. Suppose the spatial and temporal window sizes are increased to $w = a$ and $h = b$ at the $(k+1)$-th iteration, where $1 \leq a \leq w_{max}$, $1 \leq b \leq h_{max}$ and $(a, b) \neq (1, 1)$. The upper bound $\overline{D}_{i,j}[k+1]$ at the $(k+1)$-th iteration is then computed by

$$\begin{aligned} \overline{D}_{i,j}[k+1] &= \frac{1}{\sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} \delta_w \alpha_h} \left(\sum_{h \in H_{k+1}} \sum_{w \in W_{k+1}} d_{i,j,w,h} \delta_w \alpha_h + \sum_{h \in H \setminus H_{k+1}} \sum_{w \in W \setminus W_{k+1}} d_{i,j,1,1} \delta_w \alpha_h \right) \\ &= \overline{D}_{i,j}[k] + \frac{1}{\sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} \delta_w \alpha_h} \left(d_{i,j,a,b} \delta_a \alpha_b - d_{i,j,1,1} \delta_a \alpha_b \right) \\ &= \overline{D}_{i,j}[k] + \frac{\delta_a \alpha_b}{\sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} \delta_w \alpha_h} \left(d_{i,j,a,b} - d_{i,j,1,1} \right) \end{aligned}$$

where $W_{k+1} = W_k \cup \{a\}$ and $H_{k+1} = H_k \cup \{b\}$.

The lower bound $\underline{D}_{i,j}[k]$ can be computed similarly by approximating $d_{i,j,w,h}$ by d_{i,j,w^*,h^*} for all $w \in W \setminus H_k$ and $h \in H \setminus H_k$, where $d_{i,j,w,h} \geq d_{i,j,w^*,h^*}$ according to Equation (10). In particular,

$$\begin{aligned} \underline{D}_{i,j}[k+1] &= \frac{1}{\sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} \delta_w \alpha_h} \left(\sum_{h \in H_{k+1}} \sum_{w \in W_{k+1}} d_{i,j,w,h} \delta_w \alpha_h + \sum_{h \in H \setminus H_{k+1}} \sum_{w \in W \setminus W_{k+1}} d_{i,j,w^*,h^*} \delta_w \alpha_h \right) \\ &= \underline{D}_{i,j}[k] + \frac{1}{\sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} \delta_w \alpha_h} \left(d_{i,j,a,b} \delta_a \alpha_b - d_{i,j,w^*,h^*} \delta_a \alpha_b \right) \\ &= \underline{D}_{i,j}[k] + \frac{\delta_a \alpha_b}{\sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} \delta_w \alpha_h} \left(d_{i,j,a,b} - d_{i,j,w^*,h^*} \right) \end{aligned}$$

In the special case when $k = 1$, we have

$$\begin{aligned} \underline{D}_{i,j}[1] &= \frac{1}{\sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} \delta_w \alpha_h} \left(d_{i,j,1,1} \delta_1 \alpha_1 + \sum_{h \in H \setminus \{1\}} \sum_{w \in W \setminus \{1\}} d_{i,j,w^*,h^*} \delta_w \alpha_h \right) \\ &= d_{i,j,w^*,h^*} + \frac{\delta_1 \alpha_1}{\sum_{h=1}^{h_{max}} \sum_{w=1}^{w_{max}} \delta_w \alpha_h} \left(d_{i,j,1,1} - d_{i,j,w^*,h^*} \right) \end{aligned}$$

The proof is complete now.

Acknowledgments

This work was partially supported by NSF Grants 1714519, 1714826, 1839707 and 1839804. It was also supported by the Texas Comprehensive Research Fund.

References

[1]Xie, J. and Wan, Y., "Scalable Multidimensional Uncertainty Evaluation Approach to Strategic Air Traffic Flow Management," *Proceedings of the AIAA Modeling and Simulation Technologies Conference*, AIAA Paper 2015-2492, Kissimmee, Florida, January 2015.

[2]Taylor, C., Wanke, C., Wan, Y., and Roy, S., "A framework for flow contingency management," *Proceedings of the 11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, including the AIAA Balloon Systems Conference and 19th AIAA Lighter-Than*, AIAA Paper 2011-6904, Virginia Beach, VA, September 2011.

[3]Taylor, C., Wanke, C., Wan, Y., and Roy, S., "A decision support tool for flow contingency management," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2012-4976, Minneapolis, Minnesota, August 2012.

[4]Wan, Y., Taylor, C., Roy, S., Wanke, C., and Zhou, Y., "Dynamic queuing network model for flow contingency management," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 14, No. 3, 2013, pp. 1380–1392, doi: 10.1109/TITS.2013.2260745.

[5]Zhou, Y., Wan, Y., Roy, S., Taylor, C., Wanke, C., Ramamurthy, D., and Xie, J., "Multivariate probabilistic collocation method for effective uncertainty evaluation with application to air traffic flow management," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 44, No. 10, 2014, pp. 1347–1363, doi: 10.1109/TSMC.2014.2310712.

[6]Xie, J., Wan, Y., and Lewis, F., "Strategic air traffic flow management under uncertainties using scalable sampling-based dynamic programming and Q-learning approaches," *Proceedings of the 11th Asian Control Conference (ASCC)*, IEEE, City of Gold Coast, Australia, December 2017, pp. 1116–1121.

[7]Xie, J. and Wan, Y., "A Network Condition-Centric Flow Selection and Rerouting Strategy to Mitigate Air Traffic Congestion under Uncertainties," *Proceedings of the 17th AIAA Aviation Technology, Integration, and Operations Conference*, AIAA Paper 2017-3427, Denver, Colorado, June 2017.

[8]"FAQ: Weather Delay," https://www.faa.gov/nextgen/programs/weather/faq/, [Online; accessed 7-June-2017].

[9]Xie, J., Zhou, Y., Wan, Y., Mitchell, A., and Roy, S., "A Jump-Linear Model Based Sensitivity Study for Optimal Air Traffic Flow Management under Weather Uncertainty," *Proceedings of the AIAA SciTech Conference (Infotech@ Aerospace)*, AIAA Paper 2015-1573, Kissimmee, Florida, January 2015.

[10]Kuhn, K. D., "A methodology for identifying similar days in air traffic flow management initiative planning," *Transportation Research Part C: Emerging Technologies*, Vol. 69, 2016, pp. 1–15, doi: 10.1016/j.trc.2016.05.014.

[11]Delle Monache, L., Eckel, F. A., Rife, D. L., Nagarajan, B., and Searight, K., "Probabilistic weather prediction with an analog ensemble," *Monthly Weather Review*, Vol. 141, No. 10, 2013, pp. 3498–3516, doi: 10.1175/MWR-D-12-00281.1.

[12]Xie, J., Zhou, Y., Wan, Y., Tien, S.-L., Taylor, C. P., and Wanke, C. R., "A Multi-resolution Spatiotemporal Scenario Clustering Algorithm for Flow Contingency Management," *Proceedings of the 14th AIAA Aviation Technology, Integration, and Operations Conference*, AIAA Paper 2014-2029, Atlanta, GA, June 2014.

[13]Xie, J., Wan, Y., Zhou, Y., Tien, S.-L., Vargo, E. P., Taylor, C., and Wanke, C., "Distance Measure to Cluster Spatiotemporal Scenarios for Strategic Air Traffic Management," *Journal of Aerospace Information Systems*, Vol. 12, No. 8, 2015, pp. 545–563, doi: 10.2514/1.I010353.

[14]Kisilevich, S., Mansmann, F., Nanni, M., and Rinzivillo, S., "Spatio-temporal clustering," *Data mining and knowledge discovery handbook*, 2010, pp. 855–874.

[15]Tork, H. F., "Spatio-temporal clustering methods classification," *Doctoral Symposium on Informatics Engineering*, 2012, pp. 199–209.

[16]Jara, A. J., Genoud, D., and Bocchi, Y., "Big data for cyber physical systems: an analysis of challenges, solutions and opportunities," *Proceedings of the Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, IEEE, Birmingham, UK, July 2014, pp. 376–380.

[17]Xue, M., Roy, S., Zobell, S., Wan, Y., Taylor, C., and Wanke, C., "A stochastic spatiotemporal weather-impact simulator: Representative scenario selection," *Proceedings of the 11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, including the AIAA Balloon Systems Conference and 19th AIAA Lighter-Than*, AIAA Paper 2011-6812, Virginia Beach, VA, September 2011.

[18]Tien, S.-L., Taylor, C. P., and Wanke, C. R., "Identifying representative weather-impact scenarios for flow contingency management," *Proceedings of the 2013 Aviation Technology, Integration, and Operations Conference, AIAA Aviation Forum*, AIAA Paper 2013-4216, Los Angeles, CA, August 2013.

[19]Tien, S., Taylor, C., and Wanke, C., "Comparing and clustering ensemble forecast members to support strategic planning in air traffic flow management," *Proceedings of the 94th American Meteorological Society Annual Meeting*, American Meteorological Society, Atlanta, GA, February 2014, pp. 1–9.

[20]Kuhnert, M., Voinov, A., and Seppelt, R., "Comparing raster map comparison algorithms for spatial modeling and analysis," *Photogrammetric Engineering & Remote Sensing*, Vol. 71, No. 8, 2005, pp. 975–984, doi: 10.14358/PERS.71.8.975.

[21]Costanza, R., "Model goodness of fit: a multiple resolution procedure," *Ecological modelling*, Vol. 47, No. 3-4, 1989, pp. 199–215, doi: 10.1016/0304-3800(89)90001-X.

[22]Comer, D., "Ubiquitous B-tree," *ACM Computing Surveys (CSUR)*, Vol. 11, No. 2, 1979, pp. 121–137, doi: 10.1145/356770.356776.

[23]Hadjieleftheriou, M., Manolopoulos, Y., Theodoridis, Y., and Tsotras, V. J., *R-trees: A dynamic index structure for spatial searching*, Springer, Boston, MA, 2008, pp. 993–1002, doi: 10.1007/978-0-387-35973-1_1151.

[24]Bentley, J., "Multidimensional binary search trees used for associative searching." *Communications of ACM,*, Vol. 18, No. 9, 1975, pp. 509–517, doi: 10.1145/361002.361007.

[25]Izbicki, M. and Shelton, C., "Faster cover trees," *Proceedings of the International Conference on Machine Learning*, Lille, France, July 2015, pp. 1162–1170.

[26]Sellis, T., Roussopoulos, N., and Faloutsos, C., "The R+-Tree: A Dynamic Index for Multi-Dimensional Objects." Tech. rep., 1987.

[27]Houle, M. E. and Nett, M., "Rank-based similarity search: Reducing the dimensional dependence," *IEEE transactions on pattern analysis and machine intelligence*, Vol. 37, No. 1, 2015, pp. 136–150, doi: 10.1109/TPAMI.2014.2343223.

[28]James, D. L. and Pai, D. K., "BD-tree: output-sensitive collision detection for reduced deformable models," *ACM Transactions on Graphics (TOG)*, Vol. 23, No. 3, 2004, pp. 393–398, doi: 10.1145/1015706.1015735.

[29]Gionis, A., Indyk, P., Motwani, R., et al., "Similarity search in high dimensions via hashing," *Proceedings of the 25th International Conference on Very Large Data Bases*, ACM, Edinburgh, Scotland, September 1999, pp. 518–529.

[30]Indyk, P. and Motwani, R., "Approximate nearest neighbors: towards removing the curse of dimensionality," *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, ACM, Dallas, TX, May 1998, pp. 604–613.

[31]Houle, M. E. and Sakuma, J., "Fast approximate similarity search in extremely high-dimensional data sets," *Proceedings of 21st International Conference on Data Engineering*, IEEE, Tokoyo, Japan, May 2005, pp. 619–630.

[32]Ilyas, I. F., Beskales, G., and Soliman, M. A., "A survey of top-k query processing techniques in relational database systems," *ACM Computing Surveys (CSUR)*, Vol. 40, No. 4, 2008, pp. 11:1–11:58, doi: 10.1145/1391729.1391730.

[33]Guntzer, J., Balke, W.-T., and Kießling, W., "Towards efficient multi-feature queries in heterogeneous environments," *Proceedings of the International Conference on Information Technology: Coding and Computing*, IEEE, Las Vegas, NV, April 2001, pp. 622–628.

[34]Du, J., DiMego, G., Toth, Z., Jovic, D., Zhou, B., Zhu, J., Chuang, H., Wang, J., Juang, H., Rogers, E., et al., "NCEP short-range ensemble forecast (SREF) system upgrade in 2009," *19th conference on numerical weather prediction and 23rd conference on weather analysis and forecasting*, American Meteorological Society, Omaha, NE, June 2009, pp. 1–5.

[35]Du, J., DiMego, G., Zhou, B., Jovic, D., Ferrier, B., Yang, B., and Benjamin, S., "NCEP Regional Ensembles: Evolving toward hourly-updated convection-allowing scale and storm-scale predictions within a unified regional modeling system," *Proceedings of the 22nd Conference on Numerical Weather Prediction/26th Conference on Weather Analysis and Forecasting*, American Meteorological Society, Atlanta, GA, February 2014, Paper J1.4.

[36]Bright, D. R., Racy, J. P., Weiss, S. J., Schneider, R. S., Levit, J. J., Huhn, J. J., Duquette, M. A., Kain, J. S., Coniglio, M. C., Xue, M., et al., "Short Range and Storm Scale Ensemble Forecast Guidance and Its Potential Applications in Air Traffic Decision Support," *Proceedings of Preprint, Aviation, Range, Aerospace Meteorology Special Symposium Weather-Air Traffic Management Integration*, American Meteorogical Society, Phoenix, AZ, 2009, Paper P1.7.