NO BOUNDARY FOR SPATIAL INTERACTIONS — EXPLORATORY SPATIAL FLOW DATA ANALYSIS

by

Ran Tao

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Geography and Urban Regional Analysis

Charlotte

2017

Approved by:

_____
Dr. Jean-Claude Thill

_____
Dr. Harrison Campbell

_____
Dr. Wenwu Tang

_____
Dr. James Walsh

ABSTRACT

RAN TAO. No boundary for spatial interactions — exploratory spatial flow data analysis. (Under the direction of DR. JEAN-CLAUDE THILL)

Spatial interaction (SI) represents meaningful human relations between areas on the Earth's surface, such as the reciprocal relations and flows of all kinds among industries, markets, regions, cities, or logistics centers. With the widespread adoption of location-aware technologies and the global diffusion of geographic information systems (GIS), spatial interaction data have been remarkably enriched. In this dissertation research, I develop three unique but closely related exploratory spatial flow data analysis (ESFDA) methods, as an answer to the challenges and opportunities brought by the recent data revolution. Each new method stems from one or more of the following methodological subfields: geovisualization, spatial data mining, and spatial statistics.

The first method, dubbed Flow K-function, is a spatial statistical approach to detect spatial clustering patterns of flow data. In other words, it upgrades the classical hot spot detection method to the stage of "hot flow" detection. A set of spatial proximity measures are designed for flow data by integrating endpoint location, length, and direction. The measures can extract both intra-relationships and inter-relationships of flows and serve as the basis of Flow K-function. The second approach, Flow HDBSCAN, is a hierarchical and density-based spatial flow cluster analysis method. Not only can it extract flow clusters from various situations including varying flow densities, lengths, hierarchies, but it also provides an effective way to reveal the potential hierarchical structure of the clusters. The last method is called FlowAMOEBA. It is a data-driven and bottom-up approach for identifying regions of anomalous spatial interactions, based on which it creates a spatial

flow weights matrix. It upgrades A Multidirectional Optimum Ecotope-Based Algorithm (AMOEBA) (Aldstadt and Getis 2006) from areal data to spatial flow data through a proper spatial flow neighborhood definition. The method breaks the tradition that spatial interaction data are always collected and modelled between two comparable predefined geographic units, as it delineates the boundaries of anomalous interacting regions regardless of size, shape, scale, or administrative level. The spatial flow weights matrix based on the identified regions can be used to account for network autocorrelation, thus improving confirmatory studies using spatial interaction modeling.

These newly developed methods can be utilized individually for data exploration, pattern detection, and hypothesis development. They can also be used jointly to the same application to take advantage of each method. The results of these methods can further be used to form new hypotheses based on explored interesting patterns, to challenge old theories so as to form new ones, to deepen understanding of spatial interaction process, and to improve related confirmatory studies, thus improving related policy-making or problem solving strategies.

Three different use cases are presented as to demonstrate the use of each of the methods. The data include a set of motor-vehicle theft and recovery flows, a set of online iPhone transaction flows on the eBay platform, and county-to-county migration flows. Advantages and limitations of each method are tested and discussed thoroughly. Practical usefulness and application implications are also explored and discussed in each scenario.

ACKNOWLEDGEMENTS

My deep gratitude goes first to my advisor Dr. Jean-Claude Thill, who expertly guided me through my graduate education and who shared the excitement of five years of discovery. Without your guidance and persistent help this dissertation would not have been possible. I have learned so much from you, and I will keep learning from you to become a knowledgeable scholar and a better person.

I would also like to thank the other members of my dissertation committee: Dr. Wenwu Tang, Dr. Harrison Campbell, and Dr. James Walsh for your encouraging support on this dissertation work, on the courses I took from you, and on the research projects I have collaborated with you.

My appreciation extends to my friends and colleagues as well: Zhaoya, Mona, Kailas, Diep, Jae Soen, Danny, Pooya, Daidai, Jing, Wenpeng, Alex, Adam, and many more, your company and support have made my time at Charlotte colorful and enjoyable.

Last but not least, I am grateful for my families. Mom, Dad, and my little brother: thank you for the endless love and the support throughout my life. And finally, thank you Jieyan, my loving wife. I would not be as happy as I am if I did not meet you three years ago. Marrying you is the greatest achievement of my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1 What Is Spatial Interaction

Spatial interaction (SI) represents meaningful human relations between areas on the Earth's surface, such as the reciprocal relations and flows of all kinds among industries, markets, cultures, or logistics centers (Ullman 1954). It is usually represented as a dynamic flow process from one location to another, referring to the movement of human beings such as daily commuters, or traffic in goods such as raw materials, or even flows of intangibles such as information (Haynes and Fotheringham 1984). Given that it is extremely versatile to represent physical or socioeconomic processes driven by essential forces with flows, spatial interaction data has been a critical component and an enduring object of research in a wide range of fields of research and decision-making including epidemiology, economics, geography, transportation, and emergency management (Guo 2009).

Typically a flow event in geography consists of two components, i.e. the spatial component represented as a vector between two endpoints, as well as the nonspatial component which encapsulates the type or value of the flow (Tao and Thill 2016). Taking the common example of daily commuting flows, the spatial component is a directed line representing the dynamic movement process from the origin to the destination, i.e. from home to workplace, but ignoring the actual trajectory route in between (Zhu and Guo 2014). On the other hand, the nonspatial component includes all the related information such as

information of the traveler, cost of the trip, transport mode, time stamp and length of the movement, and so on.

In general, there are two types of flow data, namely individual (discrete) flow and aggregated flow (Murray et al. 2011). The former pertains to individual activities, for example one person taking the subway from home to work on a weekday morning. In contrast, the latter represents the movement or interactions of a group of people or objects, for example a group of elks residing in the northern region of Yellowstone National Park and migrating to lower altitudes before winter arrives. Another taxonomy of spatial interaction data is to categorize flows according to whether there are constraints in their actual paths (Marble et al. 1997). If there exist explicit channels e.g. river systems, rail lines, road networks that the moving objects must follow in their paths from origin to destination, such flows are constrained flows. On the other hand, if only the origin and destination are known while the paths in between are not available or simply not important to be noted, such flows are called unconstrained flows. Take the refugee flows to Europe as an example; unconstrained flows contain the information of refugee's original places and destination places, e.g. from Syria to Germany, but ignore the actual trajectory route. In contrast, the constrained flows convey more information regarding which paths refugees are taking for instance pass Turkey first or cross the Mediterranean Sea first.

1.2 Previous Theories and Methodologies of Spatial Interaction

As an enduring study object in various scientific disciplines, a number of theories and methodologies regarding spatial interaction have been developed. In geography, the concept of spatial interaction was first introduced by French geographers' notions of "géographie de circulation" (Cavaillès 1940), including both the movement of physical

objects and the communication of intangible ideas. But the full development of SI as a fundamental topic that improves the understanding of the development of distinctive regional geography was initiated in the 1950s by Edward Ullman, who later brought up one of the most well-known SI theories, namely the "three bases" of spatial interactions. In this theory, most spatial interaction processes are driven or influenced by **complementarity**, **transferability**, and **intervening opportunity**. Complementarity refers to the rationale of spatial flow process as the demand or deficit at one location is satisfied by the movement of corresponding supply or surplus at another location. The explainable cases include spatial flows in both human geography and physical geography. For example, commuting flows represent human resources moving from residential places to where the job opportunities are; winds can be seen as air displaced from a high-pressure zone complementing a low pressure zone. Spatial interactions driven by complementarity could happen between places with extremely long distance, such as petroleum transported from the Middle East to the US, or between very close ones, for example students walk a few hundred yards from home to the nearby school. However, with all other things being equal, shorter distances hold an advantage over the longer ones. This is also called the "friction of distance" known as the second base of spatial interaction, i.e. transferability. Taking the example of a person shopping for groceries, he or she will be more likely to choose the nearest store over a distant one if all of them offer the same goods at the same price. On the other hand, if a distant store offers better goods at a lower price that compensate the costs of traveling a longer distance, it has the potential to win the customer back. Clearly distance plays an important but most likely negative role here as the spatial interaction activities have to overcome the travel costs. Nevertheless, spatial interactions

in the real world are usually more complex than a binary activity between a pair of places. The above grocery shopping case shows that the second store can potentially intervene in the original flow between the first store and the customer. It indicates that if there are more than one destination (origin) having supply (demand), the final spatial interactions are not only decided by complementarity and transferability between origins and destinations, but the internal relationship among destinations (origins) as well. Such effects are captured by the third base of SI called intervening opportunity. In general, interaction between two places would happen if their complementarity is strong enough to overcome the distance, and at the same time there exists no stronger intervening opportunity nearby. This "three bases" SI theory has had a profound influence on later research. A lot of research has been conducted to investigate, validate, or complement this theory in fields like economics, transportation, business management, urban planning, etc.

An important family of methods dealing with spatial flow data is so-called spatial interaction (SI) modeling. The distinctive contribution of SI models to flow and movement analysis is to separate explanatory factors into three multiplicative classes, site attributes of origins, site attributes of destinations and measures of relative distance/travel time separating origins and destinations, sometimes collected under the heading "impedance" effect (Roy and Thill 2004). Methods of SI modeling have been continually developed for several decades. From the early gravity models (Tobler 1981), to using causative matrices to monitor system change (Plane and Rogerson 1986), and to using more general concepts of entropy or information theory (Roy and Thill 2004), SI models have been developed to help understand underlying patterns or causality of spatial interaction data.

1.3 Massive Novel Spatial Flow Data Bring Both Opportunities and Challenges

With the widespread adoption of location-aware technologies and the global diffusion of geographic information systems (GIS), spatial interaction data have been enriched in several respects (Yan and Thill 2009; Guo et al. 2012). On one hand, traditional flow events such as flows of people and flows of commodities are recorded and are becoming available in tremendous data size and fine spatiotemporal granularity. For instance United Parcel Service (UPS) Inc. and FedEx Inc. delivered a combined 947 million packages[1] from the warehouses of an electronic commerce, e.g. Amazon Inc., to a customer's front door between Thanksgiving and Christmas Eve last year. The delivery processes can be seen as nearly a billion spatial flow events with trackable high-resolution spatiotemporal information, and this includes attributes pertaining the transaction merchants and agents.

On the other hand, emerging types of interaction activities, especially information exchange on the Internet, have enhanced the richness of flow data. For example, one can easily make some online donation from one's home to help the children suffering in Syria[2]; a breaking news can be transmitted from its origin place to millions of people's smartphone using "#" and "@" on Twitter. Neither charitable donations on the worldwide web nor news transmission through social media requires physical movements in space, but interactions have indeed taken place between people from different places.

Undoubtedly, we have entered a new era of spatial flow data. Naturally a number of questions have emerged. For instance, are the traditional theories of spatial interaction still valid and valuable to explain the new types of flow events? Specifically, are there any new

[1] http://www.sgvtribune.com/business/20151222/how-fedex-handles-millions-of-packages-during-the-christmas-season
[2] https://www.unicefusa.org/donate/help-syrian-children/

types of spatial interactions driven by some other reasons than complementarity? Is it possible that transferability does not function as impedance in spatial interactions? Can nearby supply locations no longer act as competitors in the new online trading activities? In addition, can we quantitatively model the new flow data in a better way? Or even can we explore unprecedented patterns from the new flow data, thus proposing new theories? All these questions remain interesting but uncertain at this stage. As a conclusion, the increased availability of massive volumes of new forms of flow data inevitably brings unprecedented opportunities to improve our understanding of SI processes and thus enriching the SI theories. At the same time, great intellectual challenges exist for grasping these opportunities.

The most direct challenge is to handle the large volume of data with existing computing resources: the current capability of computation techniques cannot handle big flow data well. For example, a normal personal computer with 4 Gigabyte memory can only load a twenty thousand by twenty thousand matrix, which is significantly insufficient to handle millions of OD flows that are quite common nowadays. Without the capability to store, compute, analyze, and visualize the data, almost no further research can be carried out to obtain meaningful results. Many scholars have been seeking solutions to solve these problems. In general, taking advantage of most advanced computing techniques such as high performance computing, and developing new or improving existing methods are two common ways.

GeoComputation, the process of applying computing technology to geographical problems, is the solution to many pressing issues, including massive spatial flow data. In their book, Openshaw and Abrahart (2000) provide the example of the earliest uses of

parallel computing in geography, which was concerned the parallelization of the spatial interaction model. In this example, the predicted flow value from origin to destination is implicitly highly parallel since it can be computed independently. A parallel version of spatial interaction model including 10,764 OD commuting flows in the UK has been run on the KSR parallel supercomputer at Manchester and later ported on to the Cray T3D (Turton and Openshaw 1996). Although ten thousand flows are not considered big any more from today's perspective, this start is quite meaningful to seek help from fast-developing computating technology to handle large geographic datasets and speedup computing processes, and also to stimulate geographers to develop more micro models rather than aggregate, dynamic rather than static, non-linear rather than linear. Today it has become quite common that geographers make use of CyberGIS and GeoComputation techniques such as OpenMP, Message Passing Interface (MPI), MapReduce, Graphics Processing Units (GPU) to facilitate their studies of spatial flows (Wang 2010; Tang et al. 2015).

In addition to solving the data volume issue with advanced GeoComputation techniques, scholars have also been making efforts to overcome an even more urgent intellectual challenge, i.e. to transitioning from deductive reasoning guided by existing theories to inductive reasoning, which allows exploring unlimited possibilities. In particular, it requires developing more data-driven approaches tailored for spatial interaction data (Yan and Thill 2009). Exploratory spatial data analysis (ESDA) is a promising area to make contribution.

1.4 ESDA on Flows: A Promising Area to Make Contribution

ESDA, as defined by (Anselin 1994, 1998, 1998a), refers to a collection of techniques to describe and visualize spatial distributions; identify atypical locations or spatial outliers; discover patterns of spatial association, clusters, or hot spots; and suggest spatial regimes or other forms of spatial heterogeneity. ESDA is a subset of exploratory data analysis (EDA) (Tukey 1977), but with an explicit focus on distinguishing characteristics of geographical data (Anselin 1989). In contrast with confirmatory analysis such as regression modeling, ESDA is extremely useful in assessing the existence and location of nonrandom local patterns in spatial data, but at the same time it is limited by a lack of mechanism to "explain" the observed patterns (Anselin 2000). In geographical research these two types of analytical methods are commonly used in combination. ESDA can be used to explore the data and "suggest" potential associations between variables and elicit hypotheses, especially when the data is not fully understood or the target research questions remain unclear. The suggested patterns or hypotheses can then be formally tested by confirmatory analysis, for example multivariate spatial regression modeling, following a deductive modality (Anselin and Getis 1992; Yan and Thill 2009).

With respect to SI data, most previous research is deductive as spatial structure and spatial interdependencies are very much handled in an ad hoc fashion by spatial interaction modeling (Roy and Thill 2004). However SIM has gradually been found unadaptable to the current "extremely data rich and increasingly computational powerful but theory poor and hypothesis-free environments" (Openshaw 1995). As a new research trend, a number of ESDA methods have been designed for SI data in recent years. And some of them have already been applied to understanding emerging types of flow events such as airline origin-

destination flows (Yan and Thill 2009), taxi pick-up and drop-off flows (Guo et al. 2012; Zhu and Guo 2014; Liu et al. 2015), telephone call (Gao et al. 2013), and spatial interactions embedded in social media (Cao et al. 2015). In general, ESDA methods on SI data can be placed into one or more of the following categories: geovisualization, spatial data mining (SDM), and spatial statistics.

1.4.1 Geovisualization

Visualizing spatial flow data, also referred as flow mapping, has always been an important task or goal of geographers. In a flow map, flows are commonly represented by a number of straight or curved lines connecting origin and destination locations (Zhu and Guo 2014). Accompanied with well-designed color schemes, labels, or symbols, it can be used as a visual analytic method to represent the dynamics of movement between two pair-wise interacting geographical regions (Cao et al. 2015).

The first known map of spatial flows can be traced back to 1837 in which Lt. Harness depicted bidirectional traffic flows between major Irish urban centers (Marble et al. 1997). While the first experiment of flow mapping (migration flows) with the assistance of a computer is done by Tobler (1987). Since then, considerable efforts have been made to design new layout of flow maps, to increase the manageable data size, to enhance the drawing speed, to emphasize important thematic information, and to integrate user-friendly features such as interactive selection and brushing.

Notwithstanding, scholars have soon found that flow mapping is much more complex than a pure cartographic technique. Problems emerge even when mapping a relatively small dataset from today's perspective (Marble et al. 1997). Severe visual cluttering caused by

massive intersections and overlapping of flows easily turns the map unreadable. The reason is that unlike mapping point or polygon data which are discrete spatial objects, mapping flows is to visually represent the dynamic processes or relationships between two sets of geographical locations, which can easily reach a massive size. For example, a single type of flow e.g. migration between the fifty states of the United States has 2,500 different combinations of origin and destination state (including the case when both origin and destination belong to the same state). If migrants are divided into six groups by age, the maximum number of total flow increases to 15,000. The size keeps growing if more categories are added, such as ethnicity, marital status, reason of moving, etc. Moreover, the size of the flow OD matrix increases exponentially when the number of flows' starting or end location grows. Using again the same migration flow example, if upgrading the geographical resolution to the county level or even the census track level, the total flows can possibly be more than 9 million or 5 billion, respectively. Considering the fact that an increasing amount of flow data are collected at the individual level, flow mapping becomes even more troublesome. Traditional cartographic techniques are still valid in flow mapping but it only works for very small datasets only. For example drawing an arrow line to represent each flow and using color, width, and shape of the symbols for flow type and volume. Beyond a nominal dimensionality such flow maps result in clutter problems and a substantial loss of information. A number of geovisualization approaches have been proposed to address this problem for flow mapping.

As a pioneer scholar in this area, Tobler (1987) suggests that information aggregation and removal is an important part of identifying patterns through visualization. For example, Tobler (1987) observes that 75% of migration flow connections on the small side contain

less than 25% of the flow volume. Therefore, filtering out small amount of flows but only visualizing the majority is a common solution. However, the choice of which to keep and which to remove is usually arbitrary and can result in a loss of key information. Visualizing flows by aggregating those with common origin or destination is another common way. The aggregation of endpoint can be based on common large administrative units for example aggregating county level flows to the state level. Tobler (2004) generates a series of flow maps of migration from California between 1995 and 2000 using a computer. In his maps, straight lines with arrows are representing migration flows as the width of lines and arrows correspond to flow magnitude and flow directions. In his later analysis of migration patterns, he chose to show flows starting from one common origin or destination point, e.g. California. Such selective and aggregative visualization methods prove to be effective in many cases, especially when users can interactively choose flows from or to which specific endpoints to explore. When data have accurate numeric coordinates or lack of region information, endpoint aggregation can be processed with other techniques. Andrienko and Andrienko (2011) utilize a point clustering method to group flow origins and destinations before drawing the flow maps. It works extremely well for visualizing discrete flows as it takes advantage of high spatial resolution of the data. Guo (2009) proposes another approach to aggregate locations into regions based on the flow topology with a graph partitioning technique. This approach also manages to discover the natural regions from massive individual flows instead of using pre-defined political boundaries. While endpoint location aggregation is effective at reducing the clutter, it also suffers from the modifiable areal unit problem (MAUP) (Openshaw 1983) as selecting a “perfect” geographic scale or region to aggregate the endpoints is impossible. Aggregating to big

regions would result in the loss of short-distance flows whose origin and destination locate within the same region. While aggregating to small regions may not be able to remove cluttering effectively. Even when there is an appropriate scale for aggregation, using the same one everywhere may smooth out much of the interesting local spatial structure of the spatially heterogeneous flow data.

Another type of flow mapping method is to bundle nearby edges together in order to minimize edge crossing in flow maps. In contrast with previous methods built on flow endpoint locations, edge bundling methods make use of geometric characteristics of flows (edges) directly. Phan et al. (2005) present a method using hierarchical clustering to create a flow tree that connects a source (the root) to a set of destinations (the leaves). Their algorithm attempts to minimize edge crossings to create multiple-source flow maps by preserving branching substructure across flow maps with different roots that share a common set of nodes. Qu et al. (2006) propose an edge-clustering framework by grouping links based on their intersections in the Delaunay triangulation of the endpoints. Cui et al. (2008) later on develop an improved edge-bundling framework with mesh generation method that can better capture the underlying graph patterns than using Delaunay triangulation. This line of method remains active as we see more recent contributions to the literature (Holten et al. 2009; Verbeek et al. 2011; Cao et al. 2015). These flow mapping methods are designed for visualizing datasets containing both hierarchical structures and adjacent relations. The flow maps resulting from edge bundling are usually no longer straight line graphs but in the road-map-style. The limitation of edge rerouting or bundling approaches is obvious. While it improves the overall visual clarity, it inevitably compromises the accuracy of both spatial and attribute information of the data. For

example, the endpoint location and length information of a flow is lost if it is bundled to a set of nearby flows.

As a new trend, more flow-related geovisualization methods are backed by spatial data mining (SDM) techniques or spatial statistical analyses. For instance, one can perform hierarchical cluster analysis on flows first and then visualize only the flow clusters (Zhu and Guo 2014); alternatively, one can detect spatial autocorrelation in the flow dataset and then highlight the only parts with positive patterns on maps (Liu et al. 2015); generalize space-time kernel density of movement points and then utilize volumetric visualization to illustrate the density (Demšar and Virrantaus 2010). Especially in order to visualize a large volume of individual spatial flows, geovisualization is often combined with other ESDA approaches such as spatial data mining and spatial statistical analysis on spatial interaction data, which will be summarized in detail in the following sections.

The purpose of visualization is not only to illustrate the geographic entities themselves, but also to discover underlying stories as a powerful exploratory tool. Therefore, visual analytic methods are frequently applied in studying massive flow data. Yan and Thill (2009) use self-organizing maps (SOM) as the data mining engine of how the characteristics of the air transport system interact with the spatial interaction (SI) system to create relationships and structures within the US domestic airline market. Guo (2009, 2012) also uses SOM in studying migration and vehicle movement. In addition, parallel coordinate plots (PCP) and density maps are applied to explore flow data from more perspectives. Demšar and Virrantaus (2010) introduce a novel concept of 3D space–time density of trajectories to solve the problem of clutter in the space–time cube. The three dimensions

and temporal information make this work noticeable but it requires trajectory data rather than simple OD flow data. The concept of cartogram can also find its application in spatial interaction studies. Thill (2011) introduces the concept of relative space to illustrate migration patterns. In those distorted maps, destination states are allocated further or closer to migration origins, in coordination with different amounts of migrants.

Last but not least, developing handy tools for visualization and data mining on flows is an enduring important task. Beginning with Tobler's Flow Mapper (Tobler 1987; Tobler 2007), many scholars have developed various software applications for this purpose. Well-known cases include Glennon's Flow Data Model Tools in a series of ArcGIS 9 Visual Basic for Applications (VBA) macros (Glennon 2005), Flow Mapping with Graph Participating and Regionalization (Guo et al. 2009), and jFlowMap (Boyandin et al. 2010). Commercial software such as Gephi and VisIt also have flow mapping capability. Nowadays it is the trend that more applications are web-based because it has advantages such as light, portable, and highly-interactive. Some popular open source projects, e.g. R and D3.js, advance very fast in this direction. Popular contributions include FlowingData based on R (http://flowingdata.com/) by Dr. Nathan Yau, and several flow mapping examples of D3.js (http://d3js.org/). As open source projects are getting popular in both academics and industry, more web-based applications of visualizing and exploring flow data will be available in the coming years.

1.4.2 Spatial Data Mining

Spatial data mining (SDM) is the application of data mining techniques to spatial data, in order to discover previously unknown, but interesting and potentially useful patterns from high volume and nonhomogeneous spatial datasets. Due to the nature of the

geographic space and the complexity of spatial data, spatial data mining holds uniqueness from several aspects (Dao 2013). First, spatial objects are embedded in a continuous geographic space, which serves as a measurement framework for all spatial attributes. Second, the two basic effects of spatial data, namely spatial dependence and spatial heterogeneity, violate the fundamental assumption of many data analysis methods that every observation is independent and each process is consistent. Therefore, simply migrating data mining techniques to the spatial domain while ignoring these two effects will end up with biased results with limited external validity. Third, data with spatial dimensions cannot be easily reduced to points without information loss because spatial characteristics such as size, shape, and topological relationships, can have significant influence on the study process. In order to handle these unique features of spatial data listed above, various types of spatial data mining techniques have been developed, including spatial classification and prediction, spatial clustering, spatial outlier detection, and spatial association rule mining.

In terms of spatial interaction data, spatial outlier detection, defined as the technique to extract a spatially referenced object whose spatial or non-spatial attributes appear to be inconsistent with other objects within its spatial neighborhood (Shekhar et al. 2003a), has been intensively applied. Yue et al. (2011) extract the pick-up and drop-off OD flows among one-week GPS trajectories of around 12,000 taxis in Wuhan, China, and single out the trips with destination of shopping mall. With little or no land-use data, such as population, employment, or other survey data, they successfully demonstrate the feasibility of identifying shopping center attractiveness using taxi trajectories and help understand consumer shopping choices, consumer travel (by taxi) behavior, as well as urban traffic

management. A similar data set has been used in another study in which the influences of critical transportation links, i.e. major road bridges, are illustrated (Fang et al. 2012). Three exploratory analysis functions are developed to examine and visualize traffic flows in an integrated spatial and temporal environment, and alternative travel paths for those bridges are identified. Besides consumers' behavior patterns, taxi OD flows are also used to uncover taxi drivers' behavior patterns. Liang Liu and his colleagues (2010) have conducted such research, utilizing a very large dataset including one-year taxi trajectories (48 million trips) in Shenzhen, China. They classify taxi drivers by their income, and relate this to their driving habits mined from millions of trips: such as operation time, average length of single trip, activity space coverage, capability of avoiding congestion, etc. Interestingly they found the "secrets" of top-earning cabdrivers: long operation time; good sense of business (short time intervals between trips); always avoid congestion; choose fastest path rather than shortest or longest ones.

Another SDM technique that has been commonly used in studying spatial flows is spatial cluster analysis. The classical K-means algorithms have been proved very effective with respect to multi-location spatial data (Ossama et al. 2011; Genolini and Falissard 2010). Density-based clustering methods such as DBSCAN (Ester et al. 1996), OPTICS (Ankerst et al. 1999), and their variants have also been adjusted to flow data (Nanni and Pedreschi 2006; Lee et al. 2007; Zhu and Guo 2014) as density-based methods are the most suitable for discovering clusters of arbitrary shapes and filtering out noise. The key of the method is to define a set of distance functions tailored for line-segment that can measure both positional and directional differences. Hierarchical clustering can also be used for flows. For instance, Zhu and Guo (2014) develop an approach that can generalize flows to

different hierarchical levels and has the potential to support multi-resolution flow mapping. In general, spatial flow methods are designed to group observations into "clusters" based on similarity. Unlike directly aggregating flows to predefined regions such as administrative units, cluster analysis methods are able to explore the data and find similar groups of flows that are usually previously unknown. Therefore, the impact of uneven density levels or ad hoc zoning definition of flow endpoints can be handled well. It is worth mentioning that cluster analysis methods are frequently combined with visualization techniques when analyzing spatial flows, as the extracted flow clusters are essential information that deserves visual emphasis on the map.

1.4.3 Spatial Statistics

While spatial data mining techniques are capable of discovering knowledge from large databases, an important question is whether it is possible to derive some understanding, explore relationships and develop hypotheses associated with observed movements and spatial interactions (Murray et al. 2011). In order to further examine these questions in a confirmatory way, spatial statistics are favored owing to their ability to establish inferential properties. The preponderance of the literature on spatial point pattern analysis treats each point as an event independent of all the others. Spatial flow data, however, encompass at least two points (polygons), one corresponding to the origin location (region) or start of the flow and one for the destination location (region) of the flow. Flow data, therefore, differ fundamentally from point data or polygon data and methods designed to handle the points and polygons cannot be directly applied to flow data. Several endeavors have been undertaken in previous research to fill this gap. Berglund and Karlström (1999) applied the G_i statistics introduced by Getis and Ord (1992) and Ord and Getis (1995) to identify local

spatial association in flow data. Although several different spatial weight matrices were proposed in this paper to address spatial non-stationarity, only the simplest binary spatial weight matrix based on identical origins or destinations was implemented, which certainly limits its usage. Lu and Thill (2003) proposed an ad hoc and partially qualitative approach in which they apply point cluster detection methods to analyze origin and destination points respectively, and combine the two sets of results via a relationship table to conclude on the patterns exhibited by the flows. Related issues such as sensitivity to scale and neighborhood definition were discussed in their later work (Lu and Thill 2008). While decomposing one-dimensional flows into zero-dimensional points can considerably simplify the problem, this approach would inevitably overlook the simultaneity of some critical information, such as flow direction and flow length. Murray et al. (2011) departed from this approach by combining exploratory spatial data analysis and confirmatory circular statistics to analyze the similarities of flow direction and length. However, they sacrifice the actual locational information in the process so that little knowledge on spatial relationships between movements can be extracted. More recently, Liu et al. (2015) extended both global and local Moran's I statistics to a flow context, considering movement distances and directions at once. Nonetheless, their approach is still based on the spatial proximity relationship of either set of end points rather than entire vectors. Therefore, it remains within the scope of measuring spatial autocorrelation of vectors/flows in parts rather than as a whole. Novel statistical method that not only fully considers flow characteristics, i.e. end points, length, and direction, but also builds on proper measurement of spatial proximity relationship between entire flows, is anticipated.

The spatial statistical approaches listed above still belong to the broad ESDA family as they aim at exploring the data and detect patterns such as spatial clustering, spatial autocorrelation, and spatial heterogeneity. However, it is still one step away from taking these findings to confirmatory studies in order to reach solid explanatory conclusions. Taking spatial dependence as an example, many researchers have found that this type of spatial effects exists among OD flow data and specifically refer to it as network autocorrelation (Black 1992; Griffith 2007; LeSage 2008; Chun 2008). Implementing traditional spatial interaction models such as the gravity model without taking account of this effect tends to result in incorrect parameter estimation and unsound conclusions as it violates one of the key assumptions of those models, i.e. independence of observations. LeSage (2008) overcomes this problem by proposing spatial weight structures that model dependence among OD flows that are consistent with standard spatial autoregressive models. The spatial weight structures consist of three spatial connectivity matrices capturing origin, destination, and origin-to-destination dependences. Griffith and Chun come up with another solution by using eigenfunction-based filters for accommodating spatial autocorrelation effects within a spatial interaction model (Griffith 2007; Chun 2008; Chun and Griffith 2011). They have proved the effectiveness of using eigenvector filtering to improve spatial interaction modeling in various application cases. Except for common migration flows, scenarios such as journey-to-work commuting flows (Griffith 2009), interregional commodity flows (Chun et al. 2012), and space-time crime incidents flows (Chun 2014) have all been successfully tested. Accounting for spatial autocorrelation in traditional spatial interaction modeling provides an example of incorporation of exploratory spatial statistical considerations into confirmatory hypothesis-testing research.

More efforts should be made along this line to discover the usefulness of interesting findings of emerging ESDA methods.

1.5 Statement of Research

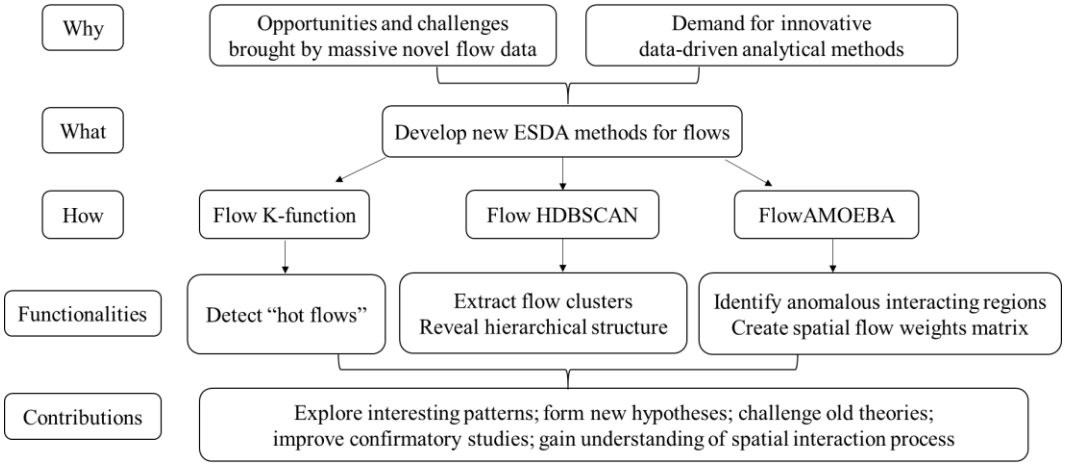


Figure 1: Roadmap of dissertation

Figure 1 is the roadmap of this dissertation showing the motivations, the means, the functionalities, and the contributions. As summarized above, ESDA is tailored for spatial flows. In other words, exploratory spatial flow data analysis (ESFDA) is the solution to grasp the opportunities and overcome the challenges brought by the revolution of spatial flow data. In the meanwhile, as an emerging and fast-developing methodological area, there is plenty of room for new contributions to ESFDA. In this doctoral dissertation research, I propose to develop three unique but closely related ESDA methods tailored for analyzing spatial flow data. These methods are designed for detecting flow's spatial patterns, extracting flow clusters and revealing their hierarchical structure, identifying regions of anomalous spatial interactions and create a spatial flow weights matrix. Their results can further be used to form new hypotheses based on explored interesting patterns,

to gain knowledge of spatial interaction process, and to improve related confirmatory studies.

The first method, named Flow K-function, is a spatial statistical approach to detect spatial clustering patterns of flow data. In other words, it upgrades the classical hot spot detection method namely Ripley's K-function, to the stage of "hot flow" detection. A set of spatial proximity measures is designed for flow data by integrating endpoint location, length, and direction. The measures are capable of assessing both intra-relationships and inter-relationships of flows and serve as the basis of this flow clustering detection approach. Flow K-function is designed to fill the gap that there is no such spatial statistical approach dedicated to detecting local spatial distribution patterns of flow data, in contrast with abundant methods available for point and polygon data. Experiment will be done on Motor vehicle theft and recovery data in Charlotte, NC. The detected "hot flows" link the common theft places and recovery places, which can certainly help law enforcement fight against such crime.

The second one, called Flow HDBSCAN, is a hierarchical and density-based spatial flow cluster analysis method. It is an extension of hierarchical clustering and density-based clustering methods in data mining area to the context of spatial flows. The method not only can extract spatial flow clusters, but reveal their hierarchical data structure if there is any. Through experimentation with both a synthetic dataset and an eBay online trade dataset, the method proves its effectiveness of extracting flow clusters from various situations including varying flow densities and flow lengths. It is also robust to avoid problems like MAUP, false positive errors, and uneven density levels or ad hoc zoning definition of flow

endpoints. Moreover, its sole-parameter design saves the dilemma of parameterization and makes it easier to apply.

The last approach is a data-driven and bottom-up approach dubbed FlowAMOEBA. It upgrades A Multidirectional Optimum Ecotope-Based Algorithm (AMOEBA) from areal data to spatial flow data through a proper spatial flow neighborhood definition. It has two major functionalities. The first one is to identify origin and destination regions that capture anomalous spatial interactions happening in between. It breaks the tradition that spatial interaction data are always collected and modelled between two comparable predefined geographic units, as it delineates the boundaries of anomalous interacting regions, regardless of the size, shape, scale, or administrative level. The second one is to create a spatial flow weights matrix based on the identified regions. The matrix can be used to account for network autocorrelation, thus improving confirmatory studies using spatial interaction modeling. Experiment has been carried out with both synthetic dataset, and a county-to-county migration dataset.

These methods can be used jointly to the same application as well. For given flow data, the “hot flow” detection method Flow K-function can be applied to detect both global and local spatial patterns of the data to examine if there exist anything interesting, i.e. patterns that are statistically significant. Next the cluster analysis method Flow HDBSCAN can be used to extract where the flow clusters locate in what kind of form, and at the same time to reveal the data structure e.g. hierarchical or flat. In order to dig out more detail on the clustered flows, the third method can help accurately quantify and visualize the anomalously interacting places with clear boundaries. Therefore, using these methods

jointly for the same application can take advantage of each method to better solve the problem.

CHAPTER 2: STUDY I. FLOW K-FUNCTION: A "HOT FLOW" DETECTION SPATIAL STATISTICAL METHOD

2.1 Overview

In the first study, I aim to develop a new spatial statistical approach to detect spatial clustering patterns of flow data with the aim of understanding their spatial relationships, while preserving the integrity of the flow data. The general principle is to extend the well-known point data analysis method, namely the Ripley's K-function, to the spatial flow context. In other words, to upgrade the classical hot spot detection method to the stage of "hot flow" detection. To this end, a set of new spatial proximity measures tailored for flow data are designed, which integrate a flow's complete spatial components including endpoint location, length, and direction. The measures are capable of extracting both intra-relationships and inter-relationships of flows and serve as the basis of this flow clustering detection approach. Specific aspects of the method are discussed to provide evidence of its robustness and expandability, such as the multi-scale issue and relative importance control. The experimental dataset consists of a set of vehicle theft and recovery location pairs in Charlotte, NC.

2.2 Motivations

The major contribution of this study to the literature is the innovative "hot flow" detection method itself. The method fills the gap that there is no such spatial statistical approach dedicated to detecting local spatial distribution patterns of flow data, in contrast

with abundant methods that have been continuously developed for point and polygon data. The method also meets the challenges brought by the emerging breadth of massive flow data, as it utilizes the accurate spatial characteristics of individual flows and it also leaves the room to integrate the semantic information. Not only can global spatial patterns of the entire study area be detected, but also the local pattern between different OD location pairs across scales can be revealed by the method as well. Therefore, the results can be easily visualized on maps.

In addition, a set of spatial proximity measures conceived specifically for flow data is critical to the soundness of the approach. Specifically, the measures are created not only to be measures of spatial proximity, but also as an effective solution for the inclusion of the multi-location interaction objects within the scope of well-developed point pattern spatial statistics, namely the local K-function. Unlike approaches treating spatial flows as two separate sets of endpoints, these measures calculate a flow distance that regards flows as inseparable objects. Moreover, controlling for the impact of flow length can be useful and sometimes necessary to avoid the false positive detection of flow clusters so that the measures also include flow length. Last but not least, a pair of coefficients is added to offer some flexibility in measuring real flow data. By adjusting the parameters of endpoint coordinate pairs, the study emphasis can be purposely placed on the spatial associations between either flow origins or flow destinations. The usage of the spatial proximity measures is not limited to this particular approach. For example, other methods of exploratory spatial data analysis such as the local Moran's I and G statistics for flow data analysis can use these measures to calculate spatial weight matrices.

Furthermore, this study contributes to the literature as a case where several ESDA approaches are combined to analyze flow data. Except for the proposed spatial statistics, geovisualization and geocomputation techniques can also be applied to improve the whole analysis process. For instance, the detected hot flows can be illustrated on map showing where exactly cars are frequently stolen and where are they transported to. Police are able to select a common stolen location and view the corresponding common recovery locations, or vice versa. On the other hand, the computing efficiency can also be boosted via high performance computing (HPC) techniques. For example, parallel computing technique OpenMP (Open Multi-Processing) can be easily applied to accelerate the Monte-Carlo simulation of statistical significance test with a prevalent multi-core computing environment.

Last but not least, application usefulness is an important aspect that this new approach can contribute to. “Hot flow” detection can help identify heavily interactive location pairs, and thus help link the commonly stolen places and recovery places. Police can use the preliminary results to further investigate who are the criminals behind the hot flows, and their behavior patterns regarding target, place, and time. Therefore, more effective actions can be taken against gang crime activities. Citizens, on the other hand, can be informed by the results where vehicle-theft crimes are more likely to happen so that they can avoid parking in these high-risk places.

2.2 Related Methods

In spatial analysis, cluster detection is an approach to second-order analysis that is designed to examine spatial dependence, or spatial relationships between events (Getis and Franklin 1987). The first step is to choose an appropriate measure of spatial proximity

between events, for which distance is a common choice. Ripley's K-function, the Geographic Analysis Machine, the Nearest Neighbor Index and many other statistical approaches are all distance-based methods. Aside from the default Euclidean distance, other kinds of distance are also applied in some cases, for instance the network distance (Yamada and Thill 2007). With spatial flow data, there is no natural mean to measure spatial proximity due to the multi-location nature of flow records and this is arguably the biggest difficulty in analyzing spatial patterns of flow data. In other words, with appropriately measured spatial proximity, cluster detection on flows boils down to the same algorithmic processes as for points or polygons. Although various distance measures have been proposed in data mining studies of trajectories, for example using the Hausdorff distance to extract clustered line segments of trajectories (Lee et al. 2007; Chen et al. 2011), I argue that these distances are not suitable to measure proximity between flows which have explicit and meaningful location correspondence. Accordingly, we devise a new proximity measure called the "Flow Distance" and a variant called the "Flow Dissimilarity". Then I extend a well-developed spatial point statistic, namely Ripley's K-function, to the spatial flow context based on the newly defined proximity measures. Inferential properties are established through significance tests by Monte Carlo simulation against the null hypothesis of spatial randomness. Several aspects such as the multi-scalar relevance, relative importance control, and flow value, are discussed in detail here to demonstrate that this method is versatile and practical.

2.3 Method in Detail

2.3.1 Flow Model

The first step is to define the study object, namely the spatial flow process. Figure 2 shows two instances of a spatial process *F* that starts at location *O* and ends at location *D*. Basic characteristics of *F* include length: $l = |\overrightarrow{OD}|$; direction: same as the direction of vector $\overrightarrow{OD}$; type: *T* (e.g. commuting flow); and value *W* (e.g. the number of commuters). This basic model is used to represent spatial flow processes in the rest of the chapter.

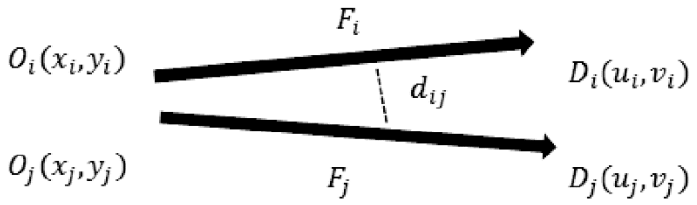


Figure 2: Basic flow model

2.3.2 Flow Proximity

As mentioned earlier, defining an appropriate proximity measure is key to decoding spatial flow patterns. Here I introduce such measures based on which both intra-relationships and inter-relationships of flows can be extracted.

Let us take the simple case of measuring the spatial proximity between flow F_i (with origin point O_i (x_i ,y_i) and destination point D_i (u_i ,v_i)) and flow F_j (from point O_j (x_j ,y_j) to point D_j (u_j ,v_j)) in a two-dimensional space (Figure 2). Measuring distance between these two spatial flows following the approaches advocated so far in the literature would generally be inadequate because distance between either origin points or destination points cannot fully represent the closeness between flows in their entirety. For instance, when

both origins are a short (or long) distance to each other and the same can be said of destinations, it is expected that F_i and F_j are also close (or distant, respectively). However, things become less trivial when the two endpoint pairs show dissimilar spatial closeness, i.e. origins are close while destinations are distant, or vice versa. Using categorical descriptions is certainly one way to associate distances among origins and destinations. For instance, both distances being short (or both endpoint pairs belong to the same region) would correspond to "high" spatial association between flows while only one pair of end points being close (or belonging to the same region) would correspond to a "medium" degree of association (Berglund and Karlström 1999; Lu and Thill 2003; Zhu and Guo 2014). While such approaches make sense to some extent, they are very sensitive to the ad hoc description standards and exhibit limited external validity.

Unlike approaches treating spatial flows as two separate sets of endpoints, I propose to calculate a flow distance that regards flows as inseparable objects. A flow process F_i with origin point O_i (x_i, y_i) and destination point D_i (u_i, v_i) can be seen as a vector point with four coordinates F_i (x_i, y_i, u_i, v_i) in a four-dimensional space. Derived from the general function of Euclidean distance I define the Flow Distance between flows F_i (x_i, y_i, u_i, v_i) and F_i (x_j, y_j, u_j, v_j) as Equation (1):

$$FD_{ij} = \sqrt{\alpha[(x_i - x_j)^2 + (y_i - y_j)^2] + \beta[(u_i - u_j)^2 + (v_i - v_j)^2]}.$$

$$or\ simplify\ as:\ FD_{ij} = \sqrt{\alpha {d_O}^2 + \beta {d_D}^2}. \quad (1)$$

where FD_{ij} denotes the distance between these two flows; d_O and d_D are the Euclidean distances between the two origins and two destinations, respectively; the coefficients α and

β serve to control the relative importance of either sets of endpoints ($\alpha > 0$; $\beta > 0$; $\alpha + \beta = 2$; by default $\alpha = \beta = 1$). Through this definition, both the closeness of origins and of destinations make a contribution to the calculation of the Flow Distance. For example in Figure 3a, $FD_{12} = \sqrt{2^2 + 2^2} = \sqrt{8}$. The value of Flow Distance becomes larger (or smaller) if both endpoints are moved further (or closer) to their counterpart at the same time, e.g. FD_{12} increases to $\sqrt{18}$ in Figure 3b while it decreases to $\sqrt{2}$ in Figure 3c. This corresponds to the general sense that proximities of endpoints are positively correlated to the flow closeness.

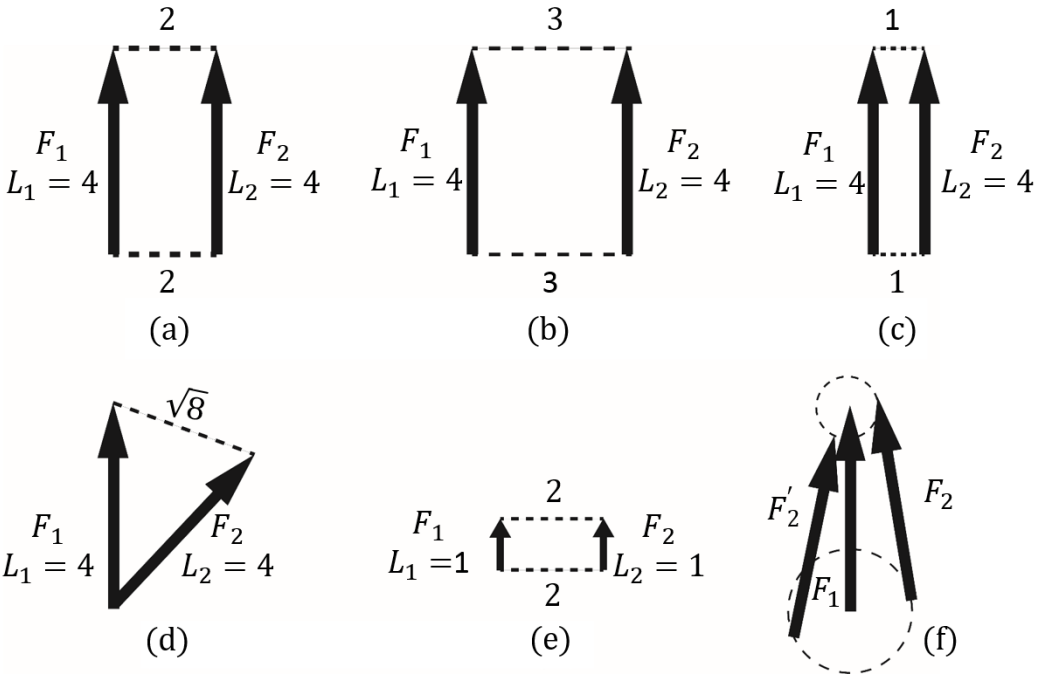


Figure 3: Flow distance examples

More importantly, the distance between origins and the distance between destinations are integrated by the same square root transformation so their variations are captured continuously and consistently, which leads to greater accuracy than qualitative descriptors.

For instance, compared with Figure 3a, Flow F_2 in Figure 3d has its origin moved towards F_1's and has its destination moved away from F_1's. According to previous methods, whether these two flows in Figure 3d are as close as they are in Figure 3a completely depend on the definition of endpoint's contiguity relationship. In other words, if two points are defined as contiguous when their distance is less than or equal to 2, F_1 and F_2 would have two contiguous endpoint pairs in Figure 3a but only one in Figure 3d. As a result, the proximities between F_1 and F_2 are radically different. In contrast, by our definition of Flow Distance, measuring proximity between two flows is not subject to the definition of endpoint's own region or the description of the combined endpoint's closeness. Instead, I capture the variation of all locations seamlessly and let the flow data decide its own spatial neighbors for itself. Accordingly, the distance between F_1 and F_2 can be calculated and compared directly as FD_{12} equals$\sqrt{8}$ in both Figure 3a and Figure 3d scenarios.

Nevertheless, only using the location information of endpoints may sometimes be inadequate because a flow does not only represent the interaction or movement between two locations, but also indicates how far and in what direction the interaction or movement happens. As shown in Figure 3e, two flows have exactly the same endpoint distances as Figure 3a, therefore the Flow Distances are the same according to Equation (1). It would be controversial to say that the two flows in Figure 3e are as close as the ones in Figure 3a given that they are separated much more, relative to their lengths. Controlling for the impact of flow length may be necessary to avoid false positive detection of flow clusters. To this end, I propose an extended version of Flow Distance that involves a rescaling, as provided by Equation (2). By dividing by the geometric mean of two flow lengths, a flow pair with longer average length would be measured closer, ceteris paribus. Therefore, the

distance between the short flows F_1 and F_2 in Figure 3e becomes four times longer as the one in Figure 3a. The rationality behind this adjustment is that under many circumstances it is more difficult or rarer to witness spatial interaction or movement happen between two distant locations than close locations. For example, wild animals are more likely to travel to a nearby river than a distant one to seek water. Incorporating flow length into the measure is one way to adjust the criterion of clustering detection for flows with unequal lengths. Given the adjustment would impair some of the metric properties of distance, I name the adjusted Flow Distance as Flow Dissimilarity, short for FDS in the rest of this paper. Also, I choose to use the geometric mean over the arithmetic mean of flow lengths because the former is more capable to attenuate the impact of extremely unequal length values. In addition, it avoids the limit case of zero-length flows.

$$FDS_{ij} = \sqrt{\frac{\alpha[(x_i - x_j)^2 + (y_i - y_j)^2] + \beta[(u_i - u_j)^2 + (v_i - v_j)^2]}{(L_i L_j)^\gamma}}.$$

$$or: FDS_{ij} = \sqrt{\frac{\alpha {d_O}^2 + \beta {d_D}^2}{(L_i L_j)^\gamma}}. \quad (2)$$

Similar to Equation (1), d_{Oij} and d_{Dij} refer to the Euclidean distance between origins O_i and O_j, and between destinations D_i and D_j, respectively. L_i and L_j are flow lengths. The rationale is that this metric integrates all the spatial elements of a flow, i.e. a pair of endpoints, length, and direction (implicitly). The numerator leverages the accuracy of endpoint coordinates and captures the variation of distances continuously and consistently. The denominator assigns advantage on longer flows given that under most circumstances spatial interaction between distant locations is scarcer due to the "friction of distance"

between origin and destination. The exponent γ offers flexibility to account for the effect of flow length. By default I assign $\gamma = 1$ by using the geometric mean of two flow lengths as the denominator. In addition, Equation (1) can be seen as a special case that $\gamma = 0$ when considering no effect of flow length.

Although considering flow length in spatial pattern detection can be very useful and sometimes necessary, I am not arguing that this is a better approach in all situations. Instead, I believe that they both make sense under certain circumstances. Evidence can be found in literature that flow length was not discussed in some research (Lu and Thill 2003 and 2008; Berglund and Karlström 1999; Zhu and Guo 2014), while it was taken into consideration in some others (Murray et al. 2011; Liu et al. 2014). In this research experiments have been conducted with both Flow Distance (Equation 1) and Flow Dissimilarity (Equation 2) for comparison, and details are provided in the case study section below.

Besides endpoint locations and flow length, the only remaining spatial element of a flow is its directionality. Although I do not directly measure directionality in Equation (1) and (2), its impact is implicitly accounted for. As illustrated in Figure 3f, to maintain F_2 at the same distance from F_1, according to our Flow Dissimilarity equation it is sufficient to keep its origin and destination at a constant distance from F_1's two endpoints, i.e. to keep its endpoints situated on circles centered on F_1's two endpoints (the dashed rings), e.g. F_2'. Given this geometric constraint, there are in fact few degrees of freedom in directionality for flows that exhibit a tendency towards clustering. Therefore, I argue that it is not necessary to discuss flow direction alone since it is heavily dependent on the endpoint locations and flow length. Our test results have also demonstrated this argument by identifying clusters of similar-direction flows.

Finally, the coefficients (α; β) in the distance and dissimilarity functions are designed to offer some flexibilities in measuring real flow data. The basic functions by default (α = β = 1) assign equal importance to the origin location and destination location of each flow. However, the research objectives may lead us to pay closer attention to one set of endpoints over the other. For instance, in a study of settlement of foreign immigrants in New York City in relation to national origin, socio-spatial patterns and processes would be better informed if more weight is put on where immigrants choose to reside rather than where they come from. As another example, the manager of a shopping center would be more interested in where customers come from so that more targeted and effective advertising strategies can be designed. The inconsistent spatial scale of flow origins and destinations may be another justification to rebalance the relative importance of origins and destinations in the Flow Distance and Dissimilarity measures. For example, different land uses are known to be spatially distributed differently across cities; in particular employment sites tend to be more clustered geographically than residential land uses. Therefore, to avoid a statistical bias, a spatial analysis of commuting flows should control for the spatial distribution of potential flow origins and destinations. With appropriate calibration, the same distance (e.g. 500 meters) would have the same impact on describing the proximity between two origin locations or between two destination locations.

By adjusting the values of α and β, the Flow Distance or Dissimilarity can receive different contributions from origins and destinations. For example, if assigning α = 1.5 and β = 0.5, the Flow Distance or Dissimilarity would be more sensitive to the change of origin locations and the corresponding spatial pattern would put more weight on where flows start. In addition, I restrict that $\alpha + \beta = 2$ to ensure the results with different coefficients are

comparable. They both must also have positive value to match the reality of flow datasets rather than points.

2.3.3 Flow K-function

Using our Flow Distance (or Flow Dissimilarity) as the spatial proximity measure, it becomes possible to apply well-developed distance-based methods to detect spatial clustering in flow data. In this study, I choose to adjust the original and local version of Ripley's K-function. As a classical clustering detection method, the K-function has been continuously implemented and enhanced since it was redefined by Ripley in 1976 (Ripley 1976; Okabe et al. 2007). The fundamental idea of the K-function is to count the number of events within a certain distance threshold of randomly selected event locations. This number is then used to calculate the K-function value after dividing by the event density; the analysis is repeated for other distances within a set interval. To obtain statistical conclusions, the K-function value needs to be compared with the expected value given by the null hypothesis, for example Complete Spatial Randomness (CSR). If the observed value is higher than expected, the study events exhibit a tendency toward clustering; or dispersed, if it is lower. Monte Carlo simulation is a frequently applied technique to assess statistical significance (Openshaw et al. 1987). One of the meaningful extensions of K-functions was introduced by Getis and Franklin (1987) based on second-order neighborhood analysis of mapped point patterns, which has been known as local K-function analysis.

Here I adjust the original K-function as Equation (3) to calculate the global flow clustering pattern of the study area, and adjust the local K-function as Equation (4) to detect the local flow clustering pattern or "hot flows". Instead of counting point events, flow

events are counted within a certain Flow Distance (or Flow Dissimilarity) r of flow Fi to represent the function value:

$$K(r) = \frac{A}{n} E(number\ of\ other\ flow\ events\ within\ r\ of\ an\ arbitrary\ flow). \quad (3)$$

$$LocK_i(r) = E(number\ of\ other\ flow\ events\ within\ r\ of\ flow\ i). \quad (4)$$

where $K(r)$ is the original or global K-function value at scale r. A is the area of study region and n is the total number of flow events. $LocK_i(r)$ is the local K-function value of flowF_i at scale r. The scale r, also known as the detection window radius or threshold distance, has always been a crucial factor in spatial statistics, especially the K-function, which is even known as "multi-distance cluster analysis" (Boots and Getis 1998). In this approach I implement the local K-function at multiple scales as well. By increasing the magnitude of scale r within a certain range deemed suitable to the process under study, e.g. from 0.1 mile to 1 mile when using Flow Distance or from 0.1 to 1.0 when using Flow Dissimilarity, it is convenient to detect multi-scale clustering patterns at once.

As with other spatial statistical methods, statistical inference is an important part of reaching any conclusion. Given the nature of flow data, normal approximation is not an appropriate null hypothesis (Lu and Thill 2003 and 2008; Liu et al. 2014). Random permutations with Monte-Carlo simulation can better serve this purpose. In a two-dimensional space, there is normally more than one way to simulate a set of flows. On the one hand, it can be proceeded by setting the location of two endpoints for each simulated flow. Alternatively, I could use observed flows as objects and move or rotate them in the study area according to some randomization procedure. Whatever the technique used, the theory or basic assumptions behind the simulation must be fully spelled out.

The simplest way is to simulate two sets of points randomly and independently based on a Poisson distribution, and then pair and connect them as flows. However, the customary null hypothesis for point data, i.e. Complete Spatial Randomness, may not be the best option for flows. A more sensible way is conditional spatial randomness, which has been used widely for computing the pseudo P-value in spatial statistics (Anselin 1995). In terms of flow data, the "condition" should be considered when the endpoints are restricted to the distribution of an at-risk population. For instance, to simulate commuting flows according to residence distribution and workplace distribution (Lu and Thill 2003); to simulate car accident points on the road network and adjust by annual average daily traffic (Yamada and Thill 2010). In addition to endpoint locations, the distribution of flow length and flow direction can also be conditional. Liu et al. (2014) simulate a set of flows by moving one flow to another randomly selected flow's endpoint location so that only flows' locations are changed while the lengths and directions are kept the same. They propose another way by randomly pairing two points, one from observed origins and the other from observed destinations, to form simulated flows. This approach keeps endpoint locations the same but reshuffles the lengths and directions as opposed to the first approach. In sum, there is no unique way to simulate spatial flows for significance testing. It is subject to the data to make an appropriate assumption (e.g. restricted to at risk population). In addition, it is up to the analyst to choose which aspect to examine (e.g. to examine the contribution of flow location to the general flow clustering pattern by only randomizing locations, while fixing direction and length). Fundamentally, cluster detection is an exploratory method of analysis. The clusters identified can reflect the respective underlying geographical processes and can

also help us contemplate unknown ruling attributes contributing to the spatial pattern. The detailed algorithm is presented step by step as follows.

2.3.4 Algorithm Steps

(1) Calculate Flow Proximity

(a) Prepare flow events as vectors with the coordinates of origin and destination points. For example, flow F_i with origin $O_i\ (x_i, y_i)$ and destination $D_i\ (u_i, v_i)$ is formatted as $F_i(x_i, y_i, u_i, v_i)$.

(b) Apply Equation (1) or (2) to calculate the Flow Distance or Flow Dissimilarity between every two flows. Thus an N by N distance matrix is computed for subsequent use.

(2) Calculate clustering detection statistics.

Calculate global and local K-function using Equation (3) and (4) for all the flow events using a series of scales r_i (t = 1, 2, …, 10; $r_i = r_1 \times i$). The unit of $r_1$is chosen on the proximity equation used in previous step.

(3) Evaluate statistical significance.

(a) Randomly simulate a set of N flows in the study area.

(b) Calculate global and local K-function value for each simulated flow same as step (1) and (2).

(c) Repeat previous two steps n times (e.g. 1,000 times for 0.1% significance level).

(d) Sort results of the n-time simulations for each flow at each scale. Set the smallest and largest ones as the lower and upper envelopes.

(e) Compare the actual result with the corresponding significance envelopes. If the observed value surpasses the upper envelop, or is below the lower envelope, the observed pattern is said to be clustered or dispersed, respectively.

(4) Visualize and discuss the results.

2.4 Experiment

2.4.1 Data Description

In this study, I test the new flow K-Function method and its algorithmic implementation using a dataset of vehicle theft and recovery location pairs in Charlotte, North Carolina. Given the determinate relationship and chronological order of the data, the locations where theft happened and the places where the vehicles were recovered can be regarded as flow origins and destinations, respectively. According to the crime report released by the Charlotte-Mecklenburg Police Department (CMPD), there were 14,064 vehicle theft cases within the city from 09/01/2008 to 08/31/2014. Of all these cases, 6,960 have correct corresponding recovery locations somewhere else in the city. In the data cleaning process, I excluded the records with identical theft and recovery locations to exclude the cases of attempted break-ins, damage to the vehicle, interrupted stealing, or other incomplete theft crimes. The final study dataset consists of 6,810 theft-recovery flow events. From the map shown as Figure 4 we can observe the distribution of these locations. Overall, both theft and recovery locations have similar distribution across the city: there is a concentration around the city center, except for the southern portion, which is known to encompass more affluent neighborhoods.

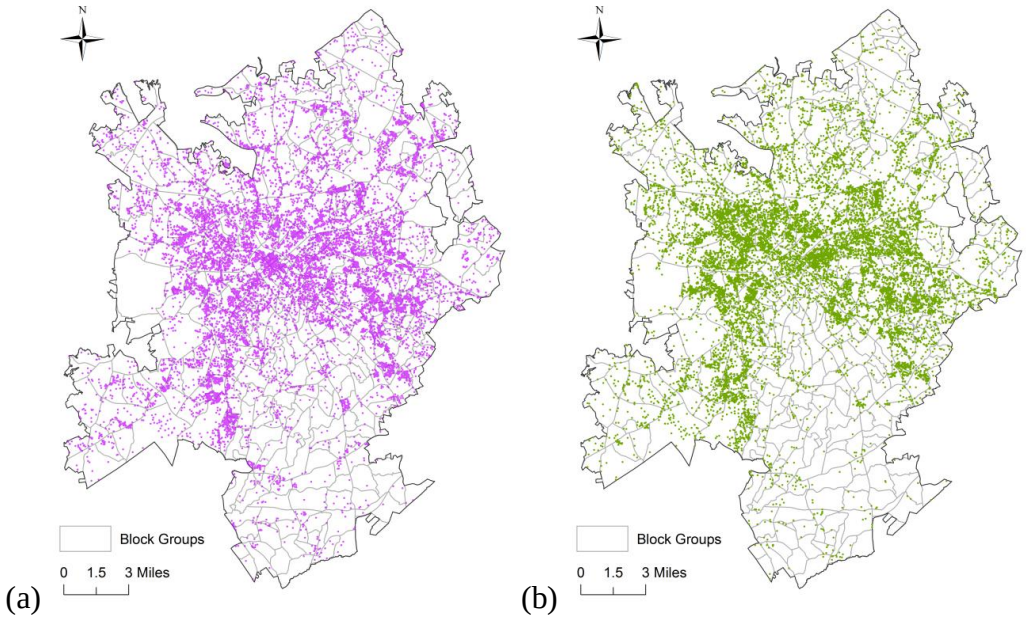


Figure 4: (a) Vehicle theft locations in Charlotte; (b) vehicle recovery locations in Charlotte

To gain a more intuitive knowledge of the data I also estimated the kernel density (KDE) for both sets of locations. The hotspot pattern is most obvious on the map with a cell size of 400 square feet and bandwidth of 0.5 mile (Figure 5). The KDE maps indicate that many car thefts happened in the eastern and northern areas near the city center, while a significant part of them were recovered in the northwestern region, where Charlotte Douglas International Airport is located.

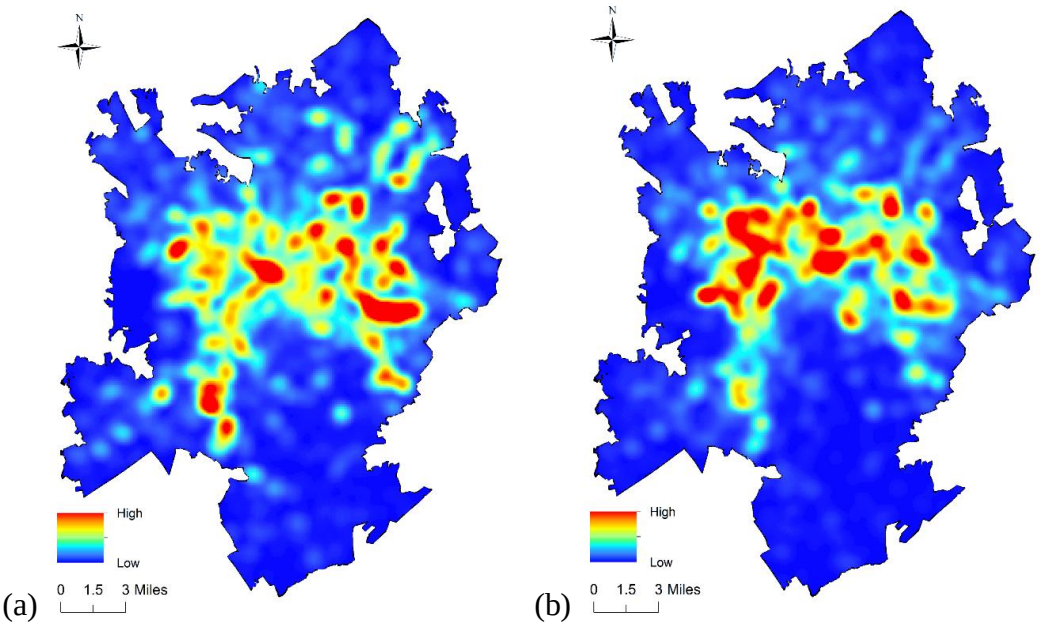


Figure 5: Kernel density estimation of (a) theft locations; (b) recovery locations

However, based on point pattern analysis only, it is difficult to build connections between theft locations and corresponding recovery locations. According to popular criminological theories of vehicle theft crimes, such as rational choice theory and routine activity theory, most criminals have meticulously designed their target places and destination places in advance based on their cost-benefit analyses (Lu 2008). As the new trend indicates, more vehicles are stolen by criminal gangs for money-making business rather than joy-riding (McGoey 2000). Thus, it would be extremely useful to discover the spatial patterns of how stolen vehicles are transported from their offense place to their destination.

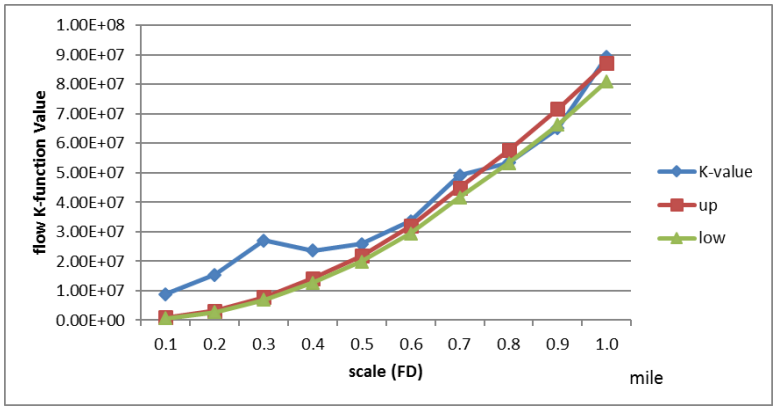
Following the complete algorithm given in the previous section, I implement this flow clustering detection approach on these crime data step by step. The null hypothesis of flow distribution is that car thefts and recoveries can happen anywhere on the street network within the Charlotte city limits. Therefore the 1,000-time Monte-Carlo simulation is

proceeded by randomly locating flows' endpoints on the city's street network. The reason to choose such assumption is that there is little prior knowledge about motor vehicle theft crime to add more restrictions to the distribution of car theft and recovery event locations, or to the flow lengths and directions. Not imposing constraints on the spatial characteristics of flows in the simulation process has the advantage of not excluding any possible contributions to the final cluster results. Edge effects are corrected by reducing the analysis area by a distance equal to the largest distance band used in the analysis (1 mile in this case study). Only the flows with both endpoints within this shrunk area are selected to computing the algorithm, while the background flow spatial process and the simulated flows remain within the original area. The implementation program is written in C/C++ and parallel computing technique OpenMP is also applied to accelerate computation, especially the simulation part. Results are visualized via software ArcMap 10.1 and jFlowMap (Boyandin et al. 2010).

2.4.2 Results and Discussion

Figure 6 shows the global Flow K-function results using two different flow proximity measures. With Flow Distance (Equation 1), the global Flow K-function value is above the upper envelope at the 0.1% significance level at several scales (Figure 6a). Especially at small scales such as 0.1, 0.2, and 0.3 mile, the clustering patterns are very significant. The global Flow K-function using Flow Dissimilarity (Equation 2) also reflects the flow distribution pattern of the entire study area across scales (Figure 6b). The two results are not dramatically different. However, global Flow K-function values are meaningful to indicate at which scale(s) the clustering pattern of the study area is more significant. Hence emphases can be anchored at these scales when looking at the local patterns.

(a)



(b)

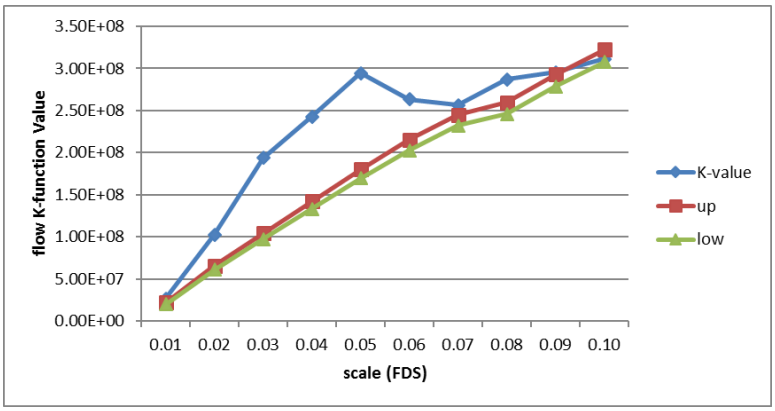


Figure 6: Global Flow K-function results using different flow proximity measures at the 0.1% significance level. (a) uses Flow Distance (Equation 1) with detection scale equal from 0.1 mile to 1 mile. (b) uses Flow Dissimilarity (Equation 2) with detection scale

Figure 7 shows the local flow clusters detected with our method at selected scales according to the global Flow K-function results. The flows on the maps represent the local clusters detected by our new approach as significant at the 0.1% level. Each flow has one end colored in red to denote the theft location and the other end in green to show the recovery location. In order to avoid visual clutter, I aggregate nearby flow clusters into the census block groups where their end points are situated.

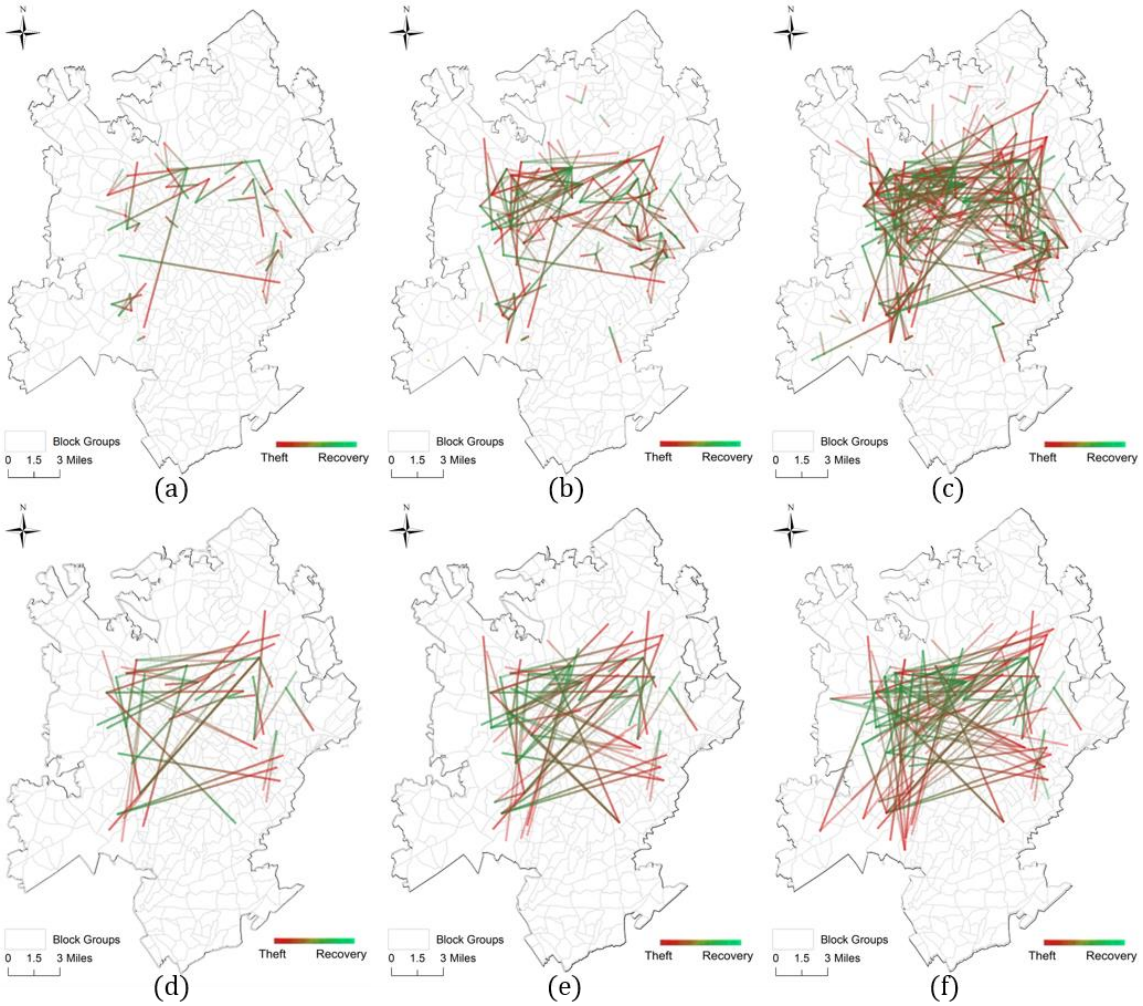


Figure 7: Detected flow clusters using different flow proximity measures. (a), (b), (c) use Flow Distance (Equation 1) with detection scale equal to 0.1 mile, 0.2 mile, and 0.3 mile, respectively; (d), (e), (f) use Flow Dissimilarity (Equation 2) with detection scale equal to 0.03, 0.04 and 0.05, respectively

The results are analyzed from two aspects. First, I compare the results obtained using the same equation of flow proximity measure, but different parametrizations. The first three results use Flow Distance with scale of different magnitudes, i.e. 0.1, 0.2, and 0.3 of a mile. As the magnitude of the scale increases, more flows are detected as local clusters. The same pattern can be found in the other set of results using Flow Dissimilarity. The variance caused by scale magnitude is consistent with the basic feature of the K-function that the

spatial pattern is partly dependent upon the size of the detection window. The increasing number of local flow clusters indicates that more nearby flows are included to contribute to the local K-function value as the detection window becomes larger. At the same time, the increase of scale does not have an equivalent impact on the background distribution which represents our null hypothesis. It is because I simulate the background distribution by randomly placing the flow events on the street network without further specific control, e.g. crime risk; therefore the simulated flows are distributed more sparsely throughout the city. As a result, the increase of scale has a positive impact on the number of local flow clusters that are detected. As in other K-function related research, choosing the optimal magnitude of scale remains an open question. It is typically selected in relation to how the results can make sense to explain context-dependent research questions. In this case, Figure 7f presents some interesting patterns about vehicle theft and recovery flows. Vehicles stolen from the area in the Southwestern section of the city are usually found somewhere far away and their transport directions vary considerably. In addition, there is another group of clusters in the Southeast showing much shorter transport distances and with similar directions towards the North. One possible reason is that for the vehicles stolen in the Southwest area there are only a few “favorable” places nearby for criminals to dispose of them. Therefore, these cars are transported over a long distance to places like chop shops for selling or to places like the airport. Routine criminals who steal from the Southeast area may find it much easier because there are sites nearby in the North to dispose of the cars.

On the other hand, it is useful to compare the results using different types of flow proximity measures, namely the Flow Distance and Flow Dissimilarity. Comparing the two series of maps in the top and bottom row of Figure 7 for a similar number of local clusters,

the most obvious difference is the average length of clustered flows. The results using Flow Distance contain many short flows, while the results using Flow Dissimilarity tend to indicate longer flows as local clusters. Taking a closer look, it is clear that some flows, especially shorter ones within the same cluster identified using Flow Distance do not share many geographic and geometric similarities with their neighboring flows, e.g. quite different flow directions and flow lengths. In contrast, flows within the same cluster using Flow Dissimilarity tend to be very similar to each other. The reason behind this difference is that, when flow length is not considered in measuring flow proximity, short flows need not be as similar in endpoint locations, length and direction to each other as longer ones to have the same flow distance. Therefore, they are more readily detected as the locus of a significant cluster than long ones, all other things being equal. It results in false positive detection since some flows are detected as local clusters simply because they are short enough to be captured by the detection window.

On the contrary, local clusters identified with Flow Dissimilarity include flows with close vehicle theft sites, close vehicle recovery sites, and similar movement directionality and distance. for selected scales within the range of statistical significance. The pattern is consistent throughout the study region. Moreover, the results would be of practical use to law enforcement agencies to detect routine gang-related crimes with locational preference for stealing and selling/disposing of vehicles in the city. As a conclusion, I argue that the algorithm using Flow Dissimilarity to measure flow proximity is less likely to lead to false positive errors as it controls for one source of spurious cluster detection. Besides, it provides a meaningful alternative to the traditional distance scale in solving the instability or inequality in cross-scale flow clustering detection.

So far I have only discussed experiments with the basic version of the flow proximity measures. Further usefulness of the measures can be explored by changing its parameter value. In both Equation (1) and Equation (2), I specify two coefficients, i.e. α and β, to control the relative importance of origins and destinations. The expectation is that changing the relative value of these coefficients can purposely create a tendency for alternative cluster detection results. To test this hypothesis, I adjust the approach by changing the coefficient values in Flow Distance. I assign $\alpha = 1.5$ and $\beta = 0.5$ for the first group and $\alpha = 0.5$ and $\beta = 1.5$ for the second. The sum of the coefficient values is controlled as 2, for the sake of the comparability of the results.

Figure 8 includes two comparable result maps. Figure 8a shows the clusters detected by the Flow Dissimilarity with $\alpha = 1.5$ and $\beta = 0.5$, while Figure 8b shows the outcomes setting $\alpha = 0.5$ and $\beta = 1.5$, both using Flow Dissimilarity measure with a scale equal to 0.04. Comparing these two maps and also comparing them with Figure 7d for which $\alpha = \beta = 1$ by default, I find that Figure 8a contains more unique clusters with very close theft locations (red end) but relatively distant recovery locations (green end), while Figure 8b tends to show the opposite pattern. In other words, flows with close theft locations are easy to be detected as clusters in Figure 8a and flows with close recovery locations are favored in Figure 8b.

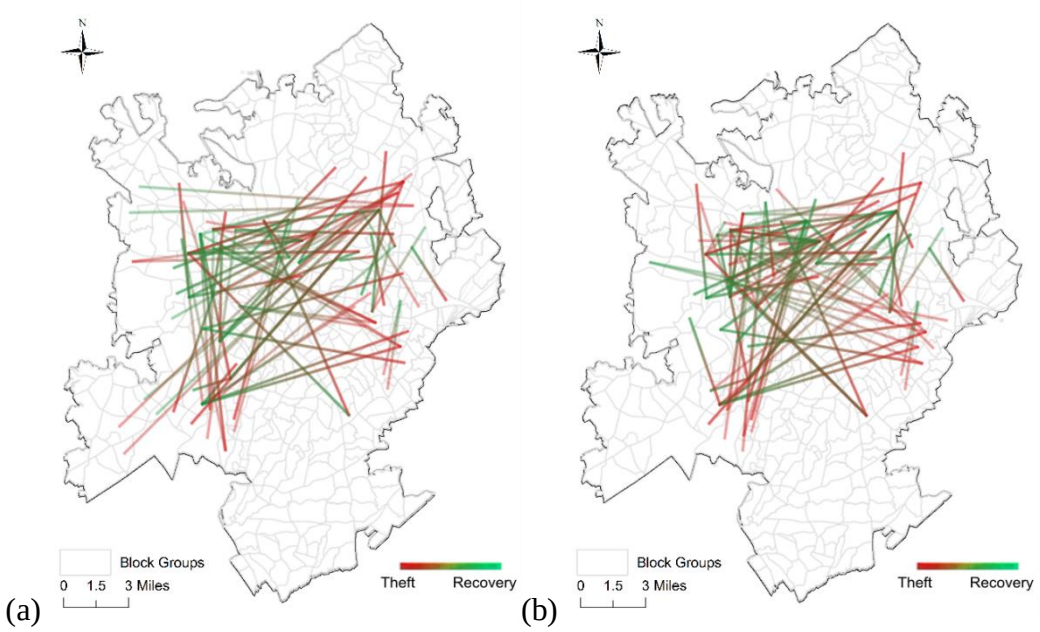


Figure 8: Flow clusters with different endpoint emphases. (a) clusters more focused on theft locations ($\alpha = 1.5;\ \beta = 0.5$); (b) clusters more focused on recovery locations ($\alpha = 0.5\ ;\ \beta = 1.5$)

These observations are in line with our premise that changing the value of Flow Distance coefficients can lead to results with different emphases, which can cater to people with different interests. In terms of practical usefulness, citizens would be more interested in looking at Figure 8a which can inform where vehicle-theft crimes are more likely to happen so that they can avoid parking in these highly risky places. On the contrary, police would find Figure 8b more useful in order to know where the concentrations of car-disposal places are and where they should search for the lost vehicles. By comparing the result maps with Google Maps I found that the neighborhoods surrounding the main campus of UNC Charlotte correspond to the cluster of theft sites in the northeastern part of Figure 8a, which indicates that this area is a popular car theft locus. Some clusters of recovery places near the city center in Figure 8b match the locations of savage vehicle yards or chop shops, where stolen cars can be quickly transacted with cash and be sold again in parts.

2.5 Summary and Future Directions

Spatial statistical approaches to clustering detection have been continuously developed for decades. In contrast with abundant methods designed for point and polygon data, approaches well suited to handling spatial flow data have not been well developed so far. To fill this gap and also to meet the challenges brought by the emerging breadth of massive flow data, this research has developed an innovative spatial statistical method for flows. A pair of particular spatial proximity measures called the Flow Distance and Flow Dissimilarity have been designed. Based on these measures the original and local version of the K-function is adjusted and implemented to examine the second-order effects of spatial flows. The global Flow K-function indicates at which scale(s) the flow clustering pattern of the entire study area is significant. By comparing the observed local K-function value with the statistical confidence envelops generated via Monte Carlo simulation, the local clustering pattern of each flow event can be identified at a certain statistical significance level. The new method is an intuitive extension of the principles embedded in the K-function for one-dimensional point events and is applicable to all types of flow data.

To test the effectiveness and usefulness of our method, a series of experiments have been implemented using a real dataset of vehicle theft-recovery flows in Charlotte, NC. The results demonstrate that our method is capable of identifying local clusters from the several thousands of tangled flows. Specifically, the measures I designed proved not only to be measures of spatial proximity, but an effective solution for the inclusion of the multi-location interaction objects within the scope of well-developed point pattern spatial statistics, namely the local K-function. By adjusting the parameters of endpoint coordinate pairs, the study emphasis can be purposely placed on the spatial associations between either

flow origins or flow destinations. In addition, the impact of flow length has also been thoroughly discussed. To overcome the statistical bias brought by flow lengths, I introduced a variant of Flow Distance called Flow Dissimilarity. The experiment shows that the algorithm using Flow Dissimilarity leads to more stable spatial patterns and is adaptive to flows with varied lengths across the study region. Overall, the method designed in this research has fully utilized the spatial characteristics of flow data, and it is demonstrated to be capable of investigating spatial associations of flow events across scales. The results examined with this method have practical implications as well. In this vehicle-theft crime example, it can inform not only where frequent car theft and recovery happen, but how the stolen cars are moved from one place to another in the form of spatial flow clusters. The results are especially useful to devise effective police responses to routine gang crime activities.

The proposed analytic method can be extended in several ways. First, further work can be done to expand the capability of this method to include additional event characteristics, for example considering flow type and value in "hot flow" detection. A plausible idea is to use the local cross K-function (Boots and Okabe 2007) instead of the traditional local K-function to detect clusters of flows with different types, e.g. rescue goods flow spatially associated with refugee flow; and to accumulate the total value of nearby flows instead of simply tallying their frequency in calculating the local K-function so as to adjust the contribution of flows with unequal value, e.g. a one-thousand-people commuting flow versus a single-person commuting flow. Also, the Flow Distance and Flow Dissimilarity measures can be shown to be effective with other methods of exploratory spatial data analysis including the local Moran's I and G statistics for flow data analysis. Equation (1)

and Equation (2) can also be modified by using other types of distance such as network distance to calculate d_O and d_D, in order to better solve context-related questions.

Furthermore, the principles of the flow proximity measure can be further expanded to higher dimensionality for the space-time analysis of flow data. For example Equation (3) below is an extension to Equation (2) to measure the spatiotemporal dissimilarity between two flows, where t_{Oi} and t_{Oj} denote the starting time of two flows, $t_{Di}$and t_{Dj} are the ending time of these flows (after normalization). With the same rationale of adjusting flow length, the temporal lengths are factored into the denominator as $T_i T_j$ with an exponent parameter γ_t for adjusting the weight.

$$FDST_{ij} = \sqrt{\frac{\alpha[(x_i - x_j)^2 + (y_i - y_j)^2] + \beta[(u_i - u_j)^2 + (v_i - v_j)^2] + \alpha_t(t_{Oi} - t_{Oj})^2 + \beta_t(t_{Di} - t_{Dj})^2}{(L_i L_j)^{\gamma}(T_i T_j)^{\gamma t}}}. \quad (5)$$

Incorporating temporal dimension can potentially upgrade the findings from pure spatial patterns to spatiotemporal patterns. The advantage of this new method would include overcoming not only the MAUP, but the Modifiable Temporal Unit Problem (MTUP) (Cheng and Adepeju, 2014) as well. And the hot flows are detected based on not only flows' spatial similarity, but the time of their co-occurrence. In order to extend the spatial clustering detection to spatiotemporal clustering detection, I can adjust the local K-function by adding the temporal dimension as Equation (6):

$$LocKt_i(r_t) = E(number\ of\ other\ flow\ events\ within\ r_t\ of\ flow\ i). \quad (6)$$

where $LocKt_i(r)$ is the space-time local K-function value of flow Fi at scale r_t. This version of local K-function is capable to detect the "hot flows" clustered in space and happen closely in time. Accordingly, the scale r_t is the threshond distance combining both

spatial and temporal proximity. The spatiotemporal proximity is measured by Equation (5), of which the input variables need normalization and the parameters need to be adjusted after proper sensitivity analysis.

CHAPTER 3: STUDY II. FLOW HDBSCAN: A HIERARCHICAL AND DENSITY-BASED SPATIAL FLOW CLUSTER ANALYSIS METHOD

3.1 Overview

Understanding the patterns and dynamics of spatial origin-destination flow data has been a long-standing goal of spatial scientists. As a common family of data mining methods, cluster analysis has proved useful in exploratory analysis of large sets of spatial flows. This study aims at developing a new flow cluster analysis method called Flow HDBSCAN (hierarchical DBSCAN) that not only can extract spatial flow clusters from various situations including varying flow densities, lengths, hierarchies, but also avoids problems like MAUP, false positive errors, and loss of spatial information. The method combines density-based clustering and hierarchical clustering approaches from data mining area and extends them to the context of spatial flows. The flow proximity measures proposed in the first study of this dissertation are used again here to accurately calculate flow density. Moreover, the method is designed to effectively reveal hierarchical structures of the data, for example one flow cluster might be composed of several smaller ones. Moreover, the sole-parameter design guarantees its ease of use, and a special index is designed to relieve the challenge of selecting clusters as the final results in some complex situations. Experiments are conducted with both a synthetic dataset and an eBay online trade flow dataset in the contiguous U.S.

3.2 Motivations

In this study I develop a hierarchical and density-based clustering approach for disaggregated spatial flow data. Compared with other flow clustering methods, Flow HDBSCAN stands out as in several ways. One type of related methods measures the spatial relationships among origins and destinations, respectively, before combining them, as the basis for clustering flows. Here, spatial relationships can be contiguity or proximity of origin or destination regions (Guo 2009; Zhu and Guo 2014). However, these methods are sensitive to uneven density levels or ad hoc zoning definition of flow endpoints; besides they are prone to false positive errors on short-distance interactions. Another type of related approaches uses flow geometry to bundle nearby ones (Cui et al. 2008). While the results usually have desirable visual clarity, these methods compromise the accuracy of both spatial and attributive information of the data. For example, the endpoint location and length information of a flow would be lost if it is bundled to a set of nearby flows. Flow HBDSCAN inherits the strengths of density-based methods in the sense that it can extract flow clusters in various situations including varying flow densities, lengths, and hierarchies. Common problems for flow clustering approaches, like MAUP, false positive errors, and loss of information, are well handled. The method also inherits the unique advantage of hierarchical cluster analysis methods as it can reveal the implicit flow data structure, which is meaningful to understand structural relationships embedded in massive and cluttered sets of individual flows. In comparison with the “hot flow” detection method developed in the first study, this method is different as it embraces the principle of data mining to group observations into “clusters” based on similarity (Waller 2009), in spite of the overlapped terminology of “spatial flow cluster”.

3.3 Method in Detail

3.3.1 Theoretical Bases

Of various clustering methods, I choose to design this flow clustering method based on density-based clustering because of its capability to discover clusters of arbitrary shape and to filter out noise. More specifically I borrow the two classic notions of density-based clustering measures, namely core distance and reachability distance (Ester et al. 1996; Ankerst et al. 1999). Core distance (*CoreDist*), calculated as Equation (7), refers to the distance between an object to its $(MintPts - 1)th$ nearest neighbor, within a search radius of ε. $MintPts$ is the minimal size of a cluster. In Figure 9, the core distance of point *o* is the distance to its third nearest neighbor, w.r.t. $MintPts = 4$.

$$CoreDist_{\varepsilon,MinPts}(\mathrm{o}) = Dist(p, (MintPts - 1)th\ neighbor\ of\ i) \quad (7)$$

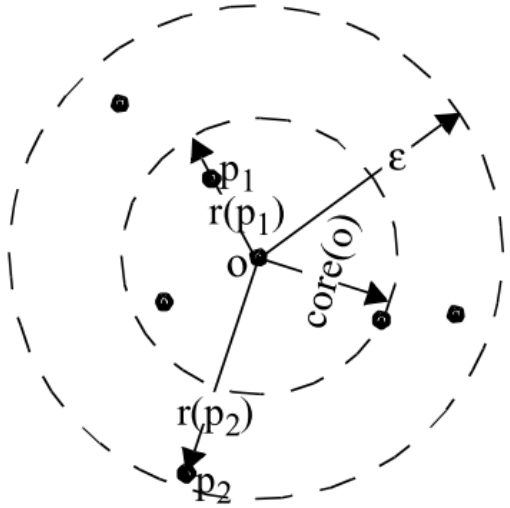


Figure 9: Core distance and reachability distance from Ankerst et al. (1999)

The other important density measure is called reachability distance (*ReachDist*), calculated as Equation (8). Taking the example in Figure 9, if a point p_1 is located inside

the core distance range of point *o*, the reachability distance between them equals $CoreDist_{\varepsilon,MinPts}(o)$. Alternatively, if a point p_2 falls outside the core distance range of point *o*, their reachability distance is the actual distance between them.

$$ReachDist_{\varepsilon,MinPts}(p,o) = max(CoreDist_{\varepsilon,MinPts}(o), Dist(p,o)) \qquad (8)$$

These two density metrics are essential to density-based clustering methods. Figure 10 is reachability plot from the article of the classic density-based method OPTICS (Ankerst et al. 1999). In this plot, all objects are listed horizontally in a way that those sharing similar reachability distance stick together. The height denotes the reachability distance value. The plot shows a mountain-like space where the "valleys" and "peaks" correspond to clustered and non-clustered objects, respectively. A deeper "valley" indicates a higher density of the corresponding cluster. To extract clusters a global parameter *rd* is introduced, which acts as a cutoff threshold of reachability distance to extract the "valleys" below it. For example in Figure 10, setting the threshold at the level of *rd1* leads to four clusters. Lowering the threshold to the level of *rd2* implies extracting clusters of a higher density. The leftmost "valley" is not identified as cluster anymore, while on the right side a big cluster is broken down into three smaller ones. As seen, the reachability plot is an effective way to visualize clusters of varying densities and potential hierarchical structures (nested clusters). However, choosing the best global threshold to extract clusters is a difficult task in many situations.

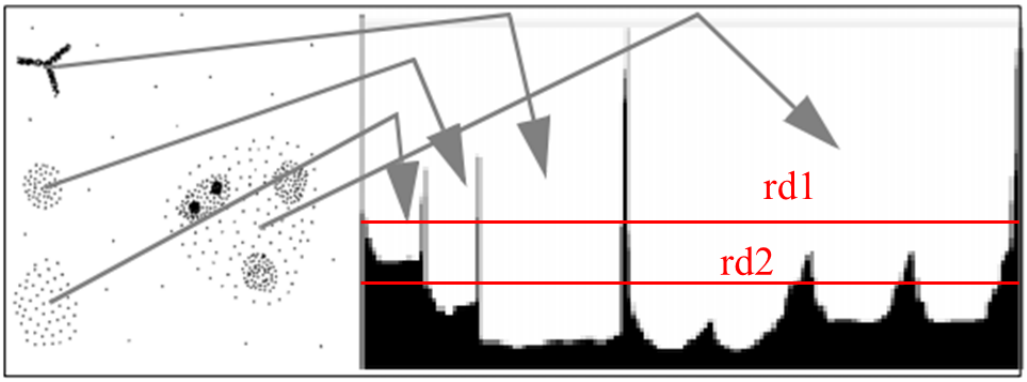


Figure 9: Reachability plot adapted from Ankerst et al. (1999)

In this study I choose not to follow the convention of density-based method to extract flow clusters. One reason is that choosing a global threshold to extract clusters is always arbitrary. On the other hand, I aim to reveal more information of the data, for example the potential hierarchical data structure, in addition to extract flow clusters. Therefore after calculating the cored distance and reachability distance, I depart from the standard practice of density-based clustering and choose to embrace the principles of hierarchical clustering in order to extract the hierarchical structures exhibited by clusters, if any.

3.3.2 Calculate Flow Density

To calculate the density measures for flows, choosing an appropriate distance measure is the very first step. Cluster analysis critically rests on an appropriate distance metric. Regarding spatial flow data, there exists no 'natural' metric. In the preceding chapter, I introduced a set of flow proximity metrics integrating corresponding endpoint coordinates as Equation (1), with additional considerations for the impact of flow length as seen in Equation (2), which are used again in this study. Flow Distance $FDist_{i,j}$ denoting the distance between flows $F_i\,(x_i, y_i, u_i, v_i)$ and $F_j\,(x_j, y_j, u_j, v_j)$ is calculated as Equation (9) or Equation (10) as below:

$$FD_{ij} = \sqrt{\alpha[(x_i - x_j)^2 + (y_i - y_j)^2] + \beta[(u_i - u_j)^2 + (v_i - v_j)^2]}. \quad (9)$$

$$FDS_{ij} = \sqrt{\frac{\alpha[(x_i - x_j)^2 + (y_i - y_j)^2] + \beta[(u_i - u_j)^2 + (v_i - v_j)^2]}{(L_i L_j)^\gamma}}. \quad (10)$$

where FD_{ij} and FDS_{ij} are the two versions of Flow Distance $FDist_{ij}$. The latter addresses the impact of flow lengths L_i and L_j while the former does not. The coefficients α and β serve to control the relative importance of either sets of endpoints ($\alpha > 0$; $\beta > 0$; $\alpha + \beta = 2$; by default $\alpha = \beta = 1$).

Based on the flow proximity metrics, flow density measures can be calculated. Following Ankerst et al. (1999), I do not set a search radius threshold for CoreD, like DBSCAN does (Ester et al. 1996), as it is usually arbitrary and has little impact on final results. Instead, CoreD here (Equation 11) is only decided by the minimum cluster size MinFlows, or the minimal number of flows a cluster must have. For example in Figure 11 if setting MinFlows = 3, the core distance of flow F_1 ($CoreD_1$) is given by the flow distance between F_1 and its second nearest neighbor F_3. Every flow object has its own core distance. A small CoreD suggests tight connections to one's neighbors, thus a likely belongingness to a cluster. In Figure 11, $CoreD_1$ is smaller than $CoreD_{11}$ for MinFlows = 3; this suggests a higher likelihood of F_1 to be detected as a cluster member compared with F_{11}.

$$CoreD_i = FDist_{i,\ (MinFlows-1)th\ nearest\ neighbor\ of\ i} \quad (11)$$

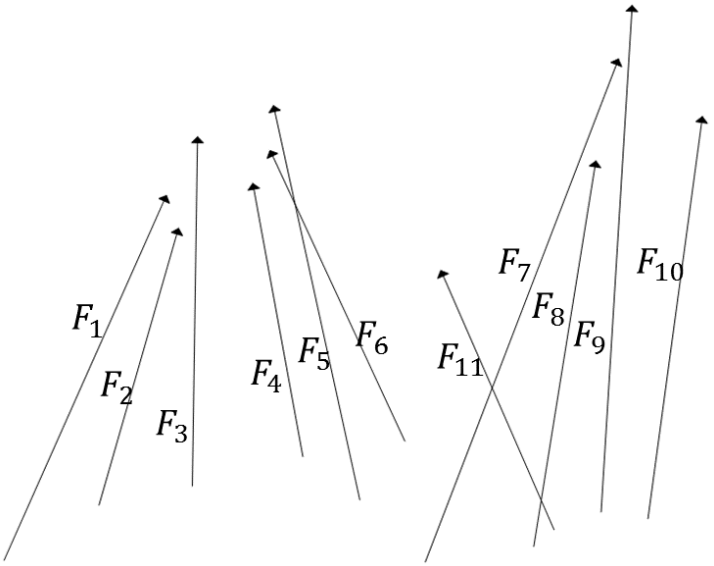


Figure 10: A sample set of flows

I choose to calculate the other density measure using the developed version of original reachability distance, called mutual reachability distance (MReachD) (Campello et al. 2013). Calculated as Equation (12), $MReachD_{ij}$ between two flows F_i and F_j I need flow distance $FDist_{ij}$ and two core distances $CoreD_i$ and $CoreD_j$. The largest value of these three distances is the MReachD between two flows.

$$MReachD_{ij} = \max(CoreD_i, CoreD_j, FDist_{ij}) \qquad (12)$$

3.3.3 From Density to Hierarchy

As mentioned earlier, the reachability distance can be plotted to reveal clusters and noises of the dataset, but choosing a global threshold to extract clusters is arbitrary. Therefore I design this method by integrating ideas and techniques of hierarchical clustering, for example using a tree-like structure to help illustrate hierarchical structure and facilitate the cluster-extracting process. Based on the calculated MReachD values, I build a minimum spanning tree (MST) (Campello et al. 2013) as Figure 12. The vertices

of this tree represent all flow objects; they are connected by edges with lengths proportional to the MReachD value between the two end vertices. In practice, this tree is built by sequentially adding the shortest edge that connects the current tree to a vertex not yet in the tree, starting from an arbitrarily selected vertex. In other words, the MST is constructed by connecting vertices with shortest possible edges. Figure 12 illustrates the MST for the sample data presented in Figure 11. All eleven flows are connected by at least one edge and together they compose the minimum spanning tree. To help explanation I choose a set of unitless numbers (1, 2, 3, and 4) to represent relative magnitudes of MReachD. Flows that are more likely to form a cluster, such as F_1, F_2 and F_3 are connected with short edges, whereas potential outlier F_{11} is linked with a long edge that suggests weak connection to the rest of the tree.

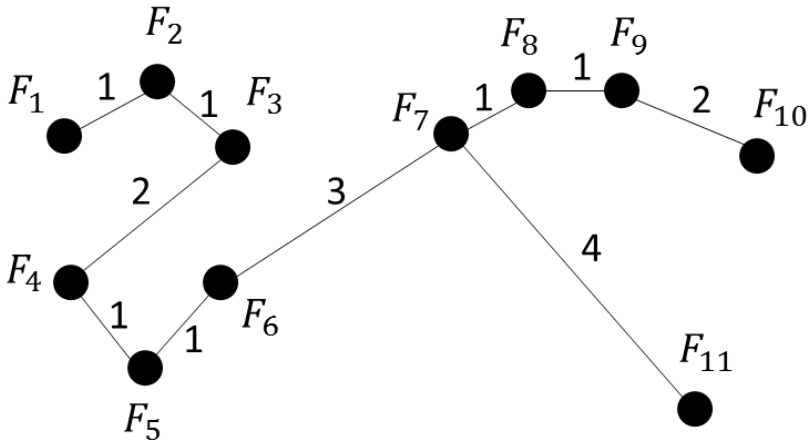


Figure 112: Minimum spanning tree

Then by sorting the edges of the tree in an increasing order of MReachD value, the MST is converted to a dendrogram that connects all vertices in a single hierarchical structure (Figure 13). This dendrogram reflects the density-based mutual reachability distance of the entire dataset in a hierarchical fashion. However, this hierarchy contains the

entire set of flow objects. It still needs a further step to discriminate vertices belonging to a cluster from noise. An analogy to this process is to prune a tree by removing the unwanted leaves and twigs (noises) and retaining the meaningful trunk and branches (clusters).

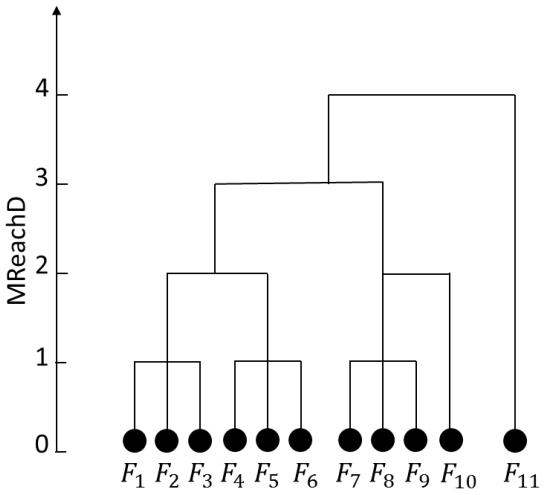


Figure 13: Dendrogram

3.3.4 Simplify the Hierarchy

Pruning the minimal spanning tree is an iterative process. The algorithm iterates through the dendrogram in reverse, from the highest MReachD, and decides at each split whether it should be removed. The minimum cluster size MinFlows serves as the only criterion. If the sizes of two descendant sets of a split are both greater than MinFlows, I maintain this split as both descendant sets can be standalone clusters. On the other hand, if one of the descendant sets contains fewer than MinFlows flows, I remove this split from the dendrogram, drop the small descendant set, maintain and keep processing the larger one. Continue this process until no more split can be removed.

In the example of Figure 13, if setting MinFlows = 3 the first split emerges at the level of MReachD = 4. One of the descendant sets contains flow F_{11} alone, so I remove this split

and drop F_{11} from the tree. For the next split at level of MReachD = 3, both descendant sets have more than three flows. Therefore I keep this split and both descendent sets. Keeping iterating the dendrogram, there are two splits at level of MReachD = 2. On the left side one set splits into two as each contains three flows, which is the minimum cluster size. On the right side only one descendant set (F_7, F_8, and F_9) remains while F_{10} has been dropped. This iteration process continues until every flow object is singled out. In this case the last step of split happens at the level of MReachD = 1. While iterating through the whole dendrogram, the algorithm marks down at which level each flow was dropped. Those dropped earlier at a high MReachD level such as F_{11} are more likely noises, while those separated in the end at a low MReachD level such as F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8, and F_9 are potential members of clusters.

3.3.5 Extract Flow Clusters

The simplified dendrogram reflects the data structure in a hierarchical way and suggests the potential clusters as well as noise. However it is still one step away from presenting the result clusters. Following the case above, F_{11} is less disputed as a noise since it cannot form a cluster with any nearby flows. The identity of F_{10} is however debatable. It can either form a four-member cluster together with F_7, F_8, and F_9, or be labelled as a noise. Confusion exists with respect to the other six flows (F_1, F_2, F_3, F_4, F_5, and F_6) as it is arguable to split them into two small clusters or keep them as a whole. To solve these confusions and decide the final clustering results, a special index called cluster stability inspired by Campello et al. (2013) is introduced as Equation (13):

$$S(C_i) = \sum_{F_j \in C_i} \lambda_{stay}(F_j) = \sum_{F_j \in C_i} (\lambda_{end}(F_j) - \lambda_{begin}(F_j)) \quad (13)$$

where $S(C_i)$ is the cluster stability of cluster C_i, $\lambda = 1/MReachD$, $\lambda_{begin}(F_j)$ and $\lambda_{end}(F_j)$ correspond to the smallest and largest λ value that flow F_j belongs to cluster C_i, respectively. And $\lambda_{stay}(F_j)$ means the range of λ value in between.

The idea is that if a flow stays with a cluster for a large range of λ values, it is considered a loyal member of this cluster. If a cluster contains many loyal members, it is considered stable. To solve the confusion like whether to split one cluster into smaller ones, or whether to drop disputed cluster members as noise, we only need to calculate and compare the cluster stability of the two ambiguous situations. Figure 14 shows the simplified dendrogram as a hierarchical cluster tree. Only noise F_{11} has been dropped (hollow vertex) and the corresponding split has been removed (dash line). Clusters are highlighted by red boxes. According to the cluster stability index, F_1, F_2, F_3, F_4, F_5, and F_6 split into two three-member clusters, and F_{10} sticks with F_7, F_8, and F_9 as a four-member cluster.

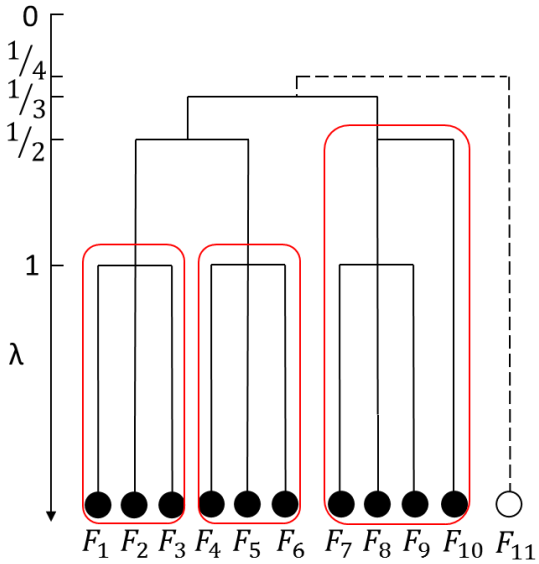


Figure 12: Hierarchical cluster tree with red boxes denoting extracted clusters

3.3.6 Algorithm Steps

(a) Calculate an N by N flow distance matrix with FDist (Equation 9 or 10);

(b) Calculate CoreD with a selected minimum cluster size MinFlows (Equation 11);

(c) Calculate an N by N MReachD matrix (Equation 12);

(d) Build a minimum spanning tree (MST) based on MReachD;

(e) Sort the MST to obtain a hierarchical dendrogram;

(f) Simplify the dendrogram based on MinFlows. More specifically, iterate through the dendrogram from the highest MReachD. If one of the descendant sets is smaller than MinFlows, drop it from the dendrogram, remove the split, and keep processing the large descendant set; if both descendant sets are equal or larger than MinFlows, keep the split and continue iterating both sets;

(g) Extract flow clusters from the hierarchical cluster tree through calculating and comparing cluster stability (Equation 13).

3.4 Experiment

3.4.1 Evaluation with Synthetic Data

To test the effectiveness of this method, I designed a synthetic spatial flow dataset of 3,000 flows in various group configurations. Figure 15 depicts the spatial distributions of flows. The legend indicates the color and size of each group of flows. All the flows' endpoints are distributed in a two-dimensional space with x and y both within the range from 0 to 1,000. Table 1 describes the size of each group and the range of the flows' four coordinates (x_o, y_o, x_d, y_d). Within its range, a coordinate is randomly generated. Group 1 are the "background" flows. The origin and destination points of these 1,000 flows are randomly distributed within the entire area. Group 8 shares the similar characteristics with

group 1, but the flows are encased in a very limited area. Given that most existing flow clustering methods mistakenly identify short flows as clusters regardless of their distinct spatial characteristics such as direction, length, and endpoint location, group 8 is designed to test whether Flow HBDSCAN can avoid such false positive errors. Each of groups 2 to 7 is a group of clustered flows with similar directions. Group 2 and 6 have relatively low density in comparison with group 3, 4, 5, and 7. This varying density design is to test whether Flow HDBSCAN is able to extract all of these clusters. If so, to check if clusters of different densities are extracted at the same time or with different parameter settings. Hierarchical clusters, namely nested clusters, are also designed. For instance group 3 and 4 have the potential to form one cluster. Group 2, 3, 4, and 5 can potentially compose an even bigger cluster. This hierarchical design is to test another claimed functionality of Flow HBDSCAN: to reveal potential hierarchical data structure of flow clusters.

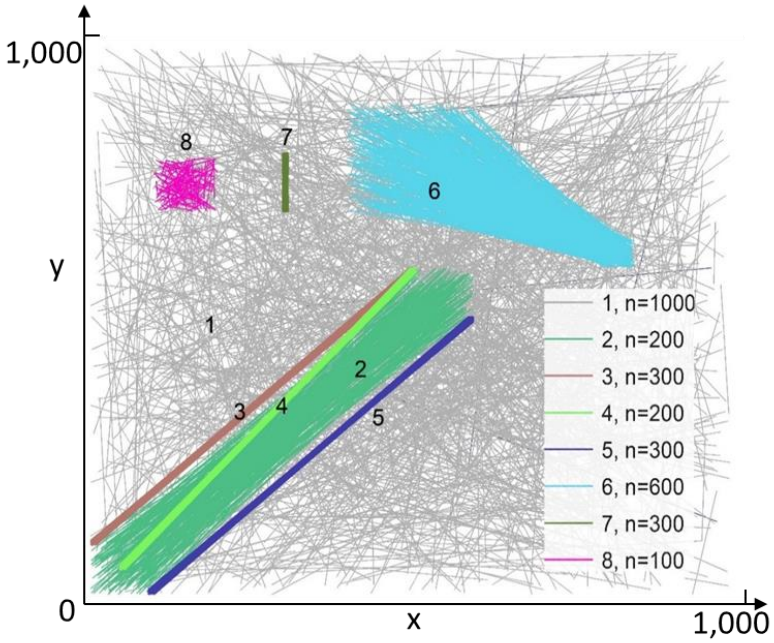


Figure 13: Map of synthetic flow dataset

Table 1: Details of synthetic flow dataset

Group No.	Size	Range of x_o	Range of y_o	Range of x_d	Range of y_d
1	1000	0 – 1000	0 – 1000	0 – 1000	0 – 1000
2	200	0 – 100	0 – 100	500 – 600	500 – 600
3	300	0 – 10	90 – 100	500 – 510	590 – 600
4	200	45 – 55	45 – 55	500 – 510	590 – 600
5	300	90 – 100	0 – 10	590 – 600	500 – 510
6	600	400 – 600	700 – 900	800 – 850	600 – 650
7	300	300 – 310	700 – 710	300 – 310	800 – 810
8	100	100 – 200	700 – 800	100 – 200	700 – 800

Performing Flow HDBSCAN with FDS (Equation 10) as flow distance metric, results w.r.t. MinFlows = 50 and 250 are picked out as examples for discussion. Figure 16a shows the map of four detected clusters for MinFlows = 50. Figure 16b reveals the hierarchical structure of the clusters. All the branches remaining on the hierarchical tree qualify as clusters w.r.t. MinFlows = 50, however not all of them are extracted as final results. The width of each branch represents the cluster size. And the height λ value denotes the density level. The figure clearly shows at which density level a cluster begins to form, and how a cluster gradually loses members (shrinks width) until it disappears or splits into smaller clusters while the density level increases (λ decreases). According to the cluster stability index, only four of the branches are selected as final clusters. Figure 16c illustrates the composition of each final cluster. Cluster C1 contains the entire group 6 and 18 nearby flows from group 1. C2 consists of the entirety of groups 3 and 4, with 37 nearby flows

from group 2. Similarly, C3 contains the entire group 5 with 15 flows from group 2 as well. C4 is identical to group 7.

The results reflect the correctness of Flow HDBSCAN as there are only a few false positive (FP) errors but no false negative (FN) errors. False positive errors in this context mean the flows designed as noises but detected as clusters. Such errors only exist in C1, where 18 out of 618 are from randomly distributed group 1. On the other hand, false negative errors represent the designed flow cluster members detected as noises. Figure 16c shows that the extracted clusters include the complete set of groups 3, 4, 5, 6, 7. The special case is group 2; some of its flows join cluster C2 and C3, while the rest of its members are detected as noise. This is due to the choice suggested by cluster stability rather than failure of detection. To quantify the correctness of the results, I use an index called cluster correctness (CC) adapted from Guo and Wang (2011). For each extracted cluster, CC is calculated as follows:

$$\mathrm{CC} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP} + \mathrm{FN}) \qquad (8)$$

where CC as cluster correctness; TP is short for true positive, or the number of flows that are both designed and identified as cluster members; FP and FN denotes the number false positive errors and false negative errors, respectively. An index close to 1 indicates that there is very limited FP or FN. The four clusters in this set of results all have very high CC, as CC1 = 0.97 while the CC for the other three all equal to 1.

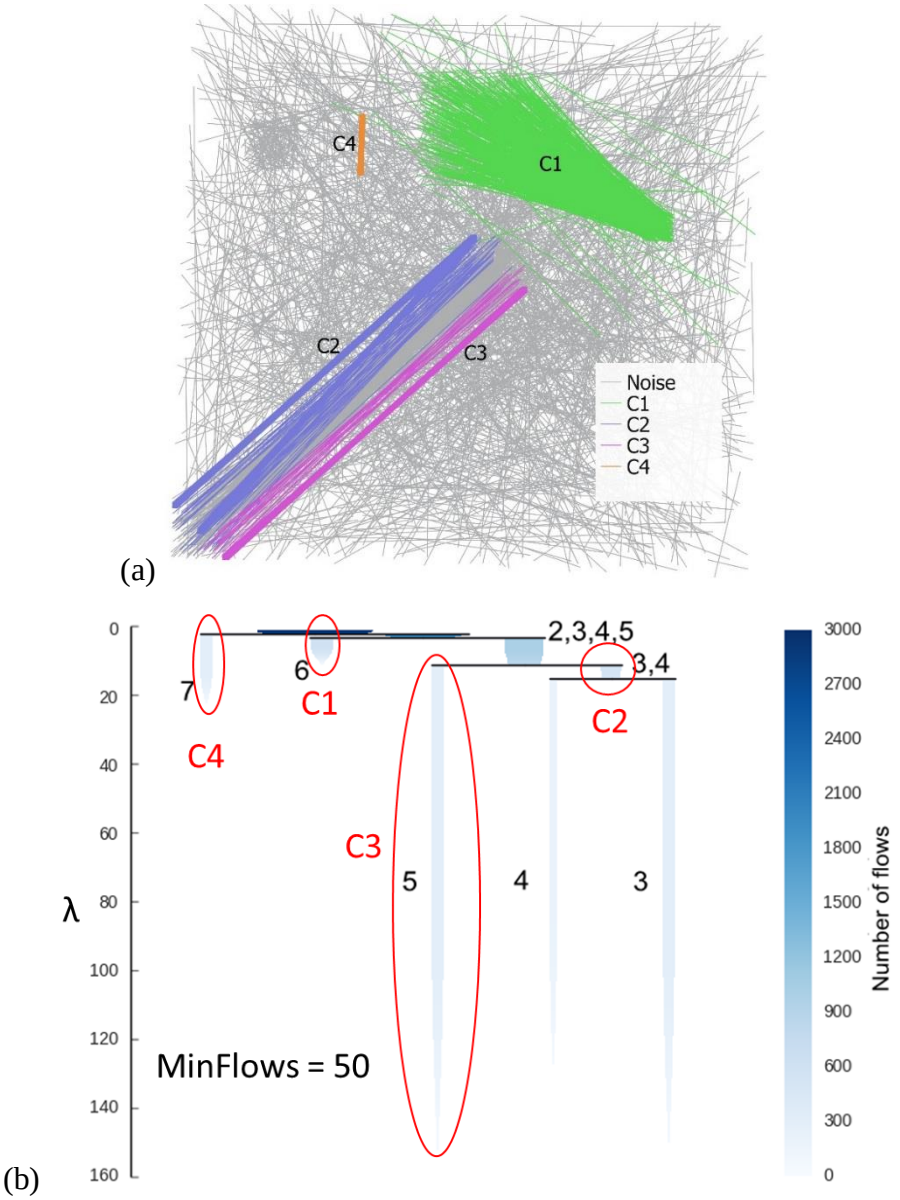
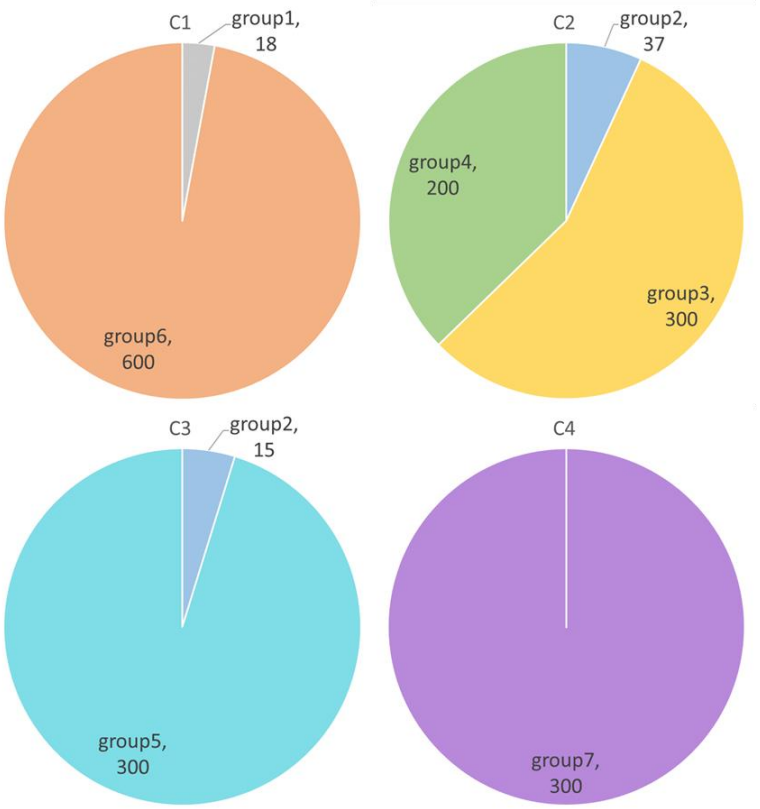


Figure 16: Results w.r.t. MinFlows = 50. (a) Result map; (b) hierarchical cluster tree; (c) composition of each flow cluster



(c)

Figure 14 continued

Another set of results is presented in Figure 17 w.r.t. MinFlows = 250. Compared with the previous result w.r.t. MinFlows = 50, only three final clusters are extracted (Figure 17a). Given that group 4 has only 200 flows which is not enough to form a cluster, there exists no branch of group 4 on the hierarchical cluster tree (Figure 17b) any more. Figure 17c shows that C1 and C3 are almost identical as for MinFlows = 50. C2 is a huge cluster combining the complete groups 2, 3, 4, and 5, with a few false positive errors from group 1. It indicates that w.r.t. MinFlows = 250 the algorithm chooses to extract the nested clusters as a whole instead of smaller ones. This time the cluster correctness index for each cluster is also very high, as CC1 = 0.97, CC2 = 0.95, CC3 = 1.

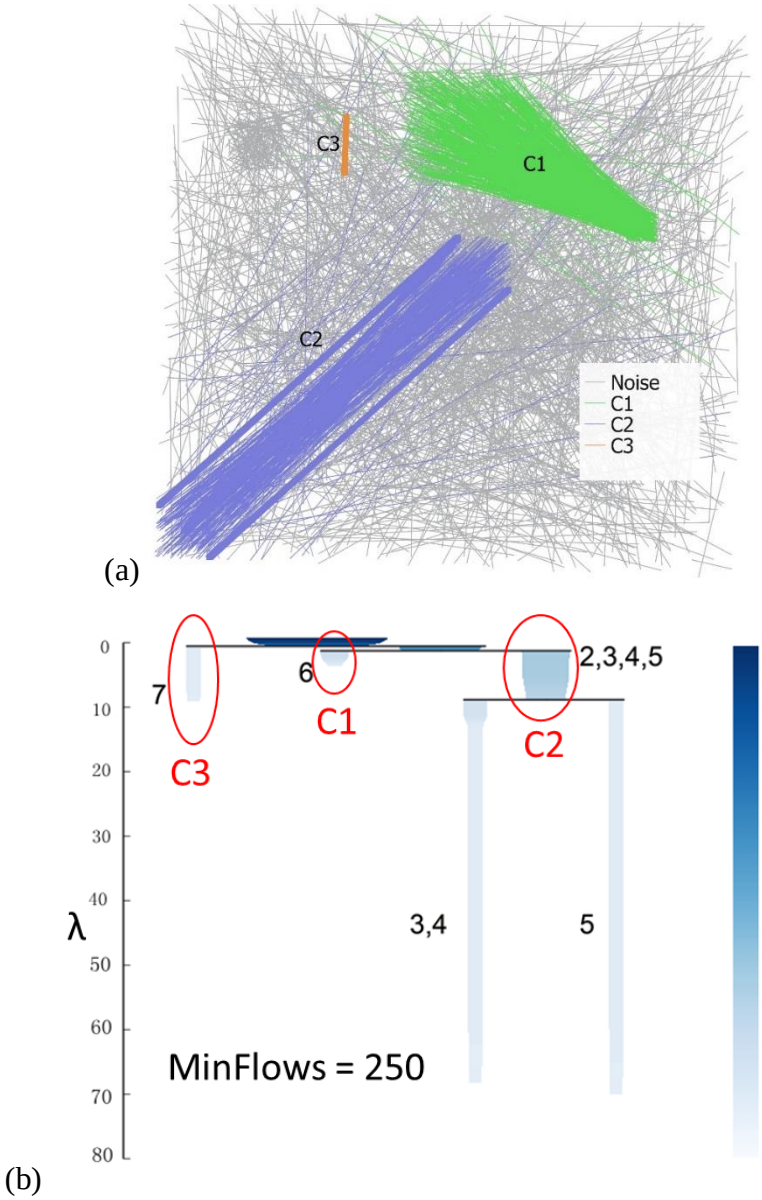
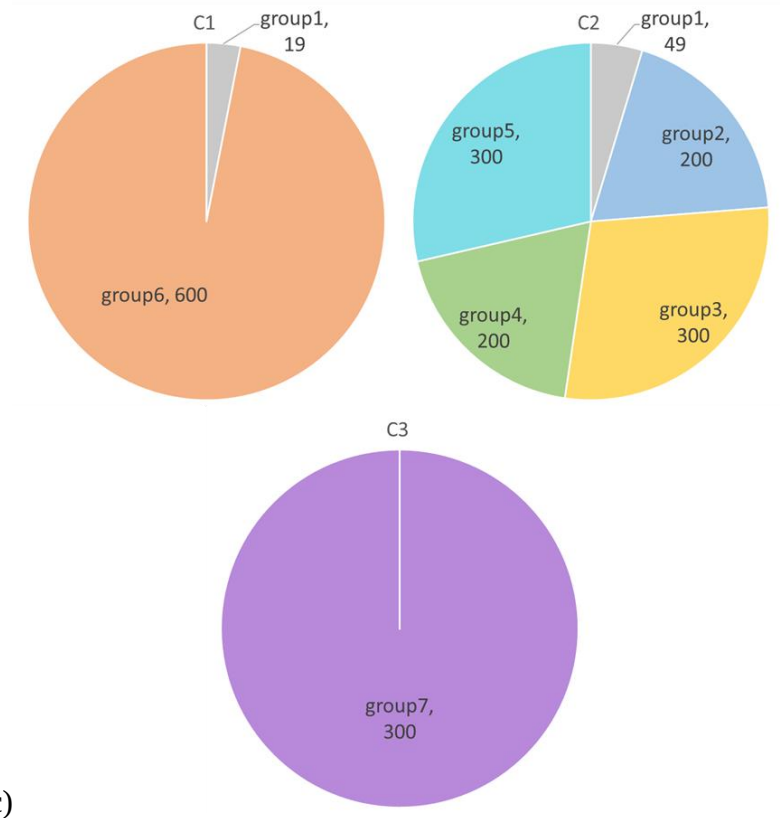


Figure 17: Results w.r.t. MinFlows = 250. (a) Result map; (b) hierarchical cluster tree; (c) composition of each flow cluster



(c)

Figure 15 continued

Sensitivity analysis is conducted for the single parameter MinFlows. Table 2 is an overview of the test results. The second row reports how many flow clusters are extracted at each MinFlows value. For example, “3 (56)” means the algorithm extracts 56 clusters w.r.t. MinFlows = 3. Four clusters are detected w.r.t. MinFlows = 30, 50, 100, or 200, and three clusters w.r.t. MinFlows = 250. As seen, the number of clusters is consistent for MinFlows value from 30 to 200.

Table 2: Results of Flow HDBSCAN on Synthetic Flow Dataset

Group No.	Size	% of clustered flow members when MinFlows =					
		3 (56)	30 (4)	50 (4)	100 (4)	200 (4)	250 (3)
1	1000	28.6	1.8	1.9	2.2	2.1	6.8
2	200	48.0	26.0	26.0	25.5	23.5	100
3	300	100	100	100	100	100	100
4	200	100	100	100	100	100	100
5	300	100	100	100	100	100	100
6	600	100	100	100	100	100	100
7	300	100	100	100	100	100	100
8	100	47.0	0	0	0	0	0

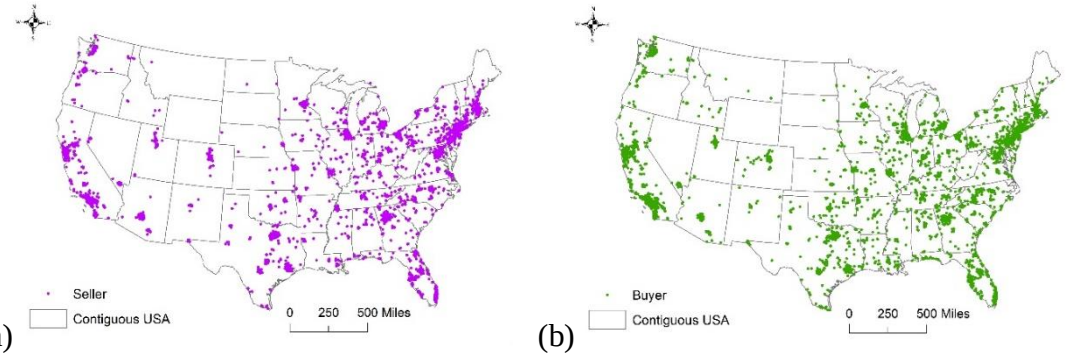
The main part of Table 2 lists the proportion of each group identified as member of any cluster, with respect to every MinFlows value. These statistics also reflect consistent results within a large parameter value range. Group 1 and Group 8 show similar patterns as both contain flows distributed randomly within a certain square region. The difference is that the longer flows of group 1 have a slight chance to join a cluster as false positive errors. While none of group 8 flows is detected as cluster when MinFlows value is equal or more than 30. It proves that Flow HDBSCAN overcomes the common false positive errors that may be caused by short flows. The statistics of the other groups are the same 100 percent of group members are detected as cluster. It verifies there is no false negative error. The only exception is group 2. Part of it forms clusters with other groups when MinFlows is equal to or smaller than 200. This was explained earlier as a choice made by cluster stability, rather than failure of detection.

As a simple sensitivity test, the sole parameter of Flow HDBSCAN, namely MinFlows, has been tested with multiple values. Within the range of 30 to 200, the results remain quite stable in terms of total cluster number and outcome within each group. Additional small clusters are detected when setting MinFlows very small, while fewer clusters are identified with a very high requirement of minimal cluster size. It shows that the algorithm is not sensitive to the changes of its sole parameters. In practice, the rule of thumb is to select an appropriate MinFlows value at a relatively small level. But it is always recommended to conduct sensitivity analysis.

3.4.2 eBay On-line Trade Flow

In order to test our method in a real situation and discover its practical usefulness, I further experiment with an eBay online trade flow dataset. The dataset contains 8,607 transaction records of the first generation of iPhone in 2007 within the contiguous US states. The locations of all the sellers and buyers are available so that each seller-buyer pair can be viewed as an individual spatial flow. In terms of the flow direction, both buyer and seller can be seen as the origin or destination. In fact, sellers post their iPhone on eBay for sale in the form of an auction. By the time an auction is scheduled to end, usually a day or two after posted, the buyer who bid with the highest price can successfully purchase it. If the focus is the online transaction, then the buyers are seen as the origins as they transfer the money to the sellers. On the other hand, the merchandise itself, namely iPhone in this case, will be shipped by the seller to the buyer after the online transaction so that the actual spatial movements of cellphones start from the sellers. In this study, I set seller to buyer as the flow direction.

Figure 18a and Figure 18b show the distribution of sellers and buyers, respectively. Overall the patterns of their distribution are very similar. Both sellers and buyers locate heavily in the most populous regions such as the coastal areas, while remote regions have few or no activities. The difference between these two distributions is not obvious. But the buyers seem to locate according to a pattern that is slightly more spatially scattered than the sellers. The point distributions however offer very limited information about patterns of the flows, which the method designed in this study can help analyze and present.



(a) (b)

Figure 16: Distribution of (a) sellers and (b) buyers

Figure 19 shows the flow clusters extracted by Flow HDBSCAN for MinFlows = 30 and FDS (Equation 10) as flow distance metric. There are in total 39 flow clusters and most of them are between big cities. More patterns are discovered with the flow length distribution as follows.

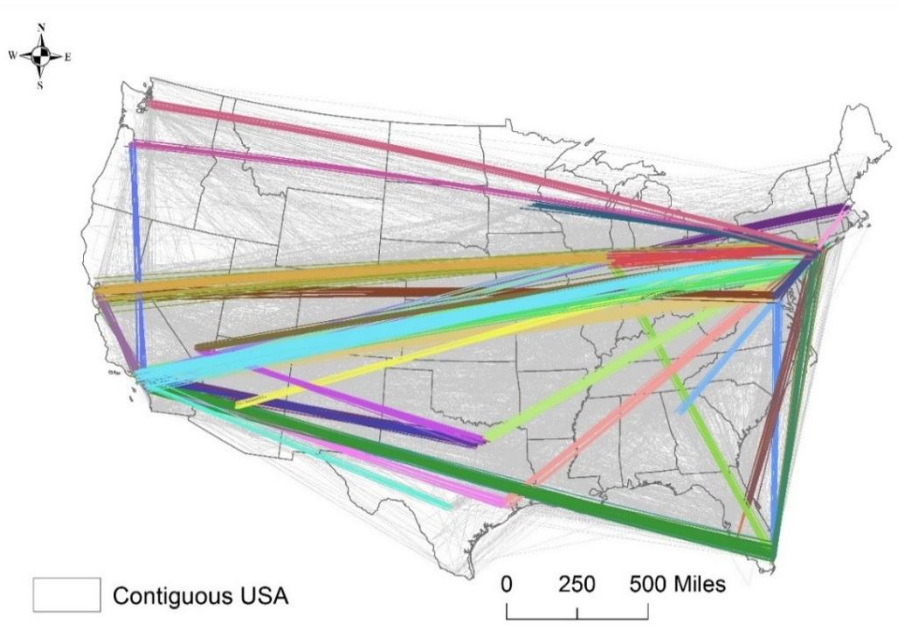


Figure 17: Map of eBay trade flow clusters

The distances separating the seller and buyer are clearly not normally distributed (Figure 20a). In fact, it is flat at most distances but extremely high in very short distance range, which means that a lot of transactions happened within the city where the seller is located, even though it is online shopping. It also has some local peaks and this becomes obvious when short flows are filtered out. The peak at 300 km could be flows between cities on the east coast such as New York City and Boston or New York City and Washington, DC, while the peak near 4,000 km probably corresponds to coast to coast flows e.g. Los Angeles and New York City or San Francisco and New York City.

To cancel bias brought by flow length thus avoiding false positive errors created by the massive short flows, FDS (Equation 10) is chosen as the flow distance metric. The result shows that flow clusters between iPhone supply and demand individuals can help reveal interesting patterns of these online eBay transaction activities. The length distribution of

clustered flows shown in Figure 20b validates that the peaks in Figure 20a are picked out. The three peaks of flow lengths correspond to the most popular location pairs of buyers and sellers in the US, i.e. two places within the same metropolitan area, two places of the same regions, and two places between East Coast and West Coast. This distribution reveals a unique pattern of online trade, that is its insensitivity to physical distance. Weak evidence has been found to support transferability of Ullman's (1953) spatial interaction theory since the long physical distance does not seem to heavily impede clustered online trade flows.

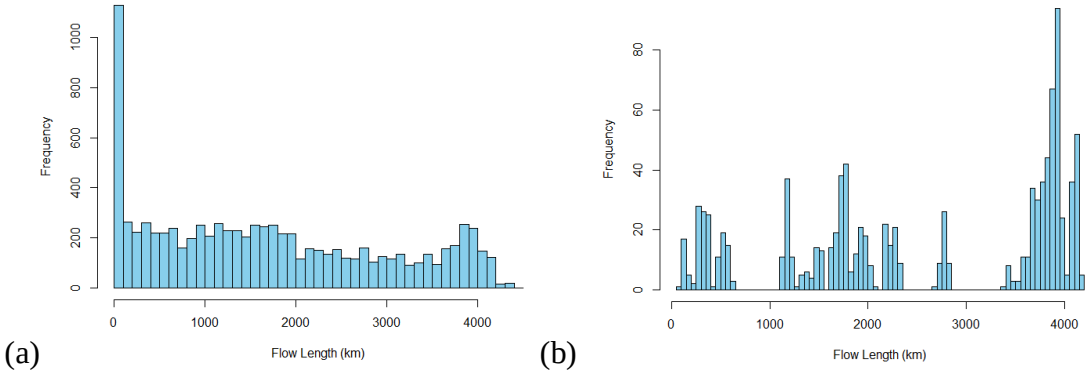


Figure 18: Distribution of length of (a) all flows; (b) clustered flows

Another major functionality of Flow HDBSCAN, namely to reveal the data structure, is also validated in this experiment. The flow clusters (Figure 21a from New York metropolitan area (Tri-State Area) to San Francisco Bay Area (Bay Area) is chosen to discuss this aspect. The hierarchical cluster tree in Figure 21b shows that this cluster can be broken down into two smaller ones. Zooming onto the destination area (Figure 21c) it shows that the buyers are from two separate regions, i.e. the North Bay Area including San Francisco and Oakland, and the South Bay Area including San Jose and Santa Clara. Notwithstanding, the origins (Figure 21d) of these flows are concentrated around NYC. This means this hierarchy is caused by the two separated regions of buyers alone.

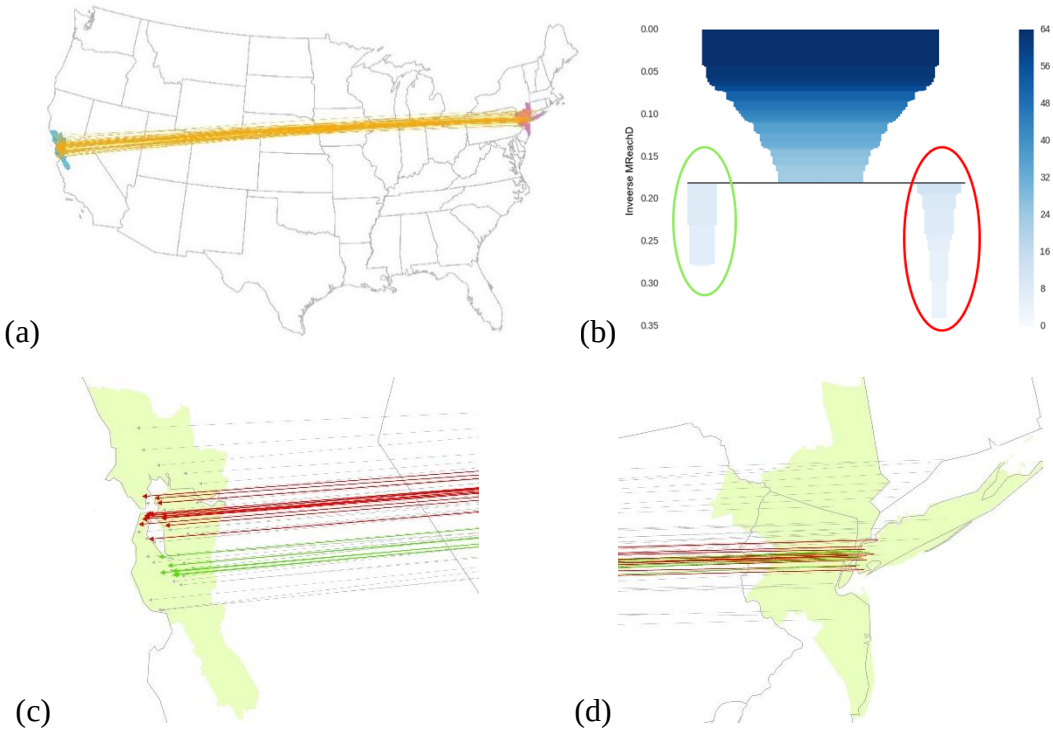


Figure 19: Result flow cluster from Tri-State Area to Bay Area. (a) Map of flow cluster; (b) hierarchical cluster tree; (c) flow destinations at Bay Area; (d) flow origins at Tri-State Area

3.5 Summary and Future Directions

This study developed an innovative spatial flow cluster analysis method called Flow HBDSCAN. The method is a combination of density-based clustering and hierarchical clustering methods and it is tailored to spatial flow data. It has the potential to be an effective tool of exploring massive spatial flow data. Experiments have been done with both a synthetic dataset and an eBay online trade dataset. The results demonstrate its capability to extract flow clusters and reveal hierarchical data structure at the same time.

Compared with other related methods, Flow HBDSCAN has some clear advantages. First, it has only one parameter which is the minimum size of cluster. This avoids introducing many arbitrary parameters and it saves the dilemma of parameterization. In

addition, the experiment shows it is not sensitive to the choice of parameter value within a large range. Second, the method overcomes the difficulties brought by varying flow densities and flow lengths. It avoids potential false positive errors of mistaking short flows as clusters by using the Flow Dissimilarity (FDS) as flow distance metric. In addition, it is designed for individual flows with fine spatial resolution so that problems like MAUP, loss of spatial information, and uneven distribution or hoc zoning definition of flow endpoints do not exist. More importantly, the method can reveal the internal data structure of flow clusters. By visualizing the hierarchical cluster tree it provides the full information of identified clusters and their internal relationships, for example one flow cluster might be composed of several smaller ones. Moreover, an index called the cluster stability is designed to help decide which clusters as the final results.

In terms of future work, one possible direction is to extend the current method from unsupervised clustering to supervised classification. By introducing the sample training process from the domain of machine learning, the method can be more pragmatically useful in real scenarios for example target on flow clusters of specific characteristics.

3.6 Comparison with Hot-Flow Detection

As stated earlier, Flow HDBSCAN has some overlaps in terminology with the hot-flow detection method namely Flow K-function introduced in the previous chapter, as they both center on spatial flow and cluster. Their methodological bases, however, are quite different. Flow HDBSCAN belongs to the family of spatial data mining, while Flow K-function stems from spatial statistics. Geovisualization techniques have been incorporated to both for better illustration of cluster results. Although they are different types of methods, they are within the common scope of exploratory spatial data analysis. This means that the two

methods share the common goal to explore spatial flow data, thus discovering interesting patterns, describing distributions, identifying outliers, and validating spatial effects such as spatial association and spatial heterogeneity. Therefore, they can be applied to the same dataset to obtain complementary findings. To shed light on their complementarity in joint applications as well as to deepen the understanding of their uniqueness, I produce the results of both methods on the same dataset and compare them side by side (Figure 22). The experiment dataset is the same one used in the previous chapter. It consists of 6,810 motor vehicle theft-recovery flow events within the city of Charlotte, NC from 09/01/2008 to 08/31/2014. Figure 22a and Figure 22b illustrate the result of Flow K-function and Flow HBDSCAN, respectively.

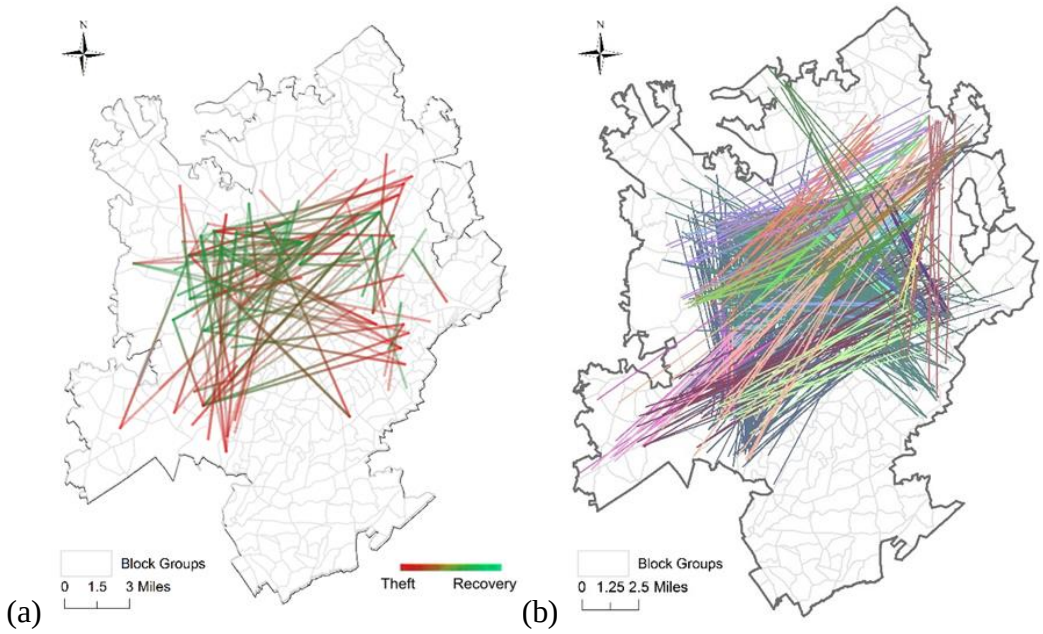


Figure 20: Comparison between Flow K-function and Flow HDBSCAN. (a) local Flow K-function at scale of 0.05 w.r.t. Flow Dissimilarity and (b) Flow HDBSCAN w.r.t. MinFlows = 10 on the Charlotte motor-vehicle theft and recovery dataset

Despite that both results are visualized as clusters of flows in a map, it is necessary to clarify the difference on what those cluster results stand for. For short, Flow K-function detects clustering as a pattern of flow distribution, while Flow HDBSCAN groups flows into clusters based on similarity. More specifically, a clustered flow in Figure 22a means that vehicle theft and recovery event are significantly more likely to happen between its origin and destination or in their vicinity, with respect to the overall average. While flows of the same color in Figure 22b are the groups of nearby flows who satisfy the requirements of minimal cluster size and density level.

By understanding the meanings of the flow clusters, the differences of the two sets of results can also be better explained. First, both methods have their own parameters or criteria. The choices of significance level, flow distance metric, size of detection window, and minimal size of a cluster are all important to the outcomes, even though in this case I customized the choices to produce results that would as similar as possible. Second, some areas in Figure 22b have more flow clusters than their counterparts in Figure 22a. This is because Flow HDBSCAN extracts and visualizes every member of a cluster, no matter the cluster size is ten or one hundred. On the other hand, Flow K-function uses a moving detection window technique such that every time the window centers at one flow while the algorithm counts the neighboring flows within it. In contrast with those big flow clusters detected by Flow HDBSCAN, fewer flows have enough amount of neighbors within its detection window to be statistically significant. This standard can be higher than the minimal cluster size adopted in Flow HDBSCAN, which explains a more scattered pattern in Figure 22a. Third, as a statistical method, Flow K-function considers edge effect while the other does not. That is why very few clustered flows are observed near the border of

the study area in Figure 22a. Fourth, Flow HDBSCAN has the advantage of visualizing clusters with different colors according to their unique identities. The differences on two sets of results shed light on the characteristics of these methods. A better understanding of their advantages and disadvantages can certainly help choose the better one to suit the context. In addition, it helps understand how these methods can complement each other if being used jointly.

The key difference is that Flow K-function assesses the space while Flow HDBSCAN focuses on the flow events themselves. In analogy to the widely used hot spot detection in crime analysis, which assesses the distribution of point-based crime events and draws conclusion such as this type of crimes are more likely to happen near a commercial place. Hot flow detection assesses the distribution of spatial flow events, and obtains conclusion such like it is statistically more likely to observe such flow phenomena near a pair of locations. The inputs are flow events, while the conclusion is clustering pattern of the space, which can be the entire study area of local places, corresponding to the global and local version of Flow K-function. However, like any other statistic approach, the distribution of the population is essential as it serves the benchmark of clustering pattern. Although techniques such as Monte-Carlo simulation can be used to avoid biased results based on normal distribution, it is not intuitive especially for beginners to design an appropriate simulation test.

Flow HDBSCAN decides whether a flow event can be grouped with others which share similarity with it. If so, it identifies which group it belongs to. In contrast with Flow K-function, this technique is free from any assumption so that it is unaffected by abnormal population distribution. In addition to extracting clustered flows, it provides abundant

information including the identify of each cluster and the hierarchical data structure. Moreover, users can modify the sole parameter to interactively explore the data. The downside is that not all clusters results are meaningful and easy to interpret, since the algorithm returns all flow clusters that meet the criteria.

To sum up, both methods are designed to explore spatial flow data. Flow K-function is more suitable to assess a long-term or large amount of spatial flow phenomena, as it measures the distribution pattern of the entire area or local places. Flow HBDSCAN works better on examining the data with no assumption at all. It is especially useful to focus on specific flow events, e.g. whether they have similar events happened nearby, and what is the structural relationship between them.

CHAPTER 4: STUDY III. FLOWAMOEBA: IDENTIFY REGIONS OF ANOMALOUS SPATIAL INTERACTIONS AND CREATE A SPATIAL FLOW WEIGHTS MATRIX

4.1 Overview

This study aims at developing a data-driven and bottom-up method to identify regions of anomalous spatial interactions, based on which to create a spatial flow weights matrix. Benefiting from the ready availability of individual flow data with fine spatiotemporal resolution, this method offers a solution by identifying the origin and destination regions that capture anomalous spatial interactions happening in between. The core idea of the method is to extend the well-known method of identifying spatial clusters, namely A Multidirectional Optimum Ecotope-Based Algorithm (AMOEBA) (Aldstadt and Getis 2006), to the context of spatial flow data. The method, dubbed FlowAMOEBA, starts from a "seed" flow to which neighboring flows are iteratively attached until the addition of any neighbor fails to increase (or decrease) the magnitude of the local spatial statistic, e.g. local G_i^* statistic (Getis and Ord 1992; Ord and Getis 1995). The outcome is a cluster of high (or low) value flows, and their combined origin and destination regions are called a flow ecotope. This iterative process is repeated by assigning every flow as the "seed" flows across the study region. After resolving overlaps of all identified flow ecotopes, those that pass statistical significance test are preserved. Boundaries of the resulting flow ecotopes are delineated as regions of anomalous spatial interactions, regardless of the size, shape, scale, or administrative level. The second product of this method consists of a spatial flow weights matrix, which is derived from the identified flow ecotopes. The matrix is designed

not only on the basis of flows' spatial relationships, but also spatial associations between flows within the same flow ecotope. One of its promising applications is to improve spatial interaction modeling by accounting for the common issue of network autocorrelation. The spatial contiguity relationships and growth rules are tailored for spatial flow data so that FlowAMOEBA can take advantage of the finest possible spatial resolution to represent the spatial context of interaction processes including shape, size, location, and topology. The method has the potential to dramatically change the way we study spatial interactions. First, it breaks the convention that spatial interaction data are always collected and modelled between spatial entities of the same granularity, as it delineates the OD region of anomalous spatial interactions. Second, the method creates an empirical spatial flow weights matrix that can handle network autocorrelation embedded in spatial interaction modeling, thus improving related policy-making or problem solving strategies.

4.2 Motivations

The major contribution of this study is to develop a novel method of identifying regions of anomalous spatial interactions. The method is a heavily data-driven approach as it lets the data speak for themselves. It adopts a bottom-up strategy to identify clusters of high (or low) value flows by delineating the boundaries of their origin and destination regions, regardless of the size, shape, scale, or administrative level. The usefulness of this method is twofold. First, it can serve as tool for exploratory spatial flow data analysis. Compared with Flow K-function and Flow HBDSCAN developed in the previous two chapters, this method is the only one that is capable of accounting for non-spatial attributes of flows in addition to the spatial elements. Therefore, the regions of anomalous spatial interactions identified by FlowAMOEBA not only reflect the spatial closeness of individual flows, but

the spatial association of the value carried by flows. As a bottom-up approach, FlowAMOEBA takes advantage of fine spatial resolution to delineate the boundaries of detected regions without restrictions of the size, shape, scale, or administrative level. Therefore, the results can potentially change our intuition that large volume of interactions always happen between predefined regions at the comparable level. For example in Figure 23, a large number of migrants move from a certain region of city A (polygon filled with blue lines) toward city B and its surrounding areas (polygon filled with red lines). The traditional and intuitive way of migration study would collect data between city A (yellow circle) and city B (green circle), and model interactions with characteristics of the two cities. In contrast, FlowAMOEBA can be used to delineate the regions (polygons filled with lines) between which a large volume of migration actually occurs. The explored results can be further used to solve the uncertain geographic context problem (UGCoP) in spatial interaction context, known as studied geographical units that fail to address the contextual uncertainty of actual and dynamic sociogeographic processes (Kwan 2012).

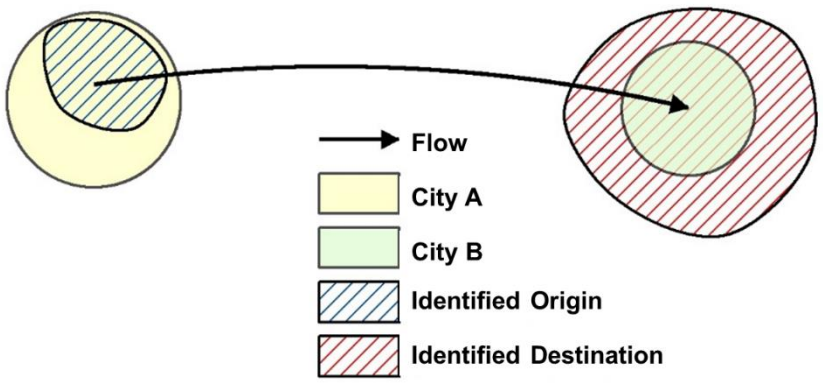


Figure 21: Comparison of predefined regions and detected regions by FlowAMOEBA

Second, the method can serve as a bridge between exploratory spatial data analysis (ESDA) and confirmatory spatial data analysis (CSDA) in the context of spatial flow data, i.e. the results of this ESDA method can be used to improve CSDA studies. The other result of FlowAMOEBA, namely the spatial flow weights matrix, can account for network autocorrelation, which is defined as the dependence of values of random variables associated with given flows on the values of the variables associated with other nearby flows (Black 1992). For example, a large number of tourists from a certain city to Yellowstone National Park may imply significant tourist flows from that city to the nearby Grand Teton National Park. Heavy traffic flows observed on a highway during rush hours may imply that equivalently heavy traffic is also likely to be observed on a nearby road parallel to the said highway. Similar to most quantitative studies with spatial regression models, spatial dependence exists as an inevitable issue that has to be solved in modeling spatial interactions, because it violates one of the key assumptions of those models that observations are independent to each other. Previous solutions include LeSage (2008) who proposed spatial weight structures that consist of three spatial connectivity matrices capturing origin, destination, and origin-to-destination dependence. Another approach is by using an eigenfunction-based filter for accommodating spatial autocorrelation effects within a spatial interaction model (Griffith 2007; Chun 2008; Chun and Griffith 2011). Using the spatial flow weights matrix derived by FlowAMOEBA can be a new solution. The idea is to use spatial weight structures of the identified flow ecotopes to capture the spatial dependence among flow data. The advantage of an approach based on FlowAMOEBA is that a single matrix suffices instead of three separate matrices. Moreover,

this solution does not require major modification of original autoregressive models and it is easy to understand.

4.3 Method in Detail

4.3.1 AMOEBA

As the theoretical basis of this method, it is necessary to first introduce how AMOEBA works. AMOEBA (A Multidirectional Optimal Ecotope-Based Algorithm) was originally proposed in Getis and Aldstadt (2004) and Aldstadt and Getis (2006) as a spatial cluster identification method and a spatial weight matrix construction tool. In contrast to other cluster detection methods such as SaTScan (Kulldorff 1997), AMOEBA does not make the implicit assumption that clusters are circular and compact regions, which may allow for the inclusion of low-value spatial events in identified clusters of high values and vice versa (Duque et al. 2010). Instead, AMOEBA follows a bottom-up strategy to identify irregular-shaped ecotopes of high/low values without having such false-positive error. It starts with one or more seed cell (spatial unit) to which neighboring cells are iteratively included until the maximum (or minimal) magnitude of the local spatial statistics, e.g. local G statistics (Getis and Ord 1992; Ord and Getis 1995) has been reached.

For a given cell i, local G statistic G_i^* is defined as follows:

$$G_i^* = \left(\sum_{j=1}^{N} w_{ij}x_j - \bar{x}\sum_{j1}^{N} w_{ij}\right) \Big/ S\sqrt{\frac{N\sum_{j=1}^{N} {w_{ij}}^2 - \left(\sum_{j1}^{N} w_{ij}\right)^2}{N-1}} \qquad (14)$$

where w_{ij} are the spatial weights that reflect the proximity between cell i and cell j, N is the total number of cells (spatial units), x_j is the value of the attribute at cell j, $\bar{x}$ is the mean of all values, and

$$S = \sqrt{\frac{\sum_{j=1}^{N} x_j{}^2}{N} - \bar{x}^2} \tag{15}$$

The nature of local G statistics is that it follows an asymptotical distribution as a normal N (0, 1). A positive (negative) value of G_i^* indicates the presence of a cluster of high (low) values of attribute *x* around cell *i*. The original AMOEBA algorithm (Getis and Aldstadt 2004; Aldstadt and Getis 2006) works as follows: first it picks a seed cell *i* and calculates its G_i^* value. A positive (negative) G_i^* value indicates that the value of the attribute *x* at cell i is larger (lower) than the overall mean $\bar{x}$. Next, the algorithm tries to expand the ecotope in space from the seed cell. G_i^* statistic is calculated for every possible combination of neighboring cells of cell *i*. After each calculation, the new G_i^* is compared with the original value at seed cell. If the absolute value of the new G_i^* is larger, then such combination is considered meaningful and the neighbor(s) are included into the ecotope.

The algorithm repeats this process for every neighboring cell of cell *i* and forms a stable ecotope by including only the neighbors that increase the absolute value of G_i^*. Next, the included cells are considered as a new region (ecotope) and the above expansion process will be carried out based on it. The neighboring cells of this ecotope are included in the calculation to determine whether or not they will be included to a larger ecotope. This iterative process of identifying sets of neighboring cells that maximize the value of G_i^* is repeated until it fails to increase the absolute value of the G_i^* statistic by addition of new neighbors. By then a stable ecotope is identified with respect to the seed cell *i*.

After the ecotope for each and every cell within the study area is identified, the algorithm keeps the non-overlapping ecotopes with the highest G_i^* values. A Monte-Carlo

simulation is performed to examine the statistical significance of each ecotope. The ecotopes (or clusters) that pass the significance level are reported as the final result.

The original AMOEBA algorithm is a bottom-up and exhaustive approach designed to identify the high-valued or low-valued ecotopes, in other words, subsets of geographically connected cells within which all spatial units carry high/low value. Despite the brilliant idea and many advantages, its computational cost is so high that it cannot be practically applied to sizable datasets. The most time-consuming step is to test every possible combination of neighboring cells in order to expand the ecotope. For example, an ecotope with 20 neighbors requires 1,048,575 iterations to fully explore the search space (Duque et al. 2010). Later, Duque et al. (2010) proposed a constructive approach as the solution. They adjust the original local G function as Equation (16).

$$G_R^* = \left(\sum_{i \in R} x_i - n\bar{x}\right) / s\sqrt{\frac{Nn - n^2}{N-1}} \tag{16}$$

The equation is almost identical to Equation (14), except that the local G statistic is calculated for the associated region, or ecotope, rather than one cell i. While expanding ecotope in space, this constructive AMOEBA approach first sorts the neighbors in the descending order based on their absolute value. Then it sequentially includes the neighbors to the ecotope and calculate the G_R^* as Equation (16). The inclusion process stops when adding the kth neighbor that cannot increase the absolute value of G_R^* anymore. Therefore, it saves the time of examining the rest of neighbors which have smaller absolute value than the kth one.

Compared with the original exhaustive search of all possible combinations, this constructive AMOEBA significantly improves the computing efficiency without losing optimality. The approach is packaged as an online Python library, namely clusterPy (Duque et al. 2011), publicly available to others. The FlowAMOEBA approach introduced in this paper is based on it. Most recently, AMOEBA has been further developed to suit the requirements of the big spatial data era (Aldstadt et al. 2012). For instance, Widener et al. (2012) developed a parallel computational implementation of AMOEBA which further boosts the computing efficiency and capability.

4.3.2 Flow Neighbor Relationships

To extend the AMOEBA algorithm to FlowAMOEBA, I consider each flow as a single object. A crucial step is to define the spatial relationship between flow events so as to regulate the ways of expanding from "seed" flows toward their neighboring flows. The principles to define this spatial relationship are straightforward: to well represent flow's spatial relationships and to serve as reference for the algorithm to iterate the process of searching and calculation.

In previous two chapters I have defined a set of flow distance metrics; here I choose to define spatial flow neighbor relationships based on contiguity. Please note that either distance or contiguity works, because they are both basic measures of spatial relationships. Distance can well handle point-based individual flows, while contiguity might be the better choice for zone-based aggregate flows. Given that FlowAMOEBA takes a bottom-up strategy, defining neighboring relationships boils down to assigning spatial weights between every two individual flows. The spatial weights are then used for deciding the

iteration process of searching and expanding from the "seed" flows. Taking the blue flow in Figure 24 as an example, several situations can be considered as follows.

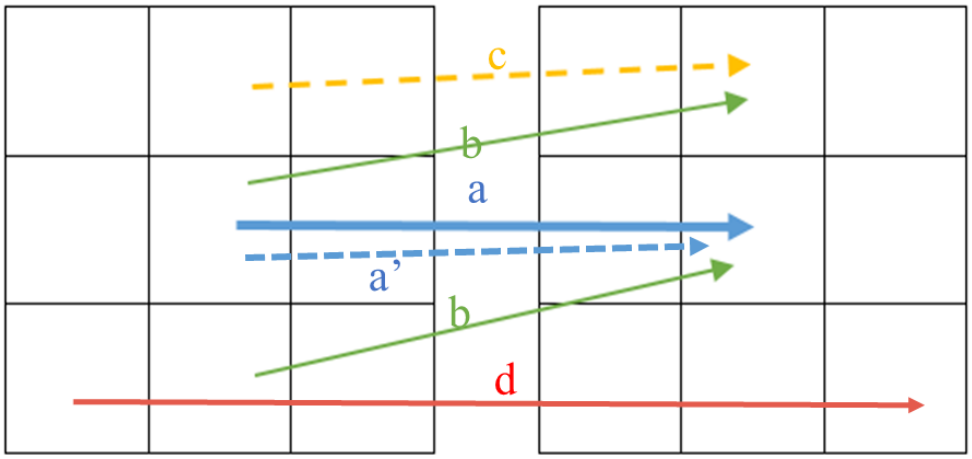


Figure 22: Different situations of flow neighboring relationship

Situation (1): same origin and destination. For example, the origins and destinations of the dashed blue flow *a'* in Figure 24 are the same as for flow *a*. In such case these two flows should be aggregated as a single flow first. The value carried by each flow should be aggregated accordingly as well, for instance to sum the number of migrants while aggregating migration flows that share the same origin and destination.

Situation (2): same origin (destination) while destinations (origins) are neighbors. For example, either of the two green flows *b* in Figure 24 shares one same end zone with flow *a*, while their other end zone are adjacent. In this case, both green flows are regarded as neighbor of flow *a* and the spatial weight between them is set to $w_{ij} = \lambda_1$ ($\lambda_1 > 0$).

Situation (3): both origins and destinations are neighbors. This also indicates the two flows are neighbors and the spatial weight between them is $w_{ij} = \lambda_2$ ($\lambda_2 > 0$). In Figure 24

the yellow flow *c* represents such situation where both origin and destination are neighbors of flow *a*.

Situation (4): neither origins nor destinations are the same nor near neighbors. This means the two flows are not neighbors and their spatial weight is $w_{ij} = 0$. In Figure 24, the red flow *d* is not considered a neighbor of flow *a* if we derive polygon contiguity based on Rook's Case. However, if Queen's case is adopted, flow *d* is considered as neighbor of flow *a*, just like flow *c*.

The situations discussed above are based on the contiguity relationship of the basic spatial unit that flows are aggregated into. This is of course not the only way to define flow neighborhood given that uncertainties remain regarding the choice of basic unit, for example rectangular or hexagonal grid cells, smallest possible administrative region, Thiessen polygons, etc. Also there are other ways to measure the relationship among basic units in addition to contiguity, such as distance and density. If one chooses to use other types of basic units or definitions of neighborhood, the same logic of categorizing situations into the above four can be applied. However, it is the goal of FlowAMOEBA to remove the dilemma of choosing the actual spatial interacting regions via taking advantage of the large volume and fine spatial granularity of individual flows.

4.3.3 Growing Process of Flow Ecotope

In this section I explain the growing process of a flow ecotope in space from the seed flow. Figure 25 shows a study area consisting of 25 spatial units (cells), and several spatial flows starting at cell 1.

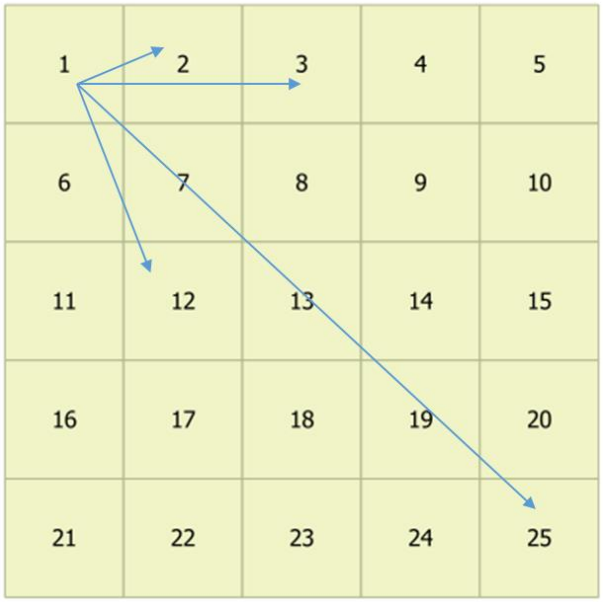


Figure 23: Example of flows within a 5 × 5 grid cells

Table 3 is an OD matrix summarizing all spatial flows within the region. The first column is the identifier of origin cell while the first row is the identifier of destination cell. Each entry in the matrix represents the value of a flow, for example flow from cell 1 to cell 2, denoted as $F_{1,2}$, has a value of 100. If the study application is migration, it means there are 100 migrants from cell 1 moving to cell 2. On the diagonal of the matrix the flow values are noted as '–'. This is because a flow cannot have both origin and destination at the same location. Theoretically, the maximal number of unique OD flows is n*(n-1) for inner-region interactions (e.g. domestic migration within the U.S.), and n*n for cross-region interactions (e.g. international migrations from Mexico to the U.S.). In example of Figure 25, there can be as many as 600 unique flows. However the actual total number of flows are usually much smaller than the theoretical maximum. For instance from 2010 to 2014 there are 4, 073 county-to-county migration flows within North Carolina, in comparison with the theoretical maximum 9,900. For the whole contiguous U.S. there are only 419,775

unique county-to-county flows, in contrast with theoretical maximum 9,656,556. Therefore, except for the diagonal there are also other '–' in the matrix, for example there is no flow starting from cell 1 and ending at cell 4. It means there exists no actual interactions between these two cells.

Table 3: OD Matrix

O\D	1	2	3	4	5	…	24	25
1	–	100	15	–	30	…	120	150
2	50	–	25	80	17	…	10	40
3	12	15	–	77	–	…	35	36
4	19	56	33	–	150	…	15	–
5	66	–	56	34	–	…	–	29
…	…	…	…	…	…	–	…	…
24	10	48	–	33	–	…	–	30
25	70	37	189	66	–	…	25	–

The OD matrix can be further converted to a dictionary-like data structure by keeping only the non-zero flows. This is an effective strategy to reduce computer memory requirements, thus boosting computational performance. In the dictionary structure flows are represented in a form like (O_i, D_i): x_i and they are separated by comma. The tuple (O_i, D_i) indicates the origin and destination of a flow and serves as the key of this flow element, while x_i corresponds to the flow value. Thus the data look as follows:

{(1, 2): 100, (1, 3): 15, (1, 5): 30, …, (25, 22): 15, (25, 23): 40, (25, 24): 10}

Applying the spatial flow neighbor relationship introduced in the previous section, a spatial flow neighbor dictionary can be obtained. The dictionary is created for fast query

of the neighbors of flows; thus, it indicates the flow ecotope where to expand in the next step. In this dictionary, each element contains a flow and its neighboring flows in the form of (O_i, D_i): [(O_1, D_1), (O_2, D_2), …, (O_n, D_n)]. The list inside the square brackets contains the neighboring flows of flow from O_i to D_i. For the case depicted in Figure 25, the flow neighbor dictionary looks like:

{

(1, 2): [(1, 1), (1, 3), (1, 7), (2, 1), (2, 2), (2, 3), (2, 7), (6, 1), (6, 2), (6, 3), (6, 7)],

(1, 3): [(1, 2), (1, 4), (1, 8), (2, 2), (2, 3), (2, 4), (2, 8), (6, 2), (6, 3), (6, 4), (6, 8)],

…

(25, 24): [(25, 19), (25, 23), (25, 25), (20, 19), (20, 23), (20, 24), (20, 25), (24, 19), (24, 23), (24, 25)]

}

As stated earlier, AMOEBA starts from a seed cell and expands the cluster of high (low) values, also known as the ecotope, to the neighboring area. Similarly, FlowAMOEBA starts from a seed flow and expands the flow ecotope towards its neighbors. Next, I illustrate how the flow ecotope grows over space step by step. Figure 26a shows the initial status of the seed flow $F_{1,25}$ with the blue cell and red cell representing origin and destination, respectively.

The first step is to calculate the local G statistic for the seed flow with Equation (16). Obtaining the local G statistic value as $G^*_{1,25} = 1.5$, which is positive and indicates that flow $F_{1,25}$ carries a higher-than-average value. Accordingly, in the following steps the algorithm will try to include more neighboring flows in order to increase this statistic value.

In the second step, the algorithm expands the flow ecotope towards the neighboring flows. Referring to the flow neighbor dictionary, the seed flow $F_{1,25}$ has eight neighboring

flows, namely $[F_{1,20}, F_{1,24}, F_{2,20}, F_{2,24}, F_{2,25}, F_{6,20}, F_{6,24}, F_{6,25}]$. Since FlowAMOEBA inherits the features of the constructive AMOEBA (Duque et al. 2010; Duque et al. 2011), these eight neighboring flows are sorted in a descending order based on their flow value. The sorted neighboring flows become a queue as $[F_{1,24}, F_{6,24}, F_{6,25}, F_{2,20}, F_{2,24}, F_{6,20}, F_{1,20}, F_{2,25}]$. Next the algorithm picks the first neighbor $F_{1,24}$ and merge it with the seed flow $F_{1,25}$ as a flow ecotope. The local G statistic is then calculated for this new flow ecotope and the value is compared with the original $G^*_{1,25}$. If the statistic becomes larger, then this neighboring flow $F_{1,24}$ is considered contributive and is confirmed to be included to the flow ecotope. The same procedure is applied for the other neighboring flows one by one in the descending order. It stops when the local G statistic of the flow ecotope fails to increase by adding new flows. For instance after successful inclusion of the first three neighbors $F_{1,24}$, $F_{6,24}$, and $F_{6,25}$, the expansion stops at $F_{2,20}$. This means the algorithm has finished searching and expanding to the first-order neighbors of the seed flow. At this point the flow ecotope evolves in the status illustrated by Figure 26b, where both origin and destination include one additional cell.

The expansion has not stopped yet. The same process as in the second step is repeated to search and include the second-order neighbors, namely the neighbors of neighbors. Again, the criterion for including any new flow is whether or not the local G statistic of the flow ecotope increases after inclusion. For instance, in this step some of the second-order neighboring flows are added and the flow ecotope now expands as Figure 26c.

This iterative and expansion process is repeated for new neighbors of the flow ecotope until it fails to increase the absolute value of the G_i^* statistic any more. By then a stable flow ecotope is identified with respect to the seed flow $F_{1,25}$.

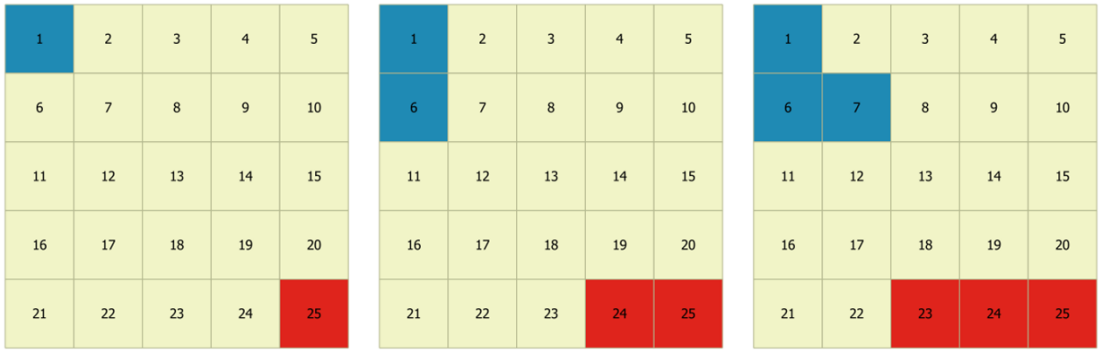


Figure 24: Growth flow Ecotope with blue denoting origin and red denoting destination. (a) Seed flow; (b) Ecotope expands to first-order neighbors; (c) Ecotope expands to second-order neighbors

The steps above illustrate how FlowAMOEBA works to extract a stable flow ecotope for a given seed flow. For flows with below-average value, the process is very similar, except that the goal is to minimize the local G value instead of to maximize. The same process has to be performed N times (N is the total number of flows). After the flow ecotope for each and every flow within the study area is identified, the algorithm keeps the non-overlapping ecotopes with the highest G_i^* values. The next and the final step is to test statistical significance, which is discussed in the following section.

4.3.4 Test Statistical Significance

Given that AMOEBA is fundamentally a spatial statistical approach, a necessary step before drawing any conclusion is to test the statistical significance of the results. In the first method Flow K-function, I have discussed the importance of significance testing and

justified that Monte Carlo simulation is the appropriate way to carry out the test for spatial flows. However, the situation here becomes more complex than for the previous two methods. This is because FlowAMOEBA is designed for aggregate flows that carry nonspatial attributes. Except for flow's spatial elements including endpoint locations, length, and flow direction, the nonspatial attribute value of flow is another critical variable to consider in the simulation process.

Again, there is no unique way to simulate spatial flows for significance testing. Given that the aggregate flows have both spatial and nonspatial elements, there are at least three general ways to design the randomization of the simulation. An analogy can be adopted here to better describe the data: we can see the flows as jars filled with jam. The jars represent flow's spatial elements, namely the vector line, and the jam represent the flow value. The first way is to control spatial elements the same while permutating the nonspatial attributes. This means that we keep all the jars unchanged, i.e. same sizes, shapes, and locations, but we permutate the amount in the jams and fill them in the jars randomly. The second way is the opposite, i.e. to control the nonspatial attributes the same but to randomize the spatial elements. In the jars and jam analogy, it means the amount of the jam in each jar stays unchanged, but the locations of jars are redistributed randomly. The third way is the combination of the previous two which means both spatial and nonspatial elements of flows are randomized.

Here I do not conclude which way is the optimal choice, because it is the goal of such exploratory spatial data analysis method to reflect the respective underlying geographical processes and can also help us contemplate unknown ruling attributes contributing to the spatial pattern. The most reasonable simulation method is always the one that fits the

context the best. No matter which way is chosen to carry out the simulation, only those flow ecotope that passed the significance level are saved as the final ecotopes.

4.3.5 Construct a Spatial Flow Weights Matrix

Spatial weights matrix (W) is a classic notion in spatial modeling. It is particularly useful to account for spatial association in spatial regression models. Getis and Aldstadt (2004) reviewed over a dozen different types of W. They summarize that there are three kinds of representations behind the designs of spatial weights matrices. The first representation is a theoretical notion of spatial association, such as a distance decay function. A *W* can also be designed as a geometric indicator of spatial nearness such as polygon contiguity. And the last one is to construct a W as descriptive expression of spatial association within a set of data, such as empirical variogram functions. In the original paper of AMOEBA, Aldstadt and Getis (2006) create spatial weights matrix following the third kind of representation. The greatest advantage is to allow study data speak for themselves, as they create *W* based on the spatial associations identified by AMOEBA.

Similarly, spatial flow weights matrix (W_f) created here is an empirical representation of network autocorrelation. It is derived from flow ecotopes identified by FlowAMOEBA. To create W_f, the first thing is to divide flows into two categories: those belong to a flow ecotope and those do not. The idea is that if a flow has spatial association with any other flow, it is certainly part of a flow ecotope because the smallest size of a flow ecotope is two. Therefore, to design W_f that can best capture spatial associations of the data, only those flow ecotope members need to be taken care of.

Taking the previous example, assume the flow ecotope shown in Figure 26c is the only one that passes significance test as identified by FlowAMOEBA, then only the flows origin from cell 1, 6, or 7, and end at cell 23, 24, or 25 have strong spatial associations with each other. Other flows such as $F_{9,16}$ belong to no flow ecotope and their spatial weights equal zero. In the W_f matrix, each row lists the spatial weights between one flow and every other flow within the same flow ecotope. Using w $[F_{i,j}, F_{u,v}]$ to represent spatial weight between flow $F_{i,j}$ and $F_{u,v}$, in this example the row for $F_{1,25}$ consists of w $[F_{1,25}, F_{1,23}]$, w $[F_{1,25}, F_{1,24}]$, w $[F_{1,25}, F_{6,23}]$, w $[F_{1,25}, F_{6,24}]$, w $[F_{1,25}, F_{6,25}]$, w $[F_{1,25}, F_{7,23}]$, w $[F_{1,25}, F_{7,24}]$, and w $[F_{1,25}, F_{7,25}]$. W_f is designed as row standardized which means the sum of weights in one row equals to 1. And the values of weights are assigned by the relative weighting scheme as follows.

$$\text{w}\left[F_{i,j}, F_{u,v}\right] = \rho_{ij,uv}\left(FDist_{max}(F_{i,j}) - FDist\left(F_{i,j}, F_{u,v}\right)\right) \tag{17}$$

where $FDist\left(F_{i,j}, F_{u,v}\right)$ denotes the flow distance between flow $F_{i,j}$ and $F_{u,v}$. $FDist_{max}(F_{i,j})$ is the maximal distance from $F_{i,j}$ to any other flow within the same flow ecotope. $\rho_{ij,uv}$ is the parameter for row standardization. This design means within a flow ecotope, a smaller flow distance leads to a larger relative spatial weight between two flows.

In the previous section I have defined flow neighbor relationships based on contiguity of origin and destination regions. So the distance between two flows in this example means the moves from one flow to the other. For example $F_{1,25}$ and $F_{1,24}$ share the same origin and their destinations are one move away, so $FDist\left(F_{1,25}, F_{1,24}\right) = 1$. Similarly, I can obtain other distances for $F_{1,25}$ such as $FDist\left(F_{1,25}, F_{6,24}\right) = 2$, $FDist\left(F_{1,25}, F_{7,24}\right) = 3$,

$FDist\left(F_{1,25}, F_{7,23}\right) = FDist_{max}\left(F_{1,25}\right) = 4$. With Equation (17) I can calculate the corresponding spatial weights, for instance w [$F_{1,25}$, $F_{1,24}$] = 3/14, w [$F_{1,25}$, $F_{1,23}$] = 2/14, w [$F_{1,25}$, $F_{7,23}$] = 1/14, and of course the total weights for $F_{1,25}$ equal to 1.

As seen, spatial flow weights matrix is designed not only based on spatial relationships such as contiguity, but spatial associations between flows within the same flow ecotope. Compared with conventional spatial weights matrix based on pure spatial relationships such as contiguity, W_f excludes those spatially adjacent flows but with no explicit spatial association, for example w [$F_{1,25}$, $F_{2,25}$] = 0. On the other hand, W_f takes into consideration of those spatial associated yet not directly adjacent flows, for example w [$F_{1,25}$, $F_{7,23}$] = 1/14.

To incorporate W_f to spatial interaction models, I can modify the common model by adding a spatial lag variable as Equation (18).

$$y = \rho_f W_f \mathrm{y} + \rho_o X_o + \rho_o X_o + \alpha\iota_N + \gamma \mathrm{g} + \varepsilon \qquad (18)$$

where y is the dependent variable or flow value. A spatial lag variable $W_f \mathrm{y}$ is added to account for spatial associations of flows, or known as network autocorrelation in the literature. X_o and X_d represent characteristics of flow origin and destination, respectively. ι_N is the constant term. g represents distance or flow length. ε is the error term. ρ_f, ρ_o, ρ_o, α, and γ are the corresponding parameters.

4.4 Experiment

4.4.1 Evaluation with Synthetic Data

A synthetic dataset is created to test this method. As shown in Figure 27, I have designed two 10 × 10 lattices, with one as the flow origin area and the other as flow destination area, respectively. Flows are created in the way that each of them originates at one cell on the left and ends at another cell on the right. Technically, there can be up to ten thousand different OD pairs. To mimic reality, I create 2,000 unique flows with non-zero values. Out of these 2,000 flows, I assign a high value to 200 of them, a low value to another 200 of them, while the rest is assigned a medium value. For a medium-value flow, I randomly pick a cell in Figure 27a as the origin, and randomly pick another cell in Figure 27b as destination (no redundant OD pairs). For anomalous-value flows, I design them as four groups, of which two are high-value and two are low-value. Figure 27 shows corresponding origin and destination cells of each group. The values are assigned following a process similar to the original AMOEBA paper (Aldstadt and Getis 2006), that is the high, medium, and low values are randomly assigned to each flow according to a normal distribution as N (150, 5), N (100, 5), and N (50, 5), respectively.

Extreme situations are carefully crafted to test the capability of FlowAMOEBA in terms of delineating the boundaries of anomalous interacting regions constructed for the synthetic dataset. As shown in Figure 27, some of flows with extreme high or low values are assigned to four groups of corresponding concentrated OD regions. They serve as targets or baits of the test. Several thoughts are behind the design. First, they all come with irregular-shaped OD regions. This is common but essential to test the accuracy of most cluster analysis methods. Second, pitfalls of false-positive errors are embedded. For

instance the “holes” (cell with regular value) at origin cell 83 and destination cell 28 might be challenging to avoid being included as positive results. Third, overlaps of flow’s O/D regions are intentionally created to add more difficulties to the task. In the origin region, there are four cells, namely cell 23, 24, 33, and 34, belong to two groups of high-value flows at the same time. On the right side there is a more extreme case that cell 53 and 63 are the destination cells of a group of high-value flow and a group of low-value flow simultaneously.

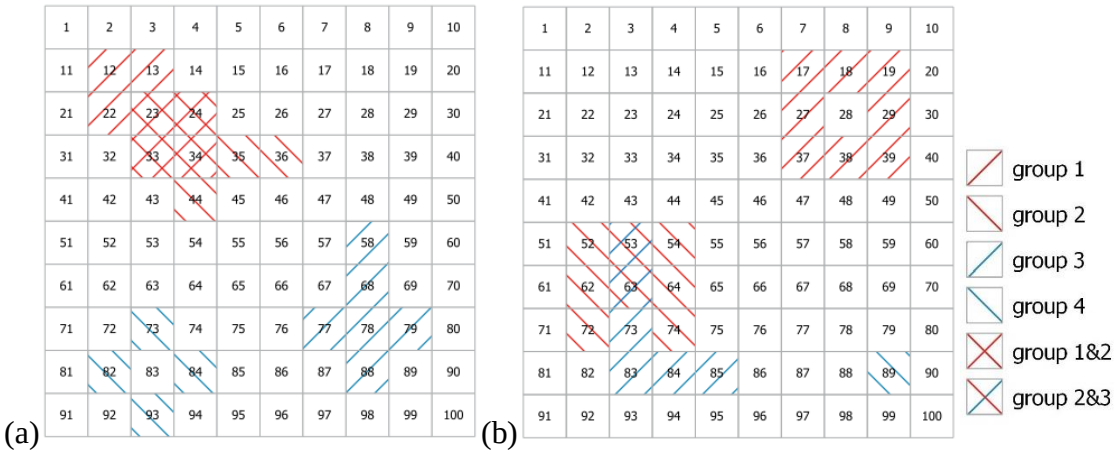


Figure 25: Synthetic dataset. (a) origin grid cells; (b) destination grid cells

FlowAMOEBA has been implemented on this synthetic dataset. Outcome of the approach is to add a group label to each of the 3,000 flows. By default flows with medium value are labeled as group 0, and extracted flow ecotopes of high (low) value are labeled as 1 (-1), 2 (-2), 3 (-3) … A 1,000-time Monte Carlo simulation is carried out to extract results at 0.1% significance level.

The results show that FlowAMOEBA has successfully extracted the designed interacting regions of extreme high (low) flow value, namely flow ecotopes. In spite of

irregular shapes of these designed regions, no cell outside regions is mistakenly included as false positive error, neither is any cell inside designed regions omitted as false negative error. Two high-value flow ecotopes and one low-value flow ecotope are identified. Flows of group 1 that start from one of the origin cells {12, 13, 22, 23, 24, 33, 34} and end at one of the destination cells {17, 18, 19, 27, 29, 37, 38, 39} are extracted as ecotope 1. Flows of group 2 that start from one of the origin cells {23, 24, 33, 34, 35, 36, 44} and end at one of the destination cells {52, 53, 54, 62, 63, 64, 72, 74} are identified as ecotope 2. And flows of group 3 that start from one of the origin cells {58, 68, 77, 78, 79, 88} and end at one of the destination cells {53, 63, 73, 83, 84, 85} are labeled as ecotope -1. However, flows of group 4 that start from one of the origin cells {73, 82, 84, 93} and end at the destination cell {89} are not identified as flow ecotope by FlowAMOEBA.

The results of this experiment demonstrate that FlowAMOEBA can accurately extract the actual spatial interacting regions under extreme circumstances. The irregular shapes of the designed targets have not been a problem as the boundaries of extracted flow ecotopes are precisely delineated. Overlaps of origin or destination regions of flow ecotopes are also resolved very well. This proves that FlowAMOEBA strictly follows the definition of flow neighborhood, which will not be affected if only one of the end regions overlapped in space. The “holes” inside target groups have not created false-positive errors or false-negative errors in the outcome. On one hand, destination cell 28 is not mistakenly included to flow ecotope 1. On the other hand, the normal value at origin cell 83 leads to the failure of detecting group 4 as a flow ecotope is also expected. The reason is that Rook’s case of contiguity rule is adopted to determine flow neighborhood in this experiment. Hence without high-value flow ending at destination cell 28, the other cells around it can still form

a connected region. But without a low-value flow starting at origin cell 83, the other four flows (73, 89), (82, 89), (84, 89), (93, 89) cannot be neighbors to each other. Accordingly, they cannot form as a larger interacting region and group 4 is not identified as flow ecotope by FlowAMOEBA. To verify this, a follow-up test is conducted. By modifying the value of flow (83, 89) to 50 (the mean of low-value flows), FlowAMOEBA detects the flows starting at one of the origin cells {73, 82, 83, 84, 93}and end at the destination cells {89} as a low-value flow ecotope. The result will also be different if using Queen's case as the contiguity rule instead of Rook's.

4.4.2 Experiment with Migration Data

To test how FlowAMOEBA can be useful in real-world applications, I choose to use county-to-county migration flow data. In recent years, the Carolinas, namely North Carolina (NC) and South Carolina (SC), have witnessed significant population growth. One reason is the large amount of inflow migrants. According to U.S. Census Bureau, during 2010 to 2014 there were 911,378 domestic residents moving in and 839,463 moving out of the Carolinas which results in 71,915 net inflow migrants. The top three most popular states of these migrants' origin are New Jersey (NJ), New York (NY), and Pennsylvania (PA) which altogether contribute 42% of the total migrations to the Carolinas. In this application, I focus on the gloss migration flows from the NJ-NY-PA region to the NC-SC region (Figure 28). With FlowAMOEBA, interesting patterns can be extracted to show where exactly these migrants come from and where have they resettled, thus quantifying and locating the actual interacting regions.

The raw migration data are downloaded from the website of U.S. Bureau of the Census. The basic spatial unit is the county. There are 150 counties in the origin NJ-NY-PA region,

and 146 counties in the destination NC-SC region. The raw data contain 2,356 unique county-to-county migration flows between these two areas, which accounts for 10.8% of the theoretical maximum number of OD pairs (150 × 146). For instance there is no person from Centre County, PA moving to York County, SC during these five years. FlowAMOEBA has been implemented with the actual amount of migration, as well as the ratios by accounting for impacts of population.

Figure 26: County-to-county migration flow from NJ-NY-PA to NC-SC

FlowAMOEBA is carried out using the Rook's case contiguity rule to determine flow neighborhoods. Monte Carlo simulation is used to test statistical significance and it is designed to suit the context. As stated earlier, there are three general ways to simulate the situation of null hypothesis, i.e. randomize spatial elements only, randomize nonspatial attributes only, and randomize them both. Here I choose to randomize the nonspatial attributes only as the simulation strategy for two reasons. First, in migration study the emphasis leans to the distribution of flows' attribute rather than the vector themselves. In

other words, it is of more interest to find abnormal number of migrants from one region to another than to find where migration can happen. Second, given the low percentage (10.8%) of non-zero OD pairs in the observation data, it would be implausible to randomly pair an origin county with a destination county, thus creating new OD flows. Because 89.2% of the chance a randomly generated OD pair carries no migration in reference to the raw data. Hence randomizing the spatial elements of flows (OD vectors) would not simulate a reasonable benchmark situation. As a result, I choose to simulate by randomly assigning migrants to the existing OD pairs.

Setting the significance level as 0.001, five high-value flow ecotopes are extracted using actual amount of migrants from NJ-NY-PA to NC-SC. Figures 29 to 33 illustrate these flow ecotopes, with blue represents origin and red represents destination. Figure 29 shows that there were many people from New York City and Long Island moving to Charlotte. This indicates concentration of migration flows from one major city to another, along with some spatial spillover effects.

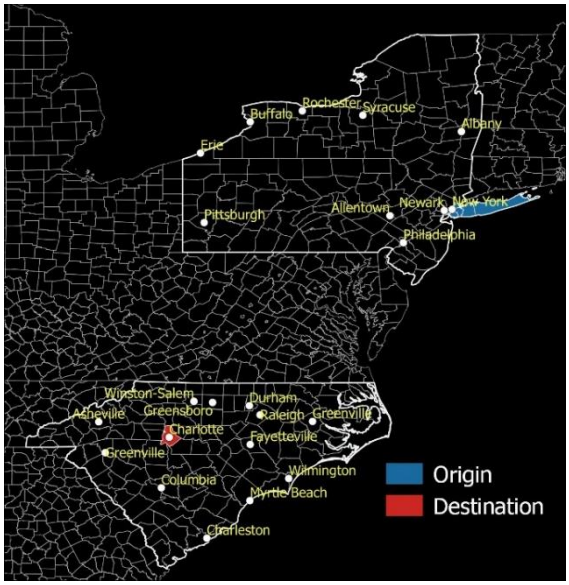


Figure 27: Migration flow ecotope 1

As the largest city in the Carolinas, Charlotte attracts migrants from other regions as well. Figure 30 shows concentrated migrations from Buffalo area to Charlotte area. This is an example of the smallest flow ecotope. It takes at least two neighboring flows with extreme value to form an ecotope with FlowAMOEBA. In this case, the destination for migrants from Buffalo contains Mecklenburg County and Lincoln County on its west.

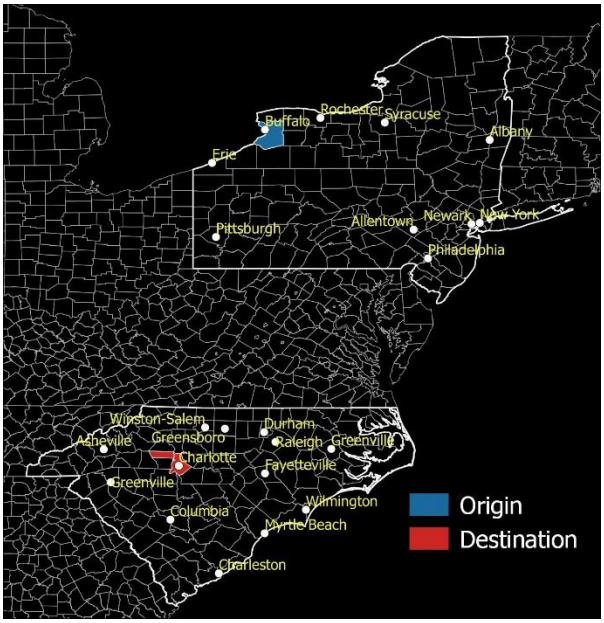


Figure 30: Migration flow ecotope 2

Figure 31 is a very good example of how FlowAMOEBA deals with irregular shapes. The origin region of this flow ecotope concentrated at New York City. On the other side, the dumbbell-like are connects Columbia area and Charleston area as the whole destination region, which validates the strictness of the algorithm on controlling flow ecotope's growth to include only the anomalous flows.

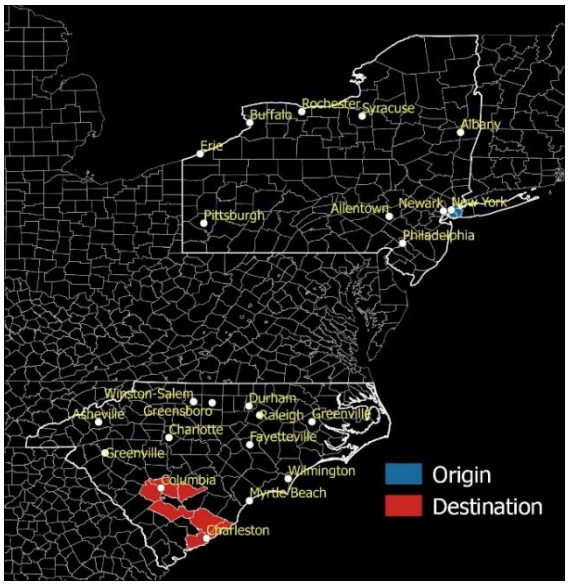


Figure 28: Migration flow ecotope 3

Figures 32 and 33 show large flow ecotopes. The two origins are the most populous area on the East Coast, namely NYC-Newark-Philadelphia mega region, with some differences about the surrounding satellite cities. The destination in Figure 32 combines Raleigh-Durham-Chapel hill and Fayetteville metropolitan area. While the destination in Figure 33 connects Columbia, Wilmington, and Myrtle Beach as a whole. Although these cities are connected as a huge destination region, their actual connections are debatable. The corridor-like destination region in Figure 33 might be a counterexample of using contiguity to decide neighbor relationships, as geographic connection does not necessarily indicate actual tight connection between two places.

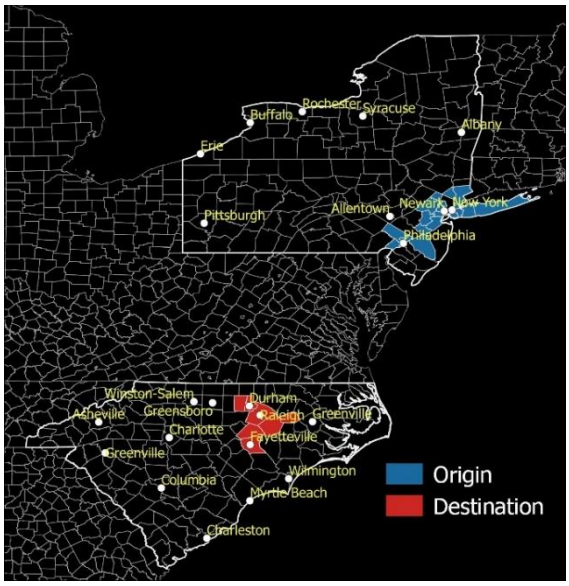


Figure 29: Migration flow ecotope 4

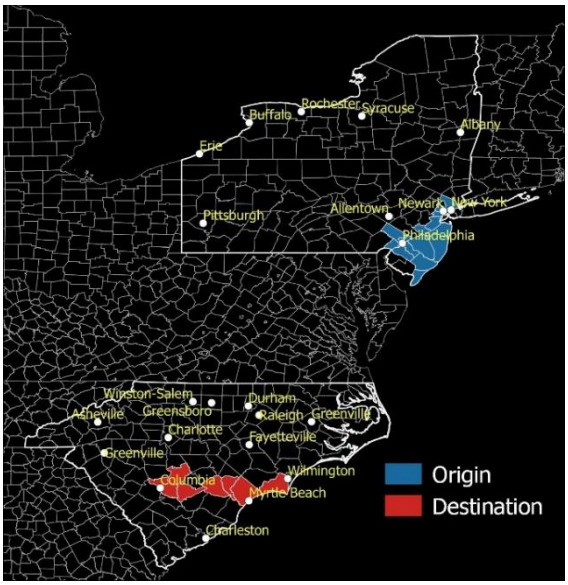


Figure 30: Migration flow ecotope 5

From this experiment with migration flows, some interesting patterns have been extracted. Comparing these results with common hypotheses of spatial interaction modeling, population's influence is obvious but distance's effect is not. Most flow ecotopes are between very populous regions as the heavily interacting regions contain one or several big cities and the surrounding area. Nevertheless, population is not the deciding factor.

Populous places such as Pittsburgh and Syracuse are not detected as popular origins, neither has Greenville-Spartanburg-Anderson been identified as popular destinations. Moreover, the formation of a flow ecotope is more than to identify places with large inflow or outflow migrants, but also how they pair with each other as origin-destination. On the other hand, there is no clear evidence to support distance's effect, which is understandable. Given that migration happens once in a long period, traveling a few hundred miles more on the way is not a big concern in comparison with other compelling reasons like job opportunity.

In order to account for population's impact on migration patterns, further tests have been done with ratios. Three ratios called Ratio_O, Ratio_D, and Ratio_OD, have been designed by dividing amount of migrants by population of origin county (Equation 19), by population of destination county (Equation 20), and by population of both origin and destination counties (Equation 21), respectively.

$$\text{Ratio\_O} = \#\ of\ migrants / origin\ population \tag{19}$$

$$\text{Ratio\_D} = \#\ of\ migrants / destination\ population \tag{20}$$

$$\text{Ratio\_OD} = \frac{\#\ of\ migrants}{(origin\ population * destination\ population)} \tag{21}$$

Figure 34 shows the only flow ecotopes identified by FlowAMOEBA using Ratio_O. Compared with results directly using amount of migrants, distinct patterns are found by removing the impacts of origin population. The origin of flow ecotopes is Seneca County, the population of which was only 35,251 according to the 2010 census. And the identified

destination contains two counties near Charleston, which are not populous either though Ratio_O does not account for destination population.

Figure 35 shows two flow ecotopes identified by FlowAMOEBA using Ratio_D. The results are similar with the ones in Figure 32 and Figure 33 to some extent. It implies weak impacts of destination population on migration patterns. In other words, large population of destination does not have an obvious positive relationship to destination attractiveness. This implication is also verified by the only result using Ratio_OD, which is exactly the same as the first one in Figure 34. Overall, patterns using ratios especially Ratio_O can better reveal the preferences of migrants. In confirmatory studies of migration, it is potentially useful to incorporate such result and spatial flow weights matrix derived from it to improve discovering, predicting, and explaining migration flows.

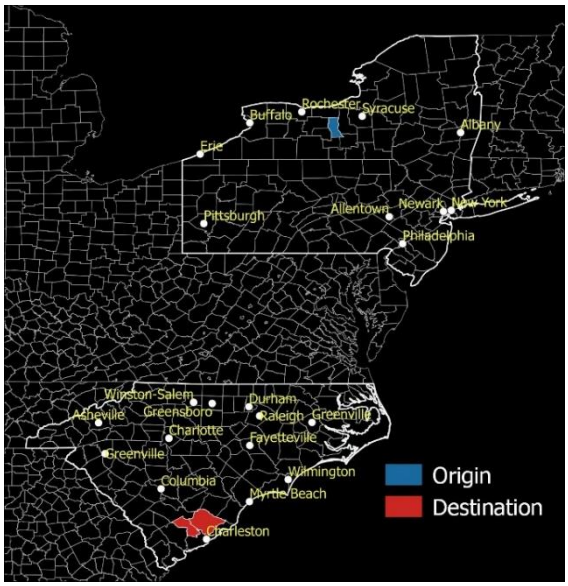


Figure 31: Flow ecotopes identified with Ratio_O

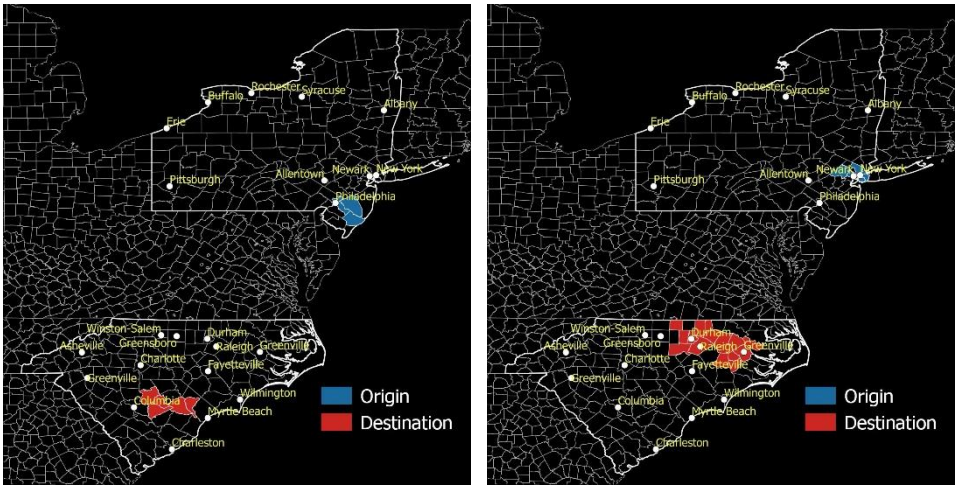


Figure 32: Flow ecotopes identified with Ratio_D

4.5 Summary and Future Directions

Experiments with a synthetic dataset and with a migration dataset verified the functionality of FlowAMOEBA and depicted its characteristics. There are several clear advantages of this novel method. First and most importantly, it can delineate regions of anomalous spatial interactions. Irregular shapes, overlaps of origin or destination regions, as well as false-positive errors and false-negative errors can be successfully overcome. Second, the method is developed based on well-accepted concepts and functions. Flow neighbor relationships are defined based on standard Rook or Queen contiguity, and the core function adopts the widely-used local G statistics. Third, the extracted ecotopes are labeled unambiguously to separate one from another. High-value ecotopes and low-value ones can be easily differentiated as well. Fourth, some designs enable the method to suitably fit the context of larger datasets. The dictionary data structure helps avoid huge memory cost of zero-value OD pairs, which in many cases are the vast majority. Adopting ideas of developed AMOEBA, namely constructive AMOEBA (Duque et al. 2010), can also dramatically lighten the computation. Flexible simulation strategies tailored for

different scenarios guarantee its applicability to more domains. Last but not least, a spatial flow weights matrix is created based on the identified flow ecotopes. And it has great potential to handle network autocorrelation embedded in spatial interaction modeling.

However, some limitations are discovered. While contiguity is easy to understand, sometimes it can create problem just like the example in Figure 33. Fortunately, there exist other options such as distance to build up flow neighborhood. Another shortcoming is the modifiable areal unit problem (MAUP) exists. Choosing the basic spatial unit is always tricky. Setting unit too big cannot take advantage of this bottom-up method, while using very small unit is not necessarily the best as the results might be too scattered.

In terms of future directions, there are two general directions. One is to improve the method itself and the other is to expand application domains. To integrate the latest geocomputation and geovisualization techniques is a natural extension. For instance to boost computation efficiency of the algorithm by processing the ecotope search at multiple seed flows in parallel. Interactive maps can help clarify final results with heavy overlaps. Furthermore, it is of great interest to conduct empirical studies to test how and to what extent that spatial flow weights matrix can improve current spatial interaction models by addressing network autocorrelations.

Applications with massive individual spatial flows is a promising direction. For example, to analyze human movement data obtained from smart fitness devices and ride-sharing system will lead to deeper understanding of mobility in a complex urban system. Beyond interactions in a physical space, interactions happening in a virtual or communication space is also a bright area of future application. For instance to apply

FlowAMOEBA to extract heavily interacted cyber community embedded in social networks has great potential to reveal the information spreading mechanism on social media platforms.

CHAPTER 5: CONCLUSIONS

This dissertation research developed three novel methods of the same subject matter, namely exploratory spatial flow data analysis (ESFDA). These newly developed methods can serve as means for effectively exploring massive new flow data, thus they make unique contributions to the literature. More importantly, they are responses to the challenges and opportunities brought by the recent data revolution, which has significantly enriched spatial interaction data in terms of accessibility, types, volume, and spatiotemporal granularity.

Given the common theme of ESFDA, these three methods share some characteristics. First, "cluster" is a common keyword to describe their functionalities, though it has different interpretations in each method. The first method Flow K-function is designed for detecting "hot flows". In other words, it can detect the local spatial clustering patterns of individual flows. Therefore "clustering" here means a spatial distribution pattern such that flows are statistically more likely to locate close to each other in space, as oppose to random or dispersed distribution. While "cluster" in the second method Flow HDBSCAN means group of flow objects that are similar to each other in terms of their spatial characteristics including location, length, and direction. Unlike the spatial statistical approach Flow K-function, which needs the distribution of the population as the benchmark to reach a conclusion about the existence of a cluster, Flow HDBSCAN is free from any assumption or null hypothesis given its root in spatial data mining. With respect to the third method FlowAMOEBA, "cluster" reflects similarity not just in space, but in nonspatial attributive

dimension as well. The identified regions of anomalous spatial interactions, also called flow ecotopes, are clusters of high (or low) value flows. Compared with the previous two methods, it not only reflects the spatiotemporal closeness of individual flows, but the spatial association of the nonspatial attribute carried by flows.

Beside the common keyword of "cluster", another important aspect is that they all take advantage of fine spatial resolution of flow data. For instance, a set of spatial proximity measures has been designed for flow data by integrating endpoint location, length, and direction. The measures can assess both intra-relationships and inter-relationships of flows and play an important role in Flow K-function and Flow HDBSCAN to overcome issues like MAUP. The bottom-up strategy adopted in FlowAMOEBA is another way to benefit from fine spatial resolution. The strategy guarantees that the boundaries of anomalous interacting regions delineated by FlowAMOEBA are free from restrictions of size, shape, scale, and administrative level, no matter how irregular the final regions look like.

Of course, each of these methods holds its own uniqueness. Flow K-function upgrades the classical hot spot detection method to the stage of "hot flow" detection. Hence it fills the gap that there is no such spatial statistical approach to detect local spatial distribution patterns of flow data, despite abundant methods for point and polygon data. Moreover, it can be easily adjusted to detect the global patterns of the entire study area as well. Flow HDBSCAN is unique in the way that it combines hierarchical clustering and density-based clustering. Therefore, it inherits the strengths of density-based methods that it can extract flow clusters in various situations, including varying flow densities, lengths, and hierarchies. On the other hand, it inherits the advantage of hierarchical cluster analysis methods to reveal the hierarchical structure of extracted clusters if there is any. In addition,

its sole-parameter design guarantees its ease of use and flexibility to explore the data interactively. FlowAMOEBA stands out as the first method to delineate anomalous spatial interacting regions, to the author's best knowledge. It not only offers a way to identify a cluster of high (or low) value flows, but it also creates a spatial flow weights matrix that can potentially improve spatial interaction modeling by accounting for the common issue of network autocorrelation.

These methods are not without limitations. Like any other statistic approach, the population distribution is essential to Flow K-function as it serves as the benchmark of clustering pattern. Although techniques like Monte-Carlo simulation can be used to avoid biased results, it is not intuitive especially for beginners to design an appropriate simulation test. Flow HDBSCAN provides an alternative way to explore flow data without any assumption or null hypothesis. However, not all extracted clusters are meaningful and easy to interpret, since the algorithm returns all flow clusters that meet the criteria. For the last method FlowAMOEBA, the modifiable areal unit problem (MAUP) exists, as setting the basic spatial unit too big cannot take advantage of this bottom-up method, while using very small units is not necessarily the best as the results might be too scattered.

This research also indicates that exploratory spatial flow data analysis (ESFDA) is a promising direction of research. In the future, I propose to build on this foundational work to enhance the ability to analyze massive data sets in ways that leverage the respective strengths of spatial analytical, spatial data mining, and visual analytical traditions. As spaces and places are created by relationships (in most respects analogous to flows), multi-scalar space-time modeling of relational entities will remain a critical priority across various sciences.

REFERENCES

Aldstadt, J., and Getis, A. (2006). "Using AMOEBA to Create a Spatial Weights Matrix and Identify Spatial Clusters." *Geographical Analysis* 38 (4): 327–343.

Aldstadt, J., Widener, M., and Crago, N. (2012). "Detecting Irregular Clusters in Big Spatial Data." *GIScience*: 2–5.

Andrienko, G. and Andrienko, N. (2002). "A General Framework for Using Aggregation in Visual Exploration of Movement Data." *Cartographic Journal* 47: 22–40.

Andrienko, N. and Andrienko, G. (2011). "Spatial Generalisation and Aggregation of Massive Movement Data." *IEEE Transactions on Visualization and Computer Graphics* 17: 205–219.

Anselin, L. (1995). "Local Indicators of Spatial Association–LISA." *Geographical Analysis* 27 (2): 93–115.

Anselin, L., Syabri, I., and Kho, Y. (2006). "GeoDa: An Introduction to Spatial Data Analysis." *Geographical Analysis* 38 (1): 5–22.

Berglund, S., and Karlström, A. (1999). "Identifying Local Spatial Association in Flow Data." *Journal of Geographical Systems* 1 (3): 219–236.

Besag, J., and Newell J. (1991). "The Detection of Clusters in Rare Diseases." *Journal of the Royal Statistical Society Series A* 154 (1): 143–155.

Black, W. R. (1992). "Network Autocorrelation in Transport Network and Flow Systems." *Geographical Analysis* 24 (3): 207–222.

Boots, B., and Okabe, A. (2007). "Local Statistical Spatial Analysis: Inventory and Prospect." *International Journal of Geographical Information Science* 21 (4): 355–375.

Boyandin, I., Bertini, E., and Lalanne, D. (2010). "Using Flow Maps to Explore Migrations over Time." In *Geospatial Visual Analytics Workshop in conjunction with The 13th AGILE International Conference on Geographic Information Science* 2 (3).

Cao, G., Wang, S., Hwang, M., Padmanabhan, A., Zhang, Z., and Soltani, K. (2015). "A Scalable Framework for Spatiotemporal Analysis of Location-Based Social Media Data." *Computers, Environment and Urban Systems* 51: 70–82.

Cavaillès, H. (1940). "Introduction à une géographie de la circulation." *Annales de Géographie* 49 (280): 170–182.

Chen, J., Wang, R., Liu, L., and Song, J. (2011). "Clustering of Trajectories Based on Hausdorff Distance." *2011 International Conference on Electronics, Communications and Control (ICECC)*: 1940–1944.

Chun, Y., and Griffith, D. (2011). "Modeling Network Autocorrelation in Space–Time Migration Flow Data: An Eigenvector Spatial Filtering Approach." *Annals of the Association of American Geographers* 101 (3): 523–536.

Chun, Y., Kim, H., and Kim, C. (2012). "Modeling Interregional Commodity Flows with Incorporating Network Autocorrelation in Spatial Interaction Models: An Application of the US Interstate Commodity Flows." *Computers, Environment and Urban Systems* 36 (6): 583–591.

Chun, Y. (2008). "Modeling Network Autocorrelation within Migration Flows by Eigenvector Spatial Filtering." *Journal of Geographical Systems* 10 (4): 317–344.

Chun, Y. (2014). "Analyzing Space-Time Crime Incidents Using Eigenvector Spatial Filtering: An Application to Vehicle Burglary." *Geographical Analysis* 46 (2): 165–184.

Cressie, N. (1993). Statistics for Spatial Data. New York: Wiley.

Cui, W., Zhou, H., Qu, H., Wong, P. C., and Li, X. (2008). "Geometry-based Edge Clustering for Graph Visualization. *IEEE Transactions on Visualization and Computer Graphics* 14 (6): 1277–1284.

Demšar, U., and Virrantaus, K. (2010). "Space–time Density of Trajectories: Exploring Spatio-temporal Patterns in Movement Data." *International Journal of Geographical Information Science* 24 (10): 1527–1542.

Dao, T. (2013). "A Comprehensive Geospatial Knowledge Discovery Framework for Spatial Association Rule Mining." (Doctoral dissertation). Retrieved from https://librarylink.uncc.edu/login?url=http://search.proquest.com/docview/1495947525?accountid=14605

Diggle, P. (1983). Statistical Analysis of Spatial Point Patterns. London: Academic Press.

Dodge, S., Weibel, R., and Forootan, E. (2009). "Revealing the physics of movement: Comparing the similarity of movement characteristics of different types of moving objects." *Computers, Environment and Urban Systems* 33 (6): 419–434.

Duda, R., Hart, P., and Stork, D. (2001). Pattern Classification, 2nd ed. New York: Wiley.

Duque, J. C., Aldstadt, J., Velasquez, E., Franco, J., and Betancourt, A. (2010). "A Computationally Efficient Method for Delineating Irregularly Shaped Spatial Clusters." *Journal of Geographical Systems*: 1–18.

Duque, J. C., Dev, B., Betancourt, A., and Franco, J. L. (2011). ClusterPy: Library of Spatially Constrained Clustering Algorithms. *Version 0.9.9*.

Fang, Z., Shaw, S., Tu, W., Li, Q., and Li, Y. "Spatiotemporal Analysis of Critical Transportation Links Based on Time Geographic Concepts: A Case Study of Critical Bridges in Wuhan, China." *Journal of Transport Geography*, 23 (2012): 44–59.

Fortin, M., and Dale, M. (2009). "Spatial Autocorrelation." In The SAGE Handbook of Spatial Analysis edited by Fotheringham, S., and Rogerson, P. London: Sage: 89–103.

Fotheringham, S. (1997). "Trends in Quantitative Methods I: Stressing the Local." *Progress in Human Geography* 21 (1): 88–96.

Fotheringham, S., and Zhan B. (1996). "A Comparison of Three Exploratory Methods for Cluster Detection in Spatial Point Patterns." *Geographical Analysis* 28 (3): 200–218.

Geary, R. (1954). "The Contiguity Ratio and Statistical Mapping." *The Incorporated Statistician* 5 (3): 115–145.

Genolini, C., and B. Falissard. (2010). "KmL: K-Means for Longitudinal Data." *Computational Statistics* 25 (2): 317–328.

Getis, A., and Aldstadt, J. (2004). "Constructing the Spatial Weights Matrix Using a Local Statistic." *Geographical Analysis*, 36 (2): 90–104.

Getis, A., and Franklin, J. (1987). “Second-Order Neighborhood Analysis of Mapped Point Patterns.” *Ecology* 68: 473–477.

Getis, A., Ord, J. (1992). “The Analysis of Spatial Association by Use of Distance Statistics.” *Geographical Analysis* 24 (3): 189–206.

Glennon, J. A. (2005). Flow data model tool for ArcGIS 9.0. Department of Geography, University of California, Santa Barbara. Flow tool available at <http:// www.alanglennon.com/flowtools/>.

Griffith, D. (2007). “Spatial Structure and Spatial Interaction : 25 Years Later.” *The Review of Regional Studies* 37 (1): 28–38.

Griffith, D. (2009). “Modeling Spatial Autocorrelation in Spatial Interaction Data: Empirical Evidence from 2002 Germany Journey-to-Work Flows.” *Journal of Geographical Systems* 11 (2): 117–140.

Guo, D. (2009). “Flow Mapping and Multivariate Visualization of Large Spatial Interaction Data.” *IEEE Transactions On Visualization and Computer Graphics*, 15 (6): 1041–1048.

Guo, D., Chen, J., MacEachren, A. M., and Liao, K. (2006). “A Visualization System for Space–Time and Multivariate Patterns (Vis-Stamp).” *IEEE Transactions on Visualization* 12 (6):1461–1474.

Guo, D., and Wang, H. (2011). “Automatic Region Building for Spatial Analysis.” *Transactions in GIS* 15 (s1): 29–45.

Guo, D., Zhu, X., Jin, H., Gao, P., and Andris, C. (2012). “Discovering Spatial Patterns in Origin-Destination Mobility Data.” *Transactions in GIS*, 16(3): 411–429.

Guo, D. (2009). “Flow Mapping and Multivariate Visualization of Large Spatial Interaction Data.” *IEEE Transactions on Visualization and Computer Graphics* 15 (6): 1041–1048.

Haynes, K. E., and Fotheringham, A. S. (1984). Gravity and Spatial Interaction Models. (Vol. 2). Beverly Hills: Sage publications.

Holten, D., and Van Wijk, J. J. (2009). “Force-Directed Edge Bundling for Graph Visualization.” *Computer Graphics Forum* 28: 983–990.

Kulldorff, M. (1997). “A Spatial Scan Statistic.” Comm*unications in Statistics - Theory and Methods* 26 (6): 1481–1496.

Lee, J. G., Han, J., and Whang, K. Y. (2007). “Trajectory Clustering: A Partition-and-Group Framework.” In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*: 593–604.

LeSage, J. P., and Pace, R. K., (2008). "Spatial Econometric Modeling of Origin-Destination Flows." *Journal of Regional Science* 48 (5): 941–967.

LeSage, J. P., and Polasek, W. (2008). “Incorporating Transportation Network Structure in Spatial Econometric Models of Commodity Flows.” *Spatial Economic Analysis* 3 (2): 225–245.

Liu, L., Andris, C., and Ratti, C. (2010). “Uncovering Cabdrivers’ Behavior Patterns from Their Digital Traces.” *Computers, Environment and Urban Systems*, 34 (6): 541–548.

Liu, Y., Tong, D., and Liu, X. (2015). “Measuring Spatial Autocorrelation of Vectors.” *Geographical Analysis*, 47 (3): 300–319.

Lu, Y., and Thill, J.-C. (2003). “Assessing the Cluster Correspondence between Paired Point Locations.” *Geographical Analysis* 35 (4): 290–309.

Lu, Y., and Thill, J.-C. (2008). “Cross-scale Analysis of Cluster Correspondence Using Different Operational Neighborhoods.” *Journal of Geographical Systems* 10 (3): 241–261.

Morstatter, F., Pfeffer, J., Liu, H., and Carley, K. M. (2013). “Is the sample good enough? Comparing data from Twitters streaming API with Twitters Firehose.” In *Proceedings of ICWSM*.

Moran, P. (1950). “Notes on Continuous Stochastic Phenomena.” *Biometrika* 37 (1): 17–23.

Murray, A., Liu, Y., Rey, S. J., and Anselin, L. (2011). “Exploring Movement Object Patterns.” *The Annals of Regional Science* 49 (2): 471–484.

Nanni, M., and Pedreschi, D. (2006). “Time-Focused Clustering of Trajectories of Moving Objects.” *Journal of Intelligent Information Systems* 27 (3): 267–289.

Okabe, A., Boots, B., and Satoh, T. (2010). "A Class of Local and Global K-functions and Their Exact Statistical Methods." *Perspectives on Spatial Data Analysis*: 101–112.

Openshaw, S., Charlton, M., Wymer, C., and Craft, A. (1987). “A Mark 1 Geographical Analysis Machine for the Automated Analysis of Point Data Sets.” *International Journal of Geographical Information Systems* 1 (4): 335–358.

Ord, J., and Getis, A. (1995). “Local Spatial Autocorrelation Statistics: Distributional Issues and an Application.” *Geographical Analysis* 27 (4): 286–306.

Ossama, O., Mokhtar, H., and El-Sharkawi, M. (2011). “Clustering Moving Objects Using Segments Slopes.” *International Journal of Database Management Systems* 3 (1): 35–48.

Phan, D., Xiao, L., Yeh, R., and Hanrahan, P. (2005). "Flow Map Layout," *IEEE Symposium on Information Visualization*: 219–224.

Qu, H., Zhou, H., and Wu, Y. (2006). “Controllable and Progressive Edge Clustering for Large Networks.” In *Proceed. of Symposium on Graph Drawing*: 399–404.

Rae, A. (2009). “From Spatial Interaction Data to Spatial Interaction Information? Geovisualisation and Spatial Structures of Migration from the 2001 UK Census.” *Computers, Environment and Urban Systems* 33 (3): 161–178.

Rinzivillo, S., Pedreschi, D., Nanni, M., Giannotti, F., Andrienko, N., and Andrienko, G. (2008). “Visually Driven Analysis of Movement Data by Progressive Clustering.” *Information Visualization* 7: 225–239.

Ripley, B. D. (1976). “The Second-Order Analysis of Stationary Point Processes.” *Journal of Applied Probability* 13: 255–266.

Sinha, G., and Mark, D. (2005). “Measuring Similarity between Geospatial Lifelines in Studies of Environmental Health.” *Journal of Geographical Systems* 7 (1): 115–136.

Symanzik, J. (2014). “Exploratory Spatial Data Analysis.” In *Handbook of Regional Science* edited by Manfred F. and Peter N. Heidelberg, Germany: Springer: 1295–1310.

Tang, W., Feng, W., and Jia, M. (2015). "Massively Parallel Spatial Point Pattern Analysis: Ripley's K Function Accelerated Using Graphics Processing Units." *International Journal of Geographical Information Science* 29 (3): 412–439.

Tao, R., Thill, J.-C., and Yamada, I. (2015). "Detecting Clustering Scales with the Incremental K-Function: Comparison Tests on Actual and Simulated Geospatial Datasets." In *Information Fusion and Geographic Information Systems* (IF&GIS'2015): 93–107.

Tobler, W. R. (1987). "Experiments in Migration Mapping by Computer." *The American Cartographer* 14: 155–163.

Tobler, W. R. (2004). Movement Mapping. http://csiss.ncgia.ucsb.edu/clearinghouse/FlowMapper.

Ullman E., and Mayer, H. (1954). "Transportation Geography" in *American Geography*: *Inventory and Prospect* edited by Preston E. James and Clarence F. Jones (Syracuse): 310–332.

Ullman E., (1980). "Geography as Spatial Interaction" in *Geography as Spatial Interaction* by Edward Ullman edited by Ronald R Boyce (University of Washington Press): 13–27.

Ullman E., (1956). "The Role of Transportation and the Bases for Interaction" in Man's Role in *Changing the Face of the Earth* edited by William L. Thomas Jr. (University of Chicago Press): 862–880.

Verbeek K., Buchin K., and Speckmann B. (2011). "Flow Map Layout Via Spiral Trees." *IEEE Transactions on Visualization and Computer Graphics* 17: 2536–2544.

Waller, L. (2009). “Detection of Clustering in Spatial Data.” In *The SAGE Handbook of Spatial Analysis* edited by Fotheringham, S., and Rogerson, P. London: Sage: 159–181.

Wang, S. (2010). “A CyberGIS Framework for the Synthesis of Cyberinfrastructure, GIS, and Spatial Analysis.” *Annals of the Association of American Geographers* 100 (3): 535–557.

Widener, M. J., Crago, N. C., and Aldstadt, J. (2012). “Developing a parallel computational implementation of AMOEBA.” *International Journal of Geographical Information Science* 26 (9): 1707–1723.

Wu, W., Wang, J., and Dai, T. (2016). “The Geography of Cultural Ties and Human Mobility: Big Data in Urban Contexts.” *Annals of the American Association of Geographers*: 1–19.

Yamada, I., and Thill, J.-C. (2007) “Local Indicators of Network-Constrained Clusters in Spatial Point Patterns.” *Geographical Analysis* 39 (3): 268–292.

Yamada, I., and Thill, J.-C. (2010). “Local Indicators of Network-Constrained Clusters in Spatial Patterns Represented by a Link Attribute.” *Annals of the Association of American Geographers* 100 (2): 269–285.

Yan, J., and Thill, J.-C. (2009). "Visual Data Mining in Spatial Interaction Analysis with Self-Organizing Maps." *Environment and Planning B: Planning and Design* 36 (3): 466–486

Yue, Y., Wang, H. D., Hu, B., and Li, Q. Q. (2011). “Identifying Shopping Center Attractiveness Using Taxi Trajectory Data.” *Proceedings of the 2011 International Workshop on Trajectory Data Mining and Analysis - TDMA*: 11–31.

Zhu, X., and Guo. D. (2014). “Mapping Large Spatial Flow Data with Hierarchical Clustering.” *Transactions in GIS* 18 (3): 421–435.