

## Chapter 2

# Many Connected Components

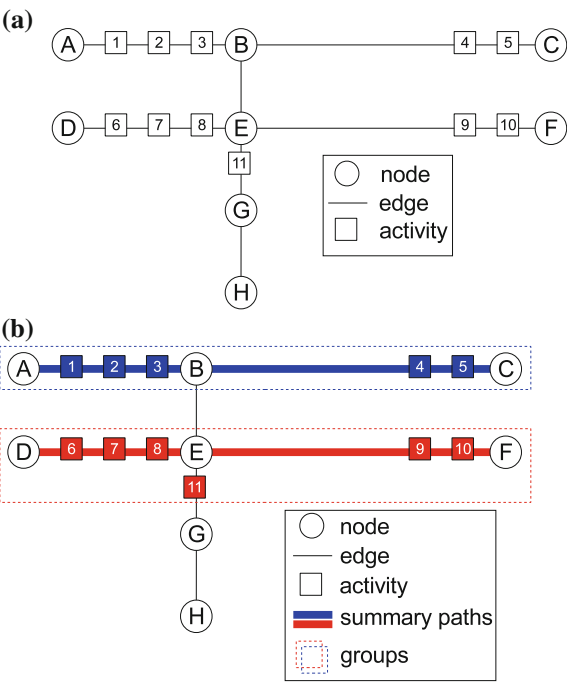
**Abstract** This chapter presents different ways of handling the first challenge of summarizing spatial network data, i.e., the large number of  $k$ -subsets of connected components in the network. This challenge is conceptualized as the spatial network activity summarization problem (SNAS) where given a spatial network, a collection of activities and their locations (e.g., placed on a node or an edge), and a desired number of paths  $k$ , SNAS finds a set of  $k$  shortest paths that maximizes the sum of activities on the paths (counting activities that are on overlapping paths only once) and a partitioning of activities across the paths.

### 2.1 Introduction

Handling the large number of connected components in a spatial network may be conceptualized as the spatial network activity summarization problem (SNAS) [1]. This problem may be formalized in different ways depending on the domain or application. For the domain of summarizing pedestrian fatalities or crime analysis, one may define the problem as follows: Given a spatial network, a collection of activities and their locations (e.g., placed on a node or an edge), and a desired number of paths  $k$ , find a set of  $k$  shortest paths that maximizes the sum of activities on the paths (counting activities that are on overlapping paths only once) and a partitioning of activities across the paths. An activity may be the location of a carjacking, a pedestrian fatality, a train accident, or any other spatial network observation. In many domains or applications such as transportation planning, the aim is usually to help people get to their destination as fast as possible; SNAS, therefore, assumes that every path is a shortest path.

An example of the spatial network activity summarization (SNAS) problem is illustrated in Fig. 2.1. The input (Fig. 2.1a) consists of eight nodes, seven edges (with edge weights of 1 for simplicity), eleven activities, and  $k = 2$ , indicating that two routes and groups are desired. The shortest paths and the activity coverage (i.e., the sum of activities on each shortest path) for this network is shown in Table 2.1. The output (Fig. 2.1b) contains two shortest paths and two groups of activities. The shortest paths are representatives for each group and each shortest path maximizes

**Fig. 2.1** Example **a** Input and **b** Output of Spatial Network Activity Summarization (Best in color)



**Table 2.1** Shortest paths from Fig. 2.1 (Activity Coverage refers to the number of activities on a path)

Source	Sink	Shortest path	Activity coverage	Source	Sink	Shortest path	Activity coverage
A	B	$\langle A, B \rangle$	3	C	E	$\langle C, B, E \rangle$	2
A	C	$\langle A, B, C \rangle$	5	C	F	$\langle C, B, E, F \rangle$	4
A	D	$\langle A, B, E, D \rangle$	6	C	G	$\langle C, B, E, G \rangle$	3
A	E	$\langle A, B, E \rangle$	3	C	H	$\langle C, B, E, G, H \rangle$	3
A	F	$\langle A, B, E, F \rangle$	5	D	E	$\langle D, E \rangle$	3
A	G	$\langle A, B, E, G \rangle$	4	D	F	$\langle D, E, F \rangle$	5
A	H	$\langle A, B, E, G, H \rangle$	4	D	G	$\langle D, E, G \rangle$	4
B	C	$\langle B, C \rangle$	2	D	H	$\langle D, E, G, H \rangle$	4
B	D	$\langle B, E, D \rangle$	3	E	F	$\langle E, F \rangle$	2
B	E	$\langle B, E \rangle$	0	E	G	$\langle E, G \rangle$	1
B	F	$\langle B, E, F \rangle$	2	E	H	$\langle E, G, H \rangle$	1
B	G	$\langle B, E, G \rangle$	1	F	G	$\langle F, E, G \rangle$	3
B	H	$\langle B, E, G, H \rangle$	1	F	H	$\langle F, E, G, H \rangle$	3
C	D	$\langle C, B, E, D \rangle$	5	G	H	$\langle G, H \rangle$	0

the activity coverage for the group it represents. For example, route  $\langle A, B, C \rangle$  is the representative for the group comprised of activities 1, 2, 3, 4, and 5, and route  $\langle D, E, F \rangle$  is the representative for the group comprised of activities 6, 7, 8, 9, 10, and 11.

SNAS may be applied in a variety of domains where observations happen along linear paths in the network. For example, crime analysts look for linear concentrations of crime (linear generators and attractors) such as speeding, street drug dealing, drunk driving, etc. to guide law enforcement [2]. Environmental engineers may try to summarize environmental change on water resources to understand the behavior of river networks and lakes [3]. Transportation officials are interested in understanding railroad accidents (e.g., derailling) to improve safety and reduce cost [4]. Emergency managers may find it useful to summarize the locations of stranded cars on highways to better understand how to allocate resources [5].

However, SNAS is computationally challenging due to the fact that if  $k$  shortest paths are selected from all shortest paths in a spatial network, there are a large number of possibilities for large  $k$ , i.e.,  $\binom{n}{k}$ , where  $n$  is the number of shortest paths. The reason for this is that different subsets of  $k$  shortest paths could be overlapping or have the same shortest paths. If paths are disjoint, the computational challenge goes away but with overlapping paths, the general problem of SNAS is NP-complete (the proof is provided in Sect. 2.3).

### 2.1.1 *An Illustrative Application Domain: Crime Analysis*

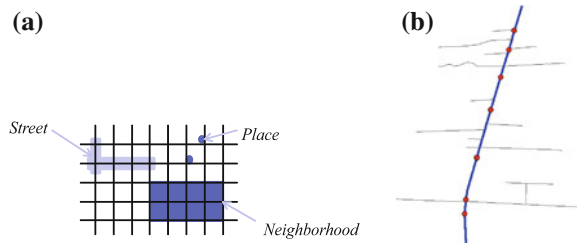
As part of their analysis, crime analysts look for linear concentrations of crime such as speeding or drunk driving to help guide law enforcement decisions (e.g., deciding which streets to patrol). A key theoretical underpinning to crime analysis is Environmental Criminology, which is the study of crime, criminality, and victimization as they relate to particular places and to the way that individuals and organizations shape their activities spatially [6]. Environmental criminology is used by law enforcement to develop an understanding of the spatial distributions (e.g., high activity places or hot-spots) of crime activities as well as location-based factors affecting activities using analytical techniques, e.g., crime mapping, and analytical tools such as Crime-Stat [7]. Police departments also leverage knowledge of environmental criminology to facilitate the design of patrol routes and provide street-based descriptions of crime attractors and generators (e.g., a bar after closing time) [8]. Locations in crime reports reference symbolic systems (e.g., street addresses, highway-mile markers) as well as numerical systems (e.g., latitude/longitude), which refer to a point. Real crime report datasets collected by police departments in formats advocated by the US Department of Justice are illustrated in Table 2.2.

Spatial theories in Environmental Criminology include Routine Activity Theory (RAT) [9] and Crime Pattern Theory (CPT) [10]. RAT postulates that the location of a crime is related to the frequently visited areas of a criminal and CPT extends this theory on a spatial model. Crime is not spread evenly across maps but instead is

**Table 2.2** Sample crime report data in formats advocated by the US Department of Justice

ID	Offence type	Date	Time	Address
96	Burglary	1/14/2006	1530	16950 GRAND AVE
477	Auto Theft	8/2/2006	2042	7950 W FAIRVIEW 1960
633	Narcotic Drug Laws	11/2/2006	1200	12950 CLEVELAND DR

**Fig. 2.2** **a** Types of crime hot spots. **b** An example of a street-based (linear) hotspot of crime in a major US city



concentrated in some areas and absent in others [2]. This knowledge is used everyday by people and is seen in the way they avoid some places and seek out others. Police also use this understanding to make decisions about how to allocate scarce resources based in part on where police demand is highest. Officers are told to be attentive in certain areas but are not given guidance in other areas where crime is scarce [2].

Crime hotspots are not only studied in terms of places and neighborhoods, but also streets [2]. For places, an explanation as to why crime events occur at specific locations is sought. For neighborhoods, analysts look at large areas and ask questions such as “Which areas are claimed by gangs and which areas are not?”. Street-based analysis deals with crimes that occur over small stretched areas such as streets or blocks and examples include street drug dealing, prostitution, and robberies of pedestrians [2]. Different types of crime hotspots are illustrated in Fig. 2.2.

The U.S. Department of Justice’s research, development, and evaluation agency (i.e., the National Institute of Justice) points out that “commonly available mapping programs make it easy to identify hot spot places or hot spot areas, but do not make linear hotspots easy to identify. Most clustering algorithms, unfortunately, will show areas of concentration even when a line is the most appropriate dimension” [2].

### 2.1.2 State of the Art

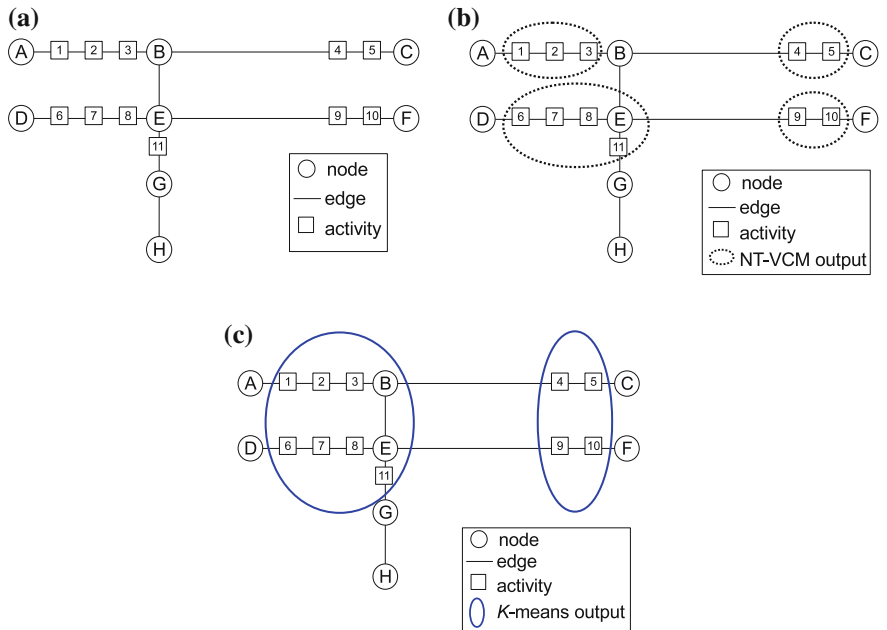
Activity summarization is a significant area in data mining and spatial computing. Most techniques can be classified as being either geometry-based [11–15] or network-based [16–25].

Geometry-based summarization techniques involve grouping similar points distributed in planar space where distance is calculated using Euclidean distance, not network distance. Such techniques focus on the discovery of the geometry (e.g.,

circle, ellipse) of high density regions [2] and include K-Means [11], K-medoid [12, 13], P-median [14] and Nearest Neighbor Hierarchical Clustering [15]. These methods are useful for providing geometric summaries such as ellipses or circles but they may fail to group activities that occur on the same street.

Network-based summarization, on the other hand, involves grouping spatial objects using network (e.g., road) distance. Examples of network based summarization techniques include Mean Streets [16], Maximal Subgraph Finding (MSGF) [17], and Clumping [18–25]. These techniques may group activities over multiple paths, a single path/subgraph, or no paths at all. For example, Mean Streets [16] finds anomalous streets or routes with unusually high activity levels. It is not designed to summarize activities over  $k$  paths because the number of high crime streets returned is always relatively small. MSGF [17] identifies the maximal subgraph (e.g., a single path,  $k = 1$ ) under the constraint of a user specified length and cannot summarize activities when  $k > 1$ . The Network-Based Variable-Distance Clumping Method (NT-VCM) [25] is an example of the clumping technique [18–25]. NT-VCM groups activities that are within a certain shortest path distance of each other on the network; therefore, a distance threshold must be specified.

Figure 2.3b illustrates an example output of NT-VCM. The threshold distance for NT-VCM is the unit distance between activities 8 and 11. NT-VCM is useful for summarizing activities on the network but the onus is placed on the user to specify a



**Fig. 2.3** Two methods of summarizing activities on a network. **a** Input. **b** NT-VCM Output [25]. **c** K-Means Output (network distance)

proper distance threshold. In this example, activities 1, 2, 3, 4, and 5, which occur on the same street would not belong to the same group, given this threshold distance.

Certain geometry-based summarization techniques may also be retrofitted to account for network distances. Figure 2.3c shows an example of the ellipses output by K-Means using network distance. The left ellipse groups activities 1, 2, 3, 6, 7, 8, and 11 whereas the right ellipse groups activities 4, 5, 9, and 10. By leveraging network distance, these techniques can be tailored to additional applications. However, even when generalized with network distances, these methods output point-based or ellipsoid-based groups, not paths or other network-based representatives such as nodes or subgraphs.

### 2.1.3 Outline of the Chapter

The rest of this chapter is organized as follows: Sect. 2.2 presents the basic concepts and problem statement of SNAS. The computational structure and a proof of why SNAS is NP-Complete is explained in Sect. 2.3. A recent approach to solving this problem, i.e., the K-Main Routes algorithm, is also presented. Section 2.4 presents a case study comparing geometry and network-based summarization techniques. A discussion is presented in Sects. 3.5 and 3.6 summarizes the chapter.

## 2.2 Basic Concepts and Problem Statement

This section defines the basic concepts and the spatial network activity summarization problem [1].

### 2.2.1 Basic Concepts

The basic concepts are defined as follows:

**Definition 1** A **spatial network**  $G = (N, E)$  consists of a node set  $N$  and an edge set  $E$ , where each element  $u$  in  $N$  is associated with a pair of real numbers  $(x, y)$  representing the spatial location of the node in a Euclidean plane [26]. Edge set  $E$  is a subset of the cross product  $N \times N$ . Each element  $e = (u, v)$  in  $E$  is an edge that joins node  $u$  to node  $v$ .

An example of a spatial network is shown in Fig. 2.1a. In the figure, circles represent nodes and lines represent edges. A road network is an example of a spatial network where nodes represent street intersections and edges represent streets. In utility networks such as electric networks, nodes represent devices such as transformers and fuses whereas edges represent lines (e.g., medium voltage lines) [27].

**Definition 2** An **activity set**  $A$  is a collection of activities. An **activity**  $a \in A$  is an object of interest associated with only one edge  $e \in E$  or one node  $n \in N$ .

In Fig. 2.1a, activities are represented as squares. In transportation planning, an activity may be the location of a pedestrian fatality; in crime analysis, an activity may be the location of a theft; and in disaster response an activity may be the location of a request for relief supplies.

**Definition 3** A **summary path set**  $\hat{P}$  is a collection of summary paths where each path  $p_i \in \hat{P}$  is a shortest path. A **summary path** imposes a partitioning on an activity set  $A$  such that  $network\ distance(a, p_i) \leq network\ distance(a, p_j) \forall p_j \in \hat{P}, \forall a \in A$ .

Figure 2.1b shows two summary paths  $\langle A, B, C \rangle$  and  $\langle D, E, F \rangle$ . Activities 1, 2, 3, 4, and 5 form a partition around  $\langle A, B, C \rangle$  because they are closer to  $\langle A, B, C \rangle$  whereas activities 6, 7, 8, 9, 10 and 11 form a partition around  $\langle D, E, F \rangle$  because they are closer to  $\langle D, E, F \rangle$ . Here the network distance between an activity and a path, i.e.,  $network\ distance(a, p_i)$ , is the network distance between  $a$  and the closest node in  $p_i$ .

**Definition 4** The **activity coverage**  $AC(p)$  of a path  $p$  is the sum of activities having network distance = 0 from an edge  $e \in p$ . The **activity coverage**  $AC(P)$  of a set of paths  $P$  is the sum of activities across individual paths  $p_i$  in set  $P$ , having network distance = 0 from each edge  $e \in P$ , counting activities that are covered several times only once. If two paths share an edge, the activities on that edge are only counted once.

An example may be seen in in Fig. 2.1a. Here the activity coverage of the shortest path from node  $A$  to node  $B$  is 3 because there are 3 activities occurring on that path. The activity coverage of the shortest path from node  $D$  to node  $F$  is 5 because there are 5 activities occurring on  $\langle D, E, F \rangle$ . If the set of paths are  $\langle A, B, C \rangle$  and  $\langle D, E, F \rangle$ , then the activity coverage is 10 because there are 10 activities on all the edges of the paths in  $P$ . If the set of paths are  $\langle A, B, C \rangle$  and  $\langle A, B \rangle$ , then the activity coverage is 5. The activities on edge  $AB$  are only counted once even though  $AB$  is an edge in both paths.

### 2.2.2 Problem Statement

The problem of spatial network activity summarization (SNAS) can be expressed as follows:

**Given:**

1. A spatial network  $G = (N, E)$  with weight function  $w(u, v) \geq 0$  for each edge  $e = (u, v) \in E$  (e.g., network distance),

2. A set of activities  $A$  and their locations (e.g., a node or an edge),
3. A desired number of summary paths,  $k$ , where  $k \geq 1$ .

**Find:**

1. A summary path set of size  $k$ ,
2. A partitioning of activities across these summary paths.

**Objective:** Maximize the activity coverage of each summary path for the group it represents.

**Constraints:**

1. Each summary path is a shortest path between its end-nodes,
2. Each activity  $a \in A$  is associated with only one edge  $e \in E$ .

In order to understand the problem statement, consider the inputs and the outputs to the problem. For the inputs, Definition 1 defines the spatial network, the activities represent objects of interest such as the locations of crimes, and  $k$  represents the desired number of summary paths. The outputs for SNAS are a summary path set of size  $k$  and a partitioning of activities across the paths. The summary paths are representatives for each group and each summary path maximizes the activity coverage for the group it represents.

**Example.** Figure 2.1a shows an example of the input, i.e., a spatial network such as a road network, composed of streets (edges) and intersections (nodes) with eleven activities (squares) and a specified value of  $k = 2$ . The goal is find two groups of activities and two routes, each route being the representative for each group. In a crime analysis scenario, identifying such routes would guide patrol efforts to mitigate crime on certain streets whereas in a transportation planning scenario, identifying such routes would guide street redesign efforts to reduce the risk of pedestrian fatalities (e.g., adding sidewalks, crosswalks, pedestrian refuges, street lighting, etc.). Figure 2.1b shows an example of the output. In this case, route  $\langle A, B, C \rangle$  is the representative for the group comprised of activities 1, 2, 3, 4, and 5; and route  $\langle D, E, F \rangle$  is the representative for the group comprised of activities 6, 7, 8, 9, 10 and 11.

When handling the challenge of many connected components, the domain often drives the way the problem is formulated. For the domains of crime analysis and transportation planning, the problem statement in its current form may be suitable to address the needs of law enforcement professionals and transportation planners. This is because maximizing activity coverage works well in these domains (e.g., finding the representative routes that cover most crimes in a group). However, for the domain of disaster response, maximizing activity coverage may not be the only desirable objective. An alternative objective in this case might be to minimize the distance that victims have to walk in order to receive assistance in the form of water, food, or medicine. The new objective may have an impact on the computational structure of the problem and different techniques may have to be employed to bring about a computationally efficient and correct solution. In the next section, the computational structure of SNAS is outlined.



## 2.3 Spatial Network Activity Summarization

This section describes the computational structure of SNAS. It also describes a new trend in this area, i.e., the K-Main Routes (KMR) algorithm and its performance-tuning decisions Network Voronoi activity Assignment, Divide and conquer Summary Path REcomputation, and Inactive Node Pruning.

### 2.3.1 Computational Structure of Spatial Network Activity Summarization

In SNAS, the optimal solution may not be unique. Additionally, among the optimal solutions there are some where every path starts and ends at active nodes. These properties are formally shown via Lemmas 2.1 and 2.2.

**Definition 5** An **active edge** is an edge  $e \in E$  that has 1 or more activities. An **active node** is a node  $u$  joined by an active edge or a node that has one or more activities, or both. An **inactive node** is a node that is not joined by any active edges.

Edges  $AB$  and  $BC$  in Fig. 2.1a are active edges because they each have at least one activity and nodes  $A, B, C, D, E, F$ , and  $G$  are all active nodes because they are all joined by active edges. By contrast, Node  $H$  is an inactive node because it is not joined by any active edges.

**Lemma 2.1** *The optimal solution for SNAS may not be unique.*

*Proof* There may be multiple solutions for different values of  $k$ . For example, given  $k = 1$  in Fig. 2.1a where all eleven activities are members of the same group, the summary path could be  $\langle A, B, E, D \rangle$  or  $\langle D, E, B, A \rangle$ , since both these paths have a maximum activity coverage of 6 based on the one group. Given  $k = 2$  and the groups shown in Fig. 2.1b, the summary paths could be  $\langle A, B, C \rangle$  and  $\langle D, E, F \rangle$  or  $\langle C, B, A \rangle$  and  $\langle F, E, D \rangle$  as both sets of paths have a maximum activity coverage of 11 based on their respective groups.

**Lemma 2.2** *Among the optimal solutions for SNAS, there exists optimal solutions where every path starts and ends at active nodes.*

*Proof* Let's begin with an arbitrary optimal solution. Let  $p$  be a shortest path that starts or ends with inactive nodes. If inactive nodes that start or end  $p$  are removed such that  $p$  starts and ends with active nodes, the resulting subpath  $p'$  is still optimal in terms of activity coverage, because no active edges were removed.  $p'$  is also still a shortest path due to the optimal substructure of shortest paths wherein subpaths of shortest paths are shortest paths [28]. In other words, eliminating inactive nodes from the beginning and end of a shortest path does not reduce coverage and does not split the path. KMR takes advantage of this property to achieve computational savings.

### 2.3.2 Proof of NP-Completeness

For simplicity, a generalized decision version of SNAS where the set of paths might be arbitrary is first defined and then shown to be NP-complete. A proof sketch showing that the decision version of SNAS with shortest paths is also NP-complete is then presented.

**Definition 6** Decision version of SNAS:

**INSTANCE:** A spatial network  $G = (N, E)$  with weight function  $w(u, v) \geq 0$  for each edge  $e = (u, v) \in E$ , a set of activities  $A$  and their locations, a set of paths  $P$ , a desired number of routes  $k$ , and a bound  $B \in \mathbb{Z}^+$ , where  $\mathbb{Z}^+$  denotes the set of positive integers.

**QUESTION:** Does  $P$  contain a cardinality  $k$  subset  $P'$  of  $P$ , i.e., a subset  $P' \subseteq P$  with  $|P'| = k$  and  $AC(P') \geq B$ , where  $AC(P')$  denotes the activity coverage of  $P'$ ?

**Theorem 2.1** SNAS is NP-complete.

*Proof* The process of devising an NP-completeness proof for a decision problem  $\Pi$  consists of the following four steps [29]:

1. showing that  $\Pi$  is in NP,
2. selecting a known NP-complete problem  $\Pi'$ ,
3. constructing a transformation  $f$  from  $\Pi$  to  $\Pi'$ , and
4. proving that  $f$  is a (polynomial) transformation

In step 1, to show that SNAS  $\in$  NP, assume that a certificate and a number  $B$  are given. The certificate consists of a spatial network  $G = (N, E)$  with weight function  $w(u, v) \geq 0$  for each edge  $e_i = (u, v) \in E$ , a set of activities  $A$ , a set of paths  $P$ , a desired number of routes  $k$ , and a cardinality  $k$  subset  $P'$  of  $P$ , i.e., a subset  $P' \subseteq P$  with  $|P'| = k$ . We can then verify in polynomial time whether  $AC(P') \geq B$  because  $AC(P')$  involves counting the number of activities in  $P'$ .

Step 2 selects the Maximum Coverage problem [30] as a known NP-complete problem  $\Pi'$ . Although the optimization version of Maximum Coverage [30] is known to be NP-Hard, its decision version is NP-Complete. The decision version is specified as follows:

**INSTANCE:** A number  $k$  and a collection of sets  $S = S_1, S_2, \dots, S_m$ , where  $S_i \subseteq \{l_1, l_2, \dots, l_n\}$ , and a bound  $B \in \mathbb{Z}^+$ , where  $\mathbb{Z}^+$  denotes the positive integers.

**QUESTION:** Does  $S$  contain a subset  $S' \subseteq S$  of sets such that  $|S'| \leq k$  and the number of covered elements  $\left| \bigcup_{S_i \in S'} S_i \right| \geq B$ ?

Steps 3 and 4 construct a transformation  $f$  from  $\Pi$  to  $\Pi'$  and prove that  $f$  is a (polynomial) transformation. The reduction entails a polynomial time transformation of the input of Maximum Coverage to the input of SNAS followed by a polynomial time transformation of the output of SNAS to the output of Maximum Coverage. The input of Maximum Coverage may be transformed to the input of SNAS using the following steps:

1. Impose a total order  $TO$  on  $n$  elements in  $L = \{l_1, l_2, \dots, l_n\}$ .
2. Convert each element in  $L$  into a node with one activity.
3. Convert each set  $S_i$  to a path  $P_i$ .

- Add edge  $(l_j, l_{j+1}) \forall j \in 1 \dots |S_i|$ .

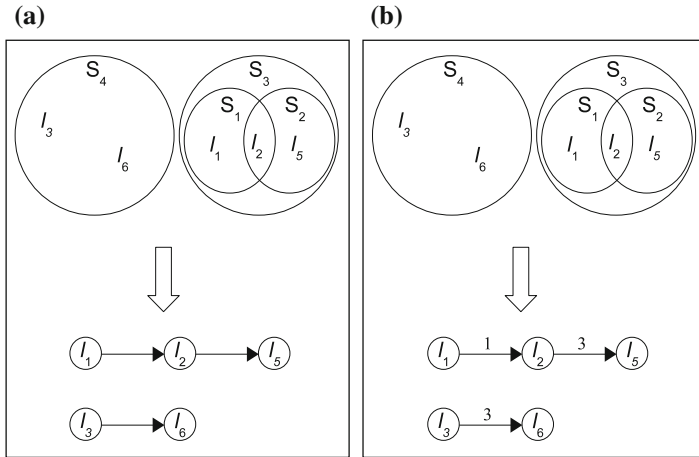
The transformation computation time is dominated by the polynomial step of sorting the elements in  $S_i$  using  $TO$ . Thus it is easy to see that the entire transformation is indeed polynomial. Consider an instance of the Maximum Coverage problem as shown in Fig. 2.4a where  $L = \{l_1, l_2, l_3, l_5, l_6\}$ ,  $k = 2$ ,  $S_1 = \{l_1, l_2\}$ ,  $S_2 = \{l_2, l_3\}$ ,  $S_3 = \{l_1, l_2, l_3\}$ , and  $S_4 = \{l_5, l_6\}$ . The resulting instance of SNAS would thus be  $P = \{(l_1 \rightarrow l_2), (l_2 \rightarrow l_3), (l_1 \rightarrow l_2 \rightarrow l_3), (l_5 \rightarrow l_6)\}$ ,  $k = 2$ ,  $A = \{a_1, a_2, a_3, a_5, a_6\}$ , and  $ActivityNode = \{a_1(l_1), a_2(l_2), a_3(l_3), a_5(l_5), a_6(l_6)\}$ .

Next, we convert the instance of SNAS output to an instance of Maximum Coverage output. The transformation, which is also polynomial, is as follows:

- For each  $k$  path  $P_i$  produced by SNAS, convert the activities on the path into elements and form a set  $S_i$ .

The candidate solutions for the instance of SNAS shown in Fig. 2.4a are  $(l_1 \rightarrow l_2 \rightarrow l_3)$  and  $(l_5 \rightarrow l_6)$ . The resulting instance of the Maximum Coverage output would be  $S_1 = \{l_1, l_2, l_3\}$  and  $S_2 = \{l_5, l_6\}$ .

The reduction of Maximum Coverage to SNAS is a polynomial time reduction since the input of Maximum Coverage can be reduced to SNAS in polynomial time, and the output of SNAS can be reduced to Maximum Coverage in polynomial time.



**Fig. 2.4** SNAS instance resulting from Maximum Coverage instance for **a** arbitrary paths and **b** shortest paths

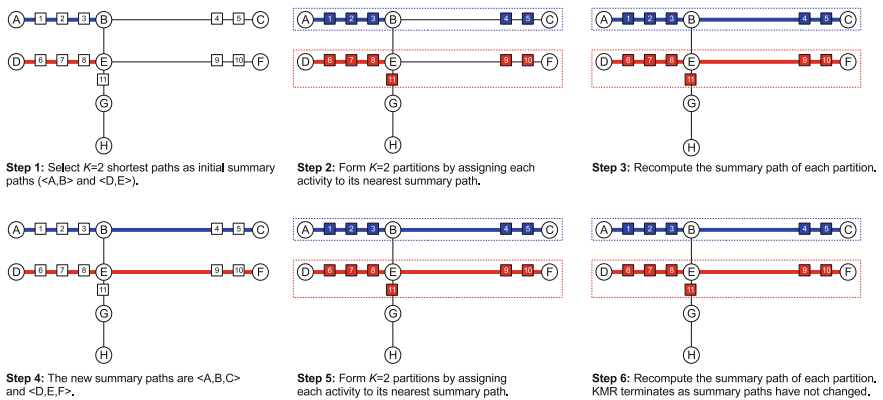
Since SNAS belongs to the class of NP and a known NP-complete problem is reduced to it, the decision version of SNAS is NP-complete.

We now present a proof sketch that the decision version of SNAS where the set of paths are shortest paths is also NP-complete. We follow the same construction as before but assign a cost for the edge  $(l_i, l_j)$  to be  $j - i$  (Fig. 2.4b). This step ensures that all paths generated by construction are shortest paths.

### 2.3.3 Trend: The K-Main Routes Algorithm

The pseudocode for the K-Main Routes (KMR) algorithm that addresses SNAS is shown in Algorithm 1. KMR’s basic structure resembles that of K-Means [11] in terms of selecting initial seeds, forming  $k$  groups, and updating group representatives until the assignments no longer change. In Line 1 of the algorithm, all shortest paths  $P$  that start and end with active nodes (inactive node pruning) are selected. In Line 2,  $k$  paths from  $P$  are selected as initial summary paths, which are the “seeds” for the algorithm. KMR then proceeds in two main phases. In the first phase,  $k$  groups are formed by assigning each activity to its closest summary path (Line 4). In the second phase, the summary path of each group is updated by calculating the shortest path that maximizes activity coverage (Line 5). These phases of assigning and updating repeat until the summary paths no longer change and the final summary paths and groups are returned (Line 8).

**KMR Example:** An example execution of KMR from first principles is shown in Fig. 2.5. In this example, the spatial network shown has eight nodes, seven edges, and eleven activities. For illustration purposes,  $\langle A, B \rangle$  and  $\langle D, E \rangle$  are selected as initial summary paths (though any set of paths may be used as the initial seeds),  $k = 2$ , and all edge weights are set to 1 (Fig. 2.5).



**Fig. 2.5** Execution trace of K-Main Routes (KMR). Circles represent nodes, lines represent edges, and squares represent activities (Best in color)

Two groups are formed by assigning each activity to its closest summary path in step 2 of Fig. 3.3. Activities 1, 2, 3, 4, and 5 are assigned to summary path  $\langle A, B \rangle$ , and activities 6, 7, 8, 9, 10 and 11 are assigned to summary path  $\langle D, E \rangle$ . Dashed lines are used to highlight the groups. If an activity's distance to multiple summary paths is equal, the activity is randomly assigned to one of the paths.

The next step involves recalculating the summary paths of each group once the groups have been formed (step 3). In the example, the new summary paths (shown in step 4) are  $\langle A, B, C \rangle$  and  $\langle D, E, F \rangle$ . As can be seen, these summary paths further maximize activity coverage, and this is the reason they are chosen.

In Step 5, the process of forming groups is repeated, but this time, the summary paths used are not the initial seeds we started with but the new summary paths  $\langle A, B, C \rangle$  and  $\langle D, E, F \rangle$  that we calculated in step 4. Again, dashed lines are used to highlight the groups. Another recalculation of the group representatives or summary paths is done in Step 6 to see if we can further maximize activity coverage. At this point, the algorithm determines that the summary paths that are recalculated do not change and as a result the algorithm terminates. We now take a closer look at each phase of the algorithm.

---

#### Algorithm 1 K-Main Routes (KMR) Algorithm

---

**Input:**

- 1) a spatial network  $G = (N, E)$ ,
- 2) a set of activities  $A$ ,
- 3) a number of routes  $k$ ,
- 4)  $\text{mode1} \in \{\text{naive}, \text{NOVA}\}$ ,
- 5)  $\text{mode2} \in \{\text{naive}, \text{D-SPARE}\}$

**Output:**

A summary path set of size  $k$  and a partitioning of activities across these summary paths, where the objective is to maximize the activity coverage of each summary path for the group it represents.

**Algorithm:**

- 1:  $P \leftarrow$  shortest paths between active nodes of  $G$
  - 2:  $\hat{P} \leftarrow k$  summary paths  $\in P$ ;  $\text{stableGroups} \leftarrow \text{false}$ ;
  - 3: **while** not  $\text{stableGroups}$  **do**
  - 4:   **Phase 1:**  $\text{currentGroups} \leftarrow \text{AssignActivities-ToSummaryPaths}(G, A, k, \hat{P}, \text{mode1})$
  - 5:   **Phase 2:**  $\hat{P}' \leftarrow \text{RecomputeSummaryPaths}(G, A, k, \text{currentGroups}, \text{mode2})$
  - 6:   **if**  $\hat{P} = \hat{P}'$  **then**  $\text{stableGroups} \leftarrow \text{true}$
  - 7:   **end if**
  - 8:    $\hat{P} \leftarrow \hat{P}'$
  - 9: **end while**
  - 10: **return**  $\text{currentGroups}$
-

### 2.3.3.1 Phase 1: Assign Activities to Nearest Summary Paths (i.e., Forming Groups)

This phase involves forming  $k$  groups by assigning each activity to its closest summary path. The pseudocode for the activity assignment algorithm is presented in Algorithm 2. The algorithm has two modes: naive and NOVA (Network Voronoi activity Assignment). In the naive mode, all distances between each activity and summary path are enumerated in order to determine the closest summary path. In contrast, NOVA avoids this enumeration and still delivers correct results.

**Performance-Tuning for Phase 1:** The Network Voronoi activity Assignment (NOVA) technique is a faster way of grouping activities, i.e., assigning each activity to its closest summary path. To understand NOVA, imagine a virtual node  $V$  was connected to every node in all summary paths by edges of weight zero (see Fig. 2.6). NOVA calculates the shortest path from  $V$  to all active nodes and discovers the closest

---

#### Algorithm 2 AssignActivitiesToSummaryPaths

---

**Input:**

- 1) a spatial network  $G = (N, E)$ ,
- 2) a set of activities  $A$ ,
- 3) a number of routes  $k$ ,
- 4) a set of summary paths  $\hat{P}$ ,
- 5) mode  $\in \{\text{naive}, \text{NOVA}\}$

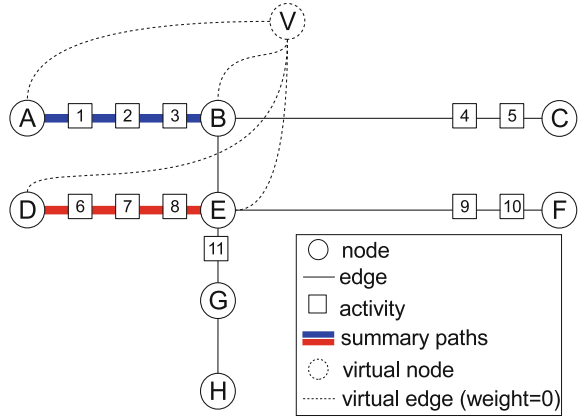
**Output:**

$k$  groups formed by assigning each activity  $\in A$  to the closest summary path  $\in \hat{P}$ .

**Algorithm:**

- 1: **if** mode = “naive” **then**
  - 2:   Enumerate all distances between each activity  
        $a_i \in A$  and each summary path  $p_i \in \hat{P}$
  - 3:    $\text{currentGroups} \leftarrow$  assign each  $a_i$  to the closest  $p_i$
  - 4: **else if** mode = “NOVA” **then**
  - 5:    $V \leftarrow$  virtual node connected to all nodes of all  
        $p_i \in \hat{P}$  with zero-weight edges
  - 6:    $\text{Open}, T_{\text{nodes}} \leftarrow V; \text{Closed}, T_{\text{activities}} \leftarrow \emptyset$
  - 7:   **repeat**
  - 8:      $n \leftarrow$  closest node in  $\text{Open}$  to any  
        $p_i \in \hat{P}; \text{Closed} \leftarrow n$
  - 9:     update  $T_{\text{nodes}}$  with  $x_i.\text{distance}$  and  $x_i.\text{sp}$  for  
       each of  $n$ 's neighbors  $x_i \notin \text{Closed}$
  - 10:    **if**  $x_i \notin \text{Open}$  **then**  $\text{Open} \leftarrow x_i$
  - 11:    **end if**
  - 12:    update  $T_{\text{activities}}$  with  $a_i.\text{distance}$  and  $a_i.\text{sp}$  for  
       each activity  $a_i \in \text{edge}(n, x_i)$
  - 13:     $\text{currentGroups} \leftarrow$  assign each  $a_i$  to the closest  
        $p_i$  based on  $T_{\text{activities}}$
  - 14:   **until** all active nodes  $\in \text{Closed}$  or  $|\text{Open}| = 0$
  - 15:   **if**  $\exists$  unassigned activities,  $a_j$  **then**  
        $\text{currentGroups} \leftarrow$  assign each  $a_j$  to any  $p_i$
  - 16:   **end if**
  - 17: **end if**
  - 18: **return**  $\text{currentGroups}$
-

**Fig. 2.6** An example of NOVA. Activity 10 gets assigned to summary path  $\langle D, E \rangle$  because the shortest path  $\langle V, E, F \rangle$  from  $V$  to activity 10 goes through node  $E$  of summary path  $\langle D, E \rangle$  (Best in color)



summary path to each activity. It is based on the observation that the shortest path from  $V$  to each activity  $a$  will go through a node in the summary path that is closest to  $a$ .

The NOVA algorithm begins by initializing all the relevant data structures. Line 5 of Algorithm 2 initializes the virtual node  $V$  connected to each node of all summary paths by edges of weight 0. Line 6 initializes both the *Open* list and  $T_{nodes}$  to  $V$  as well as the *Closed* and  $T_{activities}$  to the empty set. NOVA then expands every node in the *Open* list based on how close it is to the summary paths; closer nodes get expanded first. Once a node  $n$  is expanded, it is moved to the *Closed* list (line 8). Next, each of  $n$ 's neighboring nodes  $x_i \notin \text{Closed}$  is examined, and  $T_{nodes}$  is updated with  $x_i$ 's distance and *sp* information, where  $x_i.distance$  is the network distance of  $x_i$  from the nearest summary path, and  $x_i.sp$  is  $x_i$ 's assigned summary path.  $x_i.distance$  is calculated by adding  $n.distance$  to the distance of edge  $(n, x_i)$  (line 9). If  $x_i$  is not in *Open*, it is then added to the *Open* list (line 10).

NOVA records the activity distance to a summary path once it finds activities on an edge connecting node  $n$  to the summary path. Every activity  $a_i$  that is on edge  $(n, x_i)$  is examined, and  $T_{activities}$  is updated with  $a_i$ 's distance and *sp* information, where  $a_i.distance$  is the network distance of  $a_i$  from the nearest summary path (based on  $n$ ), and  $a_i.sp$  is the assigned summary path of  $a_i$ . Next, an activity is assigned to a summary path (line 12). If the activity was previously assigned to another summary path, it is removed from that path before being assigned to the new summary path. Once all active nodes have been added to the *Closed* list, or the *Open* list is empty, NOVA's main loop is stopped (line 13). If unassigned activities remain due to no connectivity to summary paths, these activities are randomly assigned to any summary path  $\in \hat{P}$  (line 14). NOVA then terminates and returns the current groups it finds (line 15).

Figure 2.6 shows an example of NOVA activity assignment. Virtual node  $V$  is connected by zero weight edges to nodes  $A$  and  $B$  of summary path  $\langle A, B \rangle$  and nodes  $D$  and  $E$  of summary path  $\langle D, E \rangle$  (Algorithm 2, line 5). NOVA assigns activity 10

to summary path  $\langle D, E \rangle$  because the shortest path  $\langle V, E, F \rangle$  from  $V$  to activity 10 goes through node  $E$  of summary path  $\langle D, E \rangle$ . Similarly, NOVA assigns activity 5 to summary path  $\langle A, B \rangle$  because the shortest path  $\langle V, B, C \rangle$  from  $V$  to activity 5 goes through node  $B$  of summary path  $\langle A, B \rangle$ .

### 2.3.3.2 Phase 2: Recompute Summary Paths (i.e., Selecting Group Representatives)

During phase 2, the summary path of each group is recalculated with the objective of maximizing activity coverage (recall that activity coverage is the number of activities covered by a set of paths). The pseudocode for recomputing summary paths is presented in Algorithm 3, which has two modes naive and D-SPARE. The naive mode enumerates the shortest paths between all active nodes in the spatial network while D-SPARE considers only the set of shortest paths between the active nodes of a group, which gives the correct results.

**Performance-Tuning for Phase 2:** Performance improvement for phase 2 is achieved via the Divide and Conquer Summary Path REcomputation (D-SPARE) algorithm, which chooses the summary path for each group with maximum activity coverage ( $maxPath$ ) but only considers the set of shortest paths between the nodes of a given group (Algorithm 3, lines 8–9). D-SPARE assumes rich connectivity and otherwise may return a summary path going outside a given fragment. If  $maxPath$  is null after looking at the shortest paths in a given group,  $c_i \in currentGroups$ , the shortest path which has the maximum activity coverage based on the activities in  $c_i$  is selected as  $maxPath$  (lines 10–12).  $maxPath$  is then added to  $\hat{P}$  as the new summary path for  $c_i$  (line 13). Once all groups  $c_i \in currentGroups$  have been considered,  $\hat{P}$ , which contains the summary paths with maximum activity coverage for each group, is returned.

An example of D-SPARE is shown in Fig. 2.7. The given group consists of activities 1, 2, 3, 4, and 5, and summary path  $\langle A, B \rangle$ . D-SPARE chooses a new summary path that maximizes activity coverage for every group based on the activities of each group. Summary paths  $\langle A, B, C \rangle$  and  $\langle C, B, A \rangle$  both maximize activity coverage based on activities 1, 2, 3, 4, and 5, so D-SPARE will choose one of them as the new representative for this group.

**KMR with Inactive Node Pruning Performance-Tuning:** Inactive node pruning considers only paths between active nodes (vs. paths between all nodes) which reduces the total number of paths considered (Algorithm 1, line 1). An example of inactive node pruning is shown in Fig. 2.1a. The active nodes in this network are  $A, B, C, D, E, F$ , and  $G$ . Without inactive node pruning, the number of shortest paths considered would be 56 because there are eight nodes, and the shortest path between each node and every other node is considered. With inactive node pruning, the number becomes 42, as only the shortest paths between the seven active nodes are considered.



**Algorithm 3** RecomputeSummaryPaths**Input:**

- 1) a spatial network  $G = (N, E)$ ,
- 2) a set of activities  $A$ ,
- 3) a number of routes  $k$ ,
- 4) a set of groups  $currentGroups$ ,
- 5) mode  $\in \{naive, D-SPARE\}$

**Output:**

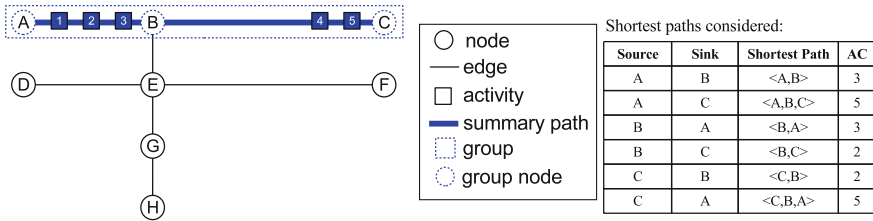
$k$  summary paths,  $\hat{P}'$ , that maximize activity coverage (AC) for each group  $\in currentGroups$

**Algorithm:**

```

1: if mode = "naive" then
2:   for each  $c_i \in currentGroups$  do
3:      $P \leftarrow$  shortest paths between active nodes of  $G$ 
4:      $maxPath \leftarrow$  path in  $P$  with Max AC based on
        $c_i$ 's activities
5:      $\hat{P}' \leftarrow maxPath$ 
6:   end for
7: else if mode = "D-SPARE" then
8:   for each  $c_i \in currentGroups$  do
9:      $P' \leftarrow$  the set of shortest paths between the active
       nodes of  $c_i$ 
10:     $maxPath \leftarrow$  path in  $P'$  with Max AC based on
        $c_i$ 's activities
11:    if  $maxPath = \emptyset$  then
12:       $P \leftarrow$  shortest paths between active nodes
        of  $G$ 
13:       $maxPath \leftarrow$  path in  $P$  with Max AC based
        on  $c_i$ 's activities
14:    end if
15:     $\hat{P}' \leftarrow maxPath$ 
16:  end for
17: end if
18: return  $\hat{P}'$ 

```

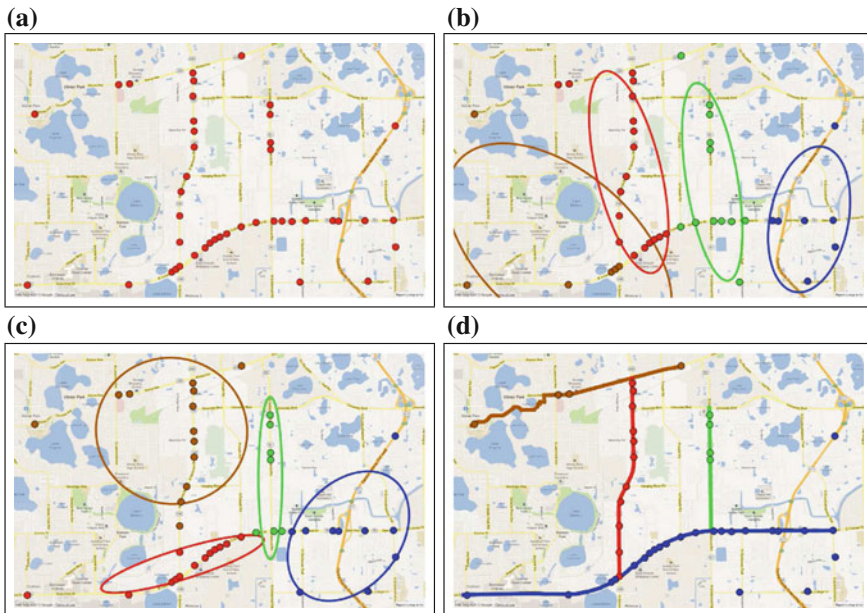


**Fig. 2.7** An example of D-SPARE. Only shortest paths between nodes in the group are considered when choosing the new summary path which maximizes activity coverage. In this case, summary paths  $\langle A, B, C \rangle$  and  $\langle C, B, A \rangle$  both maximize activity coverage based on activities 1, 2, 3, 4, and 5 so D-SPARE will choose one of these summary paths as the new representative for this group (Best in color)

## 2.4 Case Study

A qualitative evaluation comparing KMR with Crimestat K-means [11, 31] on a real pedestrian fatality data set [32] is shown in Fig. 2.6a. The data consists of 43 pedestrian fatalities (represented as dots) in Orlando, Florida occurring between 2000 and 2009. KMR uses paths and network distance to group activities on a spatial network whereas in geometry-based techniques such as K-Means, the partitioning of spatial data is based on grouping similar points distributed in planar space where the distance is calculated using Euclidean distance. Such techniques focus on the discovery of the geometry (e.g., circle, ellipse) of high density regions [2] and include K-means [11, 33–37], K-medoid [12, 13], P-median [14] and Nearest Neighbor Hierarchical Clustering [15] algorithms.

When evaluating the techniques, it is important to consider both the groups (represented by colors) and the representatives of each group (e.g., paths or ellipses). Pedestrian fatalities usually occur on streets, particularly along arterial roadways [38]. Thus this activity can be said to have a linear generator. However, the results generated by Crimestat do not capture this. From Fig. 2.8b, it is clear that the ellipse-based output is meant for areas, not streets. When Crimestat K-Means is changed to use network distance, only marginal improvement may be observed. Although the red ellipse in Fig. 2.8c is aligned with a part of the arterial road, not all the activities on this arterial



**Fig. 2.8** A comparison of KMR and Crimestat K-means when  $k = 4$  on pedestrian fatality data from Orlando, FL [32]. **a** Input. **b** Crimestat K-means with Euclidean Distance. **c** Crimestat K-means with Network Distance. **d** KMR (Best in color)

are captured. For example, the activities that occur on the road towards the bottom of the figure are split among the red, green, and blue groups. In contrast, the groups of activities in KMR capture the activities on the arterial roads (Fig. 2.8d). For example, the blue group and summary path capture the activities on the arterial road that were split across three groups in network-based K-Means. The group representatives that are paths make sense in this context due to the linear nature of the activities. In another context (in the absence of a linear generator), the geometry-based output of K-Means might make more sense; this is not the case in spatial networks.

## 2.5 Summary

This chapter explored the challenge of handling a large number of connected components in the spatial network. This challenge was conceptualized as the spatial network activity problem (SNAS), which important application domains such as crime analysis and preventing pedestrian fatalities. The chapter presented the current state-of-the-art techniques, as well as emerging trends such as the KMR algorithm. KMR uses inactive node pruning, Network Voronoi activity Assignment (NOVA) and Divide and conquer Summary PATH REcomputation (D-SPARE) to enhance its performance and scalability. A case study comparing various techniques for addressing SNAS on pedestrian fatality data was presented.

## References

1. Oliver, D., Shekhar, S., Kang, J. M., Laubscher, R., Carlan, V., & Bannur, A. (2014). A k-main routes approach to spatial network activity summarization. *IEEE Transactions on Knowledge and Data Engineering*, 26(6), 1464–1478.
2. Eck, J., Chainey, S., Cameron, J., & Wilson, R. (2005). Mapping crime: Understanding hotspots, National Institute of Justice.
3. Matthews, D. A., Effler, S. W., Driscoll, C. T., O'Donnell, S. M., & Matthews, C. M. (2008). Electron budgets for the hypolimnion of a recovering urban lake, 1989–2004: Response to changes in organic carbon deposition and availability of electron acceptors". *Limnology and Oceanography*, 53(2), 743–759.
4. Chicago Tribune, Metra argues for delay of 'fail-safe' rail system. <https://goo.gl/3bxuw0>.
5. Huffington Post, Hungary: Snowstorm strands thousands in their cars. <http://www.huffingtonpost.com/huff-wires/20130315/eu-europe-snow>.
6. Brantingham, P. J., & Brantingham, P. L. (Eds.) (1981). *Environmental criminology* (pp. 27–54). Beverly Hills: Sage Publications.
7. Levine, N. (2006). Crime mapping and the Crimestat program. *Geographical analysis*, 38(1), 41–56.
8. Scott, M. S., & Dedel, K. (2006). *Assaults in and around bars* (2nd ed.). Washington, DC: Office of Community Oriented Policing Services.
9. Cohen, L. E., & Felson, M. (1979). Social Change and Crime Rate Trends: A Routine. *American sociological review* (pp. 588–608)
10. Brantingham, P. J., & Brantingham, P. L. (1993). Environment, routine and situation: Toward a pattern theory of crime. *Advances in criminological theory*, 5, 259–294.

11. MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, 1(14), 281–297.
12. Kaufman, L., & Rousseeuw, P. J. (2009). *Finding groups in data: An introduction to cluster analysis* (Vol. 344). Wiley.
13. Ng, R.T. & Han, J. (1994). Efficient and effective clustering methods for spatial data mining. *Proceedings of the International Conference on Very Large Databases*.
14. Resende, M. G., & Werneck, R. F. (2004). A hybrid heuristic for the p-median problem. *Journal of Heuristics*, 10(1), 59–88.
15. D’Andrade, R. G. (1978). U-statistic hierarchical clustering. *Psychometrika*, 43(1), 59–67.
16. Celik, M., Shekhar, S., George, B., Rogers, J. P., & Shine, J. A. (2007). *Discovering and Quantifying Mean Streets: A Summary of Results, Technical Report 07–025*. Computer Science and Engineering: University of Minnesota.
17. Buchin, K., Cabello, S., Gudmundsson, J., Löffler, M., Luo, J., & Rote, G. et al. (2009). Detecting hotspots in geographic networks, (pp. 217–231). Berlin: Springer.
18. Roach, S. A., & Roach, S. A. (1968). *The theory of random clumping*. London: Methuen.
19. Okabe, A., Okunuki, K. I., & Shiode, S. (2006). SANET: A toolbox for spatial analysis on a network. *Geographical Analysis*, 38(1), 57–66.
20. Shiode, S., & Okabe, A. (2004). Network variable clumping method for analyzing point patterns on a network. *Unpublished paper presented at the Annual Meeting of the Associations of American Geographers*. Philadelphia, Pennsylvania.
21. Aerts, K., Lathuy, C., Steenberghen, T., & Thomas, I. (2006). Refining spatial clustering of traffic accidents using distances along the network. In *Proceedings of 19th workshop of the international cooperation on theories and concepts in traffic safety*.
22. Spooner, P. G., Lunt, I. D., Okabe, A., & Shiode, S. (2004). Spatial analysis of roadside Acacia populations on a road network using the network K-function. *Landscape Ecology*, 19(5), 491–499.
23. Steenberghen, T., Dufays, T., Thomas, I., & Flahaut, B. (2004). Intra-urban location and clustering of road accidents using GIS: A Belgian example. *International Journal of Geographical Information Science*, 18(2), 169–181.
24. Yamada, I., & Thill, J. C. (2007). Local indicators of networkconstrained clusters in spatial point patterns. *Geographical Analysis*, 39(3), 268–292.
25. Shiode, S., & Shiode, N. (2009). Detection of multiscale clusters in network space. *International Journal of Geographical Information Science*, 23(1), 75–92.
26. Shekhar, S., & Liu, D. R. (1997). CCAM: A connectivity-clustered access method for networks and network computations. *IEEE Transactions on Knowledge and Data Engineering*, 9(1), 102–119.
27. Meehan, Bill. (2013). *Modeling electric distribution with GIS*. Redlands: Esri Press.
28. Cormen, T. H. (2001). *Introduction to algorithms*. MIT press.
29. Michael, R. G., & David, S. J. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. San Francisco: W.H. Freeman.
30. Hochbaum, D. S. (1996). Approximating covering and packing problems: set cover, vertex cover, independent set, and related problems. In *Approximation algorithms for NP-hard problems* (pp. 94–143). PWS Publishing Co.
31. Levine, N. (2008). CrimeStat: A spatial statistics program for the analysis of crime incident locations, vol 3.1, Houston, TX: Ned Levine and Associates; and Washington, DC: The National Institute of Justice.
32. Fatality Analysis Reporting System (FARS) Encyclopedia, National Highway Traffic Safety Administration (NHTSA), <http://www.nhtsa.gov/FARS>.
33. Borah, S., & Ghose, M. K. (2009). Performance analysis of AIM-K-means and K-means in quality cluster generation. ArXiv preprint [arXiv:0912.3983](https://arxiv.org/abs/0912.3983).
34. Barakbah, A. R., & Kiyoki, Y. (2009). A pillar algorithm for k-means optimization by distance maximization for initial centroid designation. In *IEEE Symposium on Computational intelligence and data mining, 2009. CIDM’09* (pp. 61–68). IEEE (2009).

35. Khan, S. S., & Ahmad, A. (2004). Cluster center initialization algorithm for K-means clustering. *Pattern recognition letters*, 25(11), 1293–1302.
36. Pelleg, D., & Moore, A. W. (2000). X-means: Extending K-means with efficient estimation of the number of clusters. In *ICML* (Vol. 1).
37. Bradley, P. S., & Fayyad, U. M. (1998). Refining initial points for K-means clustering. *ICML*, 98, 91–99.
38. Ernst, M., Lang, M., & Davis, S. (2011). Dangerous by design: Solving the epidemic of preventable pedestrian deaths, Transportation for America: Surface Transportation Policy Partnership, Washington, DC.

Spatial Network Data

Concepts and Techniques for Summarization

Oliver, D.

2016, XII, 50 p. 19 illus., Softcover

ISBN: 978-3-319-39620-0