Service Management and Allocation with Modified R-Tree Indexing for Spatial Activity Clustering

Midhunlal P V[1] Jisha P Abraham[2] Surekha Mariyam Varghese[3]
[1]PG Student [2]Assosiate Professor [3]Head of Department
[1,2,3]Department of Computer Science and Engineering
[1,2,3]Mar Athanasius College of Engineering, Kothamangalam, Kerala

Abstract— The process of creating groups from a list of elements based on its properties is called clustering in datamining. Clustering can be applied to the spatial domain in which the clustered elements will be in the spatial network. In Spatial Activity Clustering the elements are the spatial locations of activities occurring in the spatial network. There is a Spatial network and set of activities, the aim is to create k linear paths which are clusters consisting of activities with minimum distances. Its applications is in the area of GIS (Geographic Information Systems) consisting of linear paths such as roadways and observations is in them. Spatial Activity Clustering uses network distance for clustering. Related works are based on geometric distances are less efficient for linear clustering. There is also works using network distance uses subgraph based approaches creates only single routes. The proposed system for Spatial Activity Clustering (SAC) computes shortest paths in spatial network which are the clusters consisting of activities. It uses End Point Joining (EPJ) for finding efficient routes. After the clustering if a need for emergency service arises, service allocation routes for these clusters are created from the service locations which may contain multiple units of service. Accessing spatial data sequentially is a cumbersome task in terms of time and cost. For minimizing the access time, a modified R Tree indexing is used to improve the performance of SAC. This tree creates neighboring links in the leaf nodes based on the spatial proximity. kNN (Nearest Neighbor) Search is used to identify the k nearest summarypaths (cluster) where multiple units can service and service routes are created.

Key words: R-Tree, Spatial Activity Clustering, data mining

I. INTRODUCTION

Spatial Activity Clustering (SAC) has its applications in GIS areas which includes linear paths such as roads, rail tracks etc. linear paths includes the observations which are used for clustering. It can be used to find road segments or lines which are hazardous to pedestrians and has to be altered [1]; transportation engineers need to identify rail lines to improve safety and reduce cost by understanding derailing [2]

Spatial Activity Clustering [3] initially has a spatial network which can be considered as a map with linear lines like roads. It can be considered as road network. Also a set activities with their locations is also given as input which are mostly occurred along the paths of spatial network. Then from these activity locations k routes/clusters are formed as linear paths. The value of k is also based on the characteristics of input.

Fig 1 and 2 displays an example input and output respectively for Spatial Activity clustering. The main components of input are nodes, edges and activities. Nodes are represented using circles and their amount is 9. There are also 16 activities represented as squares.8 edges are used to connect the nodes (with edge weights of 1 for simplicity).

In the output there are three linear clusters of activities. This linear cluster is represented through the shortest paths between their sink and source nodes. Activities 1,2,3,4 and 5 is in the cluster <A, B, C>. Route <B, E, H> represents activities 6, 7, 8, 9, 10 and 11 & route <G, H, I> represents activities 12,13,14,15 and 16. So the activities are summarized using a set of routes

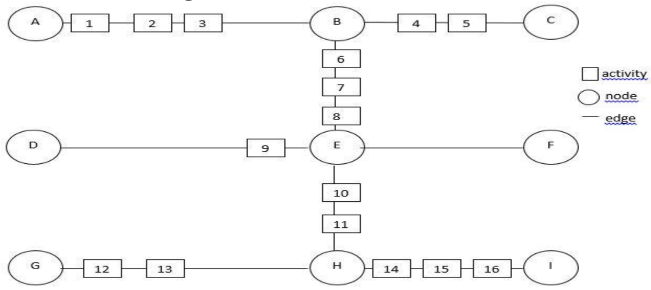

Fig. 1: Example of input of Spatial Activity Clustering

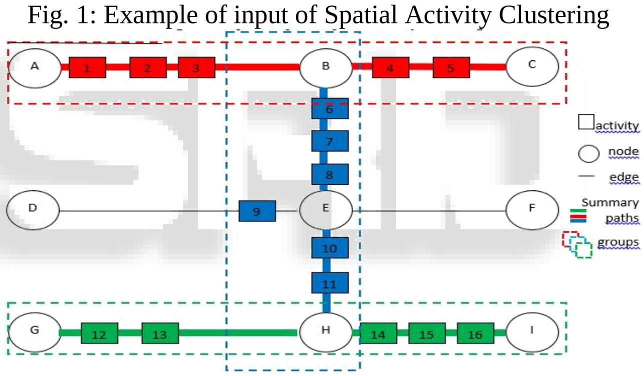

Fig. 2: Example of output of Spatial Activity Clustering

After finding the clusters/summarypaths there may be situations which needs immediate actions on these clusters. Service Route allocation is the process of allocating a service /facility to these clusters. Service points are located like the activities in spatial network. Every service station will service the nearest cluster. Service points may contain multiple service units. Routes from these service points to these clusters are created through the spatial network. The routes are the shortest path available with these points.

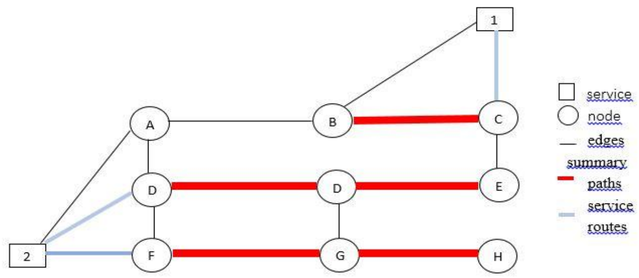

Fig. 3: Service Route allocation

Fig 3 shows the output of Service Route allocation in for spatial clusters. Spatial network contains 2 service points 1 and 2. Available service units for each service point are also 1 and 2 which means 1 can serve 1 cluster/summarypath and 2 can service 2 clusters/summarypaths. 3 clusters are calculated in spatial network for the service allocation. . In this process, for every service point the nearest route to cluster is found. First the obtained routes are indexed using Modified R-Tree Indexing [4] which is a variant of R-Tree [5]. Properties of B+-tree [6] is added to R-Tree to improve the access time for sequential access. Leaf nodes are arranged every time based on their spatial locations. Nearest service route is found using k NN (Nearest Neighbor) search [7] on the summarypaths.

II. RELATED WORK

Clusters/groups are created and summarized in many ways in datamining. They can be mainly classified as geometry based [8-12] and network based [13-14].

First one, geometry based clustering creates clusters/groups based on the Euclidean distance between the data. It creates geometrical groups of circles, ellipses, i.e. These geometrical regions consists of the high density groups of data for clustering. Spatial proximity is included in this type but spatial network proximity is not considered. Thus it will not cluster/group activities occurring on the same road and may create cluster of activities which are not interconnected by a network.

Network distance of these data is not considered for the clustering. Geometry based clustering comprises of K-Means [8], K-medoid [9], [10], P-median [11] and Nearest Neighbor Hierarchical Clustering [12].

Network based clustering uses network distance rather than the Euclidean distance. They are like distance and displacement in physics. This clustering schemes will create network structure clusters of in the form of lines/roads. Examples of these type clustering are mean streets [13] and clumping [14]. Mean streets creates clusters based on the high activity levels. It is not designed to create k clusters. The created routes usually low because of these high activity levels. NT-VCM (Network based Variable clumping distance is a clumping method which considers the distance threshold between activities to cluster them. Distance threshold must be defined for this purpose.

III. SPATIAL ACTIVITY CLUSTERING

This section describes the spatial activity clustering phase. Basic concepts are also described here. Spatial clustering algorithm also shown after the basic concepts.

Definition 1: (x, y) is a pair of real numbers which represents a spatial location n in Euclidean plane. Nodeset N is a group consists of elements like n which are called nodes (n €N). These spatial locations are interconnected through edges which consists of 2 spatial locations which defines the end points of edge. These end points are the nodes defined in N, so element e can represent as a pair of n's, e=(m, n). Elements like e (e €E) are collected into edge set E. These Nodeset N and Edgeset E together create a Spatial Network S = (N, E).

Fig 1.a shows a spatial network. Nodes are shown using circles and edges are shown using lines in the figure. An example for spatial network is a road network in a map. Junctions displayed as nodes and roads displayed as edges in the road network.

Definition 2: Activity a is spatial location like node but it occurs only in the edges or node and it is an object of interest. A is the collection of activities (a €A). Each activity is belonged to only one node (n €N) or one edge (e €N).

Squares are the activities in fig1.a. In real life activity may be a pedestrian fatality in transportation planning.

Definition3: Activity set is divided into p paths like edges. These paths are grouped into Summarypathset P (p €P). For each a in A network distance (a, pi) ≤ network distance (a, pj) where pi, pj €P.

Fig 1.2 shows the summarypathset P. P consists of 3 summarypaths. ABC, BEH, GHI are the summarypaths represented in the figure.

Activities 1,2,3,4 and 5 is in the cluster <A, B, C>. Route <B, E, H> represents activities 6, 7, 8, 9, 10 and 11 & route <G, H, I> represents activities 12,13,14,15 and 16.

Definition 4: Service point s is spatial locations like activities or nodes. Each service point has available units u along with it. S is the collection of service points along with the respected units(s, u) €S.

A real world look up for a service point is the location of a fire station and available units in that fire station are number of fire units in a fire stations. Squares in fig 3 represents service points along with units inside it.

Definition 5: ServiceRouteset R consists of a group of routes r. r €R is the shortest route through the spatial network which has one endpoint as servicepoint s €Sand anther endpoint is on the summary path p €P.

Fig 3 shows service routes as thicken lines. The service route can be considered as the route fire unit taken to reach the summary path.

A. *Problem statement*

The problem of Spatial Activity Clustering (SAC) can be expressed as follows:

Given:

- A spatial network G = (N, E) with network distance weight function w (u, v) ≥ 0 for each edge e = (u, v) ∈ E,
- An Activity set A and their locations (e.g., in a node or an edge),
- A desired number of summary paths, k, where k ≥ 1.

Find:

- A summary path set of size k,
- A partitioning of activities across these summary paths.

Objective:

- Maximize the activity coverage of each summary path for the group it represents.

Constraints:

- Each summary path is a shortest path between its end-nodes,
- Every activity is in one edge or one node.

The spatial network input for SAC is defined in Definition 1. Second input activity set is the objects of interest in the spatial network like the location of accidents. The k input represents the desired number of summary paths. The output for SAC is a partitioning of activities across the paths as a summary path set of size k and. The summary paths are representatives for each group and the activity coverage for the group it represents is maximum.

Service Route allocation problem can be defined as follows:

Given:

- A spatial network G = (N, E) with network distance weight function w (u, v) ≥ 0 for each edge e = (u, v) ∈ E,
- A set of Service points S and their locations and the available units u[s] (e.g., a node or an edge),
- A summary path set of size k obtained from SAC.

Find:

- A service route set of size k,

Objective:

- Minimize the distance for service routes Constraints:
- Each service path is a shortest path between cluster and service point,
- Each service point a ∈A is associated with only one edge e ∈E,
- Number of clusters serviced will be based on the number of service points and available units.

The spatial network input for SRA is similar to SAC. Service points will find its nearest Neighbors using kNN (nearest neighbor) search. K service paths are created with respect to each summarypaths. Service points and Service routes are defined in definition 4 and definition 5.

B. *Spatial Activity Clustering Algorithm*

Algorithm 1: Spatial Activity Clustering (SAC) Algorithm

Input:

- A spatial network G= (N, E),
- A set of activities A,
- A number of routes k,
- Mode1 ∈ {naive, NOVA_TKDE},
- Mode2∈ {naive, D-SPARE_TKDE}

Output:

- k summarypaths consists of activities belonging to activity set. Totally it can be represented as summarypathset, which can be the cluster group.

Algorithm:

1) P← shortest paths between active nodes of G
2) $\hat{P}$ ←k summarypaths/shortest paths ∈ P; stablePaths←false;
3) while not stablePaths do
4) Phase1:currentGroups←AssignActivitiesTo SummaryPaths (G, A, k,$\hat{P}$, mode1)
5) Phase 2: $\hat{P}'$←RecomputeSummaryPaths (G, A, k, currentGroups, mode2)
6) if $\hat{P}$ = $\hat{P}'$then stablePaths←true
7) if endpoints of $\hat{P}$ have duplicate entries join summarypaths with same end points goto step3
8) $\hat{P}$ ← $\hat{P}'$
9) returncurrentGroups

Algorithm 1 presents the pseudocode for the proposed Spatial Activity Clustering (SAC) approach. Basically SAC is similar to K-Means clustering, It chooses initial seeds for creating k groups, each group representative is updated in each insertion to it, and it stops when assignments do not change. All nodes which has endpoints on active nodes is selected as P in Algorithm 1 line 1(inactive node pruning). “seeds” of the SAC is selected in line2 as choosing k paths from P as initial summary paths. Next, main phases of two important processes in the algorithm is executed. First step (line 4) is the creation of k groups which are created by grouping each activity to a summarypath which is near to that activity. Second step (line 5) optimizes the summarypaths based on the criteria of maximum activity coverage. These two steps are repeated till the assignments do not change. Final summary paths and groups are returned when the algorithm terminated.

Algorithm 2: AssignActivitiesToSummaryPaths

Input:

- A spatial network G= (N, E),
- A set of activities A,
- A number of routes k,
- A set of summary paths, $\hat{P}$,
- Mode ∈ {naive, NOVA_TKDE}

Output:

- k summary paths created by the assignment of every activity a ∈ A to the nearest summary path ∈$\hat{P}$.

Algorithm:

1) if mode = “naive" then
2) Enumerate all distances between each activity, a_i ∈A and each summary path pi ∈ $\hat{P}$
3) currentGroups← assign each, a_i to the closest, p_i
4) else if mode = “NOVA_TKDE" then
5) V ← virtual node connected to all nodes of all, p_i ∈ $\hat{P}$ with zero-weight edges
6) Open,T_{nodes} ←V; Closed,$T_{activities}$ ←∅
7) repeat
8) n← closest node in Open to any p_i ∈ $\hat{P}$; Closed←n
9) update T_{nodes} with x_i.distance and x_i.sp for each of n’s neighbors x_i∉Closed
10) if x_i ∉Open then Open←x_i
11) update,$T_{activities}$ with a_i.distance and a_i.sp for each activity a_i ∈edge (n, x_i)
12) currentGroups← assign each a_i to the closest p_i based on $T_{activities}$
13) until |Open|=0 or all active nodes ∈Closed
14) for all unassigned activities, $a_j$then
    i. currentGroups← assign each a_j to any p_i
15) return currentGroups

Assignment of every activity to its nearest summary path creates k groups/clusters. Naïve and NOVA_TKDE are the two modes of activity assignment algorithm which is represented algorithm2. Assignment of activity to its nearest summary path is done by enumerating all the distances between every summary path and activity. Enumeration process of naïve mode is not done in NOVA_TKDE. Without that it gives the desired results.

Performance-Tuning for Phase 1: Assignment of activities to nearest summary path is done by Network Voronoi activity Assignment (NOVA_TKDE) technique is a faster process. Assume all summary paths are connected to a virtual node V by edges of weight zero. Nearest summary path is discovered by finding the distance all active nodes to V. A node in the summary path that is closest to a €A will be on the shortest path from a to V. All relevant data structures are initialized in the starting of pseudo code for NOVA_TKDE.

Next, virtual node V connected to each node of all summary paths by edges of weight 0 is initialized in algorithm2 line5. Line 6 initializes the Open list and T_{nodes} to V and the Closed and $T_{activities}$ to the empty set. The node which are nearer to summarypaths are expanded from the open list by NOVA_TKDE. Once a node n is expanded, it is moved to the Closed list (line 8). Next, each of n's neighbors x_i ∉Closed is examined, and T_{nodes} is updated withx_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 isx_i's assigned summary path. x_i.distance is calculated by adding n.distance to the distance of edge (nx_i,) (line 9). Line 10 ads x_i to Open List, if it is not in Open list (line 10). Activity distance to a summarypath is stored when NOVA_TKDE finds activities on an edge connecting node n to that summary path. Every activity a_i that is on edge (n, x_i) is examined, and $T_{activities}$ is updated witha_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 ofa_i. Line 12 assigns an activity to a summary path. Assignment of an activity to a summarypath is done only after removing the previos path of activity if activity was assigned earlier. Line 13 terminates main loop of NOVA_TKDE when Open list is empty or Closed list contains all active nodes. Line 14 randomly assigns activities to any summary path ∈$\hat{P}$If if there are unassigned activities remaining due to no connectivity to summary paths. Line 15 stops NOVA_TKDE and returns the current groups.

Algorithm 3: RecomputeSummaryPaths

Input:

- A spatial network G= (N, E),
- A set of activities A,
- A number of routes k,
- A set of groups currentGroups,
- Mode ∈ {naive, D-SPARE_TKDE}

Output:

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

Algorithm:

1) if mode = "naive" then
2) for each c_i∈currentGroups do
3) P← shortest paths between active nodes of G
4) maxPath← path in P with Max AC based onc_i's activities
5) $\hat{P}\,'$←maxPath
6) else if mode = "D-SPARE_TKDE" then
7) for each c_i ∈currentGroups do
8) $\hat{P}\,'$←the set of shortest paths between the active nodes of c_i
9) maxPath← path in P with Max AC based onc_i's activities
10) if maxPath=∅then
11) P← shortest paths between active nodes of G
12) maxPath← path in P with Max AC based onc_i's activities
13) $\hat{P}\,'$←maxPath
14) 14: return $\hat{P}\,'$

End Point Joining

If the obtained routes have same endpoints, join them as single summaypath and do the SAC algorithm again with obtained output as seeds

Algorithm:

1) If(Endpoints matched for summary paths)
2) Join them as single summarypath
3) Do SAC algorithm

C. Service Route Calculation Algorithm

Algorithm 4: Service Route Calculation Algorithm

Input:

Summary path set of created routes(clusters) P

Service location set of service points S ,

Set of available units for each service points U

Output:

Service route for every route(cluster) from nearest service point

Algorithm:

1) Index routes (P) using modified R-Tree Indexing
2) for each service point s in S
3) nearest routes O=NNSearch (s,U[s])
4) for each object o in O find the shortest route from service point S and store in SR
5) Return Service routes SR

Here Nearest Neighbor Search in Modified R-Tree Indexing for finding service routes.

D. Modified R-Tree

Indexing is used to access data with minimum time that means it is used to speed up data retrieval. Sql indexes are used in databases for data retrieval. The indexes are created for data in the initial step. This index is used when a retrieval of data is required.

Indexes can also be applied to spatial objects. R-tree is the most widely used spatial indeR-tree is implemented in java as the methods described by Antonio Gutman [6]. Quadratic split method which uses quadratic time for the purpose is used as splitting algorithm for splitting overflow nodes. Quadratic split is based on the two objects which consumes higher MBR (Minimum Bounding Rectangle). Rest of elements are classified relative to them.

Modified R-Tree has neighboring entries in leaf nodes. For every insertion and deletion operation the neighboring entries are updated. The criteria for neighbor entries is their spatial location. So nearby elements becomes neighbors. The leaf nodes are arranged based on their insertion in every update which includes insertion and deletion. Every operation which is done M R-Tree must look for these properties. It is shown in fig4.

kNN (k Nearest neighbor) search is included as an operation on Modified R-Tree which retrieves k nearest elements from the query point.

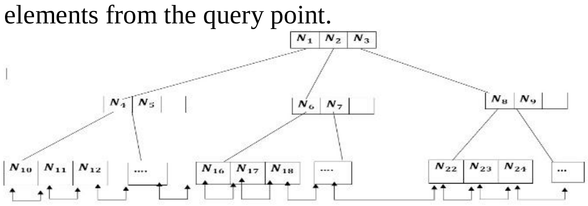

Fig. 4: Structural View of Modified R-Tree

IV. EVALUATION

Synthetic datasets are used for performance analysis. Clustering and allocation process are executed as proposed system. Spatial network consists 411 edges and 195 nodes. For anctivityset of 7 with 6 active nodes having a set of 27 shortest paths, 3 clusters/summary paths are calculated.

Table 1: Observations of Clustering

Number of clusters	2	3	4
R-Tree traversal time(s)	11.21	28.35	63.5

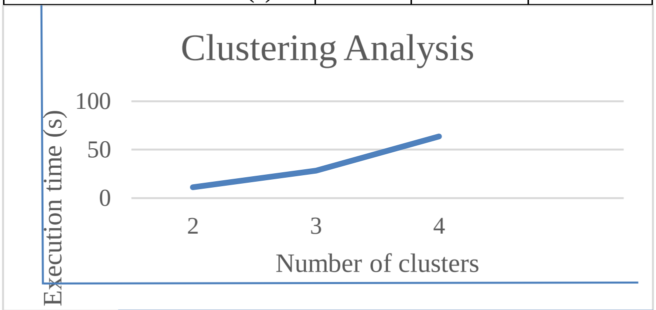

Fig 5 Execution time for Clustering, number of clusters * execution time (ns)

Performance analysis is based on indexing. The purpose of modified R-Tree is the faster access of leaf nodes. Traverse method on nodes is used for the comparison of R-Trees. Traversal time for the access of leaf nodes with clearing cache is shown in fig 6. This is the result obtained in the first attempt after restarting the datastructure. Cache doesn't have any previous data. A fresh cache is used.

Table 2: Observations of Performance of Indexes with cache clearance

Number of nodes	**R-Tree traversal time(ns)**	**Modified R-Tree traversal time(ns)**
2	919496	574743
4	1054318	654051
6	1188674	788873
8	1334225	974045
10	1226928	1037025

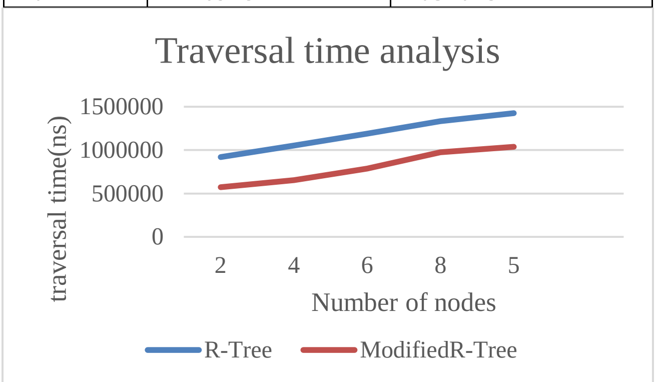

Fig 6 traversal time with clearing cache, number of nodes*execution time (ns)

Table 3: Observations of Performance of Indexes without cache clearance

Number of nodes	**R-Tree traversal time(ns)**	**Modified R-Tree traversal time(ns)**
2	440387	174942
4	578942	283173
6	865847	319561
8	965214	792138
10	975943	869579

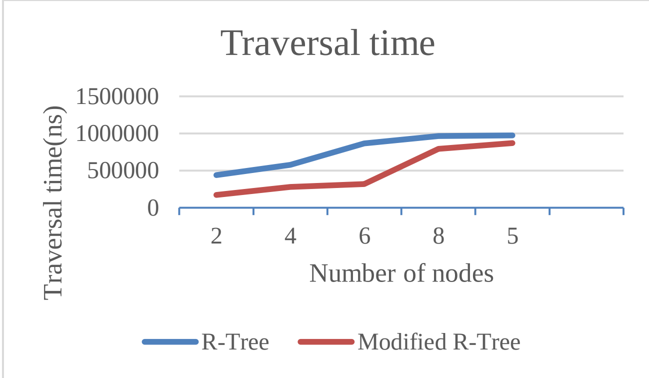

Fig 7 Traversal time without clearing cache, number of nodes*execution time (ns)

Traversal time for the access of leaf nodes without clearing cache is shown in fig 6. This output is obtained in the repeated attempts, not in the first attempt. System Cache contains previous results. The results clearly indicates the decrease in access time.

V. CONCLUSION

Spatial Activity Summarization and service route allocation problem are explored which have applications in real life scenarios like pedestrian fatalities and crime analysis. Proposed a Spatial Activity Clustering (SAC) algorithm that discovers a set of k shortest paths to group activities. SAC uses NOVA_TKDE (Network Voronoi activity Assignment), D-SPARE_TKDE (Divide and conquer Summary PAth REcomputation) and inactive node pruning techniques are incorporated in SAC to increase its and scalability and performance. SAC will yield substantial computational savings without reducing the coverage of the resulting summary paths. End Point Joining (EPJ) is included in the SAC to improve the efficiency. Service Route Allocation is the process of allocating service point to the calculated clusters and creating a route to these clusters from the service points. Service points can serve multiple clusters if they have multiple service units. Finding service routes to summary paths is implemented using k Nearest Neighbor (NN) Search in Modified R-Tree Indexing. Modified R-Tree Indexing extends R-Tree indexing by accommodating pointers in the leaf node to nearest leaf nodes.

Accessing spatial data is an important data. For that purpose indexing is used. By incorporating spatial locality to leaf nodes of R-Tree indexing, continuous access of spatial objects are taking minimum amount of time than usual R-Tree.

In future work, clustering of activities which does not belong to a point are considered (e.g., zip code level aggregated data of accidents). Use a distance-based rather than coverage-based objective function can be applied as an alternative plan. Edge weights can represent activities in the spatial network for the clustering is another method to investigate. There may be situations where we cannot define exact locations of activity, this might be useful for that kind of applications. Finding appropriate value of k can be a future work along with considering the time factor along with spatial factor. Service route allocation can be extended to create service routes for multiple clusters for a single unit in a service point. Heuristics can be used to improve the service management allocation.

References

[1] M. Ernst, M. Lang, and S. Davis. (2011). Dangerous by design: Solving the epidemic of preventable pedestrian deaths. Transportation for America: Surface Transportation Policy Partnership. Washington, DC, USA. [Online].Available:http://trid.trb.org/ view.aspx?id=114891

[2] R. Wronski. (2013, Apr. 9). Metra argues for delay of 'fail-safe' rail system. Chicago Tribune [Online]. Available: http://www. Chicago-tribune.com/news/ local/ct-met-metra-collision-prevention20130409,0, 3710438.story

[3] Dev Oliver, Shashi Shekhar, James M. Kang, Renee Laubscher, Veronica Carlan, and Abdussalam Bannur. A K-Main Routes Approach to Spatial Network Activity Summarization, IEEE Transactions on Knowledge and Data Engineering, vol. 26, No. 6, June 2014.

[4] Yonghun Park, Dongmin Seo, Jongtae Lim, Jinju Lee, Mikyoung Kim, Weiwei Bao, Christopher T. Ryu, Jaesoo Yoo. A New Spatial Index Structure for Efficient Query Processing in Location Based Services, 2010 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing.

[5] Peter Kieseberg, Sebastian Schrittwieser, Lorcan Morgan, Martin Mulazzani, Markus Huber, Edgar Weippl. Using the Structure of B+-Trees for Enhancing Logging Mechanisms of Databases. SBA Research,Vienna, Austria. Available: https://www.sba-research.org/wpcontent/ uploads/publications/ iiWAS-2011_133_Short_Kieseberg .pdf

[6] A.Gutmann, "R-trees: A dynamic index structure for spatial searching", In Proc. ACM SIGMOD intl. conf. Management of Data, 1984, pp. 599-609.

[7] Alexandr Andoni, 2009, Nearest Neighbor Search: the Old, the New, and the Impossible, Thesis, Massachusetts Institute of Technology. Available: http://www.mit.edu/~andoni/thesis/main.pdf

[8] J. MacQueen et al., "Some methods for classification and analysis of multivariate observations," in Proc. 5th Berkeley Symp. Math. Statist. Probab., vol. 1. Berkeley, CA, USA, 1967, p. 14.

[9] L. Kaufman and P. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis. Wiley Online Library, 1990.

[10] M. Resende and R.Werneck, "A hybrid heuristic for the p-median problem," J. Heuristics, vol. 10, no. 1, pp. 59–88, Jan. 2004.

[11] R. D'Andrade, "U-statistic hierarchical clustering" Psychometrika, vol. 43, no.1, pp. 59–67, Mar. 1978.

[12] M. Celik, S. Shekhar, B. George, J. Rogers, and J. Shine, "Discovering and quantifying mean streets: A summary of results," Univ. Minnesota, Minneapolis, MN, USA, Tech. Rep.07-025, 2007.

[13] K. Buchin et al., "Detecting hotspots in geographic networks," in Proc. Adv. GIScience, Berlin, Germany, 2009, pp. 217–231.

[14] P. Spooner, I. Lunt, A. Okabe, and S. Shiode, "Spatial analysis of roadside Acacia populations on a road network using the network K-function," Landscape Ecol., vol. 19, no. 5, pp. 491–499, Jul. 2004.