International Journals of Advanced Research in Computer Science and Software Engineering
ISSN: 2277-128X (Volume-8, Issue-5)

Research Article

May 2018

Route Searching using Modified k-Nearest Neighbor with Hill Climbing over Trajectories

Vineet Kumar Gupta[1], Prof. Sriram Yadav[2]

[1] Research Scholar, Department of Computer Science Engineering, MITS, Bhopal, India
[2]Assistant Professor & Head, Department of Computer Science Engineering, MITS, Bhopal, India

Abstract- Optimal planning for public transportation is one of the keys to sustainable development and better quality of life in urban areas. Based on mobility patterns, propose a localized transportation mode choice model, with which we can dynamically predict the bus travel demand for different bus routing. This model is then used for bus routing optimization which aims to convert as many people from private transportation to public transportation as possible given budget constraints on the bus route modification. It also leverages the model to identify region pairs with flawed bus routes, which are effectively optimized using our approach. To validate the effectiveness of the proposed methods, extensive studies are performed on real world data collected in Beijing which contains 19 million taxi trips and 10 million bus trips. GPS enables mobile devices to continuously provide new opportunities to improve our daily lives. For example, the data collected in applications created by Ola, Uber or Public Transport Authorities can be used to plan transportation routes, estimate capacities, and proactively identify low coverage areas. Now, study a new kind of query – Modified k-Nearest Neighbor Search with Hill Climbing (MkNNHC), which can be used for route planning and capacity estimation. Given a set of existing routes D_R, a set of passenger transitions D_T, and a query route Q, an MkNNHC query returns all transitions that take Q as one of its k nearest travel routes. To solve the problem, we first develop an index to handle dynamic trajectory updates, so that the most up-to-date transition data are available for answering an RkNNT query. Then introduce a filter refinement framework for processing MkNNHC queries using the proposed indexes. Experiments on real datasets demonstrate the efficiency and scalability of our approaches.

***Keywords:** GPS, Hill Climbing, Routing Optimization, MkNNHC, Trajectories, Efficiency & Scalability.*

I. INTRODUCTION

More and more people live in metropolitan areas or big cities due in large part to rapid development of urbanization. One major side effect is increasing traffic congestion due to limited space, and consequently unnecessary energy consumption during traffic congestion. Public transportation (e.g., bus, subway) not only saves fuel and reduces congestion, but also offers a safe, affordable, and convenient way to travel. According to American Public Transportation Association1, Americans living in areas served by public transportation save 865 million hours of travel time and 450 million gallons of fuel annually in congestion reduction alone. Households that are likely to use public transportation on a given day save more than $9,700 every year. Better public transportation planning can significantly help to foster a more sustainable development and improve quality of life. Traditional public transportation planning methods have relied on human surveys to understand people's mobility patterns and their choice among different transportation modes. Despite the substantial time and cost spent on the survey process, these methods do not provide timely and detailed analysis on different Origin-Destination (OD) pairs given the complex and dynamic settings of cities. As a result, such methods cannot accommodate real travel demands which vary over different OD pairs and often make planning inefficient in real life. If transit agencies could have an effective tool to quantify travel demand and a choice model on how people choose public transportation and private transportation (e.g., private car, taxi), then recommendations on how to better design and optimize a given public transportation network could be proposed to attract more people to public transportation.

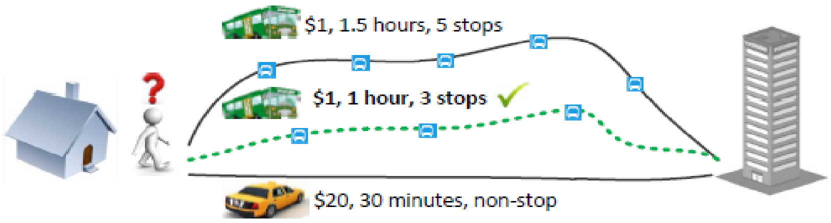


Figure 1: Attract more people to public transportation

As a result, cities would be able to better support people's travel demand through a regulated, efficient, more sustainable public transportation system Meanwhile, with the wide deployment of Automatic Fare Collection (AFC) systems on bus networks and Global Positioning System (GPS) devices on taxis, large amounts of bus transactions and taxi traces are collected, with which we can provide integrated analysis on the mobility patterns of both public and private transportation. As demonstrated in this paper, this offers the possibility of optimizing public transportation by taking overall city traffic into account. By leveraging mobility patterns of public and private transportation, public transportation services can be designed in a way that accommodates different levels of demands and by doing so, attracts more potential riders and increases utilization efficiency of the public transportation system. As shown intuitively in Figure 1, people may be more willing to choose bus over taxi when better bus routing is provided.

II. BACKGROUND

GPS enables mobile devices to continuously provide new opportunities to improve our daily lives. For example, the data collected in applications created by Uber or Public Transport Authorities can be used to plan transportation routes, estimate capacities, and proactively identify low coverage areas. In this paper, we study a new kind of query – Reverse k Nearest Neighbor Search over Trajectories (RkNNT), which can be used for route planning and capacity estimation. Experiments on real datasets demonstrate the efficiency and scalability of our approaches. To the best of our best knowledge, this is the first work to study the RkNNT problem for route planning. (Sheng Wang, Zhifeng Bao, J. Shane Culpepper, Timos Sellis, Gao Cong; 2017).

Optimal planning for public transportation is one of the keys to sustainable development and better quality of life in urban areas. This model is then used for bus routing optimization which aims to convert as many people from private transportation to public transportation as possible given budget constraints on the bus route modification. We also leverage the model to identify region pairs with flawed bus routes, which are effectively optimized using our approach. To validate the effectiveness of the proposed methods, extensive studies are performed on real world data collected in Beijing which contains 19 million taxi trips and 10 million bus trips. (Yanchi Liu, Chuanren Liu, Nicholas Jing Yuan, Lian Duan, Yanjie Fu, Hui Xiong, Songhua Xu and Junjie Wu, 2015).

The data mining is vast field with application found in many areas such that science, industrial problems and business. The data mining system architecture has the several main components database, data warehouse, or other information repositories, a server that extract the related data from repositories based on user request. The paper also uses improved Boyer Moore Horspool algorithm for evaluation of Enhanced pattern matching performance and lastly Association rules mining that finds interesting relations or associations relations between the item sets among massive amount of data. (Tahira Mahboob, Fatima Akhtar, Moquaddus Asif, Nitasha Siddique, Bushra Sikandar; 2015).

Given a set of users, a set of facilities and a query facility q, a reverse k nearest neighbors (RkNN) query returns every user u for which the query is one of its k closest facilities. RkNN queries have been extensively studied under a variety of settings and many sophisticated algorithms have been proposed to answer these queries. Furthermore, we also present a carefully developed filtering strategy that significantly improves TPL which is one of the most popular RkNN algorithms. Specifically, the optimized version is up to 20 times faster than the original version and reduces its I/O cost up to two times. (Shiyu Yang, Muhammad Aamir Cheema, Xuemin Lin, Wei Wang; 2015).

III. METHODOLOGY

The basic procedure of proposed methodology is as follows

```
Algorithm 1: MkNNHC (Q, $root_{r,}$ $root_t$)
Output: $S_{result}$          //the result set
Evaluate initial state.
Loop until a solution is found or there are no new operators left to be applied:
($S_{filter,}$ $S_{refine}$) = FilterRoute($root_r$, Q, k);
Select and apply a new operator
$S_{cnd}$ = PruneTransition($root_t$, Q, $S_{filter}$, k);
$S_{result}$ =  RefineCandidates(Q, $S_{cnd}$, $S_{refine}$);
Evaluate the new state:
If goal == found then
return $S_{result}$;
else
new current state <- better than current state
```

Algorithm 2: FilterRoute(root$_r$, Q, k)

Output:

```
S_filter        //filtering set
S_refine        // filtered node set for refinement
minheap = NULL, $S_{filter}$ = NULL, $S_{refine}$ = NULL;
minheap.push(root$_r$);
while minheap.isNotEmpty() do
    e=minheap.pop();
    if e is a node then
        if isFiltered(Q, $S_{filter}$, e, k) then
            $S_{refine.}$add(e);          // Filtered node set
            continue;
    if e is a point then
        $S_{filter.}$add(e, C(e)); // Filtering set
    else
        foreach child c of e do
            minheap.push(c, MinDist(Q, c));
return ($S_{filter,}$ $S_{refine}$);
```

Algorithm 3: IsFiltered(Q, S$_{filter}$, node, k)

Output: whether the node can be filtered

```
S <- NULL;
 For each p element of $S_{filter.}$P do
 // access list points in descending order
     if S.size > k then
         return true;
     label <-  true;
     foreach q element of Q do
         if node located in Hp:q then
             label <-  false;
     if label = true then
         S <-  S U C(p); // crossover route set
S'<- $S_{filter}$.R - S;
 foreach route belongs to S' do
     if S.size > k then
         return true;
     if VoronoiFiltering(Q, route, node) then
         S <-  S U {route};
return false;
```

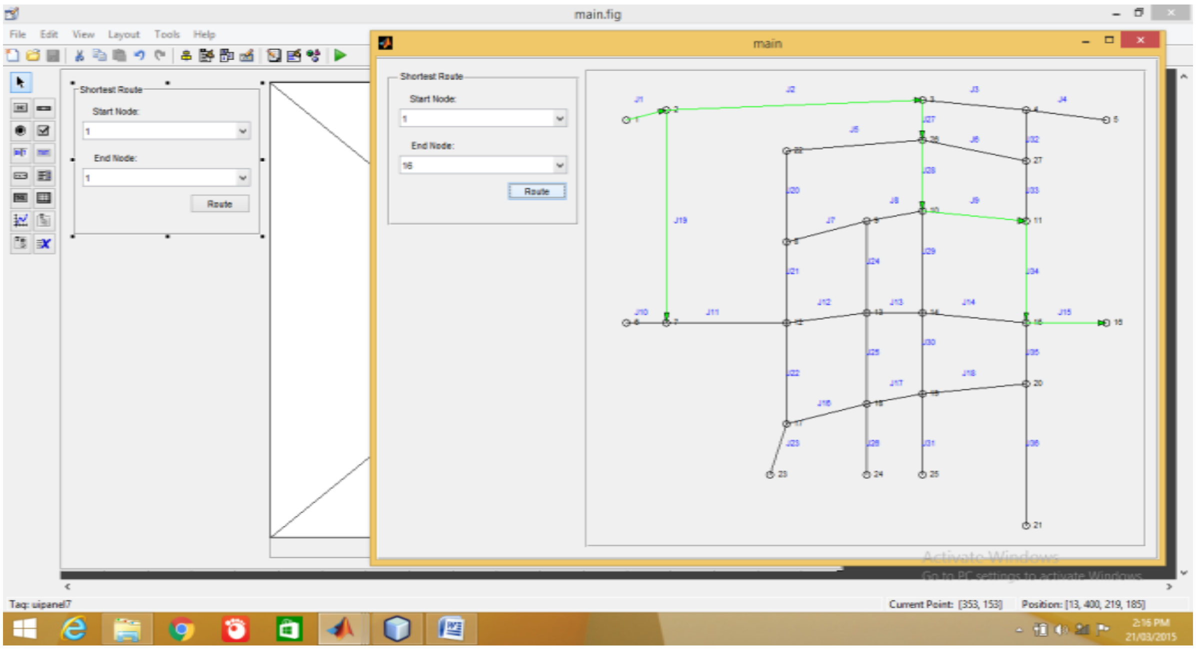
Algorithm 4: PruneTransition(roott, Q, S$_{filter}$, k)

Output: S$_{cnd}$: candidate set

```
minheap <- NULL, $S_{cnd}$ <- NULL;
minheap.push(root$_t$);
while !minheap.isEmpty() do
    e <- minheap.pop();
    if e is a Node then
        if isFiltered(Q, $S_{filter}$, e, k) then
            continue;
    if e is a point then
        $S_{cnd}$.add(e);
    else
        foreach child c of e do
            minheap.push(c, MinDist(c,Q));
return $S_{cnd}$;
```

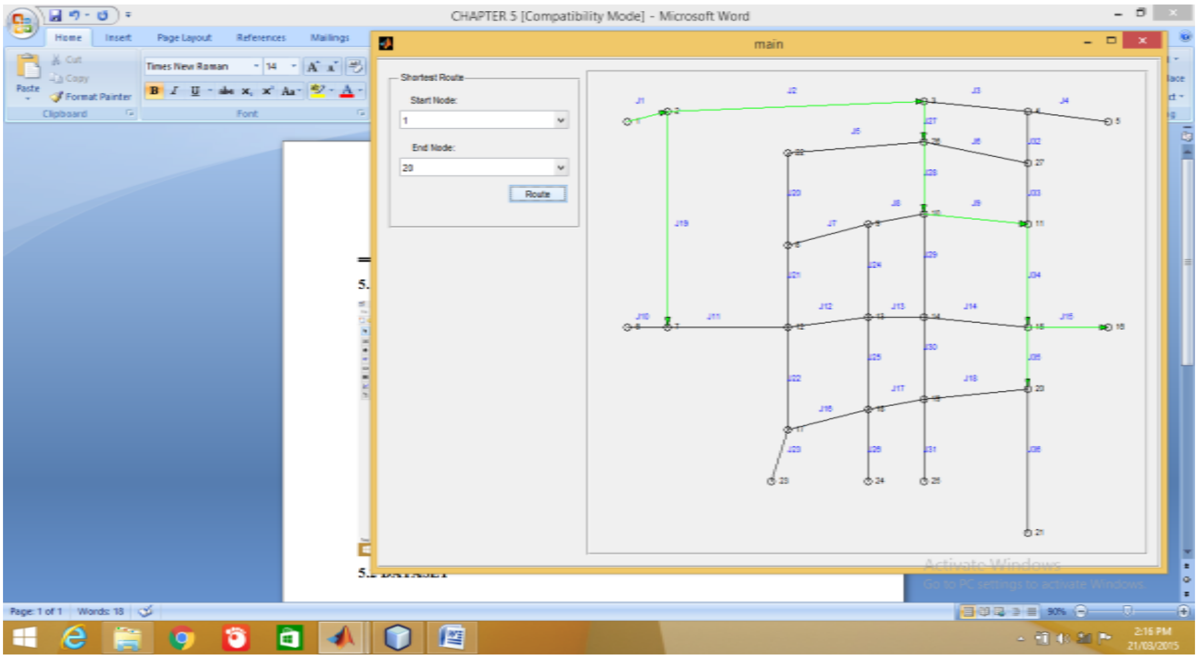
IV. RESULT AND ANALYSIS

In this chapter discuss experimental process of proposed Modified k-Nearest Neighbor Search with Hill Climbing (MkNNHC) technique for shortest route finding with optimize search. The proposed method implements in MATLAB R2013a and tested with very reputed data set from MTA (Metropolitan) UCI machine learning research center.


main.fig
File Edit View Layout Tools Help
main
Route
Tag: uipanel7
Current Point: [353, 153]
Position: [13, 400, 219, 185]
Activate Windows
Go to PC settings to activate Windows.
2:16 PM
21/03/2015

Shortest Route
Start Node: 1
End Node: 16

Figure 2: Implement route search on LA-Route


CHAPTER 5 [Compatibility Mode] - Microsoft Word
Home Insert Page Layout References Mailings
Paste Cut Copy Format Painter
Clipboard Font
Times New Roman 14
main
Route
Page: 1 of 1 Words: 18
90%
Activate Windows
Go to PC settings to activate Windows
2:16 PM
21/03/2015

Shortest Route
Start Node: 1
End Node: 20

Figure 3: Implement route search on NYC-Route

The two real bus network datasets, namely NYC-Route and LA-Route, it can be downloaded from http://web.mta.info/developers/developer-data-terms.html#data, http://developer.metro.net/gtfs/google_transit.zip respectively. We extracted the data from the GTFS datasets of New York and Los Angeles. Now implement all of the algorithms ourselves and the algorithms use common subroutines to do similar tasks. All algorithms are implemented in MATLAB and compiled to 32-bit binaries. All the experiments are run on Intel 2.66 GHz Core i3 CPU, 6GB main memory running Win 8 OS and using ST3450857SS hard disk (1TB 15K RPM SAS 6Gbps). For each algorithm, we also implemented some simple yet effective optimizations. Furthermore, after finishing our implementation, we obtained the codes from the authors to make sure that our implementation does not have any glitches that may negatively affect the algorithm. The detailed pseudo codes, the source code, data sets, and the scripts to run experiments and draw figures.

Table 1: Analysis of CPU Time in LA-Route

Interval (in sec)	RkNNT[1]	MkNNHC (Proposed)
10	57	42
20	68	61
30	86	72
40	112	87
50	142	121
60	164	139

Table 2: Analysis of CPU Time in NYC-Route

Interval (in sec)	RkNNT[1]	MkNNHC (Proposed)
10	52	39
20	78	58
30	98	64
40	131	98
50	172	114
60	211	187

In LA trajectory, CPU cost is reduce sequentially for proposed methodology as compare than CPU cost of RkNNT[1]. In a similar way, CPU cost is reduce sequentially for proposed methodology as compare than CPU cost of RkNNT[1] for given NYC-Route trajectory.

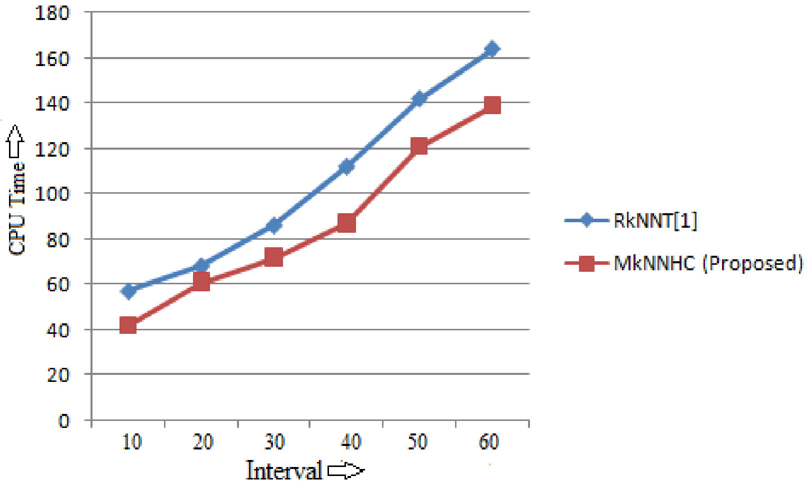


Figure 4: Analysis of CPU cost for LA-Route

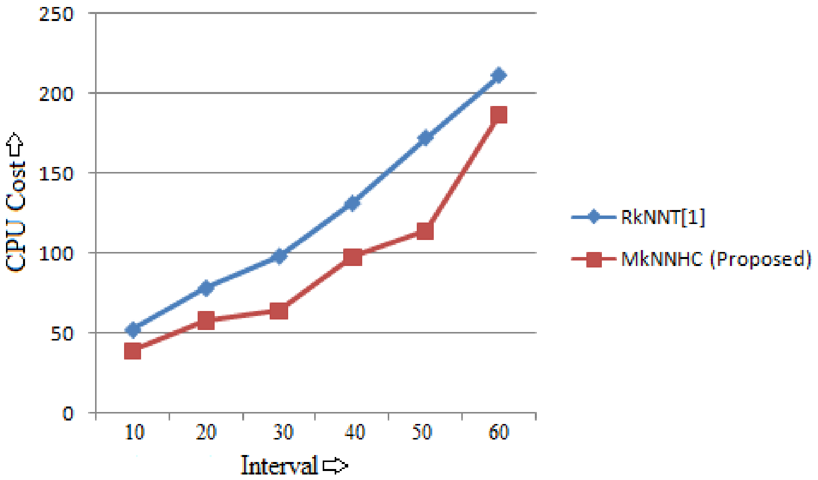


Figure 5: Analysis of CPU cost for NYC-Route

Now analyze the route distance as per running for LA-Route and NYC-Route trajectories

Table 3: Analysis of route distance in LA-Route

Running Time (in sec)	RkNNT[1]	MkNNHC (Proposed)
0	51	64
2	29	48
4	16	33
6	8	21
8	6	16
10	2	11

Table 4: Analysis of route distance in NYC-Route

Running Time (in sec)	RkNNT[1]	MkNNHC (Proposed)
0	243	287
2	122	148
4	53	89
6	41	72
8	23	52
10	11	28

In LA trajectory, route distance is take more for proposed methodology as compare than route distance of RkNNT[1]. In a similar way, route distance takes more sequentially for proposed methodology as compare than route distance of RkNNT[1] for given NYC-Route trajectory.

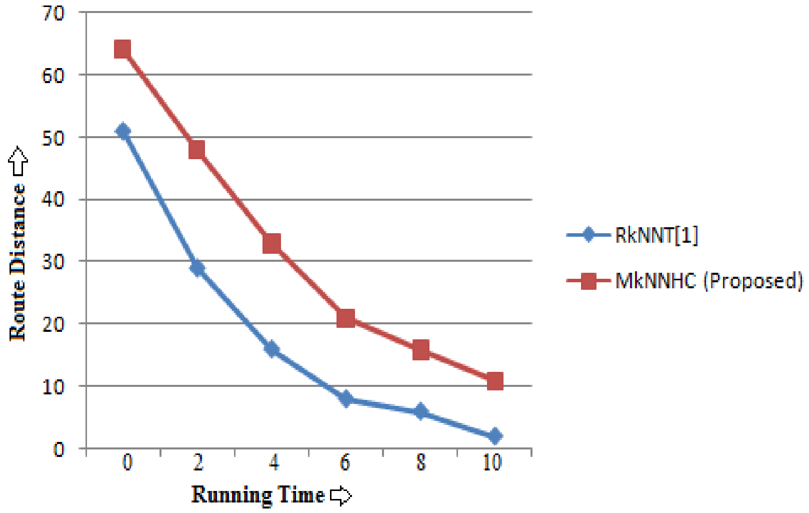


Figure 6: Analysis of Route Distance for LA-Route

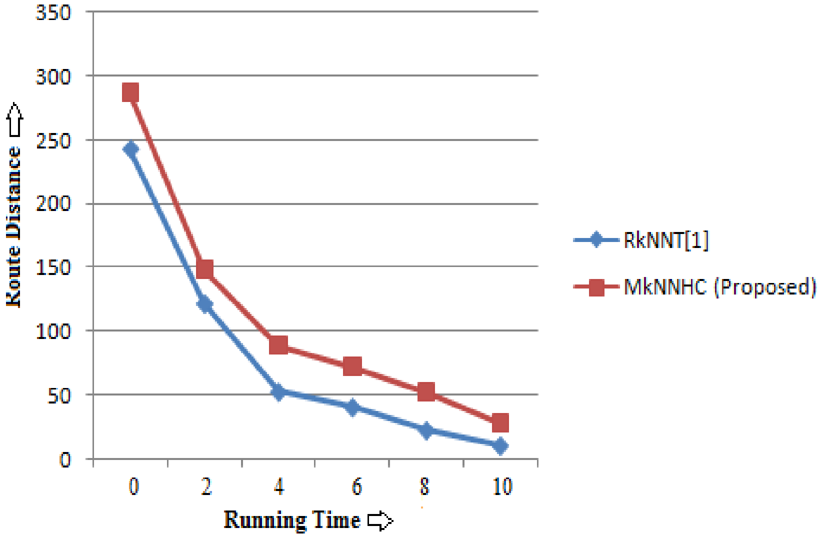


Figure 7: Analysis of Route Distance for NYC-Route

V. CONCLUSION

In this work, we proposed and studied the MkNNHC query, which can be used directly to support capacity estimation in bus networks. First, we proposed a modified hill climbing processing framework, and an optimization to increase the filtering space that improves pruning efficiency. Then we employed MkNNHC to solve the bus route planning problem. In a bus network, given a start and end bus stop, we can find an optimal route which attracts the most passengers for a given travel distance threshold. To the best of our knowledge, this is the first work studying reverse k nearest neighbors in trajectories, and our solution supports dynamically changing transition data while providing up-to-date answers efficiently. The identification and optimal planning of the flawed bus routes to improve the utilization efficiency of the public transportation service, according to the transportation mode choice model built on real data. First, we partitioned the urban area into disjoint regions on which an integrated analysis of the taxi traces and the bus transactions is conducted. Second, based on the integrated analysis, it proposed a localized transportation mode choice model, with which we can dynamically predict the bus travel demand for different bus routing. Then, we leveraged this model to optimize the bus routes by maximizing the bus ridership with budget constraints. At last, we provided a solution for the identified most flawed region pairs in the urban area. Our approach gives more effective result as compare than RkNNT. CPU cost is reduce by 19% significantly and route distance covers more by 23% significantly.

VI. FUTURE WORK

Extensive studies, which validated the effectiveness of our methods, were performed on real world data collected in Beijing which contains 19 million taxi trips and 10 million bus trips. Furthermore we can use various route filter for reduce the traffic.

REFERENCES

[1] Sheng Wang, Zhifeng Bao, J. Shane Culpepper, Timos Sellis, Gao Cong, “ Reverse k Nearest Neighbor Search over Trajectories”, IEEE Transaction on Data Engineering, 2017.

[2] Yanchi Liu, Chuanren Liu, Nicholas Jing Yuan, Lian Duan, Yanjie Fu, Hui Xiong, Songhua Xu and Junjie Wu, “Exploiting Heterogeneous Human Mobility Patterns for Intelligent Bus Routing”, IJAPTA 2015.

[3] Tahira Mahboob, Fatima Akhtar, Moquaddus Asif, Nitasha Siddique, Bushra Sikandar, “Survey and Analysis of Searching Algorithms”, IJCSI International Journal of Computer Science Issues, Volume 12, Issue 3, May 2015.

[4] Shiyu Yang, Muhammad Aamir Cheema, Xuemin Lin, Wei Wang, “Reverse k Nearest Neighbors Query Processing: Experiments and Analysis”, VLDB Endowment, Vol. 8, No. 5, 2015.

[5] Yunjun Gao, Baihua Zheng, Gencai Chen, Wang-Chien Lee, Ken C. K. Lee, Qing Li, “Visible Reverse *k*-Nearest Neighbor Queries”, ACM Journal of Engineering, 2014.

[6] Elio Masciari, Gao Shi, Carlo Zaniolo, “Sequential Pattern Mining from Trajectory Data”, ACM IDEAS ’13, October 09 – 11, 2013.

[7] Kan Jun-man, Zhang Yi, “Application of an Improved Ant Colony Optimization on Generalized Traveling Salesman Problem”, International Conference on Future Electrical Power and Energy Systems, 2012.

[8] Ugur Demiryurek, Farnoush Banaei-Kashani, Cyrus Shahabi, “Efficient Continuous Nearest Neighbor Query in Spatial Networks using Euclidean Restriction”, National Science Foundation, 2011.

[9] Dongming Zhao, Liang Luo, Kai Zhang, “An improved ant colony optimization for the communication network routing problem”, Elsivier Journal of Mathematical and Computer Modeling, 2010.

[10] Hans-Peter Kriegel, Peer Kroger, Matthias Renz, Andreas Zufle, Alexander Katzdobler, “Reverse k-Nearest Neighbor Search based on Aggregate Point Access Methods”, Int. Conf. on Scientific and Statistical Database Management, 2009.

[11] Zaiben Chen, Heng Tao Shen, Xiaofang Zhou, Jeffrey Xu Yu, “Monitoring Path Nearest Neighbor in Road Networks”, SIGMOD’09, June 29–July 2, 2009.

[12] Reza Sherkat, Davood Rafiei, “On Efficiently Searching Trajectories and Archival Data for Historical Similarities”, VLDB Endowment, ACM, 2008.

[13] Wei Wu, Fei Yang, CheeYong Chan, KianLee Tan, “FINCH: Evaluating Reverse kNearestNeighbor Queries on Location Data”, VLDB Endowment, ACM, 2008.

[14] Elias Frentzos, Kostas Gratsias, Nikos Pelekis, Yannis Theodoridis, “Algorithms for Nearest Neighbor Search on Moving Object Trajectories*”, IEEE Tran. on Data Engineering, 2008.