An Efficient Spatial Query Processing Algorithm in Multi-sink Directional Sensor Network

Zheng Ma[1,2], Jin Zheng[1], Weijia Jia[3], and Guojun Wang[1,4](✉)

[1] School of Information Science and Engineering, Central South University, Changsha 410083, Hunan, China
zhma@csu.edu.cn, zhengjin@mail.csu.edu.cn, csgjwang@gmail.com

[2] School of Information Science and Engineering, Hunan University, Changsha 410082, Hunan, China

[3] Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
jia-wj@cs.sjtu.edu.cn

[4] School of Computer Science and Educational Software, Guangzhou University, Guangzhou 510006, China

Abstract. In order to address the problem of energy- and time-efficient execution of spatial queries in directional sensor networks, an efficient hybrid spatial query processing algorithm called SQPDSN is proposed in this paper. In the majority of studies on query processing using wireless sensor networks, sensors are assumed to have an isotropic sensing and transmission model. However, in certain applications the sensors have directional sensing and directional transmission model. SQPDSN only requires each node within the query region send data message once which reduces the data messages. For achieving minimal energy consumption and minimal response time, our query processing model ensures that only the relevant nodes for the correct execution of a query are involved in the query execution. Each sector has a node which collects the sensory data in it, aggregates the data to derive partial query result and send it to the next sector. Compared with other techniques, the experimental results demonstrated an improvement of the proposed technique in terms of energy efficient query cover with lower communication cost.

Keywords: Multi-Sink wireless sensor networks · Directional transmission · Spatial query · Parallel processing

1 Introduction

In wireless sensor networks, senor nodes are deployed in a monitored region, which are capable of sensing, processing and storing environment information. It is used to query the data or events monitored or detected. A directional sensor network (DSN) is a collection of such directional sensor nodes spatially deployed in an ad hoc fashion that performs distributed sensing tasks in a collaborative manner. Unlike an omnidirectional

G. Wang et al. (Eds.): ICA3PP 2015 Workshops, LNCS 9532, pp. 13–23, 2015.
DOI: 10.1007/978-3-319-27161-3_2

sensor device, a directional one has limited range of communication and sensing capabilities as it can sense and communicate in only one direction or a certain angle.

The study of this paper is related to query-based routing protocol, so, next, we will introduce several routing protocols relevant to this paper. The tree-based approaches rely on network infrastructure (e.g., based on a spanning tree) for query propagation and processing. Such as, Tiny Aggregation (TAG) (Madden et al. 2002), the Dynamic Query-tree Energy Balancing (DQEB) (Yang et al. 2004) protocol, Gathering-Load-Balanced Tree Protocol (LBTP) (Chen et al. 2006), Semantic/Spatial correlation-aware tree (SCT) that exploits the correlation strategies was proposed by Zhu et al. (2008), the Workload-Aware Routing Tree (WART) algorithm (Andreou et al. 2011), the Energy-driven Tree Construction (ETC) algorithm (Andreou et al. 2011), and the Geometry-based Spatial Skyline Query (GSSky) (Zhang et al. 2014). These techniques rely on a network infrastructure for query propagation and processing. Some work (Tang and Xu 2006; Gnawali et al. 2009) aims to improve the query precision, besides extending the network lifetime. This centralized approach didn't generate an efficient query plan and resulted in high overhead as it requires that each node reports its metadata to the sink. Moreover the maintenance of such a network infrastructure (Chakraborty et al. 2011) is a major issue, especially when the sensor nodes failed.

The decentralized versions of spatial query execution programs are needed for sensor networks: by using a decentralized program instead of a centralized approach, it would be possible to contact only the relevant nodes for the execution of a spatial query, and hence the decentralized algorithms will incur less energy consumption (Demirbas and Ferhatosmanoglu 2003). It has been shown that, by avoiding the significant overhead of maintaining a network infrastructure, the decentralized spatial query processing techniques (Wu et al. 2007, 2008; Fu et al. 2007, 2010; Martina et al. 2014) outperform the infrastructure-based techniques. The performance (such as the query latency and the energy consumption) of itinerary-based spatial query processing techniques is dependent on the design of itineraries, such as Itinerary-Based KNN (IKNN), Density-aware Itinerary KNN query processing (DIKNN), Parallel Concentric-circle Itinerary-based KNN (PCIKNN), and Energy-Efficient and Fault Tolerant Spatial Query Processing (EEFT). With a long itinerary, long query latency and high energy consumption may be incurred due to a long itinerary traversal. On the other hand, allowing a query to run on an arbitrary number of short itineraries in parallel may result in significant collisions in the query dissemination phase.

In this paper we propose an efficient spatial query processing scheme (SQPDSN) by combining the strengths of both the infrastructure-based and the infrastructure-free query processing techniques to reduce communication collisions and alleviate the influence of nodes failure. Thus, our scheme can achieve energy-efficiency and time-efficiency.

The rest of the paper is organized as follows. We describe network model and assumptions in Sect. 2. In Sect. 3, the details of our propose SQPDSN system is presented. In Sect. 4, extensive experiments are conducted to evaluate the performance of the proposed algorithms, and we conclude our work in Sect. 5.

2 Network Model and Problem Definition

We consider a directional wireless sensor network, where sensor nodes are deployed in a two-dimensional space. Figure 1 is the example for architecture of WSN. In our assumptions, there are some pre-deployed sink nodes, which are powerful and location-aware. The sink nodes are not power constrained and stay on for the lifetime of the network. Thus, they do not have any effect on the power consumption characteristics of the network. Every sensor node can compute its sector ID according its own location and the location of every sink node via GPS or other localization techniques (Ma et al. 2011). By periodically exchanging beacon information with sensor nodes nearby, a sensor node maintains a list of neighbor nodes. Moreover, the sensed data are stored locally in sensor nodes.

We model the sector-based sensor network as a Graph G = (C, S, E), where C is the set of sectors, S is the set of sinks, and E represents the implicit network edges of the sectors in C. $E = E_1 + E_2$, $E_1 \subseteq C \times C$, $E_2 \subseteq C \times S$. In E1, if sector i can directly communicate with sector j, eij = 1, otherwise, eij = 0. And, we assume that if eij = 1 then eji = 1. In E2, if sector i can directly communicate with sink j, then eij = 1, otherwise, eij = 0. And every sink can directly communicate with every sector. There are k sink nodes, n sensor nodes and m sectors in a network. D(i) is the number of neighbor sector of sector i. N(i) is the number of nodes in sector i. τ is the time for sensor node message transmission. σ is the time for sensor node processing. In the real world, τ is much larger than σ, and σ can be ignored.

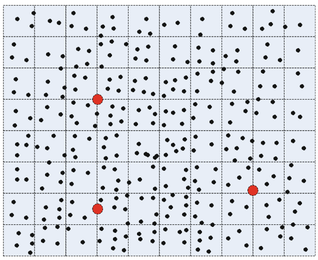

Fig. 1. Multi-sink directional sensor networks model

Each sensor has a unique sensing and communication model. These sensor nodes have a set of sensing and communication orientations or sectors. We can characterized a sensing or communication sector using the following attributes. Figure 2 is the sensing and communication model of a directional sensor node.

Rs is the maximum sensing radius and Rc is the maximum communication radius.

θs (0 < θs < 2π) is the maximum sensing angle and θc (0 < θs < 2π) is the maximum communication angle. $\underset{Vs}{\rightarrow}$ is a directional vector that divides the sensing range into two equal parts, $\underset{Vc}{\rightarrow}$ is a directional vector that divides the communication range into two equal parts.

Figure 2 is the sensing and communication model of a directional sensor node.

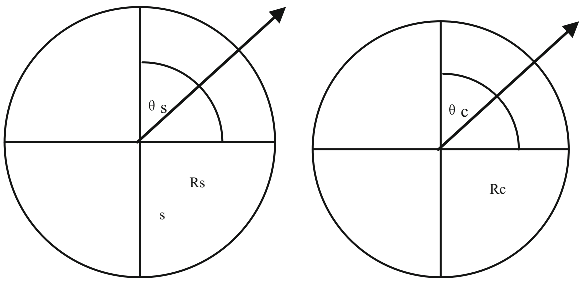

Fig. 2. Sensing and communication model of directional node

The spatial query SQL-based query language, which consists of a SELECT–FROM–WHERE clause, is commonly accepted and widely used in specifying queries for a sensor network. However, a sensor network has its own characteristics, and therefore extensions must be made to the basic SQL query. We provide some simple examples WSN query in the form of extended SQL: "*Select temp From sensors Where s.location is in R1 And Epoch Duration 20 s*" or "*Select average(temp) From sensors Where s.location is in R1 And Epoch Duration 20 s*". The former query requests the all temperature from all sensors inside the interest region, which is the row-data query. The latter one requests the average temperature from all sensors inside the interest region, which is the aggregated-data query. In this paper, we consider the latter kind.

Users of the system query the base station for data objects. For simplicity, we assume that each query asks for one data object acquired at a specific location area. We use query region to refer to the location specified in a query. A data object can be used to answer a query if the data object's spatial region covers the query region. The duration between the time the base station sending a query and the time the base station receiving the query answer is the query's response time.

Given a query Q over a sensor network, select an optimal set of sensors that satisfy the conditions of coverage as well as connectivity. The set of sensors are sufficient to answer the query such that (1) the sensing region of the selected set of sectors cover the entire geographical region of the query and (2) the selected set of sensors should form a logical routing topology for data gathering and transmission to the query source. In addition, (3) query processing except for time-constrained must incorporate energy awareness into the system to extend the lifetime of the sensor nodes and network by reducing the total energy consumption.

3 The SQPDSN Algorithm

In this section, we propose SQPDSN, a spatial query processing in directional sensor networks. The goal of SQPDSN is gathering regional aggregated values energy and time efficiently so that we can find some interest regions satisfying several conditions.

SQPDSN has two processing phases: the centralized initialization phase, the query dissemination and data collection phase, as in the prior work. In addition, SQPDSN employs several innovative ideas to improve the efficiency of spatial query result. Explicitly, the initial route, sector selection estimation, and the query dissemination are computed by the sink using its powerful capability and global knowledge. The route, the query dissemination, the data collection scheme are dynamically refined while the query is propagated using the local information. Moreover, to reduce redundant data transmission and improve the query accuracy, partial query results are collected at the some nodes and then sent back to the source node. The objective is to balance the energy consumption and time latency with the accuracy of query results constraint (Bai et al. 2012).

Next, we will elaborate the framework of SQPDSN. The framework of SQPDSN consists two phases: the centralized initialization phase, and the query dissemination and result back phase. In the former phase, the computation is done by sink nodes, and in the latter phase, the sensor nodes can obey the rule of the query processing and adjust according to the local information adaptively.

3.1 Centralized Initialization

The sink node acts as an initiator of the spatial query processing. The users register the spatial queries in one sink node. The sink node that received the query computes the cover sectors estimation, initial route, and the query dissemination and the data collection using the global information, which is based on the sector forest. The pseudo-code of sector forest construction is in Algorithm 1.

Algorithm 1. BFS Sector Forest Construction

Input: The wireless sensor networks graph *G* and a user query *Q*

Output: The sector forest *F*

Begin

Initial step:

Construct the set of object sectors *C'*, which cover the query region

For every sector in *C'*, set *label* =0

Add sink nodes *s* to *F*, every sink is the root of a tree *t*, the level of sink is 0. ψ is the estimated time cost of the tree *t*. Set ψ=0

Set $LEVEL = 0$

Loop step:

While a sector *label* ==0 in *C'* and new sector add in *F* do

For every sector *j* which its level value equals *LEVEL*

Add sector *i* to *F* and set its $level = LEVEL+1$ and $parent = j$ if it is not in *F* and is the neighbour of the sector *j*. If $i \in C'$, then set its $label = 1$, set $flag = 1$ along the path from *i* to the root until the flag value is 1 for any ancestral node of *i*

End For

$LEVEL= LEVEL+1$

End while

Critical Path Time Cost Estimation

Find the highest tree *t* in the *F* which flag value is 1

$\Psi= LEVEL*max(D(i))*2T$

If value of the ψ is larger than the epoch

Then return to user

Else send the query and the tree to the sink nodes.

End

Algorithm 1 presents the main step of this procedure from the sink nodes to the leaf sectors. The first step calculates the sector sets which cover the query region based on the global information of sector Graph, and initializes the sector forest. The second step is loop step aiming that all the object sectors are added to the forest, if the Graph G is a fully connected graph. The last step estimated the time cost for the query. If the time cost is less than the epoch, then the query task will be parallelized to related sink nodes. Otherwise, it returns to the user that the epoch is not enough and gives the user the reasonable epoch.

After the sector forest being constructed, every related sink node will receive the query and related tree. It finds the shortest route to the query region and computes its object sectors collection path, then distributes it to the child sector based on the input. Also, the sink then schedules the last end time slots to result back for every sub tree to ensure accomplishing task in epoch duration. If the value of child's flag is 1, it shows the branch has the data query sector. The pseudo-code of the algorithm is as Algorithm 2.

During the initial phase, the sink is in charge of the computation using the known global information. It only consumes the sink node's energy. The time complexity of Algorithm 2 is *O(log m)*, where m is the number of sectors. And, the query task can be parallelized into some sub query tasks according to the query requirements, aim to reduce the query processing time and balance the energy cost in networks.

Algorithm 2. Initial query dissemination

A sink node receiving a query message from other sink node

Input: Initial sink ID, tree of the current sink, the query

Output: Query messages (the initial query sector tree t, query id and information, last endtime e, sink ID)

```
Begin
 If flag ==0 then return
    Set c value is the number of child sector node which flag is 1
      For Find child i of the root
        If flag == 1
      Set endtime = endtime -c*τ
          c--
   Then send the query message to the sector i
End
```

3.2 Query Dissemination and Result Back

In this section, we describe the process how to send the query and collect the data in a network. When a node in a sector receives the query from the upper level sector, it schedules the last end time slots to result back for every child sector whose flag is 1, to ensure accomplishing task in the epoch duration, and diffuses the query to the down sectors, then waits. And if it covers part of query region, it will collect the data in its sector, then waits to fuse the part query results from the down sectors. The proposed algorithms are as shown in Algorithms 3 and 4. The back path and collect sector set will be merged in the result back message. The dynamical information will be brought to the sink nodes at last, which is benefit for whole sector structure maintenance. But in this paper, this part of work will not be described.

Algorithm 3. The query dissemination

A node in sector i receiving a query message from a node in upper sector j

Input: Query message (the query sector tree t1, query id and information, last end time e, sink ID, initial sink ID)

Output: Sub query messages (the sub query sector tree, endtime, query id and information, sink ID, initial sink ID)

```
Begin
  If sector i is the leaf node of t1
     If covers part of query region
       Then collects the data in sector i, and sends fused the query result message to the upper sector j using
certain sensing angle and sensing radius
     Else return
 If  flag ==0 Then return
    Else Set parent = j for query id
       Set endtime = e for query id
       Set c value is the number of child sector node which flag is 1
For every child k of the root i
  If flag == 1 Set endtime = endtime -c*τ c--
    Then sends the sub query message to the sector k for query id using certain sensing angle and sensing radius
End For
If i covers part of query region
    Then collects the data in sector i and fuses query result for query id
End
```

In the query dissemination phase, only the query information and query route are sent to the down sector. Now we consider the computation complexity in the query dissemination phase. There are m sectors in the query tree at most, and in each sector only one node transmits once. Therefore, it requires m transmission for query dissemination. The depth of the query tree is $log(m)$, and the scheduling time for one level of the tree is $max(D(i))$, so the time of the query dissemination is $O((log\ m)*max(D(i)))$.

Algorithm 4. The query result back

A node in sector i receiving a result message from a node in down sector k

Input: Result message (query result, back path, collect sector sets, query id, sink ID)

Output: Fused result message (fused query result, back path, query id, sink ID)

```
Begin
  Fuses the query result, back path, and collect sets from sector k for query id
    If all child have been send result message or the time is ending
      Then sends fused the query result message to the upper sector j using certain sensing angle and sensing
radius
  Else wait
End
```

In the query result back phase, only the query result and the back route are sent to the upper sector. The time of the result back is $O((log\ m)*max(D(i)))$, and it requires *m* result transmission too. So for a query, the cost of transmission is the $O(m + n)$, and the time cost is $O((\log m)*max(D(i)) + max(N(i)))$.

SQPDSN is an energy-balanced and delay-tolerated algorithm. During the initialization phase, the query has been partitioned so that multiple sinks can process it in parallel. There is no fixed head node for a sector, and every node is selected taking joint responsibility for query dissemination, data collection and query back (Wang et al. 2014). It can avoid that some nodes are dying fast in the hot space.

SQPDSN is fault tolerant due to several aspects. First, the sector structure in a wireless sensor networks will not be changed for a long time, so the sector query forest is stable. Second, node failure is masked without causing any update operation and structure change. For a dense sensor network, each sector contains several nodes and all nodes in the same sector share a common Sector ID. They can act on behalf of other nodes in the same sector. Third, SQPDSN can handle coverage holes nicely. Only if all the nodes inside a sector fail, a hole may be formed in SQPDSN. Even if the hole is formed, the node can re-route to the next sector to avoid the hole through local computation (Ma et al. 2011). Then it follows the query dissemination algorithm or data co0llection algorithm.

4 Performance Evaluation

The effects of node density and the size of the sensor network field (scalability) on query performance are investigated by changing the number nodes and/or the field size. And the proposed algorithm is evaluated by computer-based simulations compared with the GSSky, and EEFT based on the different density and different node failure rate. We compare these three algorithms in terms of lifetime, the average time cost, and accuracy rate.

The performance is evaluated based on NS2. The lifetime of network is the duration from the network initialization to the time 1/10 nodes are exhausted. The average time cost is the average time between the time a query is issued and the time the query result is returned to the sink node. The time unit is second. The accuracy rate is the ratio of the successful queries to the total queries. In our default settings, sink nodes and sensor nodes are static and location-aware. The max transmission range of sensor node is 10 units and the message delay for transmission is 30 ms. Every sensor node has the same initial energy, and energy consumption for message transmission is the same. In each round of the experiment, a series of queries are issued from randomly selected sink nodes in turn.

Table 1 shows the number of nodes on some size of sensor nodes to compare the proposed scheme with GSSky and EEFT. Every node has the same initial energy with the same max transmission range (10 units).

Table 1. Network size and number of nodes for test

Size of area(Unit)	Number of nodes			
225*225	1000	2000	3000	5000

Then, we investigate the impact of network density on the performance of the three examined algorithms. Figure 3 shows that the lifetime of network become longer when the network density increases in the three algorithms. Specifically SQPDSN has the longest life time. We can find that SQPDSN has the least latency for query and the highest accuracy in the three algorithms. In SQPDSN, the global information based on a sector is more stable, and thus the initial query plan is more proper. And SQPDSN decreases the number of messages for performing query dissemination in parallel, in order to save time. From the experimental results observed above, SQPDSN has the best performance under the experimental settings.

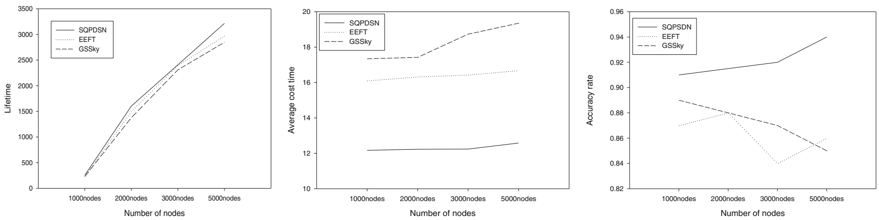

Fig. 3. Comparison with different density

Then, we investigate the impact of node failures on the performance of the three examined algorithms. The node failure rate of sensors unquestionably affects the performance of spatial query processing, especially to the infrastructure-based techniques. The network size is 225 * 225 units and the number of nodes is 3000. The node failure rate varies from 0 to 0.8. Performance study of these three algorithms is shown in Fig. 4. It shows that the life time of network becomes shorter when the node failure rate is increased in three algorithms. But SQPDSN has the longest life time among others because during the dissemination and result back phase, nodes can dynamically adjust the route to send the query and result. And the GSSky has the shortest life time. We can find that SQPDSN has the least latency for query and the accuracy is also highest in the three algorithms. The reason is that SQPDSN, which is not effected by nodes failure, is more fault tolerance.

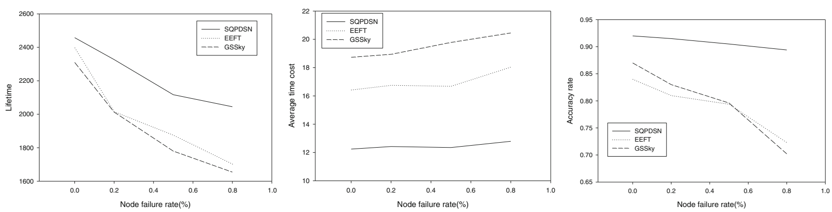

Fig. 4. Comparison with different node failure rate

5 Conclusion and Future Work

In this paper, we propose an efficient hybrid spatial query processing algorithm, SQPDSN, for spatial query processing in multi-sink directional sensor networks. This algorithm routes through the data query process and fully considers the characteristics of the data-query applications. The basic idea is to disseminate a query and collect data with pre-designed schedule and dynamic optimization. Initially, the SQPDSN is able to determine an optimal processing schedule for spatial query approximately using the backbone infrastructure. To deal with the node failure or local change during the query dissemination and data collection phase, we develop methods to bypass voids and to dynamically adjust the route. Experimental results show that SQPDSN significantly outperforms the existing query processing techniques in terms of energy consumption, query latency, query accuracy and scalability.

In future works, we intend to consider failures due to node mobility and also plan to extend for multiple queries other than snapshot queries. Finally, we intend to evaluate the proposed mechanism on a real test-bed.

References

Andreou, P., Zeinalipour-Yazti, D., Pamboris, A., Chrysanthis, P.K., Samaras, G.: Optimized query routing trees for wireless sensor networks. Inf. Syst. **36**(2), 267–291 (2011)

Bai, S., Zhang, W., Xue, G., Tang, J., Wang, C.: DEAR: delay-bounded energy-constrained adaptive routing in wireless sensor networks. In: 2012 Proceedings of IEEE International Conference on Computer Communications (INFOCOM), pp. 1593−1601 (2012)

Chakraborty, S., Chakraborty, S., Nandi, S., Karmakar, S.: A tree-based local repairing approach for increasing lifetime of query driven WSN. In: Proceedings of 14th IEEE International Conference on Computational Science and Engineering (CSE), pp. 475–482 (2011)

Chen, T.-S., Tsai, H.-W., Chu, C.-P.: Gathering-load-balanced tree protocol for wireless sensor networks. Proc. IEEE Int. Conf. Sens. Netw. Ubiquit. Trust-Worthy Comput. **2**, 8–13 (2006)

Demirbas, M., Ferhatosmanoglu, H.: Peer-to-Peer spatial queries in sensor networks. In: Proceeding of the Third International Conference on Peer-to-Peer Computing (P2P), pp. 32–39 (2003)

Fu, T-Y., Peng, W-C., Lee, W-C.: Optimizing parallel itineraries for KNN query processing in wireless sensor networks. In: Proceedings of ACM Conference on Information and Knowledge Management (CIKM), pp. 391−400 (2007)

Gnawali, O., Fonseca, R., Jamieson, K., Moss, D., Levis, P.: Collection tree protocol. In: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys), pp. 1−14 (2009)

Ma, Z., Jia, W., Wang, G.: Routing with virtual region coordinates in wireless sensor networks. In: Proceedings of 10th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 1657−1661 (2011)

Madden, S., Franklin, M.J., Hellerstein, J., Hong, W.: TAG: a tiny aggregation service for ad-hoc sensor networks. In: Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI), pp. 131–146 (2002)

Pushpam, P., Felix, E.: Energy-efficient and fault tolerant spatial query processing in wireless sensor networks. Proc. Adv. Commun. Control Comput. Technol. (ICACCCT) **43**(5), 790–794 (2014)

Tang, X., Xu, J.: Extending network lifetime for precision-constrained data aggregation in wireless sensor networks. In: Proceedings of 25th IEEE International Conference on Computer Communications (INFOCOM), pp. 1−12 (2006)

Wang, G., Wang, T., Jia, W., Guo, M., Li, J.: Adaptive location updates for mobile sinks in wireless sensor networks. J. Supercomput. **47**(2), 127–145 (2009)

Wang, G., Bhuiyan, M.Z.A., Cao, J., Wu, J.: Detecting movements of a target using face tracking in wireless sensor networks. IEEE Trans. Parallel Distrib. Syst. **25**(4), 939–949 (2014)

Wu, S-H., Chuang, K-T., Chen, C-M., Chen, M-s.: DIKNN: an itinerary-based KNN query processing algorithm for mobile sensor networks. In: Proceedings of 23rd IEEE International Conference on Data Engineering (ICDE), pp. 456–457 (2007)

Wu, S.-H., Chuang, K.-T., Chen, C.-M., Chen, M-s: Toward the optimal itinerary-based KNN query processing in mobile sensor networks. IEEE Trans. Knowl. Data Eng. **20**(12), 1655–1668 (2008)

Yang, H., Ye, F., Sikdar, B.: A dynamic query-tree energy balancing protocol for sensor networks. Proc. Wireless Commun. Networking Conf. WCNC **3**, 1715–1720 (2004)

Zhang, L., Wang, Y., Song, B.: Geometry-based spatial skyline query in wireless sensor network. In: Proceedings of Web Information System and Application Conference, WISA, pp. 27−32 (2014)

Zhu, Y., Vedantham, R., Park, S.-J., Sivakumar, R.: A Scalable Correlation Aware Aggregation Strategy for Wireless Sensor Networks. Inf. Fusion **9**(3), 364–369 (2008)

http://www.springer.com/978-3-319-27160-6

Algorithms and Architectures for Parallel Processing
ICA3PP International Workshops and Symposiums,
Zhangjiajie, China, November 18-20, 2015, Proceedings
Wang, G.; Zomaya, A.; Martinez Perez, G.; Li, K. (Eds.)
2015, LIII, 851 p. 327 illus. in color., Softcover
ISBN: 978-3-319-27160-6