

# Energy-Efficient Reverse Skyline Query Processing over Wireless Sensor Networks

Guoren Wang, Junchang Xin, Lei Chen, and Yunhao Liu, *Senior Member, IEEE*

**Abstract**—Reverse skyline query plays an important role in many sensing applications, such as environmental monitoring, habitat monitoring, and battlefield monitoring. Due to the limited power supplies of wireless sensor nodes, the existing centralized approaches, which do not consider energy efficiency, cannot be directly applied to the distributed sensor environment. In this paper, we investigate how to process reverse skyline queries energy efficiently in wireless sensor networks. Initially, we theoretically analyzed the properties of reverse skyline query and proposed a skyband-based approach to tackle the problem of reverse skyline query answering over wireless sensor networks. Then, an energy-efficient approach is proposed to minimize the communication cost among sensor nodes of evaluating range reverse skyline query. Moreover, optimization mechanisms to improve the performance of multiple reverse skylines are also discussed. Extensive experiments on both real-world data and synthetic data have demonstrated the efficiency and effectiveness of our proposed approaches with various experimental settings.

**Index Terms**—Reverse skyline, wireless sensor network, query processing, multiple queries optimization.

## 1 INTRODUCTION

As an important and popular query operator for multiple criteria decision making, skyline [1] and its variants, such as constrained skyline [2], dynamic skyline [3], [4], [5], and reverse skyline [5], [6], [7], have been applied in many applications. Given a data set  $P$ , a *traditional skyline* retrieves all the points in  $P$  that are not dominated by others. A point  $p_1$  dominates another point  $p_2$ , if  $p_1$  is not worse than  $p_2$  for each dimension  $i$  (i.e.,  $p_1[i] \leq p_2[i]$ ), and  $p_1$  is better than  $p_2$  for at least one dimension  $j$  (i.e.,  $p_1[j] < p_2[j]$ ). Fig. 1a shows an example of traditional skyline in 2D space and points  $p_1$ ,  $p_2$ , and  $p_6$  are the skyline points.

In addition to traditional skyline, *dynamic skyline* [3], [4], [5] with respect to  $q$  (denoted as  $DS(q, P)$ ) has been proposed to retrieve all the points that are not dynamically dominated by others. For the sake of simplicity, we adopt the definition of dynamic attributes proposed in [5], [6]. A point  $p_1$  dynamically dominates  $p_2$  with respect to  $q$  (denoted as  $p_1 \preceq_q p_2$ ), if it holds that: 1)  $|p_1[i] - q[i]| \leq |p_2[i] - q[i]|$ , for each dimension  $i \in D$ , and 2) there exists at least one dimension  $j \in D$ , such that  $|p_1[j] - q[j]| < |p_2[j] - q[j]|$ . In Fig. 1a, points  $p_1$ ,  $p_3$ ,  $p_4$ , and  $p_5$  are the dynamic

skyline points of  $q$ . For each point  $p_i = (p_i[1], p_i[2])$  in 2D space, it is transformed to  $p_i' = (|p_i[1] - q[1]|, |p_i[2] - q[2]|)$  where query is  $q = (q[1], q[2])$ . Another topic of interest is reverse skyline [5], [6], [7]. Given a data set  $P$ , a *reverse skyline* with respect to  $q$  (denoted as  $RS(q, P)$ ) retrieves all the points  $p \in P$  such that  $q \in DS(p, P)$ . As illustrated in Fig. 1,  $q$  belongs to the dynamic skyline of  $p_1$  (Fig. 1b), so  $p_1$  is a reverse skyline point of  $q$ . For the same reason, points  $p_3$  and  $p_5$  are also reverse skyline points of  $q$  (Fig. 1c).

Recently it is found that wireless sensor networks (WSNs) offer a very economic and effective platform to monitor the environment. To satisfy different application demands, we conduct various types of queries over WSNs, for example min, max [8], top-k [9], [10], and skyline [11], [12]. The de facto limit to processing queries in WSNs is the energy constraint, since sensor nodes are generally battery powered, and in many WSNs (e.g., an unattended and hard-to-reach environment), it is impossible or at least very difficult to change their batteries. Wireless communication is the major energy consumer in WSNs, and therefore, if we can reduce the amount of communication during query processing, the energy consumption can be significantly reduced and the lifetime of the WSNs as a whole can be prolonged.

Each type of query has its unique properties, only after understanding the properties of all kinds of queries we can handle the energy-efficient processing of multiple types of queries. Thus, almost all existing work has focused on reducing the communication cost for a specific type of query [8], [9], [10], [11], [12]. As one of the important operators in WSN applications, the energy-efficiency of RS query processing also needs to be studied in depth. Therefore, in this work, we study energy-efficient approaches to answer reverse skyline (RS) queries over WSNs.

RS is very useful for environmental monitoring applications. We have deployed a real sensor system to monitor the forest environment. By consulting the experienced forestry

- G. Wang and J. Xin are with College of Information Science and Engineering, Northeastern University, Wenhua Road 11-3, Heping District, Shenyang, Liaoning 110819, China. E-mail: wanggr@mail.neu.edu.cn, xinjunchang@ise.neu.edu.cn.
- L. Chen is with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China. E-mail: leichen@cse.ust.hk.
- Y. Liu is with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology and the School of Software, Tsinghua University, Beijing 110084, China. E-mail: yunhaoliu@greenorbs.com.

Manuscript received 28 Apr. 2010; revised 7 Sept. 2010; accepted 22 Nov. 2010; published online 7 Mar. 2011.

Recommended for acceptance by B. Cui.

For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2010-04-0243. Digital Object Identifier no. 10.1109/TKDE.2011.64.

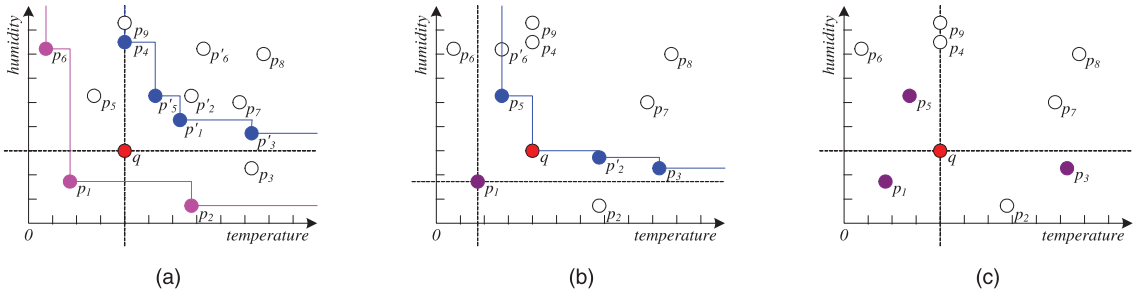


Fig. 1. Examples of Skylines. (a)  $DS(q, P_1)$ . (b)  $DS(p_1, P_1)$ . (c)  $RS(q, P_1)$ .

workers and experts, for certain dangerous patterns, we can get the critical values of the environment parameters. With these critical values, we can monitor the potential dangers may happen in the sensor networks. That is, the closer the values collected by the sensors to these critical values, the higher probability that a danger may happen in the area monitored by those sensors. In fact, we can treat each of these critical values as a “signature” of a potential danger and use these critical values as the query points, the regions where the query points are in the skyline of their sensor readings can be identified as high-risk areas. Based on the results of the RS query, further steps can be taken to minimize the risks and costs.

Another example that demonstrates the usefulness of the reverse skyline can be found, when studying the behavior of various bird species, ornithologists utilize feeder sensors to determine the places where birds appear most frequently. The sensor reading (temperature, humidity, etc.) collected by WSN reflects the preferences of the birds. If a bird prefers a place, it may prefer the places of the dynamic skyline of the place. If a new location for breeding birds is sought, the results of reverse skyline querying can be employed to identify suitable areas.

Unfortunately, implementing the reverse skyline operator in WSNs is nontrivial since RS is not *decomposable* (observation 1 in Section 3). This means that we cannot simply distribute tasks (i.e., RS queries) to subsets of data points and then compute the union of the returned RS answers, since the union is not the RS answer over all data points. Thus, existing techniques such as in-network aggregation (e.g., TAG [8]) cannot be used in WSNs to answer RS queries. In order to tackle this problem, in this paper, we propose a novel concept, called *semidominance*, which uses half distance between a query and a data point to define the dominance relationship. Based on this concept, we define *full dominance* and in turn *full skyline*, which can convert the RS problem into a full skyline problem. Interestingly, we prove that the full skyline query is decomposable. As a result, we can utilize in-network processing to reduce communication costs. Furthermore, we define a *full skyband* and use it to remove false positives during in-network processing. Several variants of RS, such as range RS and multiple RS are studied by adapting the proposed RS solutions.

In summary, we make the following contributions in this paper.

1. We propose an energy-efficient approach to evaluate reverse skyline query in WSNs, based on the

*full skyband* which contains all necessary information for the base station to reconstruct the reverse skyline. Then, the transmission of nonfull skyline points is avoided by pointing out which full skyline points belong to the reverse skyline. Furthermore, the proposed approaches are also extended to support range reverse skyline and multiple reverse skyline queries.

2. We theoretically analyze the relationship between reverse skylines on different dimensional spaces or query ranges, and propose two optimization mechanisms, vertical and horizontal optimizations, to improve the performance of multiple reverse skyline evaluation in WSNs.
3. Lastly, our extensive experimental studies using both real-world data and synthetic data show that the proposed approach can significantly reduce the communication cost among sensor nodes and save the energy consumption during the evaluation of RS queries in WSNs.

The rest of the paper is organized as follows: Section 2 briefly reviews the related work. The details of traditional and range reverse skyline query evaluation are introduced in Section 3 and Section 4, respectively. Section 5 discusses how to evaluate multiple reverse skyline queries energy efficiently. The experimental results to show the effectiveness of the proposed approaches are reported in Section 6. Finally, Section 7 concludes this paper and discusses the future work.

## 2 RELATED WORK

In the WSN literature, energy-efficient query processing technique has received considerable attention in the past few years. Madden et al. [8] proposed TAG, a typical in-network aggregation approach for WSNs. TAG first organizes all sensor nodes into a routing tree structure rooted at the base station. When a user query arrives, it sends the query along the routing tree to all sensor node, and then collects the data from the leaf nodes to the root. While processing the query, each intermediate node has to wait for messages from all its children before computing and transmitting the partial aggregation up the network. Chen et al. [11] addressed the problem of continuous skyline monitoring in WSNs and presented a hierarchical threshold-based approach, MINMAX, to minimize the transmission traffic. Xin et al. [13] investigated the problem of maintaining sliding windows skylines over WSNs, in which several types of filters are installed within each sensor to reduce the amount of data transmitted among

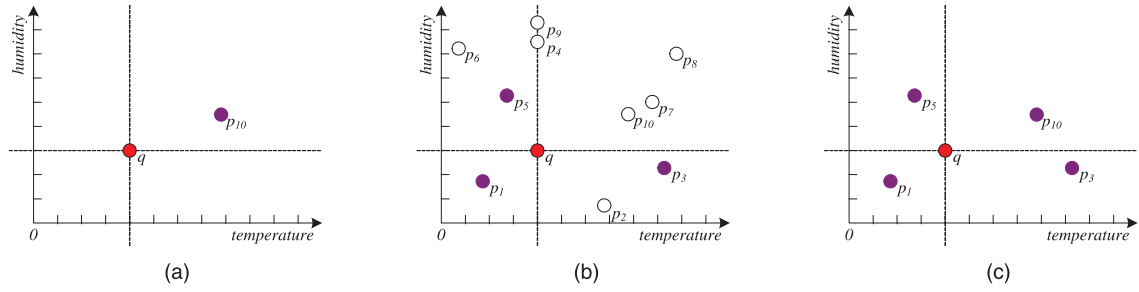


Fig. 2. Observations 1. (a)  $RS(q, P_2)$ . (b)  $RS(q, P_1 \cup P_2)$ . (c)  $RS(q, RS(q, P_1) \cup RS(q, P_2))$ .

sensor nodes. Liang et al. [14] proposed distributed algorithms for skyline evaluation and maintenance in WSNs. The proposed algorithms are based on a new concept, local skyline certificate, which has been used to find a global filter. Recently, Xin et al. [12] proposed an Energy-Efficient Multi-Skyline Evaluation algorithm (EMSE) to evaluate multiple skyline queries effectively in WSNs. EMSE utilizes both global and local optimization mechanisms to eliminate unnecessary data transmission. However, the existing works usually focus on decomposable query processing (aggregation, skyline etc.), whereas ours concentrates on the undecomposable reverse skyline query processing in WSNs.

The concept of dynamic skyline, skyline in the mapped data space, was first introduced by Papadias et al. [3], and the dynamic attributes of each mapped point were defined by several dimension functions. Deng et al. [4] studied the multisource skyline query in road networks, in which the dynamic attributes of each mapped point were defined as the relative network distances to multiple query points. Dellis and Seeger [5] considered the case that all the mapped attributes are the absolute distances to the query point. Based on dynamic skyline, Dellis and Seeger [5] proposed the reverse skyline, retrieving points whose dynamic skyline sets contain the given query point. Furthermore, they proposed a novel concept, called global skyline, which is an upper bound of the actual reverse skyline set. The proposed branch-and-bound algorithm firstly computes the global skyline as the set of candidate reverse skyline points, and then runs a window query for each global skyline point to verify whether it is a reverse skyline point or not. Lian and Chen [6] modeled the probabilistic reverse skyline query on uncertain data, in both monochromatic and bichromatic cases, and proposed effective pruning methods to reduce the search space of query processing. Wu et al. [7] investigated the bichromatic reverse skyline on precise data and proposed several nontrivial heuristics that can optimize the access order of the R-tree to reduce the I/O cost considerably. The reverse skyline processing techniques above are all proposed in a centralized scenario. To our best knowledge, there is no prior work to address reverse skyline queries over WSNs. In this paper, we explore how to energy efficiently evaluate reverse skyline queries in WSNs.

### 3 REVERSE SKYLINE QUERY PROCESSING OVER WSNs

A naive approach for answering reverse skyline queries in WSNs would be a centralized one. That is, all the sensor nodes in the WSN transmit the sensed data to the base

station, and then the base station computes the reverse skyline. Obviously, this centralized approach is the most “expensive,” since it needs to transmit all the sensed data back to the base station while the reverse skyline is only a small subset of the sensed data. As we know, if a query is decomposable, in-network aggregation techniques (e.g., TAG [8]) can be used to improve the performance [13]. Formally, we say a query operator  $op$  is *decomposable* if  $op$  has the following properties [15]:

**Definition 1.** An operator  $op$  is decomposable, if it can be computed by another operator  $g$  as follows:  $op(\bigcup_{i=1}^n D_i) = g(\bigcup_{i=1}^n op(D_i))$ .

For example, skyline and top- $k$  queries are decomposable query operators. As shown in Fig. 3, the top-2 results of  $D_1$ ,  $D_2$  and  $D_3$  are  $\{7, 3\}$ ,  $\{8, 6\}$ , and  $\{9, 5\}$ , respectively. Obviously, the top-2 result of  $\bigcup_{i=1}^n D_i$  is  $\{9, 8\}$ , and the top-2 result of  $\{7, 3\} \cup \{8, 6\} \cup \{9, 5\}$  is also  $\{9, 8\}$ . They are equal, so the top-2 query is decomposable.

Unfortunately, an RS query is not decomposable. Consider the example in Fig. 2, where the data set involved in Fig. 1a is marked as  $P_1$ , and that in Fig. 2a is  $P_2$ . Since  $p_{10}$  is the only point in  $P_2$ , the reverse skyline of  $q$  in  $P_2$  is  $\{p_{10}\}$ . The reverse skylines in data set  $P_1 \cup P_2$  are given by  $\{p_1, p_3, p_5\}$  (shown in Fig. 2b), whereas that in the union of reverse skyline sets  $RS(q, P_1)$  and  $RS(q, P_2)$  is  $\{p_1, p_3, p_5, p_{10}\}$  (shown in Fig. 2c). We can see that they are not equal. Thus, we have the following observation.

**Observation 1.** If  $P = P_1 \cup P_2 \cup \dots \cup P_n$ , then  $RS(q, P) \neq RS(q, \bigcup_{i=1}^n RS(q, P_i))$ .

In order to enable efficient RS query processing over WSNs, in this section, we first define two novel concepts, *semidominance* and *full dominance*, between two data points. Then, based on *full dominance*, we further define *full skyline*, which is proven to be a decomposable operator. Interestingly, it can be proven that all the points in the full skyline form a superset of the reverse skyline set. Therefore, we can conduct a full skyline query over a WSN, and derive the RS from the returned full skyline.

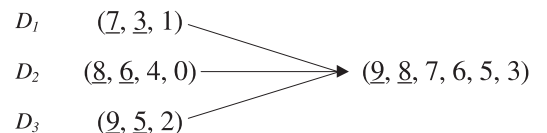


Fig. 3. Decomposability of top-2 query.

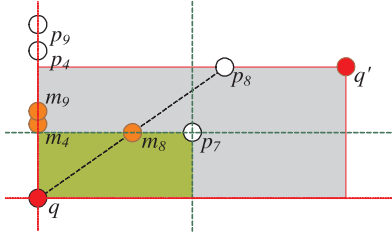


Fig. 4. Semidominance.

Nonetheless, it is annoying that we cannot completely distinguish the actual RS answers from non-RS answers in the retrieved full skyline set. To eliminate such an obstacle (i.e., to filter out all RS false positives), we carefully identify the reasons for RS false positives in the full skyline set, and propose a novel approach that obtains a (decomposable) *full skyband* set, instead of full skyline (which is a special case of full skyband, i.e., 1-full-skyband). We prove that by returning 2-full-skyband, it is enough to return all the RS answer yet distinguishing RS from non-RS answers in the 2-full-skyband set. This way, we can successfully convert our problem of searching undecomposable RS answers into an operator that finds decomposable (2-full-skyband) operators over WSNs.

### 3.1 Semidominance

In this section, we propose a novel concept, namely *semi-dominance*, for the purpose of adapting our RS problem to the WSN environment (discussed later). We use the example depicted in Fig. 4 to illustrate this concept. In particular, we denote  $q$  as a query point, and other data points (i.e.,  $p_4, p_7, p_8$  and  $p_9$ ) are to the top-right quadrant of  $q$ . Let points  $m_4, m_8$ , and  $m_9$  be the midpoints of line segments  $qp_4, qp_8$ , and  $qp_9$ , respectively.

Since  $m_8$  is located in the rectangle (including its borders) with  $q$  and  $p_7$  as its diagonal corner points, it can be proven that point  $p_8$  dynamically dominates  $q$  with respect to  $p_7$ . In other words,  $p_7$  cannot be a reverse skyline point of  $q$ . Point  $p_7$  should be a reverse skyline point only when there is no other point's midpoint contained in the rectangle.

The phenomenon that the midpoint (e.g.,  $m_8$ ) between a data point ( $p_8$ ) and a query point  $q$  is located in a rectangle with diagonal points from another point (e.g.,  $p_7$ ) and  $q$ , we will call *semidominance*.

Specifically, we say a value  $v_1$  is *semi-no-worse* than  $v_2$  with respect to  $o$  (denoted as  $v_1 \leq_o v_2$ ), if it holds that  $(v_1 - o) \cdot (v_2 - o) \geq 0$  and  $|v_1 - o| \leq 2|v_2 - o|$ ; the value  $v_1$  is said to be *semibetter* than  $v_2$  with respect to  $o$  (denoted as  $v_1 <_o v_2$ ), if it holds that  $(v_1 - o) \cdot (v_2 - o) > 0$  and  $|v_1 - o| < 2|v_2 - o|$ . We give the formal definition of *semidominance*<sup>1</sup> as follows:

**Definition 2 (Semidominance).** A point  $p_1$  *semidominates* another point  $p_2$  with respect to  $q$  (denoted as  $p_1 \leq_q p_2$ ), if it

1. The major difference with [5] is that we use " $\leq$ " while they use " $<$ ". The introduction of " $=$ " leads the necessary and sufficient condition fulfilled, so we can use it to prune non-RS points and return RS points directly. While only the necessary condition in the Lemma 2 of [5] can help to prune non-RS points. If we need RS points, a further verification step has to be taken. For example, we cannot drop  $p_7$  using Lemma 2 in [5], since there is no point satisfying the condition. A DS query is needed to verify whether  $q$  is in the DS of  $p_7$ . Compared to [5], we improve the retrieval efficiency of RS results.

holds that: 1)  $p_1[i] \leq_{q[i]} p_2[i]$ , for all dimensions  $i \in D$ , and 2) there exists at least one dimension  $j \in D$ , such that  $p_1[j] <_{q[j]} p_2[j]$ .

In the example of Fig. 4, we know that  $p_8$  semidominates  $p_7$ . The following theorem shows that we can make use of *semidominance* to conduct reverse skyline queries.

**Theorem 1.** Given a data set  $P$ , any point  $p \in P$  is a reverse skyline point of  $q$ , iff there does not exist any other point  $p' \in P$  such that  $p' \leq_q p$ .

**Proof.** 1. We prove the sufficient condition using reduction to absurdity. Assume  $p \notin RS(q, P)$ , that is  $q \notin DS(p, P)$ , then we can infer  $\exists p' \in P, p' \leq_p q$ . Therefore,  $\forall i \in D, |p'[i] - p[i]| \leq |q[i] - p[i]|$ , and  $\exists j \in D, |p'[j] - p[j]| < |q[j] - p[j]|$ .

$$\begin{aligned} |p'[i] - p[i]| \leq |q[i] - p[i]| &\Rightarrow (q[i] - p[i])^2 - (p'[i] - p[i])^2 \geq 0 \\ &\Rightarrow (q[i] - p'[i]) \cdot (q[i] + p'[i] - 2p[i]) \geq 0 \\ &\Rightarrow (q[i] - p'[i]) \cdot (2q[i] - 2p[i] - q[i] + p'[i]) \geq 0 \\ &\Rightarrow 2(q[i] - p'[i]) \cdot (q[i] - p[i]) \geq (q[i] - p'[i])^2 \end{aligned}$$

Thus,  $\forall i \in D, (p'[i] - q[i]) \cdot (p[i] - q[i]) \geq 0$ .

According to the triangle inequality, we can get  $|p'[i] - q[i]| \leq |q[i] - p[i]| + |p'[i] - p[i]|$ . Since it holds that  $|p'[i] - p[i]| \leq |q[i] - p[i]|$ , we can get  $|p'[i] - q[i]| \leq 2|p[i] - q[i]|$ .

Similarly,  $|p'[j] - q[j]| \leq |q[j] - p[j]| + |p'[j] - p[j]|$ . Since  $|p'[j] - p[j]| < |q[j] - p[j]|$  holds, we can infer  $|p'[j] - q[j]| < 2|p[j] - q[j]|$ .

Moreover, we have  $|p'[j] - p[j]| < |q[j] - p[j]| \Rightarrow p'[j] \neq q[j]$ . Thus,  $|p'[j] - q[j]| > 0$  and  $|p[j] - q[j]| > 0$ .

According to Definition 2, we have  $p' \leq_q p$ . This contradicts with  $\nexists p', p' \leq_q p$ . Therefore, the sufficient condition must be held.

2. We prove the necessary condition also using reduction to absurdity. Assume  $\exists p' \in P, p' \leq_q p$ , according to Definition 2, we can infer  $\forall i \in D, p'[i] \leq_{q[i]} p[i]$ , and  $\exists j \in D, p'[j] <_{q[j]} p[j]$ .

Since we have  $p'[i] \leq_{q[i]} p[i]$ , we can infer  $(p'[i] - q[i]) \cdot (p[i] - q[i]) \geq 0$ , and  $|p'[i] - q[i]| \leq 2|p[i] - q[i]|$ , then we can get  $(p'[i] - q[i])^2 \leq 2(p'[i] - q[i])(p[i] - q[i])$ .

Moreover,  $(p'[i] - p[i])^2 = (p'[i] - q[i] - p[i] + q[i])^2 = (p'[i] - q[i])^2 + (p[i] - q[i])^2 - 2(p'[i] - q[i])(p[i] - q[i])$ .

So, we can infer  $(p'[i] - p[i])^2 \leq (p[i] - q[i])^2$ .

Therefore,  $|p'[i] - p[i]| \leq |q[i] - p[i]|$ .

Similarly, since  $p'[j] <_{q[j]} p[j]$ , we can infer  $|p'[j] - p[j]| < |q[j] - p[j]|$ .

Thus, we can conclude that  $p' \leq_p q$  and  $q \notin DS(p, P)$ , therefore,  $p \notin RS(q, P)$ . This conflicts with condition  $p \in RS(q, P)$ , therefore  $\nexists p' \in P, p' \leq_q p$ .  $\square$

According to Theorem 1, we can rewrite the definition of reverse skyline, and obtain its equivalent semidominance based definition below.

**Definition 3 (Reverse Skyline).** Given a data set  $P$  and a query point  $q$ , a reverse skyline query (denoted as  $RS(q, P)$ ) retrieves all the points in  $P$  that are not semidominated by others with respect to  $q$ .



### 3.2 Full Dominance and Full Skyline

Giving rise to the undecomposability of reverse skyline query, the semidominance relation does not have transitivity. Thus, we cannot make use of in-network aggregation techniques. In order to tackle this serious problem over WSNs, our goal is to carefully and properly adjust the relation of semidominance, such that the modified relation becomes transitive (in turn, the query becomes decomposable). After that, we are able to take the advantage of in-network aggregation techniques to efficiently get the query results. This way, the base station can calculate reverse skyline without introducing false dismissals.

In order to achieve the goal above, we introduce another novel concept, namely *full dominance*, between any two data points. Specifically, we say that a value  $v_1$  is *full-no-worse* than another one  $v_2$  with respect to  $o$  (denoted as  $v_1 \leq_o v_2$ ), if it holds that  $(v_1 - o) \cdot (v_2 - o) \geq 0$  and  $|v_1 - o| \leq |v_2 - o|$ ; the value  $v_1$  is said to be *full better* than  $v_2$  with respect to  $o$  (denoted as  $v_1 \prec_o v_2$ ), if it holds that  $(v_1 - o) \cdot (v_2 - o) > 0$  and  $|v_1 - o| < |v_2 - o|$ . We give the *full dominance* definition below.

**Definition 4 (Full Dominance).** A point  $p_1$  fully dominates another point  $p_2$  with respect to  $q$  (denoted as  $p_1 \prec_q p_2$ ), if it holds that: 1)  $p_1[i] \leq_{q[i]} p_2[i]$ , for all dimensions  $i \in D$ , and 2) there exists at least one dimension  $j \in D$ , such that  $p_1[j] \prec_{q[j]} p_2[j]$ .

From the definition of full dominance above, we next propose a novel skyline type, called *full skyline*,<sup>2</sup> which is useful for retrieving RS answers over WSNs.

**Definition 5 (Full Skyline).** Given a data set  $P$  and a query point  $q$ , a *full skyline query* (denoted as  $FS(q, P)$ ) retrieves all the points in  $P$  that are not full-dominated by others with respect to  $q$ .

Fig. 5 illustrates a simple example of the full skyline query over data set  $P_1$  in a 2D temperature-and-humidity space. We can see that point  $p_7$  full-dominates  $p_8$ , since  $p_7$  has smaller distance to  $q$  than  $p_8$  on both dimensions. Similarly, point  $p_4$  full-dominates  $p_9$ , and point  $p_5$  full-dominates  $p_6$ . Since points  $p_1 - p_5$  and  $p_7$  are not full-dominated by any other points, we call them *full skyline points*.

Interestingly, the full-dominance relation is transitive, and the resulting full skyline query is also decomposable. Below, Lemma 1 and Theorem 2, respectively, guarantee the correctness of these two assertions.

**Lemma 1.** If  $p_1 \prec_q p_2$  and  $p_2 \prec_q p_3$ , then  $p_1 \prec_q p_3$ .

2. The detailed definition of global skyline in [5] is shown as follows: a point  $p_1$  globally dominates another point  $p_2$  with respect to  $q$ , if it holds that: 1)  $\forall i \in D, (p_1[i] - q[i]) \cdot (p_2[i] - q[i]) > 0$ , 2)  $\forall i \in D, |p_1[i] - q[i]| \leq |p_2[i] - q[i]|$ , and 3)  $\exists j \in D, |p_1[j] - q[j]| < |p_2[j] - q[j]|$ . The global skyline of a point  $q$  contains those points which are not globally dominated by another point with respect to  $q$ .

The global dominance requires  $\forall i \in D, (p_1[i] - q[i]) \cdot (p_2[i] - q[i]) > 0$ , while full dominance only requires  $\forall i \in D, (p_1[i] - q[i]) \cdot (p_2[i] - q[i]) \geq 0$  and  $\exists j \in D, (p_1[j] - q[j]) \cdot (p_2[j] - q[j]) > 0$ . Thus, full skyline is a subset of global skyline. In contrast to the full skylines in Fig. 5, global skylines not only contain all full skyline points ( $p_1 - p_5$  and  $p_7$ ), but also contain the nonfull skyline point  $p_9$  which is not globally dominated by  $p_4$ .

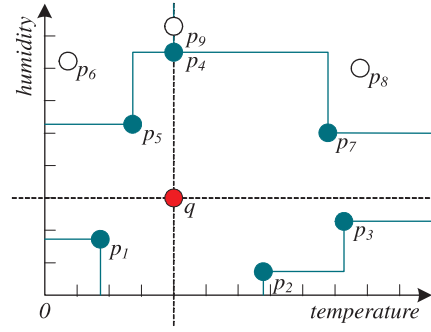


Fig. 5.  $FS(q, P_1)$ .

**Proof.** For detailed proofs see Appendix, which can be found on the Computer Society Digital Library at <http://doi.ieeecomputersociety.org/10.1109/2011.64>.  $\square$

**Theorem 2.** If

$$P = P_1 \cup P_2 \cup \dots \cup P_n, FS(q, P) = FS(q, \bigcup_{i=1}^n FS(q, P_i)).$$

**Proof.** 1. We prove  $FS(q, P) \subseteq FS(q, \bigcup_{i=1}^n FS(q, P_i))$  using reduction to absurdity. Assume  $p \notin FS(q, \bigcup_{i=1}^n FS(q, P_i))$ , then  $\exists p' \in \bigcup_{i=1}^n FS(q, P_i)$ ,  $p' \prec_q p$ . Since  $\bigcup_{i=1}^n FS(q, P_i) \subset P$ , we can infer  $p' \in P$ . Therefore,  $p \notin FS(q, P)$ .

2. We prove  $FS(q, \bigcup_{i=1}^n FS(q, P_i)) \subseteq FS(q, P)$  also using reduction to absurdity. Assume  $p \notin FS(q, P)$ , then according to Lemma 1,  $\exists p' \in FS(q, P)$ ,  $p' \prec_q p$ . Since  $p' \in FS(q, P)$ , we can infer  $p' \in \bigcup_{i=1}^n FS(q, P_i)$ . Therefore,  $p \notin FS(q, \bigcup_{i=1}^n FS(q, P_i))$ .  $\square$

Note that, since full dominance has a tighter constraint than semidominance, point  $p_1$  full-dominating point  $p_2$  can imply that  $p_1$  semidominates point  $p_2$ . Inversely, if a point  $p_2$  is not semidominated by  $p_1$ , then  $p_2$  is not full-dominated by  $p_1$ . Therefore, the set of reverse skyline points is a subset of full skylines, which can be summarized in the following theorem.

**Theorem 3.**  $RS(q, P) \subseteq FS(q, P)$ .

**Proof.** We prove the theorem using reduction to absurdity.

Assume  $p \notin FS(q, P)$ , then  $\exists p' \in P$ ,  $p' \prec_q p$ . Obviously,  $p' \prec_q p \Rightarrow p' \prec_q p$ . Thus,  $p \notin RS(q, P)$ .

Therefore,  $RS(q, P) \subseteq FS(q, P)$ .  $\square$

Theorem 3 indicates that the two sets  $RS(q, P)$  and  $FS(q, P)$  have containment relationship.

However, if we conduct a reverse skyline query in the full skyline set  $FS(q, P)$ , the resulting set of RS answers is not exactly  $RS(q, P)$ . This is summarized by Observation 2 below.

**Observation 2.**  $RS(q, P) \neq RS(q, FS(q, P))$ .

As an example in Fig. 6, we have a number of reverse skylines obtained from the full skyline of  $P_1$  (i.e.,  $RS(q, FS(q, P))$ ). In contrast to reverse skylines in Fig. 1c (i.e.,  $RS(q, P)$ ), we can see that not only data points in  $RS(q, P)$ , but also points like  $p_4$  and  $p_7$  are included in  $RS(q, FS(q, P))$ . We give a theorem below, indicating that

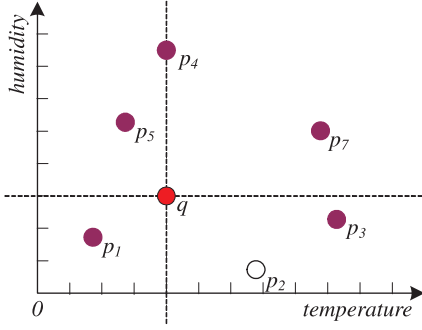


Fig. 6.  $RS(q, FS(q, P_1))$ .

sets  $RS(q, P)$  and  $RS(q, FS(q, P))$  have the containment relationships.

**Theorem 4.**  $RS(q, P) \subseteq RS(q, FS(q, P))$ .

**Proof.** Let  $p \in RS(q, P)$ , then according to Theorem 3,  $p \in FS(q, P)$ .

Since we have  $p \in RS(q, P)$ , we can infer  $\nexists p' \in P, p' \prec_q p$ .

Moreover, since  $FS(q, P) \subseteq P$ , we can get  $\nexists p' \in FS(q, P), p' \prec_q p$ . Therefore,  $p \in RS(q, FS(q, P))$ .  $\square$

From Theorem 4, we observe that  $RS(q, FS(q, P))$  is a superset of  $RS(q, P)$ , which means that the set  $RS(q, FS(q, P))$  does not miss any points belonging to the RS set. On the other hand, the set  $RS(q, FS(q, P))$  does contain points not belonging to the set  $RS(q, P)$ . These points cannot be automatically identified. To overcome this problem, we introduce the Full Skyband operator below.

### 3.3 Full Skyband

In order to return the exact RS answer set over WSNs, we now analyze the reason behind the fact that false positives exist in  $RS(q, FS(q, P))$ . We provide the key factor in the following lemma/theorem:

**Lemma 2.** If  $p_1 \prec_q p_2$  and  $p_2 \prec_q p_3$ , then  $p_1 \prec_q p_3$ .

**Proof.** For detailed proofs see Appendix, available in the online supplemental material.  $\square$

**Theorem 5.** Let  $p$  be a point not in  $RS(q, P)$  but in  $RS(q, FS(q, P))$ . Then, there must exist another point  $p' \notin FS(q, P)$  such that  $p \prec_q p'$  and  $p' \prec_q p$ .

**Proof.** Since we have  $p \notin RS(q, P)$ , we can infer  $\exists p' \in P, p' \prec_q p$ .

Moreover, since we have  $p \in RS(q, FS(q, P))$ , we can infer  $p' \notin FS(q, P)$ .

Since we have  $p' \notin FS(q, P)$ , we can also infer  $\exists p'' \in FS(q, P), p'' \prec_q p'$ .

Assume  $p'' \neq p$ , then according to Lemma 2, based on  $p'' \prec_q p'$  and  $p' \prec_q p$ , we can infer  $p'' \prec_q p$ . According to Theorem 1,  $p \notin RS(q, FS(q, P))$ . It contradicts with the condition, so we can conclude that  $p'' = p$  holds. Therefore,  $p \prec_q p'$ .  $\square$

For the sake of brevity, we call those points fully dominated by  $p$  "full-dominated points of  $p$ ," and  $p$  is the "full-dominating point" of its full-dominated points. Note

that, some full skyline points are only semidominated by their full-dominated points, and such full-dominated points have not been included in the full skyline set. As a result, the full skyline points above cannot be pruned during reverse skyline processing using the full skyline results (i.e.,  $RS(q, FS(q, P))$ ). As shown in Fig. 5,  $p_7$  is the full-dominating point of  $p_8$ , and  $p_8$  is the full-dominated point of  $p_7$ . Although point  $p_8$  semidominates  $p_7$ ,  $p_7$  is still a reverse skyline point of  $FS(q, P_1)$ , because of the absence of  $p_8$  in  $FS(q, P_1)$ .

In order to eliminate false positives in the full skyline set, we make the following observation: if we can send those data points that are fully dominated by at most one point to the base station, the problem of false positives in the set  $RS(q, FS(q, P))$  can still be easily solved. Inspired by this observation, we propose the concept of *full skyband* (Definition 6), which is similar to the skyband concept in [3] but with a different dominance semantic. By retrieving full skybands over WSNs, we can obtain the exact RS answers at the base station.

We firstly give the definition of full skyband below.

**Definition 6 (Full Skyband).** Given a data set  $P$  and a query point  $q$ , a  $k$ -full-skyband query (denoted as  $FSB^k(q, P)$ ) retrieves all points in  $P$  that are full-dominated by at most  $(k - 1)$  points with respect to  $q$ .

In the example of Fig. 5, the 2-full-skyband includes not only all full skyline points, but also the points full-dominated by exactly one point (i.e.,  $p_6, p_8$ , and  $p_9$ ). Thus, all reverse skyline points and full skyline points are included in the 2-full-skyband.

**Theorem 6.**  $RS(q, P) = RS(q, FSB^2(q, P))$ .

**Proof.** 1. Let  $p \in RS(q, P)$ , according to Theorem 3, we can get  $p \in FS(q, P)$ . Therefore,  $p \in FSB^2(q, P)$ .

Based on  $FSB^2(q, P) \subseteq P$  and  $q \in DS(p, P)$ , we can infer  $q \in DS(p, FSB^2(q, P))$ . Therefore,  $p \in RS(q, FSB^2(q, P))$ .

2. Let  $p \notin RS(q, P)$ , then we can get  $\exists p' \neq p, p' \prec_q p$ .

If  $p' \in FSB^2(q, P)$ , we can infer  $\exists p'' \in FSB^2(q, P), p' \prec_q p''$ .

If  $p' \notin FSB^2(q, P)$ , according to Definition 6, we can infer  $\exists p'' \neq p, p'' \prec_q p'$ . According to Lemma 2, since  $p'' \prec_q p'$  and  $p' \prec_q p$ , we can get  $p'' \prec_q p$ . Therefore,  $p \notin RS(q, FSB^2(q, P))$ .

In conclusion,  $RS(q, P) = RS(q, FSB^2(q, P))$ .  $\square$

Note that Theorem 6 indicates that it is sufficient to issue the 2-full-skyband query over WSNs to compute the RS query.

Similar to full skyline, the full skyband is also decomposable, which is shown by the following theorem.

**Theorem 7.** If  $P = P_1 \cup P_2 \cup \dots \cup P_n$ , then  $FSB^k(q, P) = FSB^k(q, \bigcup_{i=1}^n FSB^k(q, P_i))$ .

**Proof.** Immediately deduct from Definition 6.  $\square$

### 3.4 Reverse Skyline Processing over WSNs

After illustrating the conversion of our RS problem over WSNs into a decomposable 2-full-skyband operator, we are now ready to give our detailed query processing approach

in Algorithm 1. Specifically, a sensor node first merges the 2-full-skyband results from its child nodes into its local data set (Lines 1-2). Then, if the intermediate node is the base station, it computes the reverse skylines according to the merged data set (Lines 3-4), otherwise it obtains the 2-full-skyband answers from the merged data set (Lines 5-6). Finally, the node returns the result to the user or its parent node (Line 7).

---

**Algorithm 1:** InNetworkRS

---

**input** :  $q$  : query point,  
 $P$  : local data set,  
 $R_i$  : 2-FSB result of child sensor node  $n_i$ .  
**output**:  $R$  : local result.

1 **for** the result  $R_i$  of each child node  $n_i$  **do**  
2      $P = P \cup R_i$   
3 **if** it is the base station **then**  
4      $R = RS(q, P)$   
5 **else**  
6      $R = FSB^2(q, P)$   
7 **return**  $R_i$ ;

---

In contrast to full skyline, full skyband also sends those points having only one full-dominating point. Those points certainly do not belong to reverse skyline, since they only have the ability to semidominate their full-dominating points in the full skyline.

According to Lemma 2, the points which can be semidominated by point  $p$  can also be semidominated by  $p$ 's full-dominating point, except for  $p$ 's full-dominating point itself. Hereby, a point having one full-dominating point cannot affect the reverse skyline judgment of points in another data set, only points in the full skyline are enough, as shown in Theorem 8.

**Theorem 8.** Let  $P_1$  and  $P_2$  be two data sets, and  $p$  be a point in  $P_1$ . Then,  $p \in RS(q, RS(q, P_1) \cup FS(q, P_2))$ , iff  $p \in RS(q, P_1 \cup P_2)$ .

**Proof.** 1. The proof of sufficient condition is similar with Theorem 4. Detail is omitted due to page limitations.

2. We prove the necessary condition using reduction to absurdity. Assume  $p \notin RS(q, P_1 \cup P_2)$ , then  $\exists p' \in P_1 \cup P_2, p' \preceq_q p$ .

There are three conditions for point  $p'$ :

1. If  $p' \in RS(q, P_1) \cup FS(q, P_2)$ , we can directly infer  $p \notin RS(q, RS(q, P_1) \cup FS(q, P_2))$ .
2. If  $p' \in P_1 - RS(q, P_1)$ , we can infer  $p \notin RS(q, P_1)$ . Thus,  $p \notin RS(q, RS(q, P_1) \cup FS(q, P_2))$ .
3. If  $p' \in P_2 - FS(q, P_2)$ , we can infer  $\exists p'' \in FS(q, P_2), p'' \preceq_q p'$ . According to Lemma 2, we can infer  $p'' \preceq_q p$ . So we can get  $p \notin RS(q, RS(q, P_1) \cup FS(q, P_2))$ .  $\square$

According to Theorem 8, we can use a *flag* to indicate whether a full skyline point is also a reverse skyline point on a sensor node. Then, the parent node can validate whether such reverse skyline points are also reverse skyline points of the sub-tree only with the help of other sibling nodes' full skylines.

The full skyline having the *flag* to indicate which one belongs to reverse skyline is called the *marked full skyline* (denoted  $FS^*(q, P)$ ). Theorem 9 validates the correctness

**TABLE 1**  
Processing Details on Data Set  $P_1$

point	$R$	$F$	$D$	$D'$
$p_1$	$\{p_1\}$	$\emptyset$	$\emptyset$	$\emptyset$
$p_5$	$\{p_1, p_5\}$	$\emptyset$	$\emptyset$	$\emptyset$
$p_2$	$\{p_1, p_5, p_2\}$	$\emptyset$	$\emptyset$	$\emptyset$
$p_4$	$\{p_1, p_5, p_2, p_4\}$	$\emptyset$	$\emptyset$	$\emptyset$
$p_9$	$\{p_1, p_5, p_2\}$	$\{p_4\}$	$\{p_4\}$	$\{p_4\}$
$p_3$	$\{p_1, p_5, p_3\}$	$\{p_4, p_2\}$	$\emptyset$	$\{p_2\}$
$p_7$	$\{p_1, p_5, p_3, p_7\}$	$\{p_4, p_2\}$	$\emptyset$	$\emptyset$
$p_6$	$\{p_1, p_5, p_3, p_7\}$	$\{p_4, p_2\}$	$\emptyset$	$\emptyset$
$p_8$	$\{p_1, p_5, p_3\}$	$\{p_4, p_2, p_7\}$	$\{p_7\}$	$\{p_7\}$

of utilizing the marked full skyline to reconstruct the reverse skyline.

**Theorem 9.** If  $P = P_1 \cup P_2 \cup \dots \cup P_n$ , then  $RS(q, P) = RS(q, \bigcup_{i=1}^n FS^*(q, P_i))$ .

**Proof.** 1. Obviously,  $RS(q, P) \subseteq RS(q, \bigcup_{i=1}^n FS^*(q, P_i))$ .

2. We prove  $RS(q, \bigcup_{i=1}^n FS^*(q, P_i)) \subseteq RS(q, P)$  using reduction to absurdity. Assume  $p \notin RS(q, P)$ , we can infer  $\exists p', p' \preceq_q p$ . Without loss of generality, we assume  $p \in P_j$ . There are two cases for  $p'$  that need to be considered:

1. If  $p' \in P_j$ , we can infer  $p \in \overline{RS}(q, P_j)$ . From this, we can further infer  $p \notin RS(q, \bigcup_{i=1}^n FS^*(q, P_i))$ .
2. If  $p' \notin P_j$ , we can infer  $p' \in \bigcup_{i=1}^n FS^*(q, P_i)$  or  $\exists p'' \in \bigcup_{i=1}^n FS^*(q, P_i), p'' \preceq_q p'$ .

Whether  $p' \in \bigcup_{i=1}^n FS^*(q, P_i)$  or  $\exists p'' \in \bigcup_{i=1}^n FS^*(q, P_i)$ , we can infer  $p \notin RS(q, \bigcup_{i=1}^n FS^*(q, P_i))$ .  $\square$

According to Theorems 2 and 9, an intermediate node can get both full skyline and reverse skyline correctly merely by depending on the marked full skylines of its child nodes.

The computational process for the marked full skyline is given in Algorithm 2. The program scans the data points in data set  $P$  in a presorted order<sup>3</sup> (Line 1), which ensures that the later point cannot fully dominate the former one. Let  $F$  denote the set of points belonging to the full skyline but not belonging to the reverse skyline. If data point  $p$  is not dominated by any point in  $F$  (Line 2), we compute its full-dominating point set  $D$  from the reverse skyline candidate set  $R$  (Line 3). If  $D$  is empty, it means that point  $p$  is a full skyline point (Line 4). Next, we compare  $p$  with all the full skyline points in  $(F \cup R)$  (Line 5). If  $p$  is semidominated by any point in  $(F \cup R)$ ,  $p$  is a non-RS point, and we thus add  $p$  to  $F$  (Line 6). Otherwise, if  $p$  is not semidominated by any point in  $(F \cup R)$ ,  $p$  is an RS candidate, and we add it to  $R$  (Lines 7-8). If  $p$  has only one full-dominating point, it means that  $p$  is not a full skyline point but belongs to 2-full-skyband (Line 9). Next, we obtain those points which are semidominated by  $p$  in  $R$ , and then move those points from  $R$  to  $F$  (Lines 10-12). Finally, the algorithm returns the results of both the reverse skyline (i.e.,  $RS(q, P)$ ) as well as the full skyline excluding the reverse skyline (i.e.,  $FS(q, P) - RS(q, P)$ )

3. Any monotone scoring function would be fine. In this paper, we use the euclidean distance between the data point  $p$  and query point  $q$ .

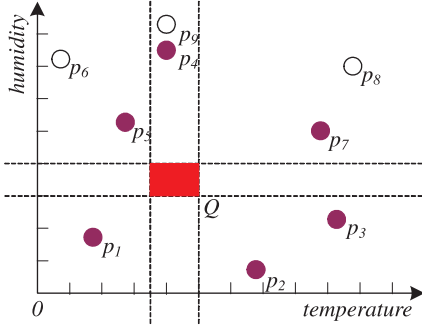


Fig. 7. Range reverse skyline.

(Line 13). Table 1 shows the computational process on data set  $P_1$  given in the example depicted in Fig. 5.

---

**Algorithm 2: LocalRS&FS**


---

**input :**  $q$  : query point,  
 $P$  : pre-sorted data set.  
**output:**  $R : RS(q, P)$ ,  
 $F : FS(q, P) - RS(q, P)$ .

```

1 for each data point  $p$  in  $P$  do
2   if  $p.isDominatedBy(F) == false$  then
3      $D = p.getDominating(R)$ ;
4     if  $|D| = 0$  then
5       if  $p.isSemiDominatedBy(F \cup R)$  then
6          $F = F \cup \{p\}$ ;
7       else
8          $R = R \cup \{p\}$ ;
9     if  $|D| \leq 1$  then
10       $D' = p.getSemiDominated(R)$ ;
11       $R = R - D'$ ;
12       $F = F \cup D'$ ;
13 return;
```

---

#### 4 RANGE REVERSE SKYLINE

When a user continuously asks for the reverse skyline while moving around, a better alternative is to submit a single query around the current location into WSN to fetch all possible results for this area. Similar to range nearest neighbor query [16], we propose a *range reverse skyline* query, which retrieves the reverse skyline for every point in a querying region. The range reverse skyline can solve the problem of the continuous RS mentioned above.

**Definition 7 (Range Reverse Skyline).** Given a data set  $P$  and a querying region  $Q$ , a range reverse skyline query (denoted as  $RS(Q, P)$ ) retrieves the union of reverse skylines of  $P$  with respect to every query point in  $Q$ .

Fig. 7 gives an example of range reverse skyline on data set  $P_1$ . Points  $p_1 - p_5$  and  $p_7$  belong to at least one reverse skyline with respect to the query points in query range  $Q$ . Thus, all of these are  $Q$ 's range reverse skyline points, and the union of these points forms the range reverse skyline of  $Q$ .

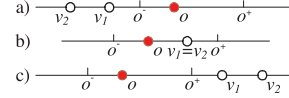


Fig. 8. Range relations.

Since the number of points in query range  $Q$  is infinite, it is impossible to carry out the range reverse skyline by first computing the reverse skyline of each point in  $Q$  and then merging all of them. As an extension of traditional reverse skyline, we want to explore if the range reverse skyline query has the same characteristics.

We denote the interval  $[o^-, o^+]$  as  $O$ . Let  $v_1 \dot{\leq}_O v_2$  stand for  $\forall o \in O, v_1 \dot{\leq}_o v_2$ , and  $v_1 \dot{<}_O v_2$  stand for  $\forall o \in O, v_1 \dot{<}_o v_2$ . Then, similar to the semidominance, we can define a concept, called *range semidominance*, as given in Definition 8 below.

**Definition 8 (Range Semidominance).** A point  $p_1$  range semidominates  $p_2$  with respect to  $Q$  (denoted as  $p_1 \dot{\leq}_Q p_2$ ), if it holds that: 1)  $p_1[i] \dot{\leq}_{Q[i]} p_2[i]$ , for all dimensions  $i \in D$ , and 2) there exists at least one dimension  $j \in D$ ,  $p_1[j] \dot{<}_{Q[j]} p_2[j]$ .

In interval  $O$ , since the possible value of  $o$  is infinite, it is difficult to judge whether or not  $v_1 \dot{\leq}_O v_2$  ( $v_1 \dot{<}_O v_2$ ) is fulfilled.

Let us first consider the example in Fig. 8. We find that  $v_1 \dot{\leq}_O v_2$  holds, if and only if at least one of the three cases below holds: 1)  $v_1 \leq o^-, v_2 \leq o^-$  and  $o^- - v_1 \leq 2(o^- - v_2)$ ; 2)  $o^+ \leq v_1, o^+ \leq v_2$  and  $v_1 - o^+ \leq 2(v_2 - o^+)$ ; and 3)  $o^- \leq v_1 = v_2 \leq o^+$ . If none of the above cases hold, then we can find an  $o \in O$  that can make  $v_1 \dot{\leq}_o v_2$  not be fulfilled. The above properties are summarized in Lemma 3.

**Lemma 3.** Given an interval  $O = [o^-, o^+]$ , it holds that  $v_1 \dot{\leq}_O v_2$ , iff we have: 1)  $v_1 \dot{\leq}_{o^-} v_2$ , 2)  $v_1 \dot{\leq}_{o^+} v_2$ , and 3)  $(v_1 - o^-) \cdot (v_1 - o^+) \geq 0$  or  $v_1 = v_2$ .

**Proof.** For detailed proofs see Appendix, available in the online supplemental material.  $\square$

The operator  $v_1 \dot{<}_O v_2$  is defined similarly to  $v_1 \dot{\leq}_O v_2$ . Clearly,  $v_1 \dot{<}_O v_2$  is not fulfilled on  $v_1 \dot{\leq}_O v_2$ ' condition 2. Thus, in order to make  $v_1 \dot{<}_O v_2$  fulfilled, one of the two following conditions is needed: 1)  $v_1 < o^-, v_2 < o^-$  and  $o^- - v_1 < 2(o^- - v_2)$ ; or 2)  $o^+ < v_1, o^+ < v_2$  and  $v_1 - o^+ < 2(v_2 - o^+)$ . The above properties are summarized in Lemma 4.

**Lemma 4.** Given an interval  $O = [o^-, o^+]$ , it holds that  $v_1 \dot{<}_O v_2$ , iff we have: 1)  $v_1 \dot{<}_{o^-} v_2$ , 2)  $v_1 \dot{<}_{o^+} v_2$ , and 3)  $(v_1 - o^-) \cdot (v_1 - o^+) > 0$ .

**Proof.** For detailed proofs see Appendix, available in the online supplemental material.  $\square$

Lemmas 3 and 4 make the judgement of range-semidominance feasible and much easier. Theorem 10 below illustrates the consanguineous relationship between range reverse skyline and range semidominance.

**Theorem 10.** Given a data set  $P$ , point  $p \in P$  is a range reverse skyline point of  $Q$ , iff there does not exist any other point  $p' \in P$  such that  $p' \dot{\leq}_Q p$ .

**Proof.** 1. We prove the sufficient condition using reduction to absurdity. Assume  $p \notin RS(Q, P)$ .



For each dimension  $i \in D$ , the value of point  $p$  must satisfy one of the following three conditions: 1)  $p[i] \geq Q[i]^+$ , 2)  $p[i] \leq Q[i]^-$ , or 3)  $Q[i]^- < p[i] < Q[i]^+$ .

Then, we construct a query point  $q'$  as follow:

If 1),  $q'[i] = Q[i]^+$ , if 2),  $q'[i] = Q[i]^-$ , if 3),  $q'[i] = p[i]$ .

Obviously, the constructed query point satisfies that  $q' \in Q$ .

Since we assumed that  $p \notin RS(Q, P)$ , we can infer  $p \notin RS(q', P)$ , thus  $\exists p', p' \dot{\prec}_{q'} p$ .

Since  $p' \dot{\prec}_{q'} p$ , we can get  $\forall i \in D, p'[i] \leq_{q'[i]} p[i]$  and  $\exists j \in D, p'[j] <_{q'[j]} p[j]$ .

If  $p[i] \geq Q[i]^+$ , we can get  $q'[i] = Q[i]^+$ . Moreover, we have  $p'[i] \leq_{q'[i]} p[i]$ , we can further infer  $p'[i] \geq Q[i]^+$ .

If  $p[i] \leq Q[i]^-$ , we can get  $q'[i] = Q[i]^-$ . Moreover, we have  $p'[i] \leq_{q'[i]} p[i]$ , we can further infer  $p'[i] \leq Q[i]^-$ .

If  $Q[i]^- < p[i] < Q[i]^+$ , we can get  $q'[i] = p[i]$ . Moreover, we have  $p'[i] \leq_{q'[i]} p[i]$ , we can further infer  $p'[i] = p[i]$ .

According to Lemma 3, we can infer  $p'[i] \leq_{Q[i]} p[i]$ . Similarly, according to Lemma 4, we can also infer  $p'[j] <_{Q[j]} p[j]$ , therefore, we have  $p' \dot{\prec}_Q p$ .

2. We prove the necessary condition using reduction to absurdity. Assume  $\exists p' \in P, p' \dot{\prec}_Q p$ , we can infer  $\forall i \in D, p'[i] \leq_{Q[i]} p[i]$  and  $\exists j, p'[j] <_{Q[j]} p[j]$ .

Since we have  $p'[i] \leq_{Q[i]} p[i]$ , we can infer  $\forall q[i] \in Q[i], p'[i] \leq_{q[i]} p[i]$ .

Since we have  $p'[j] <_{Q[j]} p[j]$ , we can infer  $\forall q[j] \in Q[j], p'[j] <_{q[j]} p[j]$ .

Thus, we get  $\forall q \in Q, p' \dot{\prec}_Q p$ , therefore, we can conclude that  $q \notin RS(Q, P)$ .  $\square$

According to Theorem 10, range reverse skyline is the set of points in  $P$  that are not range semidominated with respect to  $Q$ . It is obvious that range reverse skyline query is not decomposable either. Similar to the RS query, we can utilize a series of concepts with respect to the range reverse skyline query, such as *range full dominance*, *range full skyline*, and *range full skyband* to solve this problem.

Specifically, let  $v_1 \leq_O v_2$  stand for  $\forall o \in O, v_1 \leq_o v_2$ , and  $v_1 <_O v_2$  stand for  $\forall o \in O, v_1 <_o v_2$ . We have:

**Definition 9 (Range Full Dominance).** A point  $p_1$  range full-dominates  $p_2$  with respect to  $Q$  (denoted as  $p_1 \dot{\prec}_Q p_2$ ), if it holds that: 1)  $p_1[i] \leq_{Q[i]} p_2[i]$  for all dimensions  $i \in D$ , and 2) there exists at least one dimension  $j \in D, p_1[j] <_{Q[j]} p_2[j]$ .

**Definition 10 (Range Full Skyline).** Given a data set  $P$  and a querying region  $Q$ , a range full skyline query (denoted as  $FS(Q, P)$ ) retrieves all the points in  $P$  that are not range full-dominated by others with respect to  $Q$ .

**Definition 11 (Range Full Skyband).** Given a data set  $P$  and a querying region  $Q$ , a  $k$ -range-full-skyband query (denoted as  $FSB^k(Q, P)$ ) retrieves all the points in  $P$  that are range full-dominated by at most  $(k-1)$  points with respect to  $Q$ .

As illustrated in Fig. 8, in order to fulfill  $v_1 \leq_O v_2$ , we need to consider three cases: 1)  $v_2 \leq v_1 \leq o^-$ ; 2)  $o^+ \leq v_1 \leq v_2$ ; and 3)  $o^- \leq v_1 = v_2 \leq o^+$ . However, to fulfill  $v_1 <_O v_2$ , we only need to consider two cases: 1)  $v_2 < v_1 < o^-$ ; and 2)  $o^+ < v_1 < v_2$ . Theorem 11 formally presents this property.

**Theorem 11.** Given an interval  $O = [o^-, o^+]$ ,  $v_1 \leq_O v_2$ , iff it holds that: 1)  $v_1 \leq_{o^-} v_2$ , and 2)  $v_1 \leq_{o^+} v_2$ .  $v_1 <_O v_2$ , iff it holds that: 1)  $v_1 <_{o^-} v_2$ , and 2)  $v_1 <_{o^+} v_2$ .

**Proof.** The proof is similar to that of Lemmas 3 and 4.  $\square$

Through further analysis, range semidominance and range full dominance have similar properties, compared with semidominance and full dominance (discussed in Section 3), respectively. The details are given below.

**Theorem 12.**  $RS(Q, P) = RS(Q, FSB^2(Q, P))$ .

**Proof.** 1. Let  $p \in RS(Q, P)$ , we can get  $p \in FS(Q, P)$ . Based on this, we have  $p \in FSB^2(Q, P)$ .

Since we have  $p \in RS(Q, P)$ , we can infer  $\nexists p' \in P, p' \dot{\prec}_Q p$ . Based on this, we have  $\nexists p' \in FSB^2(Q, P), p' \dot{\prec}_Q p$ , therefore,  $p \in RS(Q, FSB^2(Q, P))$ .

2. We prove  $RS(Q, FSB^2(Q, P)) \subseteq RS(Q, P)$  using reduction to absurdity. Assume  $p \notin RS(Q, P)$ , we can get  $\exists p' \neq p, p' \dot{\prec}_Q p$ .

If  $p' \in FSB^2(Q, P)$ , we can get  $\exists p'' \in FSB^2(Q, P), p' \dot{\prec}_Q p''$ .

If  $p' \notin FSB^2(Q, P)$ , according to Definition 11, we can infer  $\exists p'' \neq p, p'' \dot{\prec}_Q p'$ .

According to Lemma 2, based on  $p'' \dot{\prec}_Q p'$  and  $p' \dot{\prec}_Q p$ , we have  $p'' \dot{\prec}_Q p$ . Therefore,  $p \notin RS(Q, FSB^2(Q, P))$ .

In conclusion,  $RS(Q, P) = RS(Q, FSB^2(Q, P))$ .  $\square$

**Theorem 13.** If  $P = P_1 \cup P_2 \cup \dots \cup P_n$ ,  $RS(Q, P) = RS(Q, \bigcup_{i=1}^n FS^*(Q, P_i))$ . Where  $FS^*(Q, P_i)$  is the marked range full skyline with respect to  $Q$ .

**Proof.** 1. Clearly,  $RS(Q, P) \subseteq RS(Q, \bigcup_{i=1}^n FS^*(Q, P_i))$ .

2. We prove  $RS(Q, \bigcup_{i=1}^n FS^*(Q, P_i)) \subseteq RS(Q, P)$  using reduction to absurdity. Assume  $p \notin RS(Q, P)$ , we can infer  $\exists p', p' \dot{\prec}_Q p$ . Without loss of generality, we assume  $p \in P_j$ . There would be two cases for  $p'$ :

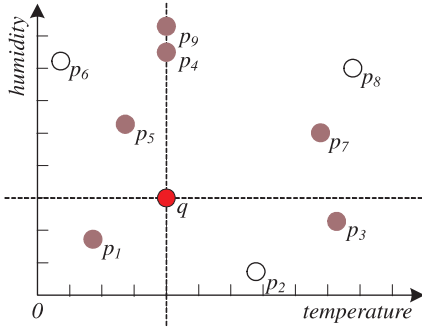
1. If  $p' \in P_j$ , we can infer  $p \in \overline{RS}(Q, P_j)$ . So, we can further infer  $p \notin RS(Q, \bigcup_{i=1}^n FS^*(Q, P_i))$ .
2. If  $p' \notin P_j$ , we can infer  $p' \in \bigcup_{i=1}^n FS^*(Q, P_i)$  or  $\exists p'' \in \bigcup_{i=1}^n FS^*(Q, P_i), p'' \dot{\prec}_Q p'$ .
3. Whether

$$p' \in \bigcup_{i=1}^n FS^*(Q, P_i) \text{ or } \exists p'' \in \bigcup_{i=1}^n FS^*(Q, P_i),$$

we can both infer  $p \notin RS(Q, \bigcup_{i=1}^n FS^*(Q, P_i))$ .  $\square$

Range reverse skyline has all the attractive properties that traditional reverse skyline has. Thus, optimization techniques used by traditional reverse skyline can be also applied in range reverse skyline directly. The only difference is that traditional relation/query needs to be replaced by range relation/query. Because the optimization principle is identical, as a particular case of range reverse skyline, traditional reverse skyline can also be processed by this algorithm without leading to any additional computational cost and data transmission.

Because of this, range reverse skyline and traditional reverse skyline are unified. Unless otherwise specified, we will not make distinction between them and both will go by the general name of reverse skyline in the following discussions.

Fig. 9.  $RX(q, P_1)$ .

## 5 MULTIPLE REVERSE SKYLINES

In real applications, different users may have different preferences, and it is quite common to have multiple queries in different subspaces which are posed into the WSN simultaneously to gather interesting information [12]. It is not efficient to individually evaluate them especially in a WSN environment where the power consumption should be minimized.

In this section, first, the basic principles of an optimization technique are introduced. Then, the optimization mechanism of multiple reverse skylines is proposed based on these basic principles.

### 5.1 Optimization Principles

In this section, we propose an effective optimization technique, called *Vertical Optimization*, for multiple queries by using the relationship between reverse skylines with respect to different subspaces but the same query range. Then, we will illustrate another technique, namely *Horizontal Optimization*, by using the relationship between reverse skylines with the same subspace but different query ranges.

#### 5.1.1 Vertical Optimization

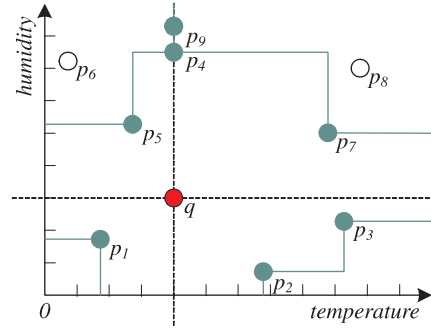
When two reverse skyline queries have the same range and their subspaces have the inclusion relationship, their RS results do not directly have the inclusion relationship.

As shown in Fig. 1, the RS set of data set  $P_1$  in 1D temperature space is  $\{p_4, p_9\}$ , whereas that in 2D temperature-and-humidity space is  $\{p_1, p_3, p_5\}$ . Thus, we cannot achieve the goal of optimization by simply merging the two queries into one.

Similar to the processing procedure for multiple skylines [17], [12], we convert semidominance into *strict semidominance*, resulting in the *extended reverse skyline*. Thus, the reverse skyline in all subspaces are included in the result set of the extended reverse skyline.

**Definition 12 (Strict Semidominance).** A point  $p_1$  strictly semidominates  $p_2$  in a subspace  $D$  with respect to  $Q$  (denoted as  $p_1 \prec_Q^D p_2$ ), if it holds that  $p_1[i] <_{Q[i]} p_2[i]$ , for all dimensions  $i \in D$ .

**Definition 13 (Extended Reverse Skyline).** Given a data set  $P$  and a querying region  $Q$ , an extended reverse skyline query in a subspace  $D$  (denoted as  $RX(Q, P, D)$ ) retrieves all the points in  $P$  that are not strictly semidominated by others with respect to  $Q$  in subspace  $D$ .

Fig. 10.  $FX(q, P_1)$ .

As shown in Fig. 9, compared with reverse skyline, the extended reverse skyline includes not only reverse skyline points  $p_1$ ,  $p_3$ , and  $p_5$ , but also those semidominated but not strictly semidominated data points  $p_4$ ,  $p_7$ , and  $p_9$ . Except for those points, no matter in which subspaces, other data points always have one extended reverse skyline point that can strictly semidominate it. The following theorem describes the above property in details.

**Theorem 14.** If  $D_1 \subseteq D_2$ , then we have  $RS(Q, P, D_1) \subseteq RX(Q, P, D_2)$ .

**Proof.** We prove the theorem using reduction to absurdity.

Assuming  $p \notin RX(Q, P, D_2)$ , we can get  $\exists p' \in P, p' \prec_Q^{D_2} p$ .

Since we have  $p' \prec_Q^{D_2} p$ , we can infer  $\forall i \in D_2, p'[i] <_{Q[i]} p[i]$ . Since we also have  $D_1 \subseteq D_2$ , we can infer  $\forall i \in D_1, p'[i] <_{Q[i]} p[i]$ . According to Definition 12, we have  $p' \prec_Q^{D_1} p$ . Based on this, we can further infer  $p' \prec_Q^{D_1} p$ . Therefore,  $p \notin RS(Q, P, D_1)$ .  $\square$

Since the extended reverse skyline is only a simple expansion of reverse skyline, we can use the same algorithm to solve it. The difference is that, we should use extended full skyline with strict full dominance, rather than the full skyline as mentioned earlier. The definitions of *strict full dominance* and *extended full skyline*<sup>4</sup> are given as follows:

**Definition 14 (Strict Full Dominance).** A point  $p_1$  strictly full-dominates  $p_2$  in a subspace  $D$  with respect to  $Q$  (denoted as  $p_1 \prec_Q^D p_2$ ), if it holds that  $p_1[i] <_{Q[i]} p_2[i]$ , for all dimensions  $i \in D$ .

**Definition 15 (Extended Full Skyline).** Given a data set  $P$  and a querying region  $Q$ , an extended full skyline query in a subspace  $D$  (denoted as  $FX(Q, P, D)$ ) retrieves all the points in  $P$  that are not strictly full-dominated by others with respect to  $Q$  in subspace  $D$ .

Fig. 10 gives an example of the extended full skyline, compared with full skyline. There are some points that are not strictly full-dominated, such as point  $p_9$ .

We can use the marked extended full skyline to accurately obtain the extended reverse skyline in WSN. However, while calculating, only the strict semidominance

4. The global dominance [5] requires  $\forall i \in D, (p_1[i] - q[i]) \cdot (p_2[i] - q[i]) > 0$  and  $|p_1[i] - q[i]| \leq |p_2[i] - q[i]|$ , while strict-full-dominance requires  $\forall i \in D, (p_1[i] - q[i]) \cdot (p_2[i] - q[i]) > 0$  and  $|p_1[i] - q[i]| < |p_2[i] - q[i]|$ . Thus, the extended full skyline is a super set of the global skyline.

relationship in space  $D$  is reserved. Moreover, the semi-dominance relation in the subspace is not completely reserved. Thus, we cannot obtain reverse skyline in each subspace accurately.

Clearly, 2-full-extended-skyband (denoted as  $FXB^2(Q, P, D)$ ) can solve this problem. We can find another point  $p' \in FXB^2(Q, P, D)$  that fulfills  $p' \preceq_Q^D p$  for any point  $p \in P$ , satisfying  $\exists D' \subseteq D, p \notin RS(Q, P, D')$ .

The purpose of transmitting points in  $FXB^2(Q, P, D) - FX(Q, P, D)$  is to judge whether or not its strict-full-dominating point belongs to the reverse skylines in some subspaces. We can optimize its transmission processes separately. These points can be divided into three possible categories:

1. Those points whose strict-full-dominating point is strictly semidominated by some point in  $FXB^2(Q, P, D)$  do not need to be transmitted, but its strict-full-dominating point however will be marked as such.
2. Those points that cannot semidominate their strict-full-dominating point in any subspace do not need to be transmitted in WSNs, because they make no contributions to the calculation of reverse skyline.
3. Since the remaining points may semidominate its strict-full-dominating point in some subspace, we need to transmit them back to the base station.

Through the above processing procedure, we can guarantee that the parent node can accurately compute the full skyline and reverse skyline in any subspace according to the obtained data, while keeping data transmission cost low.

### 5.1.2 Horizontal Optimization

According to Definition 7, when the two reverse skyline queries have the same subspace and their query ranges have the inclusion relationship, their results also have the inclusion relation, as shown in the theorem below.

**Theorem 15.** If  $Q_1 \subseteq Q_2$ , then we have  $RS(Q_1, P, D) \subseteq RS(Q_2, P, D)$ .

**Proof.** According to Definition 7, we have  $RS(Q_1, P, D) = \bigcup_{q \in Q_1} RS(q, P, D)$  and  $RS(Q_2, P, D) = \bigcup_{q \in Q_2} RS(q, P, D)$ . Since we have  $Q_1 \subseteq Q_2$ , we can infer  $\bigcup_{q \in Q_1} RS(q, P, D) \subseteq \bigcup_{q \in Q_2} RS(q, P, D)$ . Therefore,  $RS(Q_1, P, D) \subseteq RS(Q_2, P, D)$ .  $\square$

From Sections 3 and 4, we know that, by utilizing marked full skyline, reverse skyline can be calculated accurately in WSNs. However, the reverse skyline of a random subrange cannot be calculated accurately, since only semidominance relationships in the large range  $Q$  are taken into account while in the small range are not preserved.

Similar to the discussions in Section 5.1.1, 2-full-skyband can resolve the above problem. With regard to any point  $p \in P$  satisfying  $\exists Q' \subseteq Q, p \notin RS(Q', P, D)$ , we can find another point  $p' \in FSB^2(Q, P, D)$  that meets  $p' \preceq_Q^D p$ .

Equivalently, the points in  $FSB^2(Q, P, D) - FS(Q, P, D)$  are used to help determine whether its dominating points belong to reverse skyline in some subranges. Its transmission

can be optimized according to the practical situation. There are three categories for these points.

1. If its full-dominating point can be semidominated by some point in  $FSB^2(Q, P, D)$ , then it can be affiliated its full-dominating point with mark, therefore, these points need not to be transmitted.
2. If points cannot semidominate its full-dominating point in any subrange,<sup>5</sup> they do not need to be transmitted. This is because they have no contributions to the reverse skyline computation.
3. The remaining points need to be transmitted, since they may semidominate its full-dominating point in some subranges.

As a result, the above procedure not only ensures that the parent node can accurately compute full skyline and reverse skyline in any subrange by utilizing the given data, but also reduces the volume of the transmitted data.

## 5.2 Optimization Mechanism

In the same dimensional space, the query result in a small query range can be included by that in large query range; similarly, in the same query range, the query result in a space can be included by that in its superspace. Taking these aspects into account, one straightforward method is to first combine all the queries into a new one employing the largest range covered and using the highest dimensional space, and then calculate actual query results from the retrieved answers from the WSN. However, this method is not efficient. Although the number of queries is reduced by combining the subspaces or the subranges, the cardinality of the new query result increases remarkably. Therefore, it is difficult to determine directly whether or not we really should combine the queries. Thus, during the optimization, we should compare the transmission cost before and after the combination, and then choose the plan with the smallest transmission cost.

It is an important step in optimization to estimate the transmission cost of a query. The communication cost of queries cannot be estimated directly, because this cost has consanguineous relationship with the particular routing structure of the WSN. Generally speaking, the communication cost exhibits a linear relationship with cardinality of the query results, so it can be weighed by the size of query result. According to our reverse skyline query, (extended) full skyline query (sometimes, 2-full-skyband) is frequently executed in the WSNs. As such, the cardinality of the (extended) full skyline result determines the communication cost.

As shown in Fig. 11, the (extended) full skyline consists of the subresults of regions I, II, and III. Depending on the subregion as depicted in Fig. 11, we can estimate the total transmission cost as following:

**Region I:** No matter  $Q$  is a full skyline query or an extended full skyline query, all points in this region belong to the results of  $Q$ . The cardinality of this region can be estimated by multiplying the capacity of this region with the data distributing density.

**Region II:** According to the definition of the full skyline query, the set of points in this region is distributed between

5. The method to determine is similar to Lemmas 3 and 4.

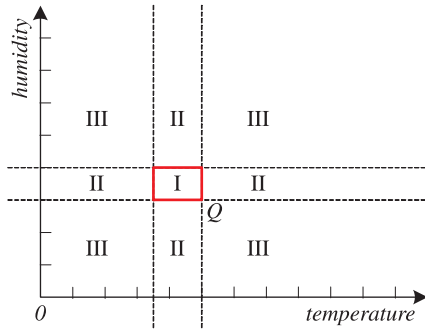


Fig. 11. Cost estimation.

$Q[i]^-$  and  $Q[i]^+$  in the space  $D' \neq \emptyset$ , and distributed out of  $Q[i]^-$  and  $Q[i]^+$  in the space  $(D - D')$  containing other dimensions. If a point  $p_1$  full-dominates  $p_2$ , then it must satisfy  $\forall i \in D', p_1[i] = p_2[i]$ . Denote the quantity of the data set  $P$  as  $N$ , data duplicate ratio in dimension  $i$  as  $\rho_i$ , and the number of points with the same value in the space  $D'$  as  $N_e = N \cdot \prod_{i \in D'} \rho_i$ . Then, the number of distinct points in  $P$  is  $N_{ne} = 1 / \prod_{i \in D'} \rho_i$ . By utilizing sampling or existing estimation methods [18], [19], the skyline cardinality of each distinct point  $p_i$ ,  $S_i$ , can be estimated. Therefore, the total number of results in region II can be estimated as  $N_{ne} \cdot S_i$ . Repeating the calculations above, all possible situations in region II can be enumerated, and the cardinality in this region can be obtained by accumulating the subresults. In practice, since  $\rho_i$  is small,  $N_e$  is small as well which makes  $N_s \approx N_e$ . So the cardinality of query result is close to the number of data points in this region.

According to the definition of the extended full skyline query, all the points in this region completely belong to the result, so the quantity of data points in this region is the cardinality of the final results.

**Region III:** The cardinality of the points in this region needs to be estimated by utilizing sampling or other existing estimation methods [18], [19].

During the optimization procedure, the data structures and corresponding algorithms for the maintenance of the original query and the synthetic query are similar to that of [12]. The only difference is in the procedure performed after a suitable super-space query is found. After some synthetic queries are found whose query spaces include the new query's subspace, we do not add the new query into any synthetic query directly, but estimate the costs before and after the combination, choosing the most suitable synthetic query to add.

## 6 EXPERIMENTAL EVALUATION

In this section, we evaluate the query performance of our proposed approaches, based on tree-based structure using both real-world and synthetic sensor data sets.

For single reverse skyline query, the following three algorithms are evaluated.

- *Centralized*: the naive approach described at the beginning of Section 3.
- *Skyband*: skyband based approach.
- *Marked*: marked full skyline based approach.

TABLE 2  
Experimental Parameters

Parameter	Default	Range
dimensionality	4	2, 3, 4
size of range	0.025	0.1, 0.05, 0.025, 0.0125, 0.00625
number of queries	30	10, 20, 30, 40, 50

For multiple reverse skyline queries, the following two algorithms are tested.

- *Naive*: evaluating multiple queries one by one.
- *Opt*: algorithm with optimization mechanism.

### 6.1 Real Traces

The real-world sensor data set adopted is obtained from a forest environment monitoring project. In the project, we adopt the TelosB Mote [20] with a *MSP430* processor and *CC2420* transceiver, and each mote is equipped with two 2,200 mAh batteries. One-hundred twenty motes are deployed in a region of 20,000 m<sup>2</sup>. We used the data collected in September 2009, which include temperature ([10°, 50°]), humidity ([10%, 90%]), light ([0 KLux, 150 KLux]), and voltage ([0 V, 3 V]). There are around 30,000 sensory data items for each type, so we have enough data for running the algorithms multiple times with different data to verify their efficiency. We test our proposed solution on this real-world data set under various parameters. Table 2 summarizes the parameters under real-world data set investigation, along with their ranges and default values. In each experiment, we vary a single parameter, while setting the remainders to their default values.

#### 6.1.1 Traditional Reverse Skyline

First, we investigate the performance of traditional reverse skyline query processing. In the experiments, we randomly generate 100 query points in the domain of the data set and report their average communication cost.

Fig. 12 shows that the communication cost of all the testing approaches is proportional to the increase of dimensionality. The reason is that an increase of dimensionality leads to the increase of the size of the individual tuples, and thus will increase the total communication

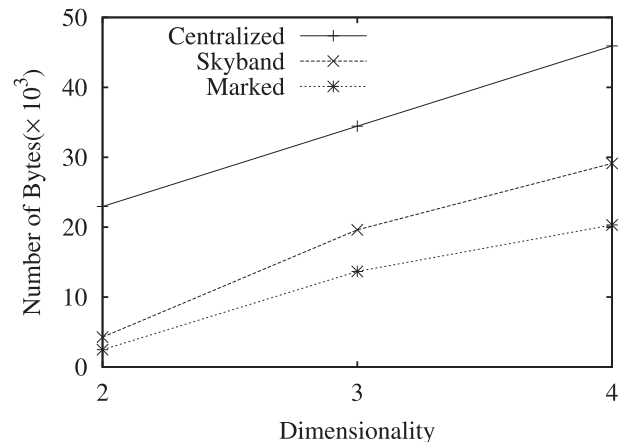


Fig. 12. Traditional reverse skyline versus dimensionality.



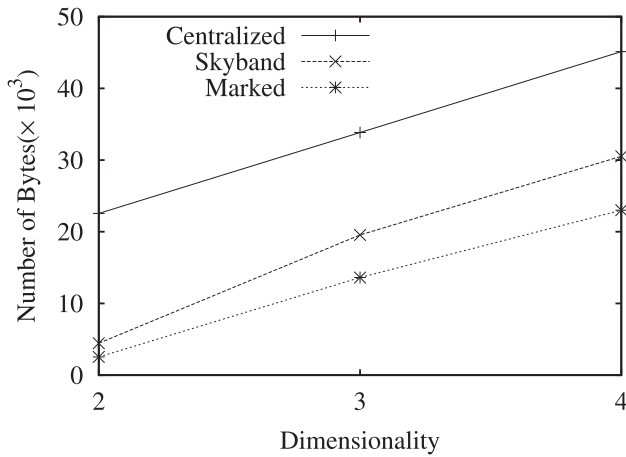


Fig. 13. Range reverse skyline versus dimensionality.

costs. Though the total number of tuples transmitted in Centralized approach is fixed, the number of messages increases accordingly due to the increase of communication cost per tuple. The increase of dimensionality also incurs a larger amount of full skyband/skyline results. Among the testing approaches, the *Centralized* approach has the highest transmission costs, whereas the *Marked* has the lowest cost. This indicates that our algorithm is efficient in reducing the communication cost.

### 6.1.2 Range Reverse Skyline

Next, we consider the influence of different parameters to the range reverse skyline query processing. We use the same set of query points generated for the traditional reverse skyline experiments. The query range on each attribute is calculated by multiplying its domain size with the size of the range.

As shown in Fig. 13, with the change of dimensionality, the communication cost trends of the three algorithms are similar, with only some slight difference in the communication cost. Since range reverse skyline is the generalization of traditional reverse skyline, their properties and regularity are similar. Range reverse skyline is the union of a set of traditional reverse skylines that have more overlaps, and thus the transmission cost increases as the dimensionality increases.

Fig. 14 illustrates the influence of query range. The communication cost of *Centralized* approach is stable, while that of *Skyband* and *Marked* decreases with decreasing the size of query range. When the query range varies, the amount of sensing points remains the same. Therefore, the communication cost of *Centralized* approach does not change. Meanwhile, the shrinking of query range leads to the decrease of the number of range full skyband/skyline points, so the communication costs drop. The cost of *Centralized* is always the highest, and that of *Marked* is always lower than that of *Skyband*, which further proves the utility of our proposed techniques.

### 6.1.3 Multiple Reverse Skylines

Next, we conduct a comprehensive performance evaluation of the multiple reverse skylines optimization. In these experiments, we mainly consider two kinds of query point distributions (*uniform* and *clustered*) and two kinds of query

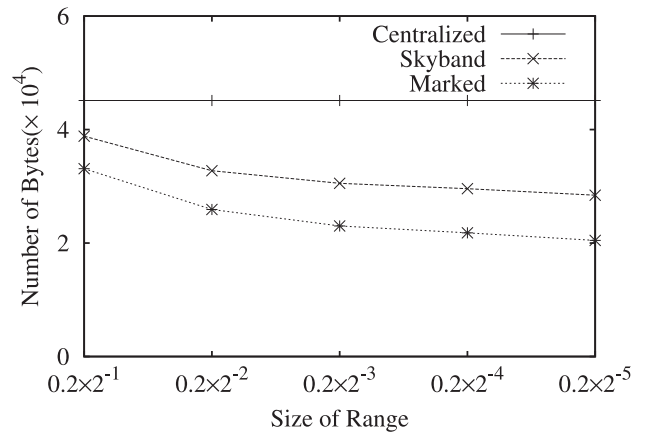


Fig. 14. Range reverse skyline versus query range size.

dimensional space distributions (*uniform* and *Zipf*). In total, there are four combinations in terms of the query point distributions and the query dimensional space distributions. Due to the similar performance, we only report the results of the two query point distributions, in the Zipf query dimensional space. The query range on each attribute is calculated by multiplying its domain size by a randomly generated decimal between zero and the size of the range.

Fig. 15 shows that the communication cost of each algorithm in multiple queries is increasing with the increase of dimensionality. Simultaneously, the communication cost of *Opt* is always lower than that of *Naive*, which illustrates that optimization mechanism is effective in avoiding the transmission of unnecessary data.

Fig. 16 shows the performance of the algorithms by changing the numbers of queries. While the number of queries increases, the communication cost of *Naive* also increases. This is because more queries lead to the increase

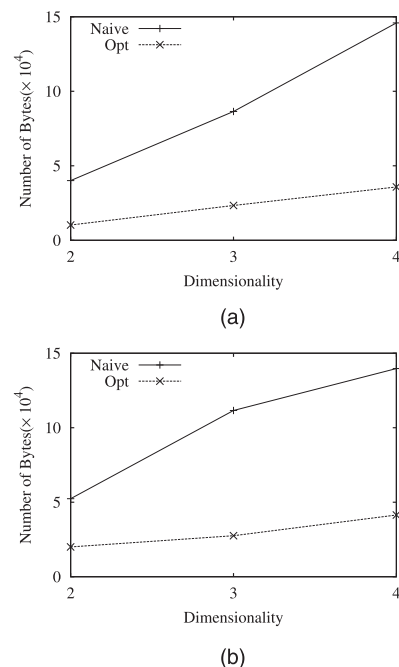


Fig. 15. Multiple reverse skyline versus dimensionality. (a) Uniform. (b) Clustered.

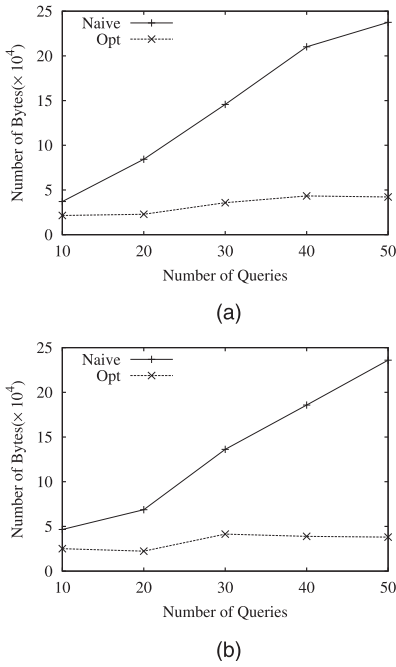


Fig. 16. Multiple reverse skyline versus the number of queries. (a) Uniform. (b) Clustered.

of the number of query results, which in turn increases the communication cost. The communication cost of *Opt* is increasing slowly with the growth of the queries. Compared to *Naive*, *Opt* is more stable, which indicates that the influence of the query number on the query performance is small by using our optimization mechanism.

Fig. 17 shows the communication cost when the maximal query range varies. Because of the reduction of the maximal range, the average range of each query is reduced, which implies that the probability that a point is range dominated is increasing. As such the number of the results is reduced. Either the cost of *Naive* or that of *Opt* is declining with the reduction of the query range. The performance of *Opt* is better than that of *Naive*, which further confirms that the optimization mechanism is effective.

## 6.2 Synthetic Traces

Due to the limited number of sensor nodes and dimensionality in the real-world sensor data, for testing the scalability of our proposed approaches, we have conducted comprehensive performance experiments using a synthetic data set. We perform our synthetic analysis with a simulator developed in C++. The simulator allows us to adjust network parameters, such as the number of nodes and the communication radius. For the sake of experimental unification, we randomly place  $n$  sensors in an area of  $\sqrt{n} \times \sqrt{n}$  units. Under these conditions, each node contains one unit space on average. The communication radii of sensor nodes are set to  $2\sqrt{n}$  units. Meanwhile, we enforce that the maximal length of packets transmitted in the network is limited to 48 bytes. All the experiments are run on a PC with 1.86 GHz Intel Core 2 CPU, 1 GB memory, and 80 GB hard disk.

The synthetic data are generated by standard data set generator for reverse skyline queries [5], including the *uniform* distributed data points and the *clustered* distributed

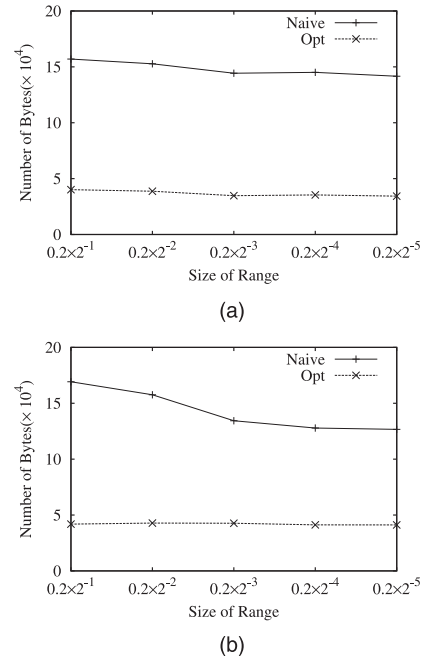


Fig. 17. Multiple reverse skyline versus query range size. (a) Uniform. (b) Clustered.

data points. The *uniform* data set consists of the points randomly generated on a unit square. However, the *clustered* data set comprises 10 randomly centered clusters, each of these contains equal number of points and follows a Multivariate Gaussian Distribution whose covariance matrix is a 0.05-diagonal matrix and mean vector is equal to the associated centroid. Due to the similar performance, the results of uniform data are omitted. Table 3 summarizes the parameters under synthetic investigation.

As shown in Figs. 18a, 19a, and 20a, the communication cost with respect to the dimensionality, in each algorithm for the synthetic data set is similar to that on the real-world data set. Figs. 18b, 19b, and 20b show that the communication cost of all approaches increase along with the increase of the number of nodes. This is because the increase of the number of nodes means an increase in the number of the sensed data points. The increase of the number of points usually incurs a larger amount of full skyband/skyline results. The performance relation of the proposed algorithms is similar to that in previous experiments, which can validate the effectiveness of our proposed approaches.

Fig. 21 shows the cardinality of the query results of Full Skyband (FSB), Full Skyline (FS), and Reverse Skyline with varying the dimensionality. Among the three query algorithms, the cardinality of the FSB result set is the highest, while the cardinality of the RS result set is the lowest. The inclusion relationships among the three queries can well explain this phenomenon. Since a RS query is not

TABLE 3  
Simulation Parameters

Parameter	Default	Range
dimensionality	5	3, 4, 5, 6, 7
number of nodes	8000	6000, 7000, 8000, 9000, 10000

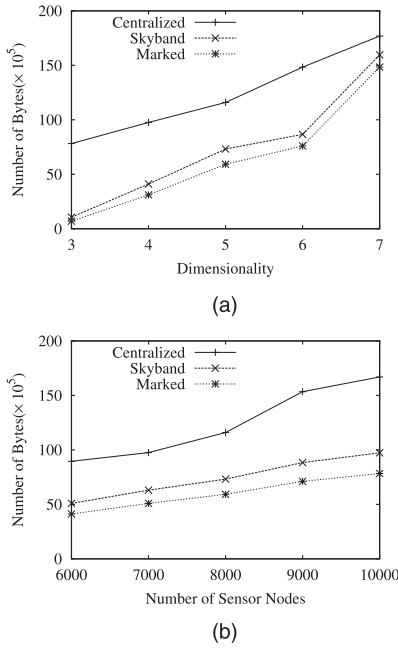


Fig. 18. Scalability of reverse skyline. (a) Dimensionality. (b) Number of sensor nodes.

decomposable, even though the cardinality of RS is much lower than those of both FSB and FS, we still need to transmit the marked full skyline, whose cardinality is a little higher than that of FS but lower than that of FSB to ensure that the RS results can be calculated accurately at the base station.

### 6.3 Performance on Local Sensor Node

We implemented the branch and bound RS algorithm (BBS) in [5] and the semidominance based RS algorithm (SDRS) using C++. As shown in Fig. 22, SDRS is much faster

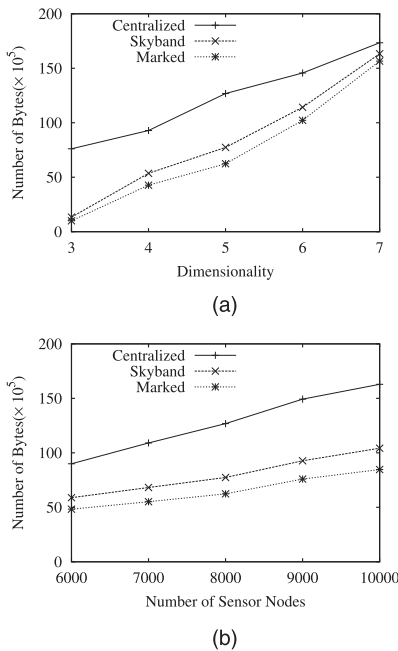


Fig. 19. Scalability of range reverse skyline. (a) Dimensionality. (b) Number of sensor nodes.

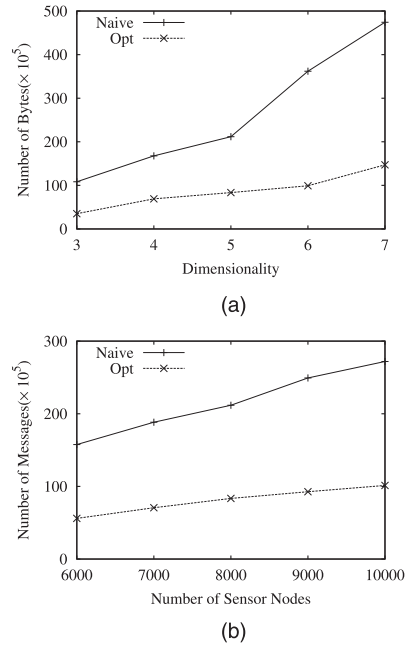


Fig. 20. Scalability of multiple reverse skylines. (a) Dimensionality. (b) Number of sensor nodes.

than BBS, because it only needs to traverse the R-tree index one round, while BBS needs to traverse the R-tree index multiple rounds (one round for a boolean window query).

### 6.4 Summary

From the experiments on both real-world and synthetic data sets, we can conclude that for traditional reverse skyline, range reverse skyline and multiple reverse skylines, our proposed approaches can achieve robust query performance with respect to various parameters, and they are energy efficient to be carried out in WSNs.

## 7 CONCLUSIONS AND FUTURE WORK

Energy is one of precious resources in WSNs, and it is mainly consumed by the wireless communication. Therefore, it has become an essential problem for sensing applications to minimize the communication cost. In this paper, we perform a comprehensive study on reverse

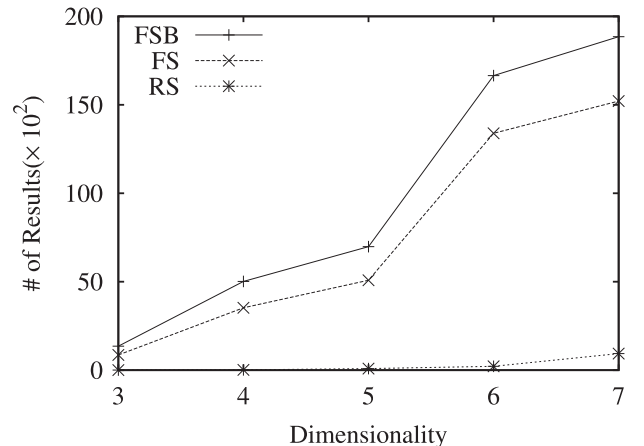


Fig. 21. Cardinalities of different queries.

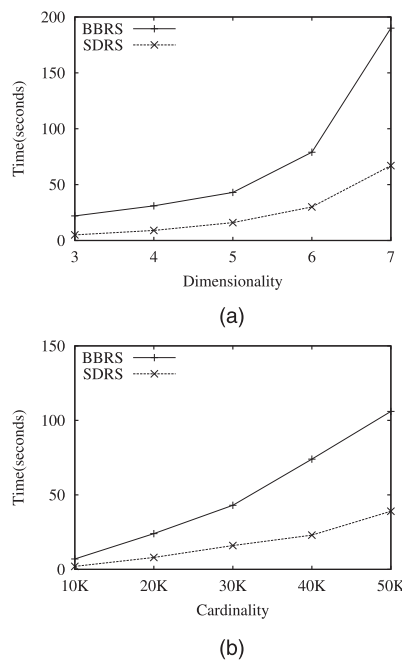


Fig. 22. Performance on local sensor node. (a) Dimensionality. (b) Cardinality.

skyline query processing in WSNs. First, we theoretically analyze the properties of reverse skyline, and introduce an energy-efficient approach based on full skyband to suppress any unnecessary data transmission. Next, the proposed approach is extended to support range reverse skyline queries. Then, optimization mechanisms for multiple reverse skyline queries are developed to save both the query propagation cost and the redundant results transmission cost. Finally, we have conducted extensive experimental studies to evaluate the performance of the proposed approaches on both real-world data and synthetic data. The experimental results show that our proposed approaches can effectively reduce communication cost and reduce the energy consumption of RS queries in WSNs.

Since data uncertainty is an inherent characteristic of WSNs [21], it is an interesting direction for our future research work to conduct queries and extract useful information from massive uncertain sensor readings.

## ACKNOWLEDGMENTS

This research is supported by the National Natural Science Foundation of China (Grant No. 60873011, 60933001, and 61025007), the National Natural Science Foundation for Young Scientists of China (Grant No. 61100022), National Basic Research Program of China (973, Grant No. 2011CB302200-G), the 863 High Technology Program (Grant No. 2009AA01Z150), and the Fundamental Research Funds for the Central Universities (Grant No. N090104001 and N090304007).

## REFERENCES

[1] S. Borzsonyi, K. Stocker, and D. Kossmann, "The Skyline Operator," *Proc. 17th Int'l Conf. Data Eng.*, pp. 421-430, 2001.

[2] E. Dellis, A. Vlachou, I. Vladimirskiy, B. Seeger, and Y. Theodoridis, "Constrained Subspace Skyline Computation," *Proc. 15th ACM Int'l Conf. Information and Knowledge Management (CIKM '06)*, pp. 415-424, 2006.

[3] D. Papadias, Y. Tao, G. Fu, and B. Seeger, "An Optimal and Progressive Algorithm for Skyline Queries," *Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '03)*, pp. 467-472, 2003.

[4] K. Deng, X. Zhou, and H.T. Shen, "Multi-Source Skyline Query Processing in Road Networks," *Proc. IEEE 23rd Int'l Conf. Data Eng. (ICDE '07)*, pp. 796-805, 2007.

[5] E. Dellis and B. Seeger, "Efficient Computation of Reverse Skyline Queries," *Proc. 33rd Int'l Conf. Very Large Data Bases (VLDB '07)*, pp. 291-302, 2007.

[6] X. Lian and L. Chen, "Monochromatic and Bichromatic Reverse Skyline Search over Uncertain Databases," *Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '08)*, pp. 213-226, 2008.

[7] X. Wu, Y. Tao, R.C.-W. Wong, L. Ding, and J.X. Yu, "Finding the Influence Set through Skylines," *Proc. 12th Int'l Conf. Extending Database Technology: Advances in Database Technology (EDBT '09)*, pp. 1030-1041, 2009.

[8] S. Madden, M.J. Franklin, J.M. Hellerstein, and W. Hong, "TAG: A Tiny Aggregation Service for Ad-Hoc Sensor Networks," *Proc. Fifth Symp. Operating Systems Design and Implementation (OSDI '02)*, pp. 131-146, 2002.

[9] A. Silberstein, R. Braynard, C.S. Ellis, K. Munagala, and J. Yang, "A Sampling-Based Approach to Optimizing Top-k Queries in Sensor Networks," *Proc. 22nd Int'l Conf. Data Eng. (ICDE '06)*, p. 68, 2006.

[10] M. Wu, J. Xu, X. Tang, and W.-C. Lee, "Top-k Monitoring in Wireless Sensor Networks," *IEEE Trans. Knowledge and Data Eng.*, vol. 19, no. 7, pp. 962-976, July 2007.

[11] H. Chen, S. Zhou, and J. Guan, "Towards Energy-efficient Skyline Monitoring in Wireless Sensor Networks," *Proc. 4th European Conf. Wireless Sensor Networks (EWSN '07)*, pp. 101-116, 2007.

[12] J. Xin, G. Wang, L. Chen, and V. Oria, "Energy-Efficient Evaluation of Multiple Skyline Queries over a Wireless Sensor Network," *Proc. 14th Int'l Conf. Database Systems for Advanced Applications (DASFAA '09)*, pp. 247-262, 2009.

[13] J. Xin, G. Wang, L. Chen, X. Zhang, and Z. Wang, "Continuously Maintaining Sliding Window Skylines in a Sensor Network," *Proc. 12th Int'l Conf. Database Systems for Advanced Applications (DASFAA '07)*, pp. 509-521, 2007.

[14] W. Liang, B. Chen, and J.X. Yu, "Energy-Efficient Skyline Query Processing and Maintenance in Sensor Networks," *Proc. 17th ACM Conf. Information and Knowledge Management (CIKM '08)*, pp. 1471-1472, 2008.

[15] J. Considine, F. Li, G. Kollios, and J. Byers, "Approximate Aggregation Techniques for Sensor Databases," *Proc. 20th Int'l Conf. Data Eng. (ICDE)*, pp. 449-460, 2004.

[16] H. Hu and D.L. Lee, "Range Nearest-Neighbor Query," *IEEE Trans. Knowledge and Data Eng.*, vol. 18, no. 1, pp. 78-91, Jan. 2006.

[17] A. Vlachou, C. Doukeridis, Y. Kotidis, and M. Vazirgiannis, "SKYPEER: Efficient Subspace Skyline Computation over Distributed Data," *Proc. IEEE 23rd Int'l Conf. Data Eng. (ICDE)*, pp. 416-425, 2007.

[18] S. Chaudhuri, N.N. Dalvi, and R. Kaushik, "Robust Cardinality and Cost Estimation for Skyline Operator," *Proc. 22nd Int'l Conf. Data Eng. (ICDE '06)*, p. 64, 2006.

[19] Z. Zhang, Y. Yang, R. Cai, D. Papadias, and A.K.H. Tung, "Kernel-Based Skyline Cardinality Estimation," *Proc. 35th SIGMOD Int'l Conf. Management of Data (SIGMOD)*, pp. 509-522, 2009.

[20] J. Polastre, R. Szewczyk, and D.E. Culler, "Telos: Enabling Ultra-Low Power Wireless Research," *Proc. Fourth Int'l Symp. Information Processing in Sensor Networks (IPSN '05)*, pp. 364-369, 2005.

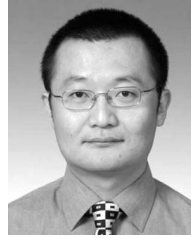
[21] R. Cheng and S. Prabhakar, "Managing Uncertainty in Sensor Databases," *SIGMOD Record*, vol. 32, no. 4, pp. 41-461, 2003.



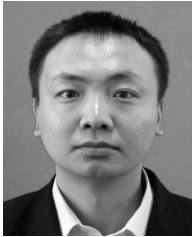


**Guoren Wang** received the BSc, MSc, and PhD degrees from the Department of Computer Science, Northeastern University, China, in 1988, 1991, and 1996, respectively. Currently, he is a professor in the Department of Computer Science, Northeastern University, China. His research interests include XML data management, query processing and optimization, bioinformatics, high-dimensional indexing, parallel database systems, and P2P data management.

He has published more than 100 research papers.



**Lei Chen** received the bachelor's degree in computer science at Tianjin University, China, in 1994, and the master's degree in computer science at the Asian Institute of Technology in 1997. He received the PhD degree in computer science at University of Waterloo, Canada. He is currently an assistant professor of computing science at Hong Kong University of Science and Technology, China. His research interests include multimedia databases, graph databases, uncertain and probabilistic databases.



**Junchang Xin** received the BSc, MSc, and PhD degrees in computer science and technology from the Northeastern University, China, in July 2002, March 2005, and July 2008, respectively. He is currently a lecturer in the Department of Computer Science, Northeastern University, China. His research interests include data management over wireless sensor network and uncertain data management.



**Yunhao Liu** received the BS degree in automation from Tsinghua University, China, in 1995, the MS, and a PhD degrees in computer science and engineering from Michigan State University in 2003 and 2004, respectively. He is a member of Tsinghua National Lab for Information Science and Technology, professor in the School of Software at Tsinghua University, and the director of the Tsinghua National MOE Key Lab for Information Security. He is also a faculty in the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology. He is a senior member of the IEEE, and is also an ACM Distinguished Speaker.

► For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).