COSE: A Query-Centric Framework of Collaborative Heterogeneous Sensor Networks

Yuan He, *Member*, *IEEE*, and Mo Li, *Member*, *IEEE*

Abstract—Demands on better interacting with physical world require an effective and efficient collaboration mechanism of multiple heterogeneous sensor networks. Previous works mainly focus on each single and specific sensor network, thus failing to address issues in the newly emerging scenario. In this paper, we propose COSE, a query-centric framework of collaborative heterogeneous **sen**sor networks, where sensor networks collaborate with each other for effective and efficient processing of queries. Finding an optimal strategy of query processing with respect to energy efficiency is a crucial issue in COSE, which we formulate into an optimization problem, called EE-QPS. We prove the NP-hardness of EE-QPS, and then design a heuristic approach named IAP by utilizing the correlation (called *implication* in this paper) among different sensor networks. The experimental results demonstrate that in the context of COSE, IAP achieves optimized energy efficiency under various settings.

Keywords—Sensor network, query processing, energy.

1 Introduction

Due to the recent advances in wireless communication and microelectronic technologies, both the price and size of sensors have decreased quickly. Today's applications for sensor networks range from personal to mission critical systems including scientific observation, digital life, home automation, environment surveillance, traffic monitoring, and so on [1], [2], [3]. Many of them are developed and promoted by governments, enterprises, and public organizations, offering continuous collection of real-time information, fulfilling the requirement of people's daily lives.

In the foreseeable future, we expect to witness the proliferation of sensor networks with a variety of functions that require a comprehensive collaboration mechanism among them. Specific designs are necessary to manipulate a fabric of multiple sensor networks, facilitate the collaboration among them, and support efficient query processing. Previous studies in sensor networks, however, mainly focus on the performance and efficiency inside a single sensor network [1], [2], [3]. In this work, we broaden the research into the scope of multiple sensor networks.

This study is indeed motivated by a practical application of Qinhuangdao Oilport in China. Timely planning is required as a critical part in the management of oil production. The oil production flow is related to many factors such as oil supplies in oil fields, the flux capacity of oil pipeline transmission, landway traffic, and environments of harbors. Formerly, we can only make relatively static decisions based on coarse estimations on these factors. The output of decisions often suffers from the dynamics of these factors, causing unredemptive loss of profit and even serious accidents. Therefore, a number of preliminary wireless sensor networks (WSNs) are deployed along the oil production flow to obtain live environmental data.

To truly utilize WSNs in the above application, however, many challenges need to be addressed. The multiple sensor networks are usually heterogeneous, namely adopting different sensors, sensing different types of data, using different communication protocols, and powered by different energy sources. Previous studies mainly concentrate on data collection and query processing in a single sensor network [1], [2], [3]. Using these approaches, we can only obtain isolated and incomplete results, inevitably leading to unilateral and even incorrect decisions. Also, the sensor networks continuously generate huge volumes of data with various attributes, simply gathering all the data and processing them in a centralized manner is communication-intensive. Thus, distributed sensing and collaborative query processing among multiple sensor networks are indispensable for the above application. Moreover, the sensor networks are likely to receive substantive complex ad hoc queries, while the sensors are usually energy constrained and not easily rechargeable. Therefore, energy-efficient query processing with multiple sensor networks is a crucial issue but has never been studied before. Considering that all the sensor networks are spatially distributed and independent, how to enable them to effectively collaborate is a challenging issue, even if the optimal strategy of query processing is provided.

To address the above challenges, we propose COSE, a query-centric framework of collaborative heterogeneous

- *Y. He is with the Tsinghua National Laboratory for Information Science and Technology and the School of Software, Tsinghua University, Room 501, Block A, Liye Building, SI-Park, Taike Park, New District, Wuxi, Jiangsu 214000, China. E-mail: he@greenorbs.com.*
- *M. Li is with the School of Computer Engineering, Nanyang Technological University, N4-02C-108, 50 Nanyang Avenue, Singapore 639798. E-mail: limo@ntu.edu.sg.*

Manuscript received 31 Aug. 2011; revised 19 Jan. 2012; accepted 26 Jan. 2012; published online 13 Feb. 2012.
Recommended for acceptance by S. Papavassiliou, N. Kato, Y. Liu, C.-Z. Xu, and X. Wang.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDSSI-2011-08-0586.
Digital Object Identifier no. 10.1109/TPDS.2012.72.

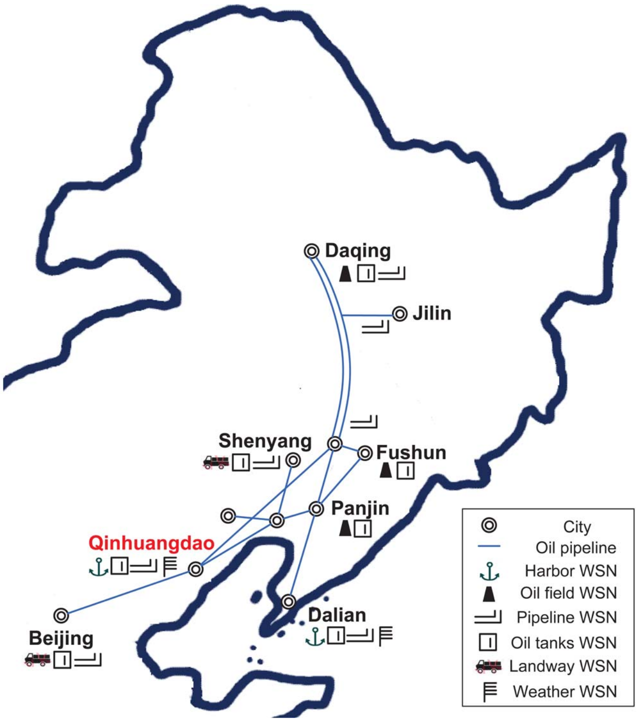


Fig. 1. Oil production map in Northeast China.

sensor networks. Our main idea is to utilize the correlations (called *implication* in this paper) among heterogeneous sensor networks to reduce the communication cost incurred by query processing and forwarding among the sensor networks. The major contributions of this work are as follows:

1. We propose the framework of collaborative heterogeneous sensor networks and an implication-aware scheme called sink-overlay to organize heterogeneous sensor networks.
2. We formulate the optimization problem of query processing in COSE and prove its NP-hardness.
3. We design a mechanism to estimate the implications among sensor networks, based on which we propose efficient algorithms to schedule the pipeline of query processing.

The rest of this paper is organized as follows: Section 2 introduces the motivating application of this paper. The design of COSE is presented in Section 3. Section 4 studies the optimization problem of query processing in COSE, proves its NP-hardness, and presents heuristic algorithms to solve it. Section 5 presents the performance evaluation results, followed by brief review of the related work in Section 6. We conclude in Section 7.

2 Motivation

Qinhuangdao Oilport is the world's largest harbor for energy sources exportation. As shown in Fig. 1, the major industrial cities and oil fields in North and Northeast China are involved in a giant oil production flow, including oil extraction, refining, storage, pipeline transmission, landway transportation, and shipping. As the juncture of North China and Northeast China, Qinhuangdao Oilport and several other harbors are in charge of transferring oil among the oil fields and refineries in different cities as well as shipping and exporting oil to foreign countries.

Timely planning is necessary to keep the entire production flow safe and efficient. It is, however, extremely challenging to plan the production flow which is affected by many factors. For example, the environments in oil fields determine the production rate and supply volume of oil. The condition in pipelines determines the safety of oil transportation and the capacity of pipeline transmission. The busyness of railway and road traffic affects the landway transportations. Atrocious weather at the harbor might lead to harbor clogging and delay the shipping. The availability of oil tanks affects the harbor capacity and thus limits the capacity of oil handlings. The siltation of the sea route in the harbor results in the delay of all the scheduled cargos. It might get even worse if the oil transportation is not scheduled in time, resulting in a chain reaction in the shortage of oil tanks, congestion of pipelines, and landway transportation in disorder. With the assistance of widely deployed sensor networks, it is much easier to make real-time and accurate decisions based on the live sensor data reflecting the dynamics of all affecting factors. Specifically, we can check the busyness of the landway traffics as well as the oil pipelines, and schedule a railway transportation of the oil from a highly productive oil field to an idle harbor in good natural conditions. Note that making such a decision relies on the collaboration among many sensor networks as the oil field WSNs for production surveillance, the oil pipeline WSNs for safety surveillance and flux monitoring, the landway WSNs for real-time traffic monitoring, the weather WSNs in the coastal cities, and the harbor WSNs for environment monitoring and harbor surveillance.

The goal of COSE is to enable the collaboration among those sensor networks deployed separately. In COSE, each sensor network is treated as a data source, which accepts external queries from the web portal and provides instant information at the cost of internal resource and power consumption. Note that most sensor networks involved in the above application are deployed in wild areas where human beings seldom reach and are expected to persist for long. The energy efficiency of sensor networks has all along been a critical concern. A straightforward solution is that we independently query every sensor network and gather all corresponding information for local analysis. Then, each query will potentially involve all the sensor nodes in each sensor network. Such a blind querying scheme incurs excessive energy consumption, which obviously ruins the sensor networks for long-term uses.

Consequently, a crucial yet challenging issue in COSE is how to efficiently query the sensor networks and obtain the desired information with the minimum total energy cost in all the sensor networks.

In this study, we show that due to the natural interdependence in the physical world, data of different sensor networks are usually correlated with each other. For example, sensor data of temperature and humidity, Ultraviolet Index and illumination, the road traffic and the busyness of parking lots, the flux of oil pipelines, and the available capacity of oil tanks. We can partially infer the data of a sensor network based on the data of another one, as long as they are correlated. We call such correlations among sensor networks *implication*.

Implications can be utilized to save the total energy cost of query processing. Specifically, when we process a query involving multiple sensor networks, the data from previously queried sensor networks can be used to partially infer the data of the subsequently queried sensor networks. Therefore, it costs the subsequent ones fewer operations (including sensing and communication) to obtain the necessary data. The total energy cost to process this query is thus reduced. For example, in the motivating application, when setting a schedule for the oil production flow, after we obtain the information from the harbor surveillance sensor networks, we may sweep off those harbors under infeasible condition. In the subsequent stages, we only query the status of traffic and oil pipelines from only a portion of sensor networks related to the feasible harbors, saving unnecessary operations.

Therefore, in a query-centric framework of collaborative heterogeneous sensor networks, it is of great significance to schedule the sequence of query processing to achieve the optimized overall energy efficiency by fully utilizing the implications among sensor networks. Achieving the optimized schedule, however, is nontrivial in COSE. In later discussion, we will first formulate the query optimization problem and prove its NP-hardness even with full knowledge of all the affecting factors. Second, it is yet hard to obtain the implications, which vary with different pairs of sensor networks along time. Third, most operations in COSE must be executed in a distributed fashion, such as data sharing among sensor networks and implication quantification.

3 Design

This section elaborates the design of COSE. Since COSE is built upon numerous sensor networks, we need to provide a general mechanism of membership management and information sharing among the sensor networks. In the following sections, we, respectively, introduce the basic infrastructure of COSE, formalize the framework of collaborative heterogeneous sensor networks as data sources with different attributes, and present the sink-overlay construction.

3.1 Basic Infrastructure

COSE consists of a central manager (CM) and numerous collaborative heterogeneous sensor networks. Both CM and the sinks of sensor networks are connected over the Internet. CM performs the following functions:

1. To manage the membership and coordinate sensor networks in the sink-overlay. CM maintains a list of all active sensor networks in COSE. The format of the information stored for each sensor network together with the formalization is described in Section 3.2.
2. To provide a uniform web portal that accepts external queries and outputs responses. Though web portal is not indispensable in COSE, it makes it convenient for the users to issue queries and obtain the live data.
3. To schedule the pipeline of query processing. Considering the aforementioned energy constraints, query processing in COSE is an optimization problem with respect to energy efficiency in sensor networks. As we elaborate in Section 4, CM plays the role of scheduling query processing. It determines the pipeline of processing, and collects data from the queried sensor networks.

On the other aspect, the sinks serve as gateways for interconnecting the sensor networks in COSE. They self-organize into a multidimensional sink-overlay. Sink-overlay construction is introduced in Section 3.3.

3.2 Formalization of Sensor Networks

COSE is an integrated framework that manipulates collaboration among sensor networks. Information exchange is frequent in the sink-overlay. We provide a uniform formalization for sensor networks including the elementary properties, such as location, scale, function, energy, and data attributes, etc. This allows us to formalize a sensor network as a data source that produces meaningful data at quantified costs. We formalize a sensor network W by the following expressions. W is a sensor network in COSE containing four elementary domains

$$\begin{aligned} W &= <Basic, Energy, Data, Overlay>, where \\ Basic &= <Name, Location, Scale> \\ Energy &= <Unit\ cost, Current\ capacity> \\ Data &= <Period\ of\ validity, Attribute\ W_1, \\ &\quad Attribute\ W_2 \ldots> \\ Overlay &= <Neighbor\ List1, Neighbor\ List2 \ldots> \end{aligned}$$

The *Basic* domain differentiates sensor networks and provides information for the potential collaboration among them. It includes the basic information of a sensor network, such as *Name, Location*, and *Scale*. *Name* is a unique ID assigned by the owner of the sensor network to distinguish from other sensor networks. *Location* [4] is the geographical location where the sensor network is deployed, for instance, longitude and latitude. *Scale* denotes the size of the sensor network, usually represented by the number of sensors contained.

The *Energy* domain is utilized to evaluate the query overhead in a sensor network, providing necessary information for the query optimization process. *Unit cost* represents the average unit energy cost for a sensor to sense and transmit data once, measured by nJ. Here, we do not differentiate sensing, transmission or other in-network operations and wholly regard them as a unit when measuring the energy consumption. *Current capacity* denotes the current total energy capacity of the sensor network. Also we neglect the diversity of energy distribution on individual sensors and focus on the energy efficiency of sensor networks and the interactions among them from a macro point of view. In practice, the unit cost of sensing may be measured by the product of the sensor's current under active sensing mode and the time needed to sense once. The current capacity may be measured by monitoring the battery voltage. There are also existing techniques [5] to precisely track the energy-related factors.

The *Data* domain describes the sensor data of a sensor network and relates sensor networks through different attributes. It is utilized in the sink-overlay construction and data sharing. *Period of validity* denotes the time for the sensed data to be valid. The succeeding sequence in the *Data* domain lists all the physical attributes of the sensor data. For example, a thermometer sensor network usually generates the data with temperature attributes. A sensor network monitoring seawater is able to sense data with three attributes, including temperature, depth, and salinity.

The *Overlay* domain reveals the structure of sink-overlay and directs data sharing among sensor networks. It specifies the neighboring sensor networks of the current sensor network in the sink-overlay. Here, we define neighbor to be a sensor network connected with the current sensor network via a sink-overlay connection, as introduced in the next section.

3.3 Sink-Overlay Construction

As mentioned in Section 3.1, sink-overlay facilitates query-centric processing in COSE. When constructing the sink-overlay, it is necessary to investigate the implications among sensor networks, which benefits query processing, especially in terms of energy efficiency. In general, sensor networks with implications to each other can be classified into two cases: 1) they are deployed with spatial proximity, i.e., geographically close to each other; or 2) they have functional proximity. In other words, they probably have common attribute(s) in their sensor data. Existing data correlations have been validated by both practical deployments and theoretical models [2], [6].

Since all components of COSE are connected through the Internet, the sink-overlay does not relate to the connectivity among sensor networks. It is an application-layer abstract which facilitates data sharing and query processing. Connections in the sink-overlay indicate the relationship of implications and potential collaboration among sensor networks.

Based on the above foundations, we construct the sink-overlay using an implication-aware method. The overlay connections are categorized into two types, *local connection* and *attributed connection*, which are built among the sinks of the sensor networks. A local connection connects a sensor network with another one in its adjacent area. An attributed connection is built between two sensor networks that have common data attribute(s).

The sink-overlay connections are constructed as follows: When a new sensor network joins COSE, it registers its information at CM. Upon receiving the registration, CM checks its local database to returns a list of candidate neighbors to the new sensor network. Each candidate neighbor has spatial proximity or(/and) functional proximity with the new one. CM determines this list according to the information of the *Basic* and *Data* domains of the sensor networks. After receiving the list, the new sensor network connects to all the neighbors in the list, either with *local connections* or *attributed connections*. This connectivity information is further used to update the *Overlay* domains of corresponding sensor networks.

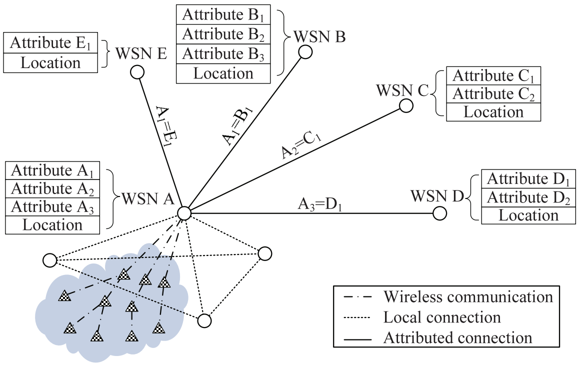


Fig. 2. The sink-overlay.

Fig. 2 shows an example of sink-overlay. Suppose sensor network A newly joins in with three data attributes $<A_1, A_2, A_3>$. First it connects with the three geographically adjacent sensor networks. Then, A connects with sensor network B because $A_1 = B_1$. In other words, sensor networks A and B have a common attribute. Similarly, A connects with sensor networks C, D, and E.

3.4 Data Sharing Policy

Data sharing is carried out among neighboring sensor networks as a manner of measuring the implications among them. Besides on-demand sensing to resolve queries, every sensor network periodically collects its sensor data to track the status in the network. The sensor data from periodical and on-demand sensing are stored at the sinks until expiration. After a preconfigured period, every sensor network exchanges the latest sensor data with all its neighbors so as to keep them updated. Note that the frequency of periodical data sensing is configured much lower than that of on-demand sensing. Thus, the overhead of data sharing through the Internet is not a significant issue compared with saving the energy costs in the energy-constrained sensor networks. Since the sink-overlay neighbors have spatial or functional proximity, data sharing among neighboring sensor networks enables local collaboration among heterogeneous sensor networks, as well as cross-area collaboration among sensor networks with similar functionalities.

4 Query Processing in COSE

We focus on processing complex ad hoc queries (or complex query in short), which has not received much attention before. A complex query acquires the data of multiple sensor networks. Different with data stream queries, complex queries are not predefined or repeated with data streams. While the sink-overlay in COSE facilitates data sharing among the sensor networks, a challenging problem is how to process a complex query involving multiple sensor networks with minimal energy cost. We address this issue by emphasizing the implication-aware collaboration among the sensor networks. Instead of digging into the concrete operations (such as data sensing, filtering, aggregation, and caching) and correlation patterns of sensor data, we focus on the impact of implication and the methodology to utilize it, so that we can minimize the total energy cost of query processing in COSE. Note that COSE determines the schedule of sensor networks for query processing along the pipeline. It does not affect the original transfer/aggregation mechanism [7], [8] of any sensor network.

In this section, we first elaborate how a query is resolved in COSE and formulate this process into an optimization

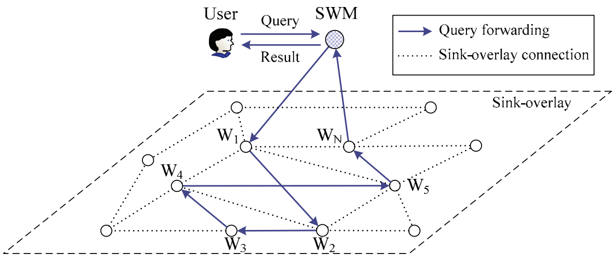


Fig. 3. The query-centric framework.

problem. We further prove it is NP-hard. Then, we propose a heuristic approach to resolve it, which achieves optimized energy efficiency.

4.1 The Query Optimization Problem

As we elaborated in Section 2, implications can be utilized to save the energy cost of query processing. Hence, in COSE, a complex query is processed by a pipeline of the involved sensor networks, so that the implications among sensor networks can be fully utilized.

Suppose a query Q concerns a subset of sensor networks in COSE, say $\{W_1, W_2 \dots W_N\}$. As shown in Fig. 3, we denote a sensor network as a node, ignoring the concrete structures. The pipeline to resolve a complex query can be depicted as a directed cycle in the graph, starting and ending at CM. At the beginning, query Q is received from the web portal and interpreted. Then, CM selects W_1 to forward query Q. After filtering the data gathered from W_1, query Q is regulated by further constraining its selection predicates (while query regulation is out of the discussion of this paper, we just enumerate all the phases of query processing). Then, query Q and the filtered data are passed to W_2. Similarly, after W_2 finishes its work, it passes query Q and the filtered sensor data to W_3. The process continues until all the N sensor networks have been accessed one by one. In the end, the complete result of query Q is returned from the last sensor network W_N to CM, and then output to the user.

It might be argued that energy efficiency of the pipeline is achieved at the cost of response latency of a query. In fact, the time required to collect all the data in a sensor network is usually less than a few seconds. Therefore, the pipeline-processing still satisfies the requirement of response latencies. Data processed by the pipeline can be transmitted from one WSN to another directly through Internet, which does not require any relaying WSN.

4.2 Problem Formulation

The Energy-Efficient Query Processing in COSE (EE-QPS) problem can be modeled with a directed weighted graph $G = (V, E)$. Suppose we have N sensor networks involved. V is the set of nodes. Let nodes $v_1, v_2, \dots, v_N$ represent the sensor networks $W_1, W_2, \dots, W_N$. E is the set of edges representing implications. Edge e_{ij} is the edge from v_i to v_j, and s_{ij} is the weight of e_{ij} $(i \neq j)$.

We define s_{ij} as the index of implication from W_i to W_j. Instead of directly measuring the correlation between W_i and W_j, s_{ij} is a factor to quantify the proportion of information in W_j that remains uncertain when the data of W_i are known. Specifically, for a certain query involving W_i and W_j, the quantity of eligible results returned from W_j is invariable. When W_i is first queried, the smaller s_{ij} is, the more information of W_j can be inferred from the data of W_i, and the less information of W_j needs to be collected afterwards. Note that s_{ij} is not necessarily equal to s_{ji}. Specially, $s_{ii} = 1$. For $i \neq j$, when W_j is completely independent from $W_i, s_{ij} = 1$; when W_j can be completely inferred from $W_i, s_{ij} = 0$.

We define C_i as the original energy cost in sensor network W_i (measured by nJ) incurred by a query, when no information of W_i is inferred from other sensor networks. According to the formalization in Section 3.2, $C_i = Unit\ cost \times Scale$ for W_i. In order to balance the load among different sensor networks, we further take the remaining energy capacity into consideration. Thus, the normalized value is used in the subsequent computations. Abusing notations, we still use C_i to denote it. $C_i = Unit\ cost \times Scale/Current\ capacity$ of W_i.

For a complex query, the involved sensor networks are correlated with each other. Therefore, the cost reduction in a subsequently queried sensor network is an accumulative effect caused by all the previously queried ones. It is difficult to accurately estimate the cumulative effect of implications among sensor networks, especially in dynamic and unpredictable environments. As a simplified case, we assume all s_{ij} are independent from each other. Thus, the aforementioned accumulative effect can be quantified by multiplying the indices of implications from all the upstream sensor networks along the pipeline to the current one.

Taking Fig. 3 as an example, we process a query through the pipeline $(W_1 \rightarrow W_2 \rightarrow \cdots \cdots \rightarrow W_N)$. The querying cost P_i incurred in sensor network W_i is calculated by

$$P_i = C_i \times \prod_{1 \leq j \leq i} s_{ji}.$$

For convenience, we set $s_{ii} = 1$. The total cost P incurred in all the N sensor networks is calculated by

$$P = \sum_{i=1}^{N} P_i = \sum_{i=1}^{N} \left(C_i \times \prod_{1 \leq j \leq i} s_{ji} \right). \tag{4.1}$$

Clearly, we have $N!$ options to schedule the pipeline of query processing. Meanwhile, due to the natural heterogeneity, the indices of implications probably vary a lot with different pairs of sensor networks. Therefore, different pipelines present great difference in the total energy costs. Toward the same query result, a well scheduled pipeline incurs much less energy cost than the poorly scheduled ones. And for any complex query, there exists an optimal pipeline scheduling, which incurs the minimum total cost in resolving it. Formally, the EE-QPS problem in COSE is formulated as follows:

INSTANCE. A sequence of positive constants $(C_1, C_2, \dots, C_n)$, where C_i denotes the normalized original cost in sensor network W_i incurred by a query. Correspondingly, there is an implication matrix

$$S_{N,N} = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1N} \\ s_{21} & s_{22} & \cdots & s_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ s_{N1} & s_{N2} & \cdots & s_{NN} \end{bmatrix}, \tag{4.2}$$

where $0 \leq s_{ij} \leq 1, s_{ii} = 1$ for all integers i and j in $[1, N]$.

SOLUTION. $(a_1, a_2, \ldots, a_N)$, which is a permutation of $(1, 2, \ldots, N)$.

MEASURE.

$$P = \sum_{i=1}^{N} \left(C_{a_i} \times \prod_{1 \leq j \leq i} s_{a_j a_i} \right),$$

which is the total cost of all the involved sensor networks. The optimal solution of the problem minimizes the value of P, i.e., achieves the best energy efficiency.

4.3 Hardness of EE-QPS

In this section, we theoretically analyze the complexity of finding the optimal solution to the EE-QPS problem.

Theorem 4.1. *The EE-QPS problem is NP-hard.*

Proof. We show a reduction from SET-COVER problem. In SET-COVER, we are given a universe U of m elements and a collection of sets $\Gamma = \{S_1, S_2, \ldots, S_n\}$,

$$S_i \subseteq U, \bigcup_{1 \leq i \leq n} S_i = U.$$

The goal is to decide, for some given k, if there exist no more than k sets in Γ whose union is U.

Given an instance of SET-COVER, we construct a directed weighted graph, where a node is mapped to a set or an element in the instance of SET-COVER. There are $(n + m)$ nodes in the graph, denoted by $v_1, v_2, \ldots, v_{n+m}$. We use $v_1, v_2, \ldots, v_n$ to denote the n sets while $v_{n+1}, v_{n+2} \ldots v_{n+m}$ to denote the m elements. Abusing notations, we also use $v_i (1 \leq i \leq n + m)$ to represent its corresponding set or element.

First, we set $C_i = 1$ if node v_i denotes a set while $C_i = n + 2$ if node v_i denotes an element. Moreover, we add an auxiliary node v_{n+m+1} into the graph, which does not represent a set or an element. We set $C_{n+m+1} = n^m$.

Second, we set the weights of edges as follows:

$s_{ij} = 0$, if v_i is set and v_j is element, $v_j \in v_i$;
$s_{n+m+1,i} = 0$, if v_i is a set;
$s_{j,n+m+1} = 1/n$, if v_j is an element;
$s_{ij} = 1$ for all the other cases.

Thus, we obtain a matrix $S_{n+m+1,n+m+1}$, where each of the first n rows/columns corresponds to a set, each of the following m rows/columns corresponds to an element, and the last row/column corresponds to the auxiliary node v_{n+m+1}

$$S_{n+m+1,n+m+1} = \begin{array}{c} \begin{array}{cccccccccc} \multicolumn{4}{c}{\overbrace{\hphantom{1\quad 1\quad \cdots\quad 1}}^{n}} & \multicolumn{4}{c}{\overbrace{\hphantom{0\quad 1\quad \cdots\quad 0}}^{m}} & \end{array} \\ \begin{bmatrix} 1 & 1 & \cdots & 1 & 0 & 1 & \cdots & 0 & 1 \\ 1 & 1 & \cdots & 1 & 1 & 1 & \cdots & 0 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & \cdots & 1 & 0 & 0 & \cdots & 1 & 1 \\ 1 & 1 & \cdots & 1 & 1 & 1 & \cdots & 1 & 1/n \\ 1 & 1 & \cdots & 1 & 1 & 1 & \cdots & 1 & 1/n \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & \cdots & 1 & 1 & 1 & \cdots & 1 & 1/n \\ 0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 & 1 \end{bmatrix} \end{array}.$$

Now we have constructed an instance for the EE-QPS problem. It is obvious that reduction above is polynomial. Next we show that solving this instance of EE-QPS also solves the original instance of SET-COVER.

First we show that if the answer to the SET-COVER instance is "yes," then there is a solution to the above instance of EE-QPS with a cost no more than $k + 1$.

Suppose $(v_1, v_2 \ldots v_t)$ is a solution to the SET-COVER problem $(t \leq k)$, we may construct a solution $(a_1, a_2 \ldots a_{n+m+1})$ to the instance of EE-QPS as

$$(a_1, a_2 \ldots a_{n+m+1}) = (v_1, Z_1, \ldots v_t, Z_t, v_{n+m+1}, V'), \qquad (4.3)$$

where every $Z_i (1 \leq i \leq t)$ represents a sequence of elements not covered by sets $v_1, v_2 \ldots v_{i-1}$ but covered by set $v_i.V'$ is the sequence of the (n-t) sets which are not selected by the solution $(v_1, v_2 \ldots v_t)$. According to the above value assignments, the total cost of the solution $(v_1, Z_1, \ldots v_t, Z_t, v_{n+m+1}, V') = t + 1 \leq k + 1$.

Next, we show that if there is a solution to the above instance of EE-QPS with a total cost no more than $k + 1$, then the answer to the original SET-COVER instance is "yes."

Suppose $(a_1, a_2, \ldots a_{n+m+1})$ is a solution to the instance of EE-QPS, which has a total cost of no more than $k + 1$. According to the above value assignments, we have the following deductions.

1. The auxiliary node v_{n+m+1} must appear after all the elements. Otherwise, the total cost is $> n + 1 \geq k + 1$, which contradicts with the presumption.
2. An element cannot appear until at least one set covering it appears. Otherwise, the total cost is $> n + 2 > k + 1$.

Third, the appearance of the auxiliary node v_{n+m+1} in $(a_1, a_2 \ldots a_{n+m+1})$ separates all the sets into two parts. Suppose there are t sets that appear before v_{n+m+1}, then the total cost of $(a_1, a_2 \ldots a_{n+m+1}) \geq t + 1$. Therefore, $k + 1 \geq t + 1$, i.e., $k \geq t$. According to the above deductions, these sets cover all the elements. Thus, we find a solution to the SET-COVER problem by selecting the t sets that appear before v_{n+m+1} in $(a_1, a_2 \ldots a_{n+m+1})$. Since $k \geq t$, step 2 is proved.

By combining 1 and 2, we have proved that the SET-COVER problem is polynomial-time reducible to EE-QPS. Hence, EE-QPS is NP-hard. □

4.4 Pipeline Scheduling

Due to the NP-hardness of EE-QPS problem, we need a heuristic approach to resolve it. The environment of COSE produces the following challenges:

First, sensor networks in COSE are deployed separately and preserve independency to each other. It is difficult to host all the sensor data on a sole server, if not impossible. Second, since COSE integrates numerous sensor networks, it is expensive to share the sensor data throughout the entire COSE. Thus, any sensor network in COSE can only have partial knowledge. Third, considering the characteristics of sensor networks, their statuses change frequently and are hard to predicate. It is nontrivial to accurately quantify the indices of implications between any two of them.

Bearing these points in mind, we design the heuristic approach, *Implication-Aware Pipeline* (IAP) as follows: a sensor network periodically quantifies the indices of implications from itself to all its neighbors. Only indices of implications are reported to CM and kept updated. Based on the latest estimated implications, we use A* algorithm to find the optimal scheduling of query processing. A greedy algorithm is also proposed for faster decisions, which outputs the close-to-optimal scheduling of query processing.

4.4.1 Quantification of Implication

Since the data in sensor networks change frequently, implications are periodically estimated in COSE. Suppose X and Y are neighboring sensor networks. We use information entropies to quantify the indices of implications from sensor network X to sensor network Y. Specifically, what we need is a factor to quantify the proportion of information in sensor network Y that remains uncertain when the data of X are known. An important point that is worth noticing here is implication is asymmetric in general. The mathematical properties of conditional entropy well fit our need in the problem formulation. Thus, we define

$$s_{XY} = \frac{H(Y|X)}{H(Y)} = \frac{-\sum_i \sum_j P(x_i, y_j) \log P(y_j|x_i)}{-\sum_j P(y_j) \log P(y_j)}. \tag{4.4}$$

We directly use X and Y to represent the data sets of X and Y while x_i, y_j are the corresponding sensor data, respectively. $H(Y)$ is the original entropy, denoting the original uncertainty of data set Y. $H(Y|X)$ is the conditional entropy, denoting the uncertainty of data set Y when data set X is already known. The conditional probability $P(y_j|x_i)$ is calculated with those x_i and y_j falling into the same *Period of validity*. For instance, $P(y_j = u|x_i = v)$ is the probability of y_j to be u, given the current value of x_i to be v. The quotient $s_{XY} = H(Y|X)/H(Y)$ denotes the proportion of information in Y that remains uncertain and needs to be measured when X is known. It appropriately expresses the physical meaning of implication from X to Y.

According to the data sharing policy introduced in Section 3.3, a sensor network knows the latest data of its neighbors, so X can easily calculates s_{XY}. After the calculations, all sensor networks upload the updated indices of implications to CM. Thus, CM obtains a global view of implications in COSE while needs not collect the detailed sensor data.

Because a sensor network shares its data only with its neighbors, i.e., a small portion of sensor networks in COSE, it is possible that some s_{XY} involved in pipeline scheduling has not been calculated before the first time it is enquired. In case this happens, CM sets the initial value of s_{XY} to be 1, which means Y is initially assumed to be independent from X. Meanwhile, CM sends a message to the sinks of X and Y, requesting them to build a new connection in the sink-overlay. The implications between X and Y can be calculated from then on. Moreover, indices of implications are updated through periodical data exchanges so that the accuracy of quantifications is adaptive to the dynamic environments of sensor networks.

4.4.2 Pipeline Scheduling of Query Processing

In this section, we first present the optimal scheduling of query processing. Then, we give a greedy algorithm that achieves close-to-optimal scheduling.

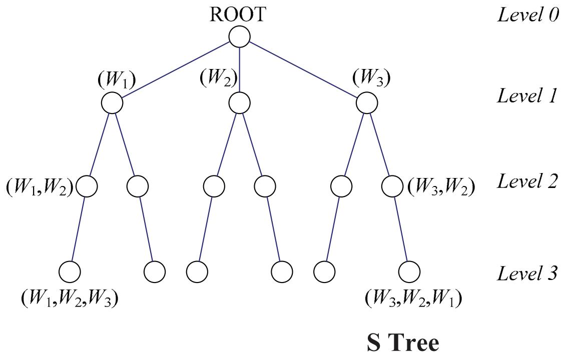


Fig. 4. An EE-QPS tree.

1. *The Optimal Scheduling*

Due to the NP-hardness of EE-QPS, it is very unlikely that one can solve the problem in polynomial time. Nevertheless, when the number of sensor networks involved in a query is small, we can adopt the A* algorithm [9] to find the optimal scheduling of the pipeline. Given an instance of EE-QPS with N sensor networks $(W_1, W_2, \ldots, W_N)$, the scheduling process can be illustrated with a tree called EE-QPS Tree. Fig. 4 shows an EE-QPS Tree to schedule 3 sensor networks. This tree roots at an empty node. The nodes at level i represent all the sequences of i sensor networks. Therefore, each leaf node in the tree, respectively, represents a candidate scheduling. Suppose M is a node at level i and M' is a node at level $i+1$. R and R' are the corresponding sequences of M and M'. There is an edge from M to M' if and only if R' can be constructed by inserting the $(i+1)$th sensor network to the tail of R.

Suppose $R = (v_1, v_2, \ldots, v_i)$, we define a heuristic function for node M as follows:

$$F(M) = G(M) + H(M), \text{where}$$

$$G(M) = \sum_{k=1}^{i} \left(C_{v_k} \times \prod_{1 \le j \le k} s_{v_j v_k} \right)$$

$$H(M) = \sum_{W_t \notin R} \left(C_t \times \prod_{1 \le j \le N} s_{jt} \right).$$

$G(M)$ is the current total cost incurred in sensor networks $(v_1, v_2, \ldots v_i)$. $H(M)$ is the estimated lower bound of total cost incurred in the subsequent sensor networks, which are not included in R. Thus, $F(M)$ is the cost corresponding to node M.

We then use the A* algorithm to find the minimum-cost leaf node in EE-QPS Tree. Since $H(M)$ is a lower bound on the future cost, the result returned by A* is guaranteed to be optimal [9]. Although in the worst case the algorithm will visit all the $O(N!)$ leaves, on most typical inputs the A* algorithm can stop much earlier.

2. *The Greedy Algorithm*

When the number of sensor networks involved in a query gets larger, it will be computationally infeasible to execute the A* algorithm. Hence, we design a greedy algorithm to find a close-to-optimal scheduling, the temporal complexity of which is only $O(N^2)$.

TABLE 1a
Implications among Different Sensors

	Temperature	Humidity	Illumination
Temperature	1	0.74	0.59
Humidity	0.28	1	0.34
Illumination	0.25	0.56	1

TABLE 1b
Implications between two WSNs in CitySee

	Temperature-NetA	CO_2-NetB
Temperature-NetA	1	0.69
CO_2-NetB	0.35	1

Given an instance of EE-QPS with N sensor networks $(W_1, W_2, \dots, W_N)$, we suppose $(V_1, V_2, \dots, V_N)$, a permutation of $(W_1, W_2, \dots, W_N)$, is the final scheduled pipeline. Then, the process of scheduling can be divided into $N+1$ states $T_0, T_1, \dots, T_N$, where T_i refers to the state when the first i sensor networks of the pipeline have been selected. Correspondingly, we define two sets R and R'. Given state T_i, R contains the first i sensor networks that have been determined in the pipeline, while R' contains the $(N-i)$ unselected ones. We define a heuristic function as follows to select the $(i+1)$th sensor network of the pipeline

$$U(v) = C_v \prod_{x \in R} s_{xv} + \sum_{y \in R' - \{v\}} \left(C_y \times s_{vy} \times \prod_{x \in R} s_{xy} \right). \tag{4.5}$$

The parameters C_v, C_x, and C_y are the original costs, which can be known from the basic information of the sensor networks. s_{xv}, s_{xv}, and s_{vy} are the indices of implications quantified as above. $U(v)$ is the sum of two parts. The first part is the energy cost in v if it is selected as the $(i+1)$th sensor network. The second part is the upper bound of total energy cost in the remaining $(N-i-1)$ unselected sensor networks, if v is selected as the $(i+1)$th sensor network. $U(v)$ denotes the maximal energy cost incurred in the remaining $(N-i)$ sensor networks if v is selected as the $(i+1)$th sensor network of the pipeline.

Therefore, the $(i+1)$th sensor network of the pipeline should be sensor network v which minimize $U(v)$, expressed as follows:

$$V_{i+1} = \arg\min_{v \in R'} U(v). \tag{4.6}$$

Subsequently, V_{i+1} is removed from R' and added into R. After N rounds of selection, the pipeline is finally decided. As soon as the pipeline is scheduled, the query is passed from CM to sensor network V_1, then V_1 to V_2, then V_2 to V_3 and so on. In the end, the query is finished on V_N. The final result is then returned to CM and output to the user.

5 Experiments

We conduct experiments to validate our scheme and evaluate its performance. In Section 5.1, we report our measurement on the implications among real sensor networks. The results demonstrate the universality of the mutual implications in practical sensor networks. We then evaluate the proposed IAP approach with large-scale simulations. The results are presented in Section 5.2.

5.1 Real-World Observations

We observe a real deployed sea monitoring system in our OceanSense project [10], which acts as a preliminary attempt toward COSE. OceanSense aims to build an integrated information system of multiple sensor networks for environment surveillance on the sea. Note that the data we provide here are to express the implication among heterogeneous data. We examine a 24-hour data set collected from OceanSense. Using (4.4), we calculate the indices of implications among three types of data (temperature, humidity, and illumination). As shown in Table 1a, entry (X, Y) denotes the index of implication from sensor network X to sensor network Y.

We also carry out a new experiment with CitySee [11]: an urban sensing system for CO_2 Monitoring. CitySee currently deploys over 1,000 sensor nodes in four sensor networks. For ease of presentation, let's name two of the sensor networks as NetA and NetB. The geographical distance between NetA and NetB is at least 1 kilometer. We then observe the temperature readings on a sensor in NetA and the CO_2 readings on another sensor in NetB for 24 hours. Using (4.4), we calculate the implications between the two sensor networks as shown in Table 1b.

The results in Table 1 typically reflect the existence of implications among heterogeneous sensor networks, validating the assumption of our scheme.

5.2 Performance Evaluation

We conduct several groups of simulations to evaluate the performance of IAP based on the data collected from OceanSense. Table 2 lists the parameters we used in the simulations. The two algorithms of IAP are, respectively, denoted as IAP-A* and IAP-Greedy.

The following basic settings apply for all the simulations: $N_W = 30$ and $N_Q = 1{,}000$, i.e., we consider the system involving 30 different sensor networks and inject 1,000 queries for performance evaluation. The sensor networks involved in each query are randomly chosen from the 30 simulant sensor networks. We vary the other relevant parameters in different simulations for comparisons.

TABLE 2
Parameters in the Simulations

Parameters	Descriptions
N_W	Total # of sensor networks.
N_Q	Total # of queries.
$c[1..N_W]$	Original query costs of sensor networks (nJ).
$s[1..N_W][1..N_W]$	Indices of implications.
$n[1..N_W]$	# of sensor networks involved in each query.

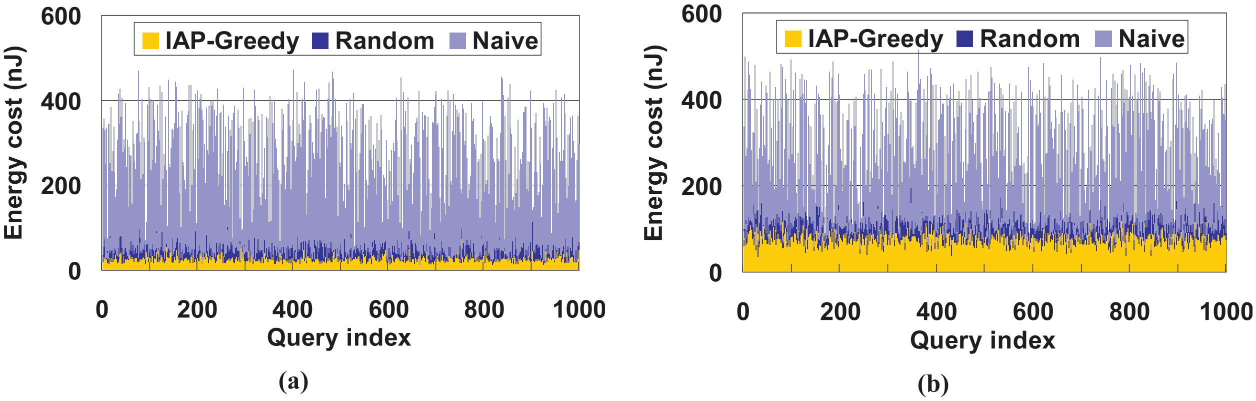


Fig. 5. Energy costs of IAP-Greedy, Random, and Naive approaches.

5.2.1 Benefit of Implication-Aware Approaches

We first evaluate the performance with a general setting, where $n[1..N_W]$ conform to a uniform distribution on [3, 15] and $c[1..N_W]$ conform to a uniform distribution on [10, 50]. This group of simulations is divided into two rounds. In round 1, $s[1..N_W][1..N_W]$ conform to a uniform distribution on [0, 1]. In round 2, $s[1..N_W][1..N_W]$ conform to a uniform distribution on [0.5, 1].

We compare IAP-Greedy with two other approaches. One of them is a naive approach, which broadcasts a query to all the involved sensor networks simultaneously and all the sensor networks process the query independently without awareness of implications among each other. Obviously the total energy cost of this approach is the sum of the original costs incurred in all the involved sensor networks. The other is a random approach, which processes a query in a randomly scheduled pipeline.

Figs. 5a and 5b plot the energy costs of all 1,000 queries using three different approaches in round 1 and round 2.

Compared with the naive approach, the percentage of cost saved by the random approach has mean 72.2 percent and standard deviation 15.7 percent in round 1, and has mean 52.6 percent and standard deviation 17.3 percent in round 2. This shows the benefit of implications among sensor networks on improving the energy efficiency of query processing.

IAP-Greedy performs even better than the random approach. Compared with the random approach, there is a further energy saving in IAP-Greedy. The percentage of cost saved by IAP-Greedy has mean 34.5 percent and standard deviation 26.1 percent in round 1, and has mean 25.8 percent and standard deviation 10.6 percent in round 2. IAP-Greedy always outperforms the random approach in all instances. This validates the fact that IAP-Greedy optimizes the pipeline scheduling by intentionally utilizing the implications among sensor networks.

It is worth noticing that in the two rounds of simulations, we conduct the comparisons using different settings of implications. The simulation results suggest that as long as the sensor networks are correlated with each other, it is always necessary and beneficial to adopt the implication-aware approaches.

5.2.2 Comparison among Approaches

In the following simulations, we compare IAP-A* and IAP-Greedy with the random approach using various settings. We evaluate the performance gains by measuring the percentage of saved cost for each query:

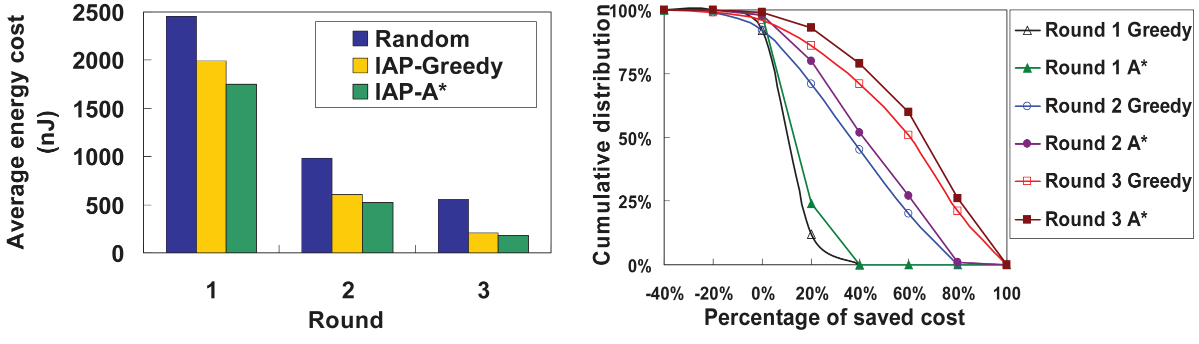
$$\text{Percentage of saved cost} = 1 - \frac{\text{Cost of IAP}}{\text{Cost of the random approach}}.$$

1. *Performance Gain versus Implication Intensity.* Let $n[1..N_W]$ conform to a uniform distribution on [3, 15] and $c[1..N_W]$ conform to a uniform distribution on [10, 1000]. We conduct three rounds of simulations. In round 1, $s[1..N_W][1..N_W]$ conform to a uniform distribution on [0.7, 1]. In round 2, $s[1..N_W][1..N_W]$ conform to a uniform distribution on [0.4, 0.7]. In round 3, $s[1..N_W][1..N_W]$ conform to a uniform distribution on [0.1, 0.4]. With such settings, the variance of implications is unchanged while the intensities are varied. Note that a smaller value of implication index indicates a stronger implication from one sensor network to another, as stated in Section 4.2.

Fig. 6a plots the average costs of the random approach, IAP-Greedy, and IAP-A* for all the 1,000 queries. Fig. 6b shows the cumulative distribution of the performance gains of IAP-Greedy and IAP-A*. From the results we can see apparent advantages of IAP over the random approach. The performance gains of IAP increase along with the implication intensity. Meanwhile, IAP-Greedy always presents comparable performance with IAP-A*, and the latter yields optimal solutions in theory. Comparatively, IAP- Greedy yields close-to-optimal solutions with much lower overhead.

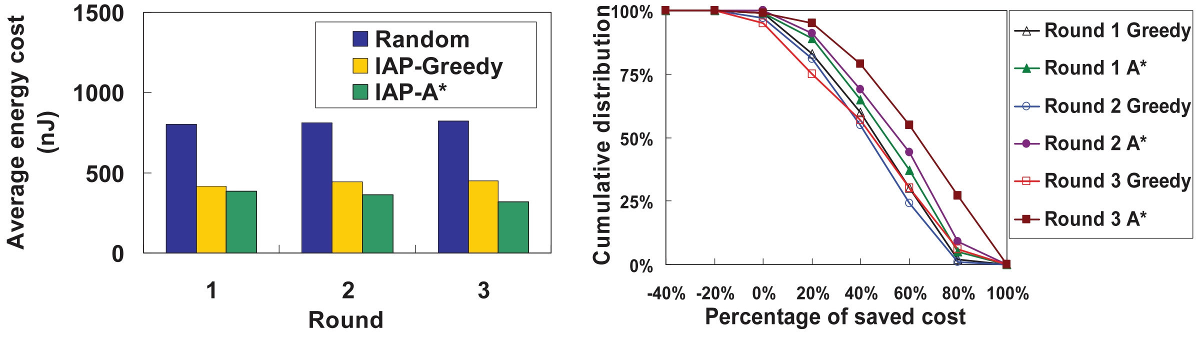
2. *Performance Gain versus Implication Variance.* Let $n[1..N_W]$ conform to a uniform distribution on [3, 15] and $c[1..N_W]$ conform to a uniform distribution on [10, 1000]. We conduct three rounds of simulations. In round 1, $s[1..N_W][1..N_W]$ conform to a uniform distribution on [0.4, 0.6]. In round 2, $s[1..N_W][1..N_W]$ conform to a uniform distribution on [0.25, 0.75]. In round 3, $s[1..N_W][1..N_W]$ conform to a uniform distribution on [0, 1]. With such settings, the implications in the three rounds have equivalent means but different variances.

Fig. 7a plots the average costs of the random approach, IAP-Greedy, and IAP-A* for all the 1,000 queries. Fig. 7b depicts the cumulative distribution of the performance gains. From the results we can see apparent and consistent advantages of IAP over the random approach with different implication variances. Again, we are pleased to see comparable performance of IAP-Greedy and IAP-A*. The



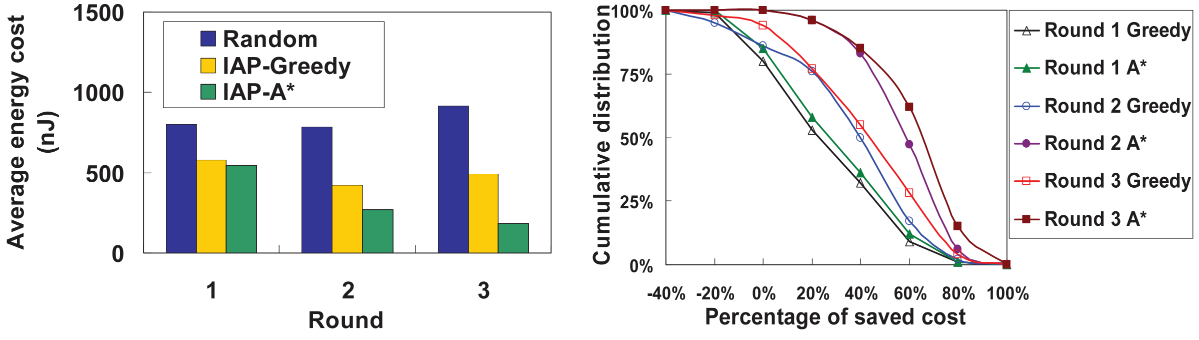
(a) (b)

Fig. 6. Performance gain versus implication intensity.



(a) (b)

Fig. 7. Performance gain versus implication variance.



(a) (b)

Fig. 8. Performance gain versus query complexity.

gap between these two algorithms increases slightly as the implication variance increases.

3. *Performance Gain versus Query Complexity.* Let $s[1..N_W][1..N_W]$ conform to a uniform distribution on [0, 1] and $c[1..N_W]$ conform to a uniform distribution on [10, 1000]. We further examine the impact of query complexity to the performance gains of IAP. For all the 1,000 queries, $n[1..N_W]$ are, respectively, set at 5, 10, and 15 in three rounds of simulations. Fig. 8a plots the average costs of the random approach, IAP-Greedy, and IAP-A* for all the 1,000 queries. Fig. 8b depicts the cumulative distribution of performance gains. From round 1 to round 3, IAP shows increasingly remarkable performance gains over the random approach, i.e., the more complex the query is, the more energy is saved by IAP. Meanwhile, the gap between IAP-Greedy and IAP-A* increases along with the query complexity. This result reveals the necessity and benefit of optimizing the pipelines of query processing, especially for complex queries involving many different sensor networks.

4. *Performance Gain versus Heterogeneity.* In this group of simulations, we evaluate the performance gains of IAP when the original energy costs of sensor networks become heterogeneous.

Let $c[1..N_W]$, respectively, conform to a uniform distribution on [10, 100], [10, 1000], and [10, 10000] in three rounds of simulations. Since $c[1..N_W]$ is uniformly distributed, a larger variance of $c[1..N_W]$ leads to stronger heterogeneity. Here, $s[1..N_W][1..N_W]$ conform to a uniform distribution on [0, 1]. $n[1..N_W]$ conform to a uniform distribution on [3, 15].

Fig. 9a plots the average costs of the random approach, IAP-Greedy, and IAP-A* for all the 1,000 queries. Note that we use logarithmic scale in Fig. 9a. Fig. 9b depicts the cumulative distribution of the performance gains. The results show that the performance gains of IAP over the

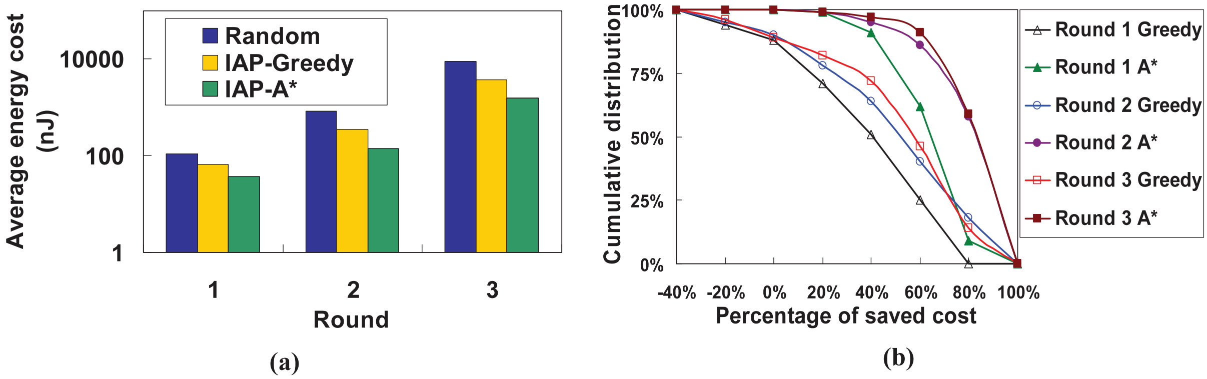


Fig. 9. Performance gain versus heterogeneity.

random approach increase along with the heterogeneity of sensor networks. Consider the practical scenarios in COSE, where different sensor networks are integrated, the cost of sensing and communication in different sensor networks is indeed quite heterogeneous. The simulation result reveals that IAP is especially effective and efficient for such environments. Moreover, it is worth noticing that the gap between IAP-Greedy and IAP-A* keeps consistent, which indicates that IAP-Greedy is applicable and adaptive to the heterogeneous environments.

5. *Running Time.* Let $c[1..N_W]$ conform to a uniform distribution on [10, 1000] and $s[1..N_W][1..N_W]$ conform to a uniform distribution on [0, 1]. For all the 1,000 queries, we repeat the simulations with varied $n[1..N_W]$ from 3 to 11. Fig. 10 shows the average running times for IAP-Greedy and IAP-A* to schedule a query. The simulations run on a PC with 2 GHz Genuine Intel(R) CPU, 1 G Memory, and Windows XP Professional Operation System. IAP-A* runs a bit faster than IAP-Greedy when $n[1..N_W] \leq 5$. But its running time increases very rapidly as $n[1..N_W]$ increases. When $n[1..N_W] = 11$, it costs 80.8 sec to schedule a query with IAP-A*, while the running time of IAP-Greedy still remains at 0.001 sec in average.

Now we briefly summarize the experimental results. IAP saves a lot of energy costs by utilizing the implications among sensor networks during query processing. IAP-A* yields the optimal scheduling, while IAP-Greedy yields close-to-optimal scheduling with much shorter running time. Therefore, we should adopt IAP-A* when the number of sensor networks involved in a query is small. And IAP-Greedy is suitable and efficient in processing queries that involve many different sensor networks.

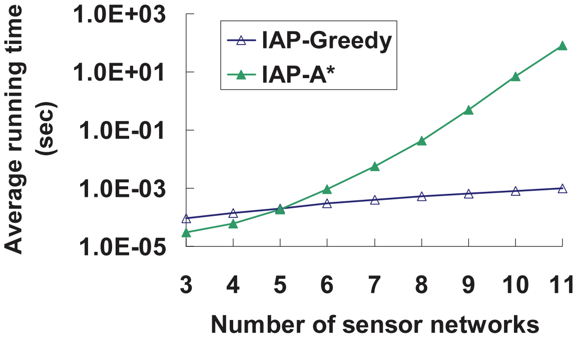


Fig. 10. Comparison of running times.

6 Related Work

There has been remarkable success in the research field of sensor networks. The state-of-art technologies, however, mainly concentrate on the intra-sensor-network issues, such as in-network sensing control, data processing, and network protocol design [12], [13]. While much attention has been put into the networking of distributed sensing, not enough work has been done for exploring mechanisms to manage, share, analyze, and understand the data among different sensor networks. We discuss several related models and applications in this section, which in some extent show comparability to our scheme of COSE.

COSE versus Query processing in a single sensor network. A school of existing works focus on the issue of query processing [14], [15] in a single sensor network. They are similar with our scheme in the exploitation and utilization of correlations in query processing. The data from multiple sensor networks in COSE, however, are very likely to be heterogeneous so that the existing schemes are no longer applicable. While our focus in this paper is not the integration of heterogeneous data, we propose the concept of sink-overlay in COSE to realize efficient data sharing among multiple heterogeneous sensor networks. Meanwhile, the model-driven approaches in [14], [15] highly rely on the coupled attributes of sensory data in a single network. Comparatively, our method is more generic by introducing information theory to estimate the implications. It is indeed applicable to both scenarios, i.e., various data attributes in a single network and heterogeneous data in multiple sensor networks. Moreover, we take load balance among sensor networks into account when formulating the query processing problem, as introduced in Section 4.2. This is also a newly emerging issue in the context of multiple sensor networks.

COSE versus "Sensor Web." The term "Sensor Web" was first introduced by Delin [16]. It is defined as a specific type of sensor network: an amorphous network of spatially distributed sensor platforms (pods) that wirelessly communicate with each other. The novelty of the Sensor Web architecture lies in the ability of the individual pieces to act and coordinate as a whole. Sensor Web is an autonomous, stand-alone, sensing entity, which does not need assemble resources among sensor networks through the presence of the World Wide Web. Different from "Sensor Web," COSE in this work defines an integrated framework of web and sensor networks. Sensor networks in COSE preserve their own independency and original functionalities, while are manipulated to collaborate with each other in addressing complex queries.

COSE versus SenseWeb. The term "SenseWeb" is sometimes used to refer to sensors connected to the Internet. Nath et al. propose SensorMap [17], which represents a new class of applications that relies on real-time sensor data and its mash-up with the geocentric web to provide instantaneous environmental visibility and timely decision support. The platform also transparently provides mechanisms to archive and index data, to process queries, to aggregate and present results on the web interface based on Windows Live Local. IrisNet proposed in [18] is another software infrastructure for worldwide web of sensors that lets users query globally distributed collections of high-bit-rate sensors. IrisNet takes a database centric approach in its design. It implements service-specific distributed XML databases of the sensor data and provides high level APIs to query and process the sensor data. Another prototype system is SensorNet [19]. SensorNet proposes a data architecture and infrastructure that supports plug-and-play sensors of various types, archival storage of sensor data, standards-based publication of sensor data, and sensor control services. It allows for the integration of dissimilar sensor systems into one system. It focuses on high speed, reliable access and delivery of sensor data inside the infrastructure.

Compared with SensorMap, IrisNet and SensorNet, COSE also proposes the presence of a website portal as an interface for query input and output. But COSE aims to provide more powerful functions than simply exhibiting the instant sensor data. Queries injected into COSE are processed by a pipeline of sensor networks. It provides fused results, making the in-network data sensing and transmission intelligent and transparent to the web users. Moreover, energy efficiency is emphasized as a crucial part in the design, because what COSE deals with is a network of sensor networks instead of a network of individual sensing devices.

COSE versus web service. Since sensor networks in COSE collaborate in addressing complex queries from external users, COSE provides similar portal as web service. The environments they are applied to, however, are totally different. The architecture of web service facilitates software development with the computing capacities provided by service providers over The Internet [20]. Services can be encapsulated and combined in a unified software process. Different from web service, COSE provides services based on live sensor data from spatially dispersed sensor networks in the physical world. Moreover, sensor networks are manipulated in collaboration and union, rather than trading and competition in the case of web service.

COSE versus distributed databases. Queries in COSE are resolved by a pipeline of sensor networks, similarly with query processing in distributed database [21]. However, they have several distinct characteristics, which make it inapplicable to migrate the existing approaches in distributed database into the framework of COSE: A distributed database is distributed into partitions/fragments that may be replicated, while each sensor network in COSE is unique. Sensor data reside only in their original sensor networks, except minor information exchanges among them. Besides, due to the nature of sensor networks, the total in-network energy consumption is the first-place metric when we optimize query processing in COSE. Oppositely, the goal of query optimization in distributed database is to minimize the cost of bandwidth, which is incurred when data are forwarded from one node to another.

7 Conclusion and Future Work

Demands on better interacting with physical world are driving the sensor networks from isolated working into collaborative operations. This paper proposes COSE, a query-centric framework of collaborative heterogeneous sensor networks, where sensor networks are interconnected and collaborate with each other in the level of sink-overlay. The problem of energy-efficient query processing in COSE is particularly studied. We prove this problem is NP-hard even with global knowledge. Accordingly, we design a heuristic approach IAP to minimize the energy costs caused by query processing in the distributed context.

Considering that the future sensor network systems supporting numerous users and applications, it is an interesting and important issue to address multiple pipelines for query processing in heterogeneous sensor networks. Because the data of a sensor network may be shared/reused by multiple queries, the optimal schedule of a single pipeline does not match the optimal solution when it is scheduled together with other pipelines. Implication is no longer the only factor to be considered. High reusability of sensing data will build up the utility of a WSN in scheduling. Other issues, such as query's priority and query aggregation are also potential research directions.

Unified systems of web and sensor networks present a promising direction for integrating various sensor network and ubiquitous computing resources to achieve powerful and intelligent functionalities. In the next step, we plan to progress the research in both theoretical and systematical aspects. We also consider the user-oriented optimization as a potential direction in the future.

Acknowledgments

This study is supported in part by the NSFC program under Grant 61170213, the National Basic Research 973 Program of China under Grant No. 2011CB302705, NAP M4080738.020 of Nanyang Technological University, NSFC Major Program 61190110, and China Postdoctoral Science Foundation under Grant 2011M500019.

References

[1] T. Gao et al., "Participatory User Centered Design Techniques for a Large Scale Ad-Hoc Health Information System," *Proc. First ACM SIGMOBILE Int'l Workshop Systems and Networking Support for Healthcare and Assisted Living Environments (HealthNet)*, 2007.

[2] R. Szewczyk et al., "An Analysis of a Large Scale Habitat Monitoring Application," *Proc. ACM Second Int'l Conf. Embedded Networked Sensor Systems (SenSys)*, 2004.

[3] P. Zhang et al., "Hardware Design Experiences in ZebraNet," *Proc. ACM Second Int'l Conf. Embedded Networked Sensor Systems (SenSys)*, 2004.

[4] Y. Liu et al., "Location, Localization, and Localizability," *J. Computer Science and Technology*, vol. 25, pp. 274-297, Mar. 2010.

[5] P.D.R. Fonseca, P. Levis, and I. Stoica, "Quanto: Tracking Energy in Networked Embedded Systems," *Proc. Eighth USENIX Symp. Operating System Design and Implementation (OSDI),* 2008.

[6] N. Xu et al., "A wireless Sensor Network for Structural Monitoring," *Proc. ACM Second Int'l Conf. Embedded Networked Sensor Systems (SenSys),* 2004.

[7] K. Liu et al., "Robust and Efficient Aggregate Query Processing in Wireless Sensor Networks," *ACM Mobile Networks and Applications,* vol. 13, pp. 212-227, Apr. 2008.

[8] H. Tan et al., "Computing Localized Power Efficient Data Aggregation Trees for Sensor Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 22, no. 3, pp. 489-500, Mar. 2011.

[9] R. Dechter and J. Pearl, "Generalized Best-First Search Strategies and the Optimality af A*," *J. ACM,* vol. 32, pp. 505-536, July 1985.

[10] OceanSense: http://www.cse.ust.hk/~liu/Ocean/, 2012.

[11] X. Mao et al., "Citysee: Urban CO2 Monitoring with Sensors," *Proc. IEEE INFOCOM,* 2012.

[12] J. Hill and D. Culler, "Mica: A Wireless Platform for Deeply Embedded Networks," *IEEE Micro,* vol. 22, no. 6, pp. 12-24, Nov./Dec. 2002.

[13] C. Intanagonwiwat et al., "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *Proc. ACM MobiCom,* 2000.

[14] A. Deshpande et al., "Model-Driven Data Aquisition in Sensor Networks," *Proc. 30th Int'l Conf. Very Large Data Bases (VLDB),* 2004.

[15] S. Babu et al., "Adaptive Ordering of Pipelined Stream Filters," *Proc. ACM SIGMOD Int'l Conf. Management of Data,* 2004.

[16] K.A. Delin, "The Sensor Web: A Macro-Instrument for Coordinated Sensing," *Sensors,* vol. 2, pp. 270-285, 2002.

[17] S. Nath et al., "SensorMap for Wide-Area Sensor Webs," *IEEE Computer Magazine,* vol. 40, no.7, pp. 90-93, July 2007.

[18] P.B. Gibbons et al., "IrisNet: An Architecture for a World-Wide Sensor Web," *IEEE Pervasive Computing,* vol. 2, no. 4, pp. 22-33, Oct.-Dec. 2003.

[19] SensorNet: http://www.sensornet.gov/, 2011.

[20] I. Foster, "Service-Oriented Science," *Science,* vol. 308, pp. 814-817, 2005.

[21] M.T. Ozsu and P. Valduriez, *Principles of Distributed Database Systems,* second ed. Prentice Hall, 1999.

Yuan He (M'06) received the BE degree in the University of Science and Technology of China in 2003, the ME degree in Institute of Software, Chinese Academy of Sciences in 2006, and the PhD degree in Hong Kong University of Science and Technology. He is a member of Tsinghua National Lab for Information Science and Technology. His research interests include sensor networks, peer-to-peer computing, and pervasive computing. He is a member of the IEEE, the IEEE Computer Society, and ACM.

Mo Li (M'06) received the BS degree in the Department of Computer Science and Technology from Tsinghua University, China and the PhD degree in the Department of Computer Science and Engineering from Hong Kong University of Science and Technology. He is currently an assistant professor in School of Computer Engineering of Nanyang Technological University, Singapore. He won ACM Hong Kong Chapter Prof. Francis Chin Research Award in 2009 and Hong Kong ICT Award—Best Innovation and Research Grand Award in 2007. His research interest includes wireless sensor networking, pervasive computing, mobile and wireless computing, etc. He is a member of the IEEE and the ACM.