POLLACK PERIODICA
An International Journal for Engineering and Information Sciences
DOI: 10.1556/606.2020.15.2.2
Vol. 15, No. 2, pp. 13–24 (2020)
www.akademiai.com

A COMPARATIVE EVALUATION OF TECHNIQUES FOR N-WAY JOINS IN WIRELESS SENSORS NETWORKS

[1] **Boubekeur DJAIL***, [2] **Walid Khaled HIDOUCI,** [3] **Malik LOUDINI**

[1,2,3] Laboratoire de la Communication dans les Systèmes Informatiques
Ecole supérieure d'informatique , Alger Algerie
e-mail: [1]b_djail@esi.dz, [2]w_hidouci@esi.dz, [3]m_loudini@esi.dz

Received 28 July 2019; accepted 23 November 2019

Abstract: In wireless sensors networks, data are sensed and recorded as databases, and then acceded by relational queries. Joins are queries that are largely used. Joins collect data from several nodes' table. These are operations that typically consume a lot of energy because they generate a large number of messages in the network. Researchers worked to decrease this consumed energy. Many strategies were proposed in this way, but most of them addressed only binary joins. N-way joins received few interests. N-way joins perform join operations between more than two tables. They cause greater energy consumption. Additionally, the number of execution order is very important; it grows exponentially with the number of considered tables.

In this paper, a comparative study is performed between existing techniques for processing N-way joins in wireless sensor networks. These are two recent techniques: N-way Local Join and N-way Local Semi-Join, and two reference techniques, Sens-join and extern join.

Several convenient parameters are selected to make this comparison. For each strategy, the ideal cases of its use are determined.

Keywords: Cost communication, In-network join, Join operation, N-way join, Wireless sensor networks

1. Introduction

A wireless sensors network is a network with nodes consisting of sensors devices, which communicate with wireless interfaces [1]. Sensors are characterized by limited

* Corresponding Author

capacities of memorization and computing, and are alimented by low power batteries. This energy is a determinant factor for sensors lifetime.

Sensors nodes detect events and record corresponding data to form a table at each node. The tables of all nodes are considered as a distributed database table [2].

The database is queried using relational queries, (The database is consulted through relational queries) like projection, selection, joins, etc.

A large number of applications, in wireless sensors networks, use join queries to collect data from several nodes of the same network. Examples of these applications are: vehicle surveillance and tracking, animal habitat monitoring, environment monitoring, home and commercial building automation, precision agriculture, and water resource management [3].

Processing join queries involve very high energy consumption, which affect the lifetime of each participating sensors node and then the whole network [4], [5].

Several research works addressed this problem and aimed to reduce the consumed energy during the query treatment. In [6] it was confirmed that consumed energy in transmission of messages between nodes is higher than that in processing data at the nodes. In this vision, it is necessary to decrease essentially the number of transmitted messages during a query execution.

Many of works were proposed to treat this query type, but most of them addressed binary joins only. N-way joins, which are performed between more than two tables, are rarely referred.

The challenge with N-way joins is that to reduce the consumed energy usually higher than that with binary joins, and to determine the best execution order among an exponential number of alternatives.

Stern et al. [7] proposed Sens-join, a technique to perform any join types, include N-way joins. Sens-join uses filters, which are determined at the nodes and transmitted to the sink where the query is executed. The filters are the list of attribute-join values that are selected by the join query. They permit to reduce considerably the number of transmitted messages. This solution remains however insufficient since the messages of the filters are all transmitted to the sink and not exploited within the network.

A comparison of four techniques is performed: Two recent techniques: N-way Local Join (NLJ) and N-way Local Semi-Join (NLSJ) and two reference techniques: Sens-join and Extern join. NLJ (N-way Local Join) [8] and NLSJ (N-way Local Semi-Join) [9] execute each intermediate join of an N-way join, locally at one of the two concerned sites. The following site is then selected as the nearest to the precedent one. NLSJ uses additionally the Semi-Join principle to filter even more, the tuples of the tables concerned by the join query.

2. Definitions

- A join is essentially a Cartesian product of the operand tables followed by a predicate selection [1];
- A theta-join is the query that contains an arbitrary comparison operator in the join predicate;
- An equi-join uses only equality operator to express join predicate;

- A binary join is performed between two tables. An N-way join considers more than two tables for its execution;
- Non-joinable tuples are the tuples that are not selected by the join query;
- A Semi-Join is join query which is executed between a table and a projection of another table on join-attributes;
- Linear trees technique: The principle is that N-way join queries are represented as trees of binary joins. The query processor determines an execution order, either from left to right or from right to left.

3. Joins features in wireless sensor networks

3.1. Implementation of join queries in wireless sensors networks

There are mostly two possible implementations of join queries in wireless sensors networks: extern and in-network join implementation [3].

With extern-join, the query is performed at the sink. All concerned nodes, by the join query, must beforehand transmit their tuples to the base station. This implementation is too simple to implement, but it introduces a very high consumption of energy due to the important volume of transmitted messages, which is conveyed from the nodes to the sink.

In-network join is the implementation that performs join operations at internal nodes of the network. It reduces considerably the quantity of transmitted messages between nodes, and then the whole consumed energy [3].

3.2. Join types in wireless sensors network

According to spatial aspect, join queries in wireless sensor networks can be classified into two types: unique-region joins and inter-region joins [3].

A unique-region join matches tuples between tables of nodes in the same region. However, an inter-region join (*Fig. 1*) performs the query between tables of nodes in two different regions.

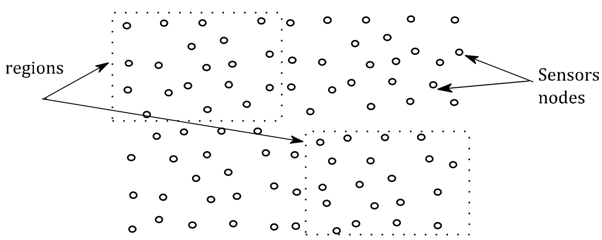

Fig. 1. Inter-region join principle

According to temporal aspect, joins queries in wireless sensor networks are divided into two categories: One-shot joins and continuous joins [3].

One-shot joins use fixed windows that are defined on a number of tuples or a time period. The join query is so performed between tuples of each window that represent tables at nodes.

Continuous joins are executed permanently by using sliding windows, which define sets of tuples to be considered at each step of the query execution. A particular case of continuous joins are the periodic joins. Periodic joins are joins which are repeated after each defined period of time.

4. Related works

Several techniques were proposed to address joins queries in wireless sensor networks. An important factor is considered to classify these techniques into two main categories. This factor is whether the techniques filter or not the tuples in order to reduce the volume of the transmitted messages.

The first category of techniques is generally the first proposed ones. These techniques did not consider filtering of tuples. Yao and Gehrke [2] performed a comparison between an extern-join and an in-network join by considering communication cost. It was resulted that in-network technique includes less dissipation of energy for low join selectivity. Bonfils and Bonnet [10] worked to determine the optimal node where to execute an in-network join. The site obtained is on the shortest path between the two nodes that participate in the query. It is nearest to the site, which has more data. Coman et al. [11] proposed local join and mediated join techniques to address an inter-region join. Local join executes locally the query at nodes of one of the two regions, whereas mediated join performs the query at an intermediate region. It has confirmed that no specific join strategy presents the best results for all queries.

The second category of proposed techniques considered filtering of tuples. These are recently the most common. Yu et al. [12] proposed synopsis join to address a one-shot inter-region query join. They used a distributed alternative of the Semi-Join approach to reduce tables' sizes. Coman et al. [11] suggested: local Semi-Join to improve the latest technique: local join. The join operation is carried out in one of the two areas, and uses Semi-Join principle. Min et al. [13] presented various plans to execute a join query and they proposed a cost model to choose the optimal plan under various conditions.

Specific joins were also addressed by many other researchers. Mo et al. [14] addressed spatial queries in wireless sensors network. Kang et al. [15] treated iceberg join query, a special type of join where only tuples whose cardinality exceeds a certain threshold are admitted to the join operation. Min et al. [16] proposed a solution based on time-windowed principle to treat continuous joins.

Most of those techniques were suggested for binary joins. Few of them were developed for N-way join. Stern et al. in [7] proposed a strategy to treat all join types, included N-way join queries. The strategy consists of performing the query join at the sink, by using filters, which are determined at internal nodes based on the relevant records. The NLJ and NLSJ techniques are recently proposed to treat N-way join queries. NLJ adopts an in-network execution to reduce the number of transmitted messages. NLSJ improves NLJ with adopting Semi-Join principle.

5. Techniques for processing N-way join queries

In this evaluation, an interest is granted with the ability of the studied techniques to reduce the quantity of transmitted messages, by comparing the communication cost. The communication cost is the parameter, which is expressed by the number of transmitted messages during a join query execution. The techniques selected for this study are presented below.

5.1. Extern join technique

Extern join executes the query totally at the sink. All tuples of concerned tables are beforehand transmitted to the station base from roots' nodes of regions (*Fig. 2*). This technique presents the advantage of simplicity, but this generates high transmission costs.

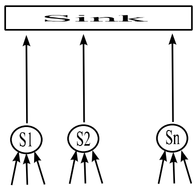

Fig. 2. Extern join execution

5.2. N-way local join technique

N-way Local Join [8] is the technique proposed to address N-way joins queries in wireless sensors networks. NLJ does not use tuples filtering. It consists in performing each intermediate join locally at one of the two selected regions. NLJ adopts the left linear tree technique to determine the execution order of the join operations [17]. This choice is also based on geographical zone positions to select the nearest region to the actually selected region.

NLJ technique runs in three phases:

Phase 1. Query dissemination

In this phase, the generated query at the sink is diffused to the root nodes at concerned regions. This dissemination is held based a location routing protocol Greedy Perimeter Stateless Routing (GPSR) for wireless networks [18], to guaranty the reception of the query message by recipient nodes.

Note that it is assumed that regions are organized hierarchically, where a node is selected as principal: root node. It is also assumed that each node recognizes its location and the locations of its neighbors, via Global Positioning System (GPS) or via localization algorithms [19].

Phase 2. Query execution

The join query is performed in several steps. At each step, an intermediate join is executed between a determinate nodes' pair. The left linear tree technique is used to fixe the execution order of joins.

For a nodes' pair (Si, Si+1), NLJ runs this two following actions (*Fig. 3*):

i) The relation R_i is transmitted to the region Si+1;
ii) The join operation is executed in the region Si+1.

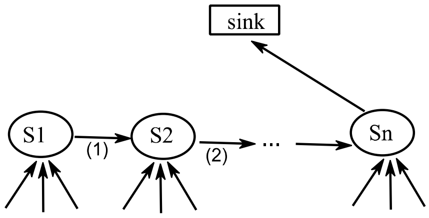

Fig. 3. N-way local join execution, (1) Relation R1, (2) (R1 join R2) result

Phase 3. Final result transmission

At the end of intermediates' joins execution, the final result is transmitted to the sink.

5.3. N-way local Semi-Join technique

N-way local Semi-Join [9] is proposed as an improvement of NLJ technique. NLSJ is a filtering technique that uses the Semi-Join principle to filter non-joinable tuples.

NLSJ runs in three phases:

Phase 1. Query dissemination

The query initiated at the sink, is transmitted to the root node of each concerned region. A location routing protocol GPSR is used to ensure that the query is received by all recipient nodes. Each root node collects tuples from nodes belonging to its region.

Phase 2. Query execution

For a nodes' pair (Si, Si+1), an intermediate join is processed as follows (*Fig. 4*):

i) The join attribute of the relation Ri+1 is transmitted to site Si;
ii) A Semi-Join is executed in Si, and the result is transmitted to Si+1;
iii) The final result of the intermediate join is performed in Si+1.

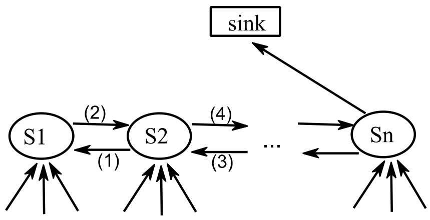

Fig. 4. N-way local Semi-Join execution
(1) Join attribute of R2, (2) Semi-Join result of proj (R2) join R1, (3) Join attribute of R3,
(4) Semi Join result of proj (R3) join (R1 join R2)

Phase 3. Final result transmission
With the last intermediate join, the determined result is communicated to the sink.

5.4. Sens-join technique

Sens-join [7] that was proposed by Stern and al. performs in five phases:

Phase 1: Query diffusion
The query is diffused by the base station to all concerned root nodes.

Phase 2: Join attributes transmission
The join attributes are transmitted by all root nodes to the sink (*Fig. 5*). These attributes will be used to determine the filter for the join query.

Phase 3: Filter determination
Based on the values of join attributes, a filter is generated, and then transmitted to all root nodes in objectify to establish the Semi-Join.

Phase 4: Semi-Join accomplishment
Using the received filter, the root nodes execute the Semi-Join with all tuples. The determined result is then conducted to the base station.

Phase 5: Final execution
At the sink, the join final result is performed after the all results were received from root nodes.

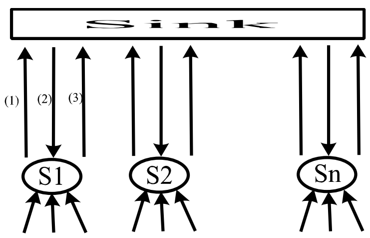

Fig. 5. Sens-join execution
(1) Join attributes; (2) Filter; (3) Semi-Join results

6. Experimentation and performance analysis

6.1. General description

In this experimentation the query example considered is a one-shot inter-region joins having the following syntax:

```
SELECT R1.attributs, R2.attributs,…,Rn.attributs
FROM  R1, R2, …, Rn
WHERE  predicat(R1)  AND predicat(R2)  … AND predicat(Rn)
AND join-exp (R1.join-attributs , R2.join-attributs ,…, Rn.join-attributs)
```

where:

Ri is the relation of the ith region.
predicat (Ri) is a selection predicate of the relation Ri.
join-exp is the join condition.

The example that is simulated in this evaluation is that for the vehicle traffic control through many geographical regions. For three regions it can be written as:

```
SELECT   V1.VId, V1.time, V2.time, V3.time
FROM      V1, V2, V3
WHERE    (V1.time IN i1) and (V2.time in i2) and (V3.time in i3) and (V1.VId = V2. VId) and
(V2. VId= V3. VId)
```

where:

i1, i2, and i3 indicate time ranges during which the Vehicles passed respectively through regions 1,2 and 3.

6.2. Experimentation environment

To perform the evaluation, the NS3 simulator is used. The four described techniques are tested considering the following parameters:

- Tuple size is 40 bytes;
- Message size is 40 bytes;
- A Column is 10 bytes;
- Result tuple size is 30 bytes.

As the cost of communication is the most important factor in the testing of joins in wireless sensor networks, tests are developed by considering this grandeur according to several other parameters that are:

- Selectivity factor;
- Number of tables;
- Size of tables.

Cost communication according to selectivity factors

For the parameter selectivity factor two intervals are considered; one for low values [10^{-5}, 10^{-4}] and another for high values [10^{-4}, 10^{-3}].

In each interval, values are generated in a random way. In the horizontal axis, the average of selectivity factors values of the intermediate joins is used.

A simulation is performed for three tables and another one for five tables. For each simulation the four techniques previously described are compared.

The table size taken in this evaluation is 2000 tuples.

Cost communication according to tables number

The number of considered tables varies from two to seven. In the first evaluation a selectivity factor value was used in the lower interval and it is equal to 0.000025. In the second evaluation, the selected value is in the high interval and it is equal to 0.00025

Cost communication according to tables size

For the evaluation, the values of the table sizes parameter are between 1000 tuples and 3000 tuples. The same values are also considered as the previous evaluation, for the selectivity factor: 0.000025 and 0.00025.

6.3. Experimentation results

Impact of join selectivity factors

The technique, which shows the best performances, in all realized tests, is NLSJ (*Fig. 6 - Fig. 9*). The reasons of its success are in its adoption of the Semi-Join principle to optimally filter the non-joinable tuples. With NLSJ, the communication cost values are the lowest, essentially for the low values of selectivity factor. From the value 0.0003, NLSJ starts to decrease in performance.

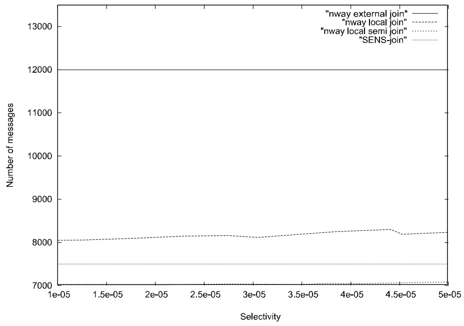

Fig. 6. The communication cost (tuples) for 3 tables in the interval [10^{-5}, 10^{-4}] of selectivity factor

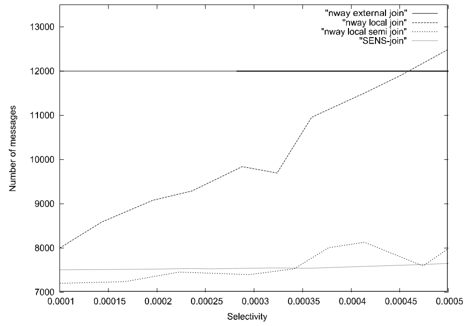

Fig. 7. The communication cost (tuples) for 3 tables in the interval [10^{-4}, 10^{-3}] of selectivity factor

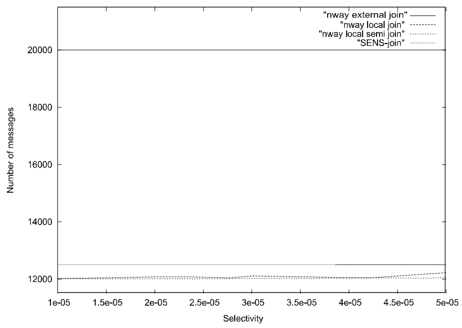

Fig. 8. The communication cost (tuples) for 5 tables in the interval [10^{-5}, 10^{-4}] of selectivity factor

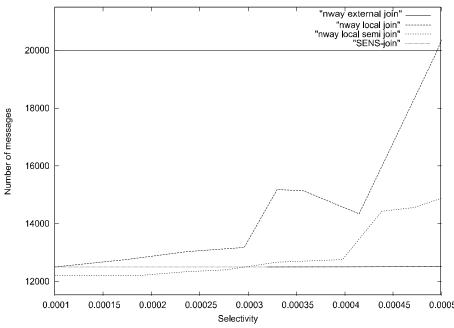

Fig. 9. The communication cost for 5 tables in the interval [10^{-4}, 10^{-3}] of selectivity factor

Sens-join technique has close performances compared to NLSJ. Sens-join uses also filters to eliminate non-joinable tuples before query execution, but the join query does not perform at sensor nodes level (in-network), hence the performances are not the same as those of NLSJ.

NLJ technique does not offer a good performance because it does not apply any form of filtering. However, NLJ adopts an in-network execution, which gives it some interest and performs better than an extern-join approach. Additionally, it is noted that with five tables and for low selectivity factor, NLJ executes better than Sens-join. This can be explained by that NLJ reduce the number of transmitted messages at each step where it executes an intermediate join with selectivity factors that are very low.

Extern-join presents low results in the tested intervals, but the results remain constant during all ranges. It was demonstrated that extern-join is interested in use with very high selectivity factors.

Impact of number of tables

NLSJ maintains its best abilities for low or high selectivity factors, regardless of the number of tables selected (*Fig. 10* - *Fig. 11*). It keeps a slope almost constant and is much lower than those of other techniques.

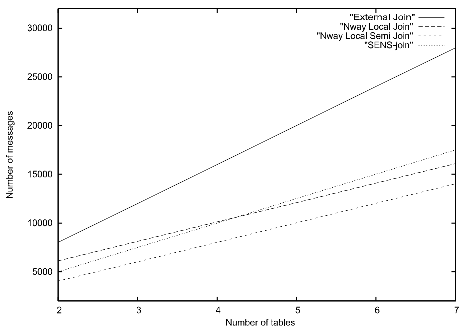

Fig. 10. Communication cost (tuples) for selectivity factor equal to 0.000025

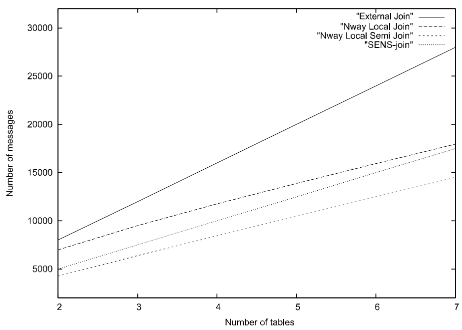

Fig. 11. Communication cost (tuples) for selectivity factor equal to 0.00025

Sens-join decreases in performance for low selectivity factors, from four tables, in favor of NLJ technique. For high selectivity factor, Sens-join performs better than NLJ, but has a high slope.

NLJ executes better than extern-join for all selectivity factors of the two intervals. Also, it has an interesting slope closer to that of NLSJ.

Extern-join is the worst in the selected intervals of selectivity factor, and presents a very high slope.

Impact of sizes of tables

NLSJ always presents the best results, with a slight slope. However, Sens-join is more efficient than NLJ for all values of the intervals tested (*Fig. 12* - *Fig. 13*).

NLJ is more effective than extern-join, with a slope that gradually grows in the form of a curvilinear curve.

6.4. Discussion

NLSJ has the best performance in terms of the selectivity factor values, or depends on the number of tables or table sizes. NLSJ presents the best performances because of its use of the Semi-Join principle in addition to a network execution.

Sens-join only takes the Semi-Join to filter non-joined tuples, so it performs poorly than NLSJ.

NLJ does not use Semi-Join, nor filtering, but only network execution. It can be interesting for high selectivity factor values.

Extern-join performs the join query completely at the base station, without filtering. This technique is recognized as attractive only for very high selectivity factor values.

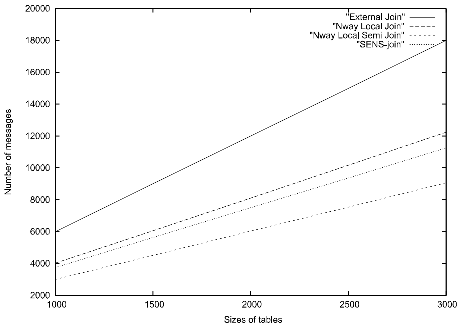

Fig. 12. Communication cost (tuples) for selectivity factor equal to 0.000025 depending to the sizes of tables (tuples)

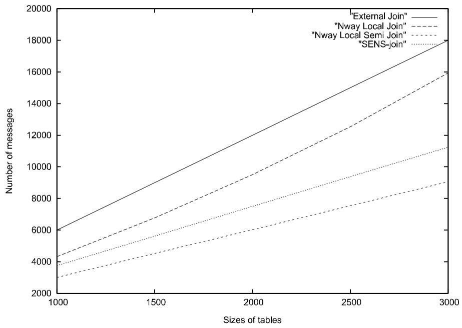

Fig. 13. Communication cost (tuples) for selectivity factor equal to 0.00025 depending to the sizes of tables (tuples)

7. Conclusion

Four techniques of N-way join queries, in wireless sensor network were compared in this paper. They are: Extern-join which executes the query at sink, NLJ and NLSJ whose perform the join locally at selected nodes, and Sens-join that proposes a combination of the two possibilities, by determining filters locally at internal nodes and by executing the query at the sink.

After realizing the tests on these techniques, NLSJ shows the best performance due essentially to its adoption of Semi-Join and the in-networks execution principles. Sens-join has performances close to those of NLSJ. However, NLJ and extern-join show the lowest results, especially for low selectivity factors. Finally it can be concluded that NLSJ is the best choice for join queries with low join selectivity factors, and extern-join is more accommodating for very high values of selectivity factor.

In the future works, studies can concern joins queries for data stream in wireless sensors networks. This is a very attractive area of study mainly in on-line industrial environment monitoring.

References

[1] Gupta H., Chowdhary V. Communication-efficient implementation of join in sensor networks, *Ad Hoc Networks*, Vol. 5, No. 6, 2007, pp. 929–942.

[2] Yao Y., Gehrke J. Query processing in sensor networks, First *Biennial Conference on Innovative Data Systems Research*, Asilomar, CA, USA, 5-8 January 2003, pp. 233–244.

[3] Kang H. In-network processing of joins in wireless sensor networks, *Sensors*, Vol. 13, No. 3, 2013, pp. 3358–3393.

[4] Györke P., Pataki B. Power supply design for solar powered harvesting-aware sensors, *Pollack Periodica,* Vol. 7, No. 3, 2012. pp. 45–54.

[5] Garai Á., Péntek I., Adamkó A., Németh Á. Methodology for clinical integration of e-health sensor-based smart device technology with cloud architecture, *Pollack Periodica,* Vol. 12, No. 1, 2017, pp. 69–80.

[6] Zhao F., Guibas L. *Wireless sensor networks: an information processing approach,* Morgan Kaufmann, 2004.

[7] Stern M., Buchmann E., Böhm K. Towards efficient processing of general-purpose joins in sensor networks, *2009 IEEE 25th International Conference on Data Engineering*, Shanghai, China, 29 March-2 April 2009, pp. 126–137.

[8] Boubekeur D., Khaled H. W. Malik L. A technique for N-way joins in wireless sensor networks, *Database Systems Journal,* Vol. 7, No. 1, 2016, pp. 3–9.

[9] Boubekeur D., Khaled H. W., Malik L. N-way local Semi-Join: A filtering technique for N-way joins in wireless sensors networks, *Journal of Electronic Systems*, Vol. 6, No. 1, 2016, pp. 7–16.

[10] Bonfils B. J., Bonnet P. Adaptive and decentralized operator placement for in-network query processing, *Telecommunication Systems*, Vol. 26, No. 2-3, 2004, pp. 389–409.

[11] Coman A., Nascimento M. A., Sander J. On join location in sensor networks, *International Conference on Mobile Data Management*, Mannheim, Germany, 1 May 2007, pp. 190–197.

[12] Yu H., Lim E.-P., Zhang J. On in-network synopsis join processing for sensor networks, *7th International Conference on Mobile Data Management*, Nara, Japan, 10-12 May 2006, pp. 32–32.

[13] Min J. K., Yang H., Chung C.-W. Cost based in-network join strategy in tree routing sensor networks, *Information Sciences*, Vol. 181, No. 16, 2011, pp. 3443–3458.

[14] Mo S., Fan Y., Li Y., Wang X. Multi-attribute join query processing in sensor networks, *Journal of Networks*, Vol. 9, No. 3, 2014, pp. 2702–2712.

[15] Kang H. In-network processing of an Iceberg Join Query in wireless sensor networks based on 2-way fragment Semi-Joins, *Sensors*, Vol. 15, No. 3, 2015, pp. 6105–6132.

[16] Min J. K., Kim J., Shim K. TWINS: Efficient time-windowed in-network joins for sensor networks, *Information Sciences*, Vol. 263, 2014, pp. 87–109.

[17] Steinbrunn M., Moerkotte G., Kemper A. *Optimizing join orders*, Citeseer, 1993.

[18] Karp B., Kung H. T. GPSR: Greedy perimeter stateless routing for wireless networks, *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, Boston, Massachusetts, USA, 6-11 August 2000, pp. 243–254.

[19] Savvides A., Srivastava M., Girod L., Estrin D. Localization in sensor networks, in *Wireless sensor networks*, Raghavendra C.S., Sivalingam K. M., Znati T. (Eds.) Springer, 2004, pp. 327–349.