Chapter 5

Performance Evaluation

In order to evaluate the performance of algorithms SB and OOB, we have implement a simulation model of the wireless sensor networks. Specifically, in Section 5.1, the simulation model is described. Then, we examine the impact of sharing partial results in Section 5.2. Performance of algorithms SB and OOB is comparatively analyzed in Section 5.3. Section 5.4 is devoted to the sensitive analysis of algorithm SB.

5.1 Simulation Model

A wireless sensor network is simulated, where there are 500 sensors randomly deployed in a 500*500 m^2 region. The sink is at the left-top corner of the region. network environment and query processing model of sensor network in Java. The tranmission range of sensors are set to 50 m. Users submit their query to the sink and that the query will utilize TAG scheme [6] to form a query tree, where the root node is the sink. Table **??** summaries the definitions for some primary simulation parameters. The number of queries running at the same time is denoted as n. The query range is referred to those sensors whose sensing data are the data sources of one query tree. Note that with higher overlap in two query ranges, these two query trees are able to have more opportunities to share partial results. Thus, we express

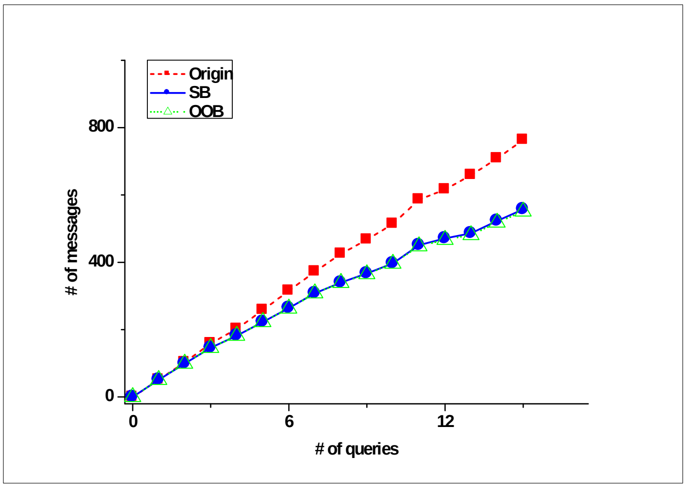

Figure 5.1: Number of messages involved under scheme origin, algorithm SB and algorithm OOB with the number of queries varied.

the non-overlap of query ranges as *random of overlap*. For the comparison purpose, scheme *origin* is referred to the scenario that queries are performed as usual without any partial result sharing.

5.2 The Impact of Sharing Partial Results

First, we investigate the impact of sharing partial query results, where randomness of overlap is set to 50% and we set query range to 100 x 100 m^2. As be seen in Figure 5.1, the numbers of messages of scheme origin, SB and OOB increase linearly with the number of queries. Note that through the partial result shares, the number of messages of SB and OOB is smaller than that of Origin. Moreover, the performance of heuristic algorithm SB is very close to the one of A* algorithm OOB.

Next, we investigate the performance of scheme origin, algorithms SB and OOB with the values of query range varied. The number of queries is set to 10 and the value of randomness of overlap is set to 50%. As shown in Figure 5.2, our proposed algorithms outperform scheme origin. Furthermore, the performance studies of algorithm SB and OOB is almost the same.

Finally, we vary randomness of overlaps and fix number of queries to 10 and query range

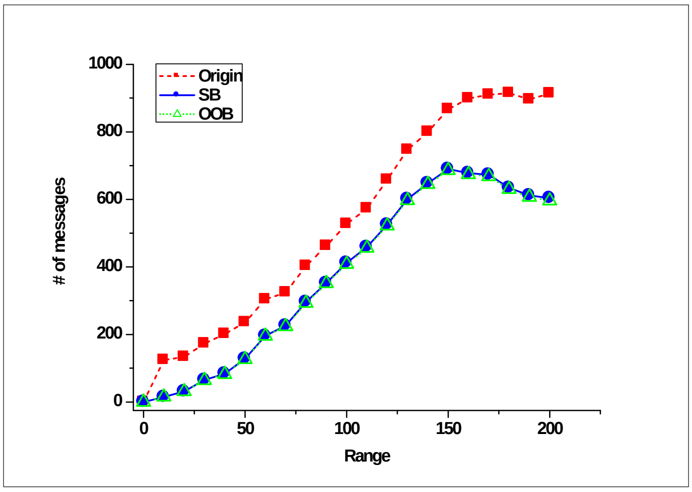

Figure 5.2: The performance comparision among origin, SB and OOB with various values of range.

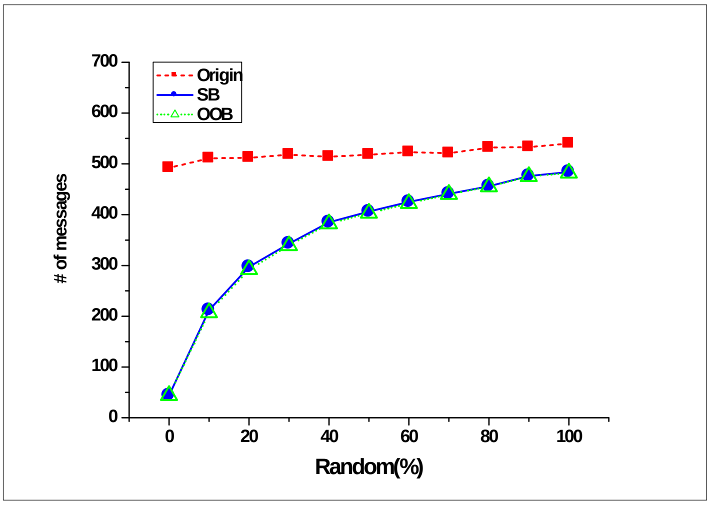

Figure 5.3: Number of messages of Origin, SB and OOB with the value of Range varied.

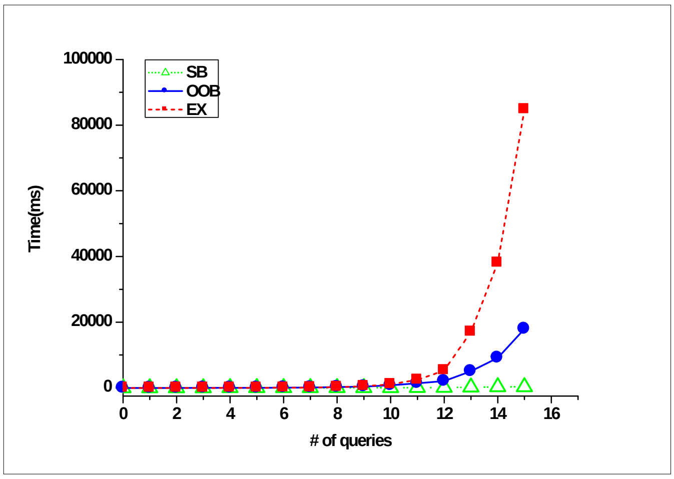

Figure 5.4: Execution time of SB, OOB and EX with the number of queries varid.

is set to 100 x 100 m^2. As can be seen in 5.3, the number of messages can be reduced by SB and OOB sharply decrease when random of overlaps goes from 0 to 50 and softly increase when the value of random of overlaps is larger than 50.

5.3 Time Complexity Comparison

In order to show the efficiency of our proposed SB and OOB, we implement an exhaustive search scheme, referred to as scheme *EX*, to compare the solution quality of SB and OOB. Moreover, we demonstrate that SB and OOB are very efficient in determining backbones without scarifying the solution quality.

In Figure 5.4, we vary the number of queries and set randomness of overlaps to 50% and query range is set to 100 x 100 m^2. As can be seen in Figure 5.4, the execution time of algorithm OOB is almost half of the one of scheme EX. Moreover, the execution time of OOB and EX both increase exponentially with the number of queries. On the other hand, the time spent for SB only slightly and linearly increases.

In Figure 5.5, we vary query range and set the number of queries to 10 and randomness of overlaps to 50%. It can be seen that the execution time of algorithm OOB and scheme EX increase linearly with query range. On the other hand, the time spent for SB only slightly

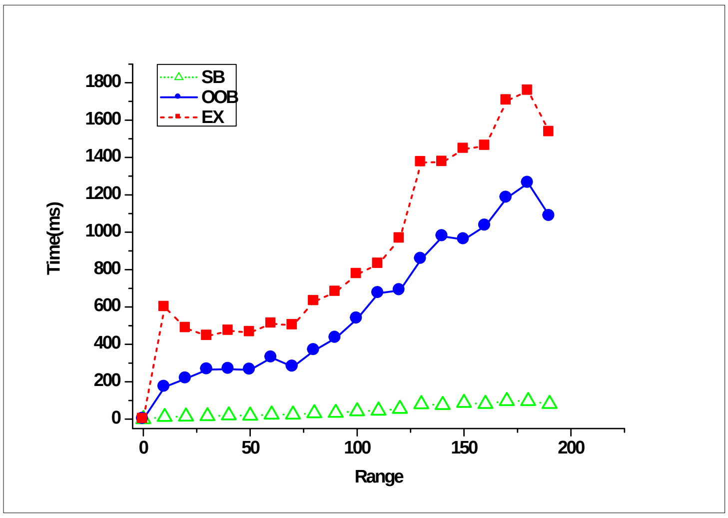

Figure 5.5: Performance results of SB, OOB and EX as Range varies.

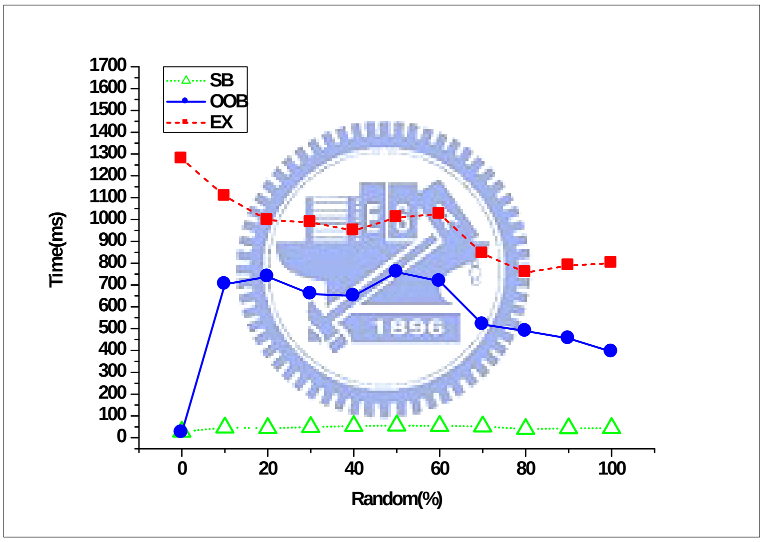

Figure 5.6: Execution time of SB, OOB and EX with various values of Random.

and linearly increases and is almost constant related to the ones of OOB and EX, showing the advantage of algorithm SB.

In Figure 5.6, we vary randomness of overlaps and set the number of queries to 10 and query range to 100 x 100 m^2. The time spent for EX is stable within a range. The one spent for OOB sharply increase while the randomness of overlaps goes from 0 to 10, and is stable after randomness of overlaps is larger than 10. On the other hand, the time spent for SB is almost constant and is far less than the ones spent for OOB and EX. In summary, the time

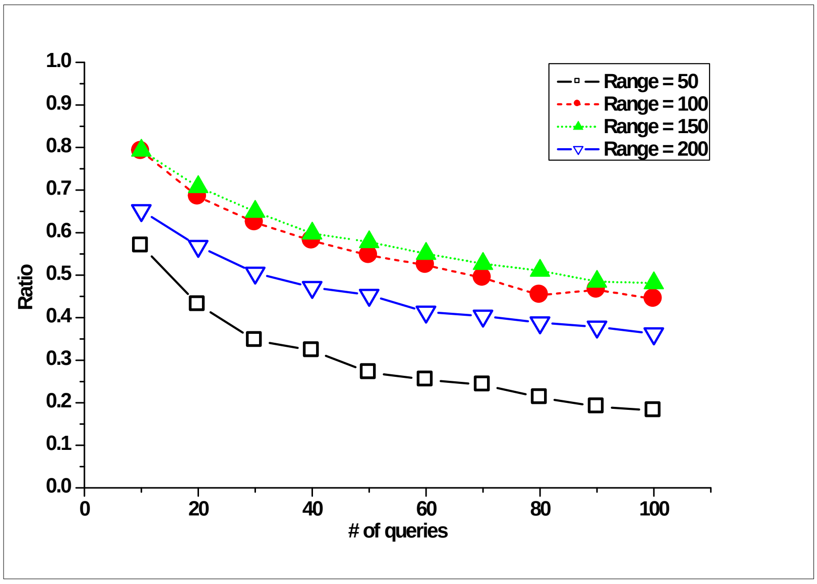

Figure 5.7: Performance study of SB with the number of queries varied.

spent for SB is extremely less than the ones of OOB and EX, but the performance of SB is very close to the ones of OOB and EX.

5.4 Performance of Algorithm SB

In this experiment, we measure the stability of algorithm SB. That is, we vary two parameters for each sub-experiment and compare the performance of SB while the values of these parameters are changing. The performance of SB is measured by a factor *Ratio*, which is the number of messages spent for all queries after applying SB divides the one before applying SB. The smaller the Ratio is, the more number of messages are reduced.

In Figure 5.7, we vary the number of queries and query range. As we can see, the Ratio slightly decreases as the number of queries increase. Moreover, the Ratio increases while query range goes from 50 to 150 and decreases after query range is larger than 150.

In Figure 5.8, we varying number of queries and randomness of overlaps. The ratio of algorithm SB increases when randomness of overlaps increases. That is, when randomness of overlaps is higher, the number of messages that SB can reduce is fewer. Moreover, Ratio

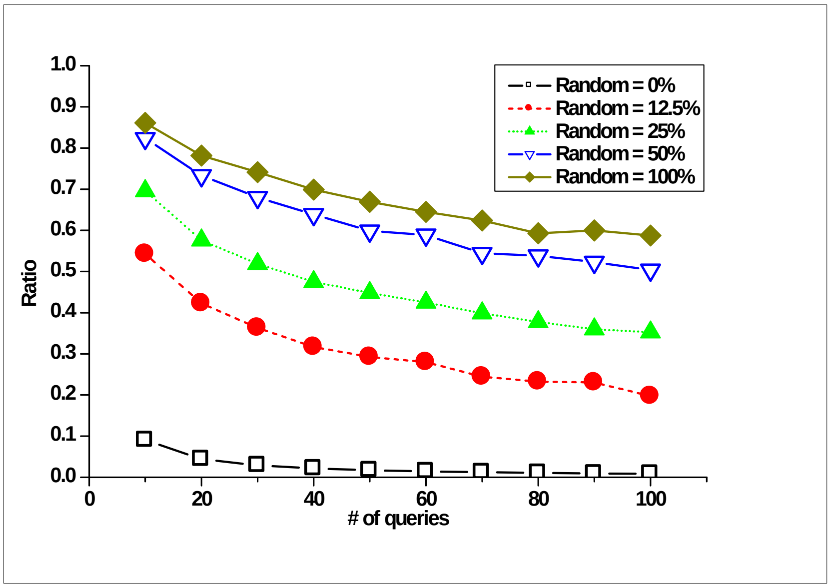

Figure 5.8: Evaluating on number of queries and Random

rise sharply when randomness of overlaps changes from lower value and rise softly when randomness of overlaps changes from higher one.In Figure 5.9, we vary randomness of overlaps and query region. When randomness of overlaps is 0, Ratio is almost a constant. When randomness of overlaps is larger than 0, Ratio increases sharply when query range goes from 0 to 150 and decreases softly after query range is larger than 150. This result is very similar to those in Figure 5.2.

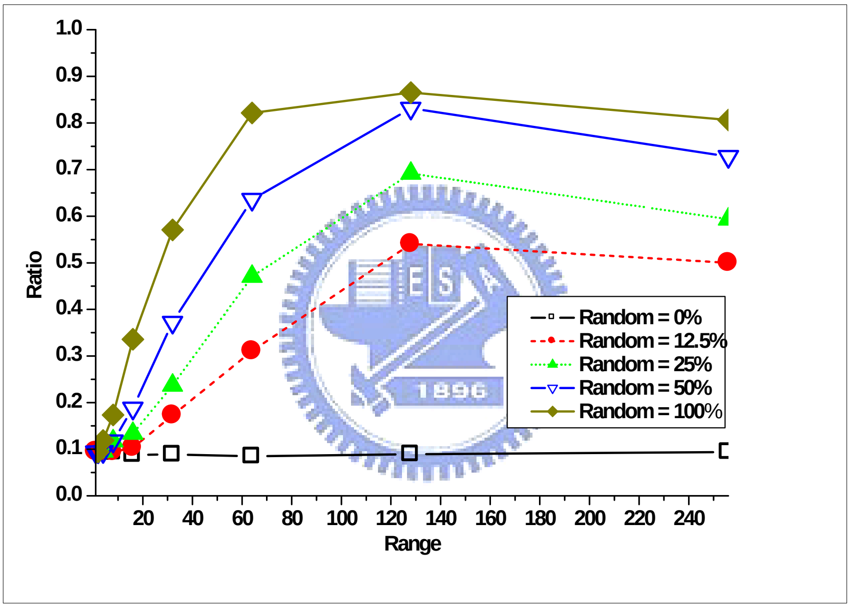

Figure 5.9: Performance study of SB with Range and Random varied.