Title	Trading Sensing Coverage for an Extended Network Lifetime
Authors(s)	Lim, Jong Chern, Bleakley, Chris J.
Publication date	2013-04
Publication information	Lim, Jong Chern, and Chris J. Bleakley. “Trading Sensing Coverage for an Extended Network Lifetime.” Springer, April 2013. https://doi.org/10.1007/s11235-011-9595-0.
Publisher	Springer
Item record/more information	http://hdl.handle.net/10197/7129
Publisher's statement	The final publication is available at www.springerlink.com.
Publisher's version (DOI)	10.1007/s11235-011-9595-0

Noname manuscript No.
(will be inserted by the editor)

Trading Sensing Coverage For An Extended Network Lifetime

Jong Chern Lim · C.J. Bleakley

Received: date / Accepted: date

Abstract One of the main benefits of using Wireless Sensor Networks (WSNs) is that they can be deployed in remote locations without any prior infrastructure. Because of this nodes are normally battery powered. This limits the performance of the network. In this paper, we propose a novel method of scheduling nodes based on a user's sensing coverage requirement. Through the use of our proposed scheduling algorithm (Ncut-GA), it is shown that the duration in which the user's coverage requirement is met can be extended. When compared with a previously published algorithm (Greedy-MSC), the proposed algorithm is able to increase the coverage duration by up to 80%. Furthermore it is also shown that the duration of which the WSN can operate till the first node dies can be improved by up to 200% through the use of Ncut-GA.

Keywords Sensing Coverage · Scheduling · Genetic Algorithm

1 Introduction

The field of Wireless Sensor Networks garnered significant research interest over the last decade because of the flexibility and potential benefits it can offer. WSNs are possible due to swift improvements in wireless communications, processor design and microelectromechanical systems (MEMS). A single wireless node normally consists of a radio, micro-controller and a variety of sensors. These nodes operate autonomously. They are able to communicate wirelessly and do not rely on external power supply. This enables them to be

Jong Chern Lim
Complex & Adaptive Systems Laboratory,
School of Computer Science and Informatics,
University College Dublin,
Dublin, Ireland
E-mail: jong.lim@ucdconnect.ie

deployed quickly, even in the harshest of environments where no infrastructure is available. Once deployed, nodes are able to self organize into smart networks which can carry out sensing tasks.

WSNs have potential uses and benefits in a wide variety of applications, such as health care, military surveillance and environmental monitoring. The general consensus from those involved with WSN research is that due to their energy and memory constraints, WSNs require application specific algorithms and software. Based on application requirements, various methods of scheduling node activity have been proposed to increase battery lifetime. For instance, if an application only requires daytime monitoring then nodes can be directed to sleep during the night. This is important because, as shown in [8], it is possible to reduce power consumption by 43% by switching the radio to sleep mode rather than idle mode.

The quality of service provided by a coverage algorithm depends on the application it is used for. Some applications might have an emphasis on coverage over lifetime while others an emphasis on lifetime over coverage. In this paper we propose a coverage algorithm suitable for applications which require some tradeoff on coverage for an extended network lifetime. For instance instead of detecting 99% of events, an application might want to extend lifetime while being able to detect 80% of events. Our proposed algorithm is evaluated in terms of two definitions of lifetime 1) the duration till the first node dies , and 2) the duration the algorithm is able to detect $x\%$ of events where x is a predefined limit.

The proposed algorithm makes use of high node densities to find disjoint subsets of nodes which can operate in a round robin fashion while maintaining a user specified sensing coverage. Disjoint subsets are such that for any two subsets C_i and C_j, $C_i \cap C_j = \phi$. Thus any given node in the network only belongs to a single subset. The purpose of using disjoint subsets is to achieve network load balancing. While picking the best nodes to monitor an area will provide the best coverage possible it will cause the premature death of these nodes. This is undesirable as these nodes may be needed for other tasks as well. Furthermore we show that through the use of disjoint subsets the duration of which the coverage meets the user's requirements can be extended.

Our proposed algorithm Ncut-GA uses a normalized cut clustering algorithm to form the initial subsets. The overall performance of these initial subsets are then improved through the use of a Genetic Algorithm. Ncut-GA is compared with Greedy-MSC which is an algorithm proposed in [3]. Greedy-MSC is a sensing coverage algorithm which builds disjoint subsets in a greedy fashion. The difference between our proposed algorithm and Greedy-MSC is that Greedy-MSC does not make full use of all nodes within the network. It only builds the best possible set of subsets and leaves the remaining nodes to do nothing. This is a waste of resources. The algorithm presented herein maximizes node participation in the overall goal of the application. As will be shown this improves performance. The details of these algorithms are presented in Section 3.

The remainder of this paper is broken into four parts. In Section 2, we discuss related work and point out the novelty of our approach. In Section 3, we describe the proposed algorithms in detail. In Section 4, we examine the results and their implications. Finally, the paper ends with a conclusion.

2 Related Work

The main goal of scheduling in the context of WSNs is to intelligently control when nodes are switched between active and sleep states in order to conserve energy while meeting application specific goals. Herein, a novel algorithm is presented which schedules node activity to conserve energy while preserving sensing coverage. Coverage scheduling can be done in both a distributed and centralized manner.

In this paper, we propose a centralized approach to coverage scheduling which is suitable for use in two-tier networks, such as TENET [5]. In a two tier network, every node is a single hop from a node with unlimited power and a long range transmitter (master node) providing connectivity to the sink. As pointed out by [5] the benefits of using a two-tier network is that there is a higher likelihood of packet delivery and that the radios of master nodes which are more powerful can improve network capacity. A centralized approach is perfect for two tier networks as it will maximize the use of the master nodes and ensure a longer lifetime for the second tier nodes.

In a multihop network, nodes which do not have direct connectivity to the sink require relaying their data from one node to another until the packet arrives at the sink. Scheduling in a multihop network must take connectivity into consideration. Thus, each subset formed must form a mini-network which enables all nodes within the subset to send data back to the sink. The algorithms presented herein are not suitable for multihop networks as they do not take connectivity into consideration. However as pointed out by [12] for a set of sensors with full single coverage over an area, the communication graph is connected if the communication radius of a node is double that of the sensing radius. Thus if the schedules produced by the algorithm meet those requirements it will work in a multihop network.

Distributed approaches have been dealt with in a number of publications. A node self-scheduling algorithm is presented in [4]. In this algorithm, each node advertises its position and listens to obtain the location of its neighboring nodes. Based on this, nodes calculate the sensing area of all neighboring nodes. If the node's sensing coverage is fully covered by its neighbors then the node can choose to sleep. In order to avoid two nodes in the same vicinity going to sleep simultaneously, thus affecting coverage area, a back-off scheme is used. CCP [12] is an algorithm which can provide different levels of coverage for the user. Similar to [4], CCP requires the location of all its neighbors. When an active node receives a HELLO message from one of its neighbors it executes the coverage eligibility algorithm to determine whether it should remain active.

pCover [10] is a distributed approach which increases lifetime by trading it for reduced sensing coverage. Each node takes into the account the location of its neighbors. The sensing range of the node is imagined to be a virtual grid. Using the virtual grid, the node calculates the number of grid points which are within its sensing range, and the number of grid points which are covered by the sensing range of the neighboring nodes. The node decides its eligibility for switching to a sleep state based on a coverage percentage threshold which is set by the user.

The centralized approach to coverage scheduling can be broken into two main categories of algorithms, those designed for multihop networks and those designed for two tier networks. DSSP [1] and [7] are designed for multihop networks. DSSP is made up of three separate algorithms. The first is a redundancy checking algorithm. Its purpose is to check whether the sensing coverage of node i can be covered by other nodes and if removing the node will cause a disconnectivity. The second algorithm selects which nodes should remain active with respect to their remaining energy levels. The third algorithm handles the routing and decides whether or not to run the second algorithm again. The algorithm proposed in [7] is based on the minimum dominating set concept. It consist of three different phases. In the first phase, node connectivity information is collected and a graph for the network is constructed in the base station. In the second stage a collection of dominating sets of nodes are found. Finally these dominating sets are used to schedule the activity of the nodes.

The scheduling algorithms for two-tier networks can be divided into two classes, Disjoint Set Covers (DSC) [11] [2] and Multiple Set Covers (MSC) [3]. In [11] and [2] the term DSC is used for a collection of C subsets where every subset cover $C_i \subseteq C$ is disjoint and fully covers a set of T Targets. Subsets are disjoint meaning that for any two subsets C_i and C_j, $C_i \cap C_j = \phi$. In [2], a maximum flow problem is used to represent the Disjoint Set Cover problem. In order to find the DSC, the maximum flow problem is modeled as mixed integer programming.

In [11] conventional binary sensing disks are not used. Instead the sensing region of a node is defined according to the probability of missing an event and the probability of a false alarm. The sensing radius is calculated based on probabilities which can be set by the user. In this paper, we assume that the user sets the sensing range. In [11] an algorithm which is able to use cooperation between sensing nodes to increase sensing coverage is also presented. The paper uses a greedy algorithm similar to the one presented in [3] in order to form the DSC.

Unlike DSC, MSC allows a node to join multiple subsets. In [3], a Greedy algorithm called Greedy-MSC is used to find the optimum MSC. Results show that, through the use of Greedy-MSC, it is possible to have an average performance which is close to the upper bound. Each node is given a predicted lifetime value, each time a node participates in a subset, a certain percentage of its lifetime is deducted. A node is allowed to continue participating in different subsets till its lifetime value reaches 0. The paper claims that because the solution space of DSC is included in that of MSC, the optimal solution of the

MSC algorithm produces better results in terms of improving lifetime. In this paper Greedy-MSC is used as a sensing area coverage algorithm by covering the area with a dense set of uniformly placed target points.

The algorithm proposed herein, is different from other centralized algorithms because it makes an attempt to maximize the use of every node. This is done by first finding the number of required nodes needed for each subset to maintain sensing coverage and then maximizing the number of subsets which can be found. By maximizing the use of every node it is shown that the duration before the first node dies can be lengthened.

The centralized algorithms used in [11] and [3] build subsets which have full coverage. The algorithm presented herein also allows trading-off on sensing coverage for an increase in the number of subsets which in turn increases network lifetime.

3 Overview of Approach

In this research the underlying assumptions are:

- node locations are known
- two-tier network is used [5]
- the user defines node coverage as a disk of radius r

Figure 1 shows the Greedy-MSC algorithm which is used to form subsets with full sensing coverage. The target used is a virtual grid with every point within the grid making up target, T. The algorithm works by first finding the node which covers the most grid points. Once found, the node is assigned to a subset. The remaining nodes and grid points are updated accordingly. This process is repeated until all of the grid points have been covered. These nodes which give full sensing coverage are taken as the first subset. The subsequent subsets are found in the same way. The algorithm ends when no more subsets providing full coverage can be found. For the purpose of comparison Greedy-MSC has been modified so that the subsets found are not only limited to full coverage subsets.

Rather than only trying to maintain a 100% coverage of the area during the full period of operation the proposed algorithm herein (Ncut-GA) allows the tradeoff of sensing coverage for a longer lifetime. Given a percentage of coverage required by the user, the algorithm first works by calculating the number of sensor nodes which are required to maintain said coverage. This is done by using an equation proposed in [13]. Given that r is the sensing radius, l and m the respective lengths of the area and n the number of nodes then the expected coverage is:

$$E[C_n] = [1 - (\frac{\frac{1}{2}r^4 - \frac{4}{3}lr^3 - \frac{4}{3}mr^3 + \pi r^2 ml}{m^2 l^2})^n]lm \quad (1)$$

Once the number of nodes required n_{req} are found, clustering is done to split the full number of nodes into clusters with each having n_{req} number of

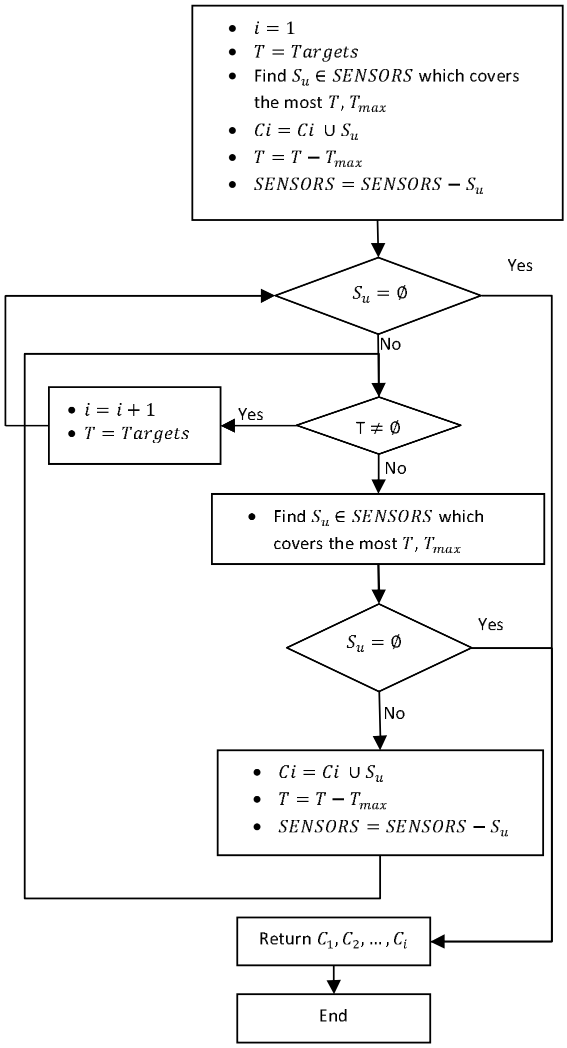

Fig. 1 Greedy-MSC Algorithm

nodes. Figure 2 shows how the proposed algorithm Ncut clustering algorithm works. The Normalized Cut (N-cut) [9] clustering algorithm first clusters the nodes based on the Euclidean distance between nodes. Once this is done the center node is chosen from each cluster. The center node is the node with the smallest average Euclidean distance within the cluster. These nodes form one subset. Once a subset has been found the nodes set is updated. The subsequent subsets are found by repeating the process. Once this is done these initial subsets are are passed through a Genetic Algorithm.

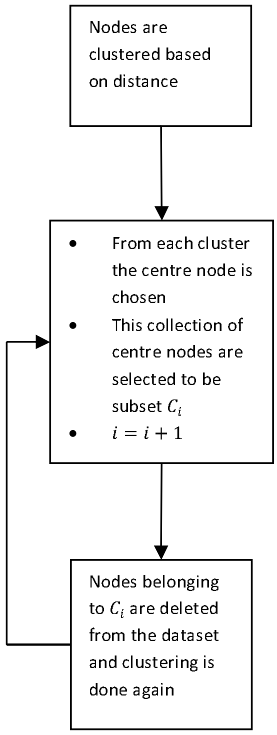

Fig. 2 Clustering Algorithm (Ncut)

As will be shown in the results, the performance of the subsets identified by Ncut decreases as the algorithm moves from the first subset identified to the last. This is not ideal as some applications require that all subsets have the same performance. A Genetic Algorithm (GA) is proposed to make the performance of the subsets equal. The algorithm works by randomly picking subsets and randomly swapping their nodes. A fitness function is used to test the newly formed subsets to decide whether the switch should be permanent. The percentage of coverage of a single subset is chosen as the fitness function. Given that t_i is a grid point belonging to the target T with x_{t_i} and y_{t_i} being its Cartesian coordinates, N_{C_i} the number of nodes in the subset C_i and S_u a node belonging to C_i with its coordinates represented by x_{S_u} and y_{S_u}, then the distance between a node S_u and a point t_i is

$$D = \sqrt{(x_{t_i} - x_{S_u})^2 + (y_{t_i} - y_{S_u})^2} \tag{2}$$

Given that N_t is the total number of target points and N_{t_i} the number of target points within the sensing radius of the nodes belonging to C_i, then the percentage of coverage given by C_i is

$$P_{C_i} = \frac{N_{t_i}}{N_t} \times 100 \tag{3}$$

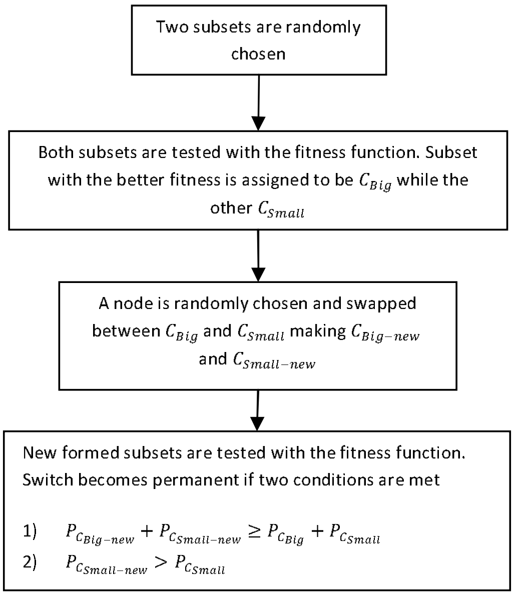

Fig. 3 Genetic Algorithm (GA)

The subsets found by Ncut-GA are used till a node dies after which the process is repeated.

4 Results

The algorithms were tested using Matlab and were applied to a random deployment of 100 nodes over an area. A virtual grid with spacing of 0.1m between each point was used as the Targets. Four hundred events were randomly simulated each day within the area. The performance of the algorithms are evaluated in terms of 1) The duration till the first node dies, and 2) The duration the algorithm is able to detect x% of events. Where x is the specified coverage requirement by the user. We assume that a node will transmit a single packet whenever it detects an event. Each node is allowed to transmit 100 times before it is marked as dead. Similar assumptions were made by [6].

Simulations were done using three different sensing radii (1.4m, 1.8m, 4.0 m). For each sensing radius, simulations were done 10 times. The results Coverage Performance and Lifetime Performance is the average taken from the 10 simulations. The performance of the different algorithms are compared with a Default Network. In the Default Network, every node is switched on. Greedy-MSC (100%) represents the full coverage of targets while Greedy-MSC (90%) is the modified version which allows only 90% of targets to be covered. Ncut-GA (90%) and Ncut-GA(80%) are 90% and 80% coverage algorithms respectively.

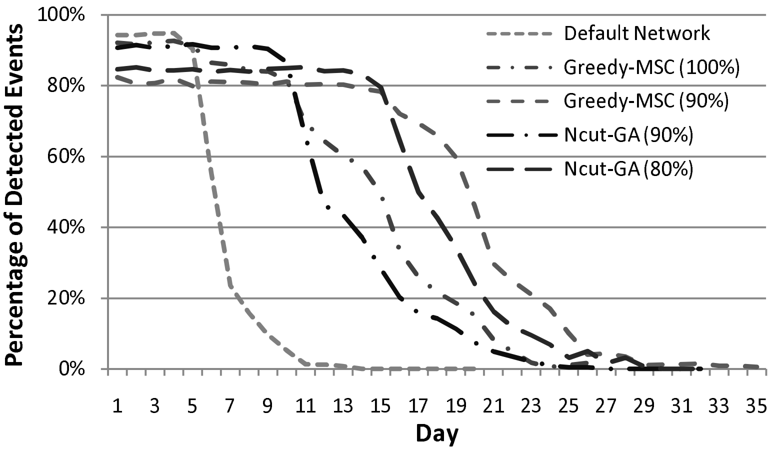

Fig. 4 Coverage Performance of Ncut-GA, Greedy-MSC and Default Network (Sensing Radius = 1.4m , Area=10x10m)

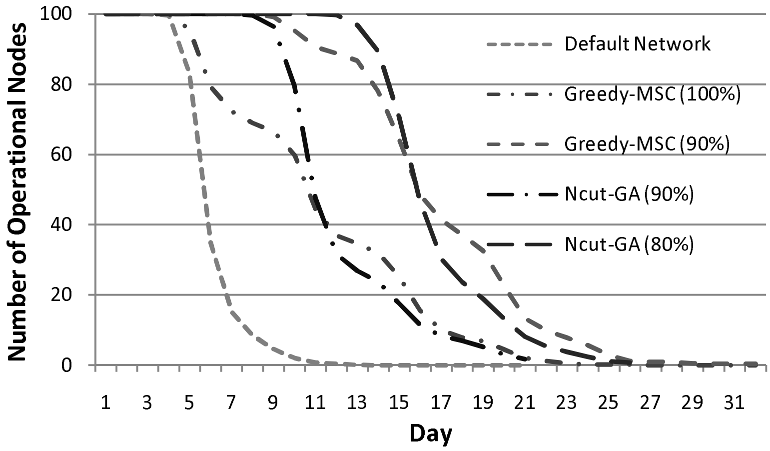

Fig. 5 Lifetime Performance of Ncut-GA, Greedy-MSC and Default Network (Sensing Radius = 1.4m , Area=10x10m)

The first three results are from simulations on a $10 \times 10m$ area while the final one is on a $15 \times 15m$ area.

Figure 4 shows the performance of Greedy-MSC, Ncut-GA, and a Default Network. In terms of percentage of detected events the Default Network performs the best. However it is only able maintain this high detection rate during the first five days. The reason for this is because the disadvantage of switching on all nodes is that each event is detected by multiple nodes causing many redundant transmissions of the same detected event. These extra transmissions quickly reduces the lifetime of the sensor node. Compared to the default network, the performance of Ncut-GA (90%) show that by sacrificing some level

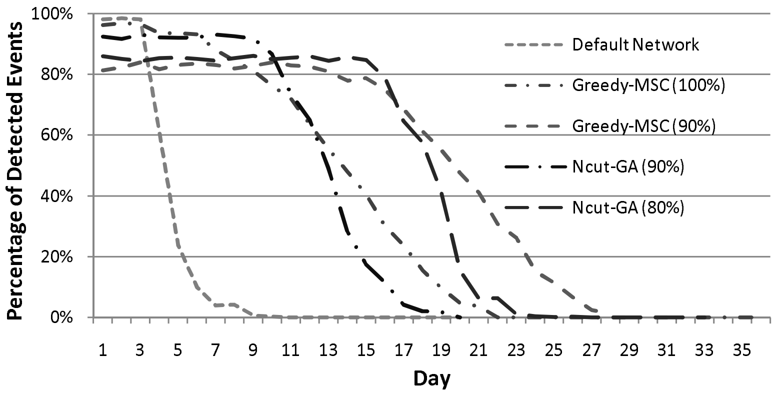

Fig. 6 Coverage Performance of Ncut-GA, Ncut, Greedy, Greedy-MSC and Default Network (Sensing Radius = 1.8m , Area=10x10m)

of coverage a high level of events can be detected for a much longer period of time. In terms of being able to detect more than 90% of events Ncut-GA (90%) performs 80% longer than the Default Network. Compared to Ncut-GA (90%), Greedy-MSC (100%) has a similar performance till day five after which the performance of Greedy-MSC (100%) drops rapidly. Ncut-GA (90%) on the other hand is able to sustain its level of performance till day nine. Next the performance of Ncut-GA (80%) is compared with Greedy-MSC(90%). As can be seen from Figure 4 the performance of Greedy-MSC (90%) using a 90% target limit doesn't map well with the actual performance. Ncut-GA (80%) performs better than Greedy-MSC (90%) until day 15 by an average of 3.5%, after which the performance of both algorithms drop rapidly.

The performance of the different algorithms in terms of the number of operational nodes throughout the operational period is presented in Figure 5. The figure shows that while being able to outperform Greedy-MSC (100%) in terms of better detecting events for a longer period of time, Ncut-GA (90%) operates 100% longer before the first node dies. Similarly when Greedy-MSC (90%) is compared with Ncut-GA (80%), Ncut-GA (80%) performs 37.5% longer before the first node dies.

Next the performance of the various sensing coverage algorithms are evaluated when the sensing radius of a node is set at 1.8m. Compared to the default network, by trading off on 10% of coverage Ncut-GA (90%) is able to detect more than 90% of events for a 200% longer duration. This is a much better performance than which was seen with a sensing radius of 1.4m. The reason for this is because as the sensing radius is increased more nodes detect the same event thus increasing the number of redundant transmissions. Included in the figure as well is the performance of Greedy-MSC (100%). Greedy-MSC (100%) performs better than Ncut-GA (90%) till day six, after which the performance drops below 90%. In that respect Ncut-GA (90%) performs 50% better. Com-

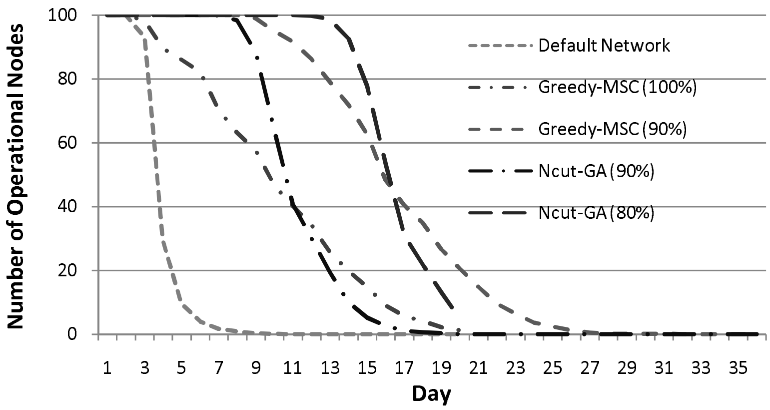

Fig. 7 Lifetime Performance of Ncut-GA, Ncut Greedy and Greedy-MSC (Sensing Radius = 1.8m , Area=10x10m)

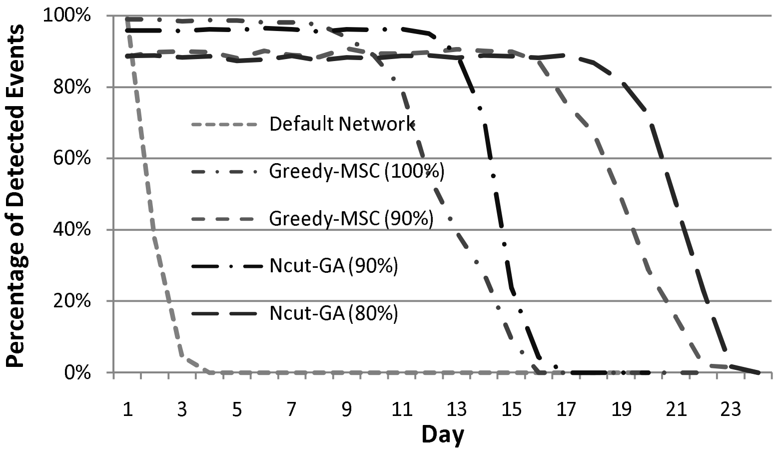

Fig. 8 Coverage Performance of Ncut-GA, Ncut, Greedy, Greedy-MSC and Default Network (Sensing Radius = 4.0m , Area=10x10m)

paring the performance of Greedy-MSC (90%) with Ncut-GA (80%) , Ncut-GA (80%) performs 3% better for a 15% longer duration above the 80% limit.

Figure 7 shows that Ncut-GA (90%) performs 200% better than Greedy-MSC (100%) in terms of duration till the first node dies. While out performing Greedy-MSC (90%) in terms of detected events, Ncut-GA (90%) also performs 42.8% longer before the first node dies.

Next we take a look at the results when the sensing radius was set to 4m. From the results (Fig. 8) it is seen that Greedy-MSC (100%) performs on average 2.5% better than Ncut-GA until the ninth day after which its performance drops below the 90% limit. Though Ncut-GA (90%) performs slightly worse it maintains a $>$ 90% detection rate for a 30% longer duration. In terms of detected events Greedy-MSC (80%) performs 1.4% better but

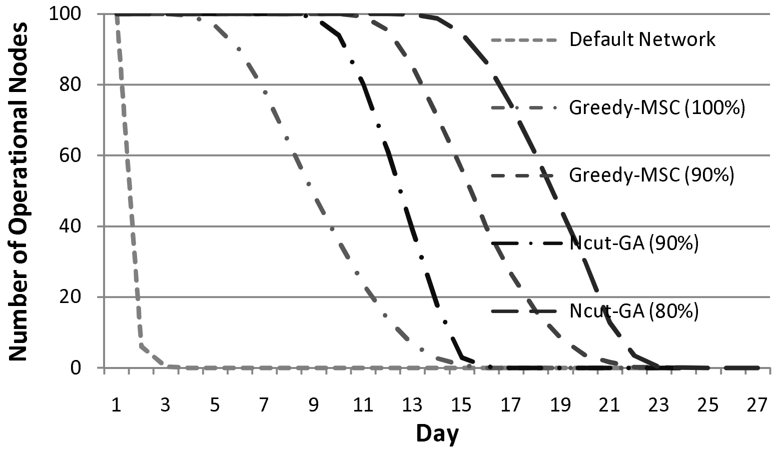

Fig. 9 Lifetime Performance of Ncut-GA, Ncut Greedy, Greedy-MSC and Default Network (Sensing Radius = 4.0m , Area=10x10m)

similarly to which was seen with a 90% limit Ncut-GA (80%) performs a few days longer above the 80% limit.

In terms of length of time till the first node dies, figure 9 shows that Ncut-GA performs longer than Greedy-MSC.

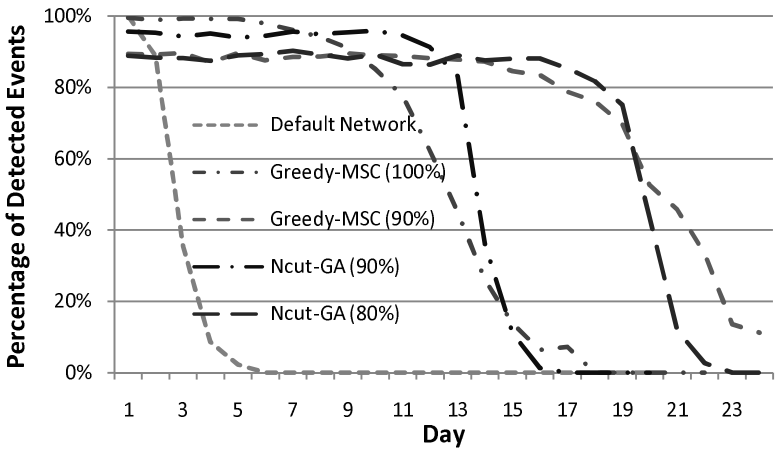

Fig. 10 Coverage Performance of Ncut-GA, Ncut, Greedy, Greedy-MSC and Default Network (Sensing Radius = 4.0m , Area=15x15m)

Finally results when the algorithm was tested on a $15 \times 15m$ area is presented. A sensing radius of 4m is used for this set of simulations. Similar results to that which was found when tested on the $10 \times 10m$ area is found. Figure 10 shows that compared to Greedy-MSC (100%), Ncut-GA (90%) performs 33.3%

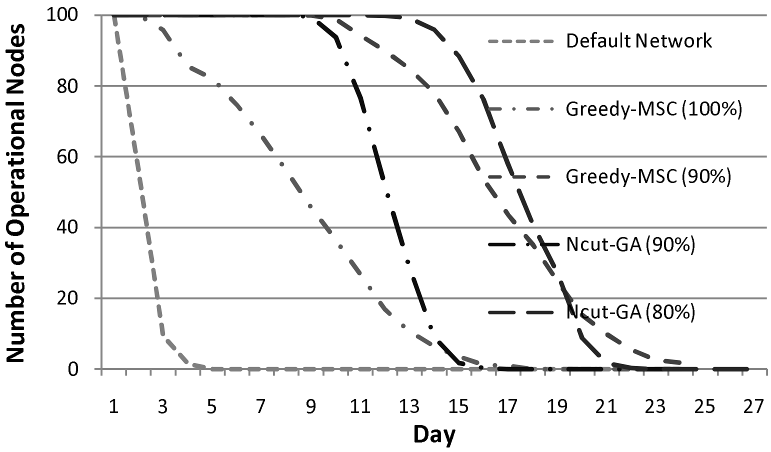

Fig. 11 Lifetime Performance of Ncut-GA, Ncut Greedy, Greedy-MSC and Default Network (Sensing Radius = 4.0m , Area=15x15m)

longer above the 90% limit. There is no significant difference between the performance of Ncut-GA (80%) and Greedy-MSC (90%) for fourteen days. On day fifteen the performance of Greedy-MSC (90%) begins to drop. Ncut-GA (80%) performs 12.5% longer above the 80%limit.

In terms of duration till the first node dies, Figure 11 shows that both Ncut-GA (90%) and Ncut-GA (80%) perform better than Greedy-MSC (100%) and Greedy-MSC (90%) respectively.

Tables 1 and 2 give the performance summary of both Ncut-GA and Greedy-MSC when compared with the Default Network. In terms of the two performance metrics used (duration till the first node dies, duration percentage of events detected are within limit) Ncut-GA outperforms Greedy-MSC in all simulation cases. Result comparison between Ncut-GA (90%) and Ncut-GA (80%) show that the coverage performance of Ncut-GA can be successfully scaled down to improve network lifetime.

As previously mentioned after the subsets have been found by the Ncut algorithm a Genetic Algorithm is used to improve the performance of the subsets. Next we take a look at the performance gained through the use of the genetic algorithm. Figure 12 shows the performance gain achieved in terms of percentage of coverage (Equation 3) from using the Genetic Algorithm. Using Ncut the performance of the last subset found performs the worse. Through the use of the Genetic Algorithm the performance of each individual subset is evened out and the overall performance is improved. The average performance gain achievable from using the genetic algorithm decreases as the number subsets found using the Ncut algorithm increases. For example on three subsets with a sensing radius of 1.6m the average performance gained from using the genetic algorithm is 9%. The performance gained from using the genetic algorithm on 10 subsets (with a sensing radius of 3.5m) using Ncut is 3.37%. Even though on average there is only a small improvement in performance a

Table 1 Improvement in Performance by Ncut-GA, Greedy-MSC With Respect to Default Network (90% Coverage Limit)

Method	Area	Number of Nodes	Sensing Radius	Duration Till 1st Node Dies	Duration % of Detected Events are Within Limit
Ncut-GA (90%)	10m x 10m	100	1.4m	100%	100%
Greedy-MSC (100%)	10m x 10m	100	1.4m	0%	0%
Ncut-GA (90%)	10m x 10m	100	1.8m	200%	200%
Greedy-MSC (100%)	10m x 10m	100	1.8m	0%	100%
Ncut-GA (90%)	10m x 10m	100	4.0m	600%	1200%
Greedy-MSC (100%)	10m x 10m	100	4.0m	200%	800%
Ncut-GA (90%)	15m x 15m	100	4.0m	700%	1100%
Greedy-MSC (100%)	15m x 15m	100	4.0m	100%	800%

Table 2 Improvement in Performance by Ncut-GA, Greedy-MSC With Respect to Default Network (80% Coverage Limit)

Method	Area	Number of Nodes	Sensing Radius	Duration Till 1st Node Dies	Duration % of Detected Events are Within Limit
Ncut-GA (80%)	10m x 10m	100	1.4m	267%	200%
Greedy-MSC (90%)	10m x 10m	100	1.4m	167%	160%
Ncut-GA (80%)	10m x 10m	100	1.8m	400%	400%
Greedy-MSC (90%)	10m x 10m	100	1.8m	250%	333%
Ncut-GA (80%)	10m x 10m	100	4.0m	1100%	1700%
Greedy-MSC (90%)	10m x 10m	100	4.0m	900%	1500%
Ncut-GA (80%)	15m x 15m	100	4.0m	1000%	1700%
Greedy-MSC (90%)	15m x 15m	100	4.0m	700%	1500%

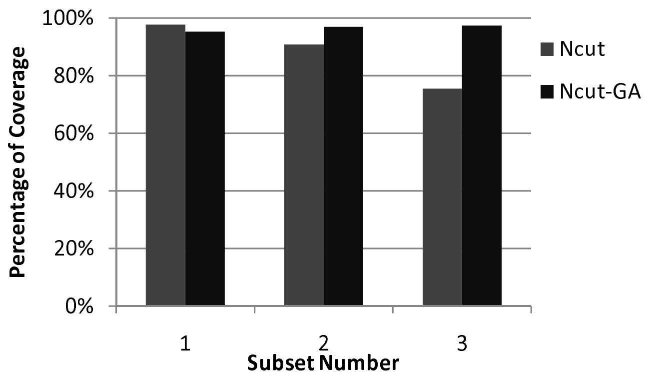

Fig. 12 Average Percentage of Coverage vs. Subset Number

closer look at the results reveals that the performance gained by the tenth subset is 21.7%. However as the first nine subsets found by Ncut have a good performance the average performance gained from using the genetic algorithm is not reflected. Thus even where a large number of subsets are needed it is still important to use the Genetic-Algorithm to ensure the performance of the last subset found is maintained at a high level.

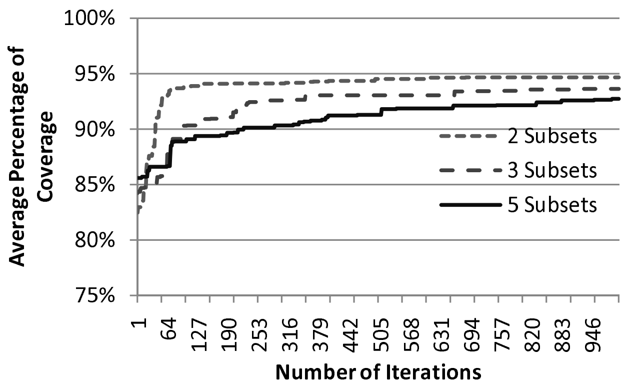

Fig. 13 Percentage of Coverage vs. Number of Iterations

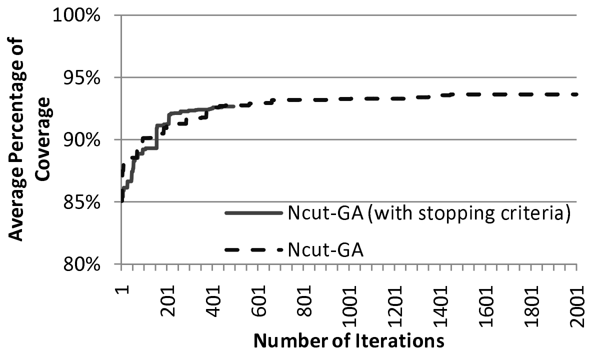

Fig. 14 Lifetime Performance of Ncut-GA, Ncut Greedy, Greedy-MSC and Default Network (Sensing Radius = 1.8m)

Figure 13 shows the relationship between the number of iterations and the average percentage of coverage. Given that N_C is the number of subsets in C then the Average Percentage of Coverage given by C is:

$$P_C = \frac{1}{N_c} \times \sum_{i=1}^{N_c} P_{C_i} \tag{4}$$

where P_{C_i} is given by Equation 3. Results show that the knee of the curve occurs at different iteration points for different subset sizes. As a stopping criteria we propose that the genetic algorithm be ended when there is no improvement in performance after a number of iterations. As an example Figure 14 shows the performance difference when a stopping criteria of 50 iterations is used. There is only a minor loss in performance from using the stopping criteria. It is also proposed that setting the stopping criteria as the specified coverage limit x not be done because as Figure 12 shows the performance of the subsets can exceed that limit.

5 Conclusion

In this paper, we proposed a disjoint sub-setting algorithm which can trade-off some coverage for an extended lifetime. Using Equation 1 the number of nodes per subset needed in order to meet the user's coverage requirement is estimated. Once found Ncut-GA is used to find the disjoint subsets. The performance of the Ncut-GA algorithm is compared with a Default Network as well as a previously proposed algorithm Greedy-MSC which provides full coverage disjoint subsets. Compared to the previously published algorithm Greedy-MSC (100%), Ncut-GA (90%) is able to increase the coverage duration by up to 80% as well as to lengthen the duration it takes for the first node dies by up to 200%. To further test the performance of Ncut-GA, modifications were made to Greedy-MSC to allow it to work with different coverage limits. Results show that Ncut-GA still extends the duration $x\%$ events are detected by up to 15% and lengthens the duration till the first node dies by up to 43%. Ncut-GA is able to achieve a much better performance because it is able to maximize the percentage of node participation when the disjoint subsets are found.

References

1. Bulut, E., Korpeoglu, I.: DSSP: A Dynamic Sleep Scheduling Protocol for Prolonging the Lifetime of Wireless Sensor Networks. In: Advanced Information Networking and Applications Workshops, 2007, AINAW'07. 21st International Conference on, vol. 2 (2007)
2. Cardei, M., Du, D.: Improving wireless sensor network lifetime through power aware organization. Wireless Networks **11**(3), 333–340 (2005)
3. Cardei, M., Thai, M., Li, Y., Wu, W.: Energy-efficient target coverage in wireless sensor networks. In: Proceedings IEEE INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3 (2005)
4. Georganas, N.: A coverage-preserving node scheduling scheme for large wireless sensor networks. In: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, pp. 32–41. ACM New York, NY, USA (2002)
5. Gnawali, O., Jang, K., Paek, J., Vieira, M., Govindan, R., Greenstein, B., Joki, A., Estrin, D., Kohler, E.: The tenet architecture for tiered sensor networks. In: Proceedings of the 4th international conference on Embedded networked sensor systems, pp. 153–166. ACM New York, NY, USA (2006)
6. Gupta, H., Navda, V., Das, S., Chowdhary, V.: Efficient gathering of correlated data in sensor networks. ACM Trans. Sen. Netw. **4**(1), 1–31 (2008). DOI http://doi.acm.org/10.1145/1325651.1325655
7. Pazand, B., Datta, A.: An energy-efficient node-scheduling scheme for wireless sensor networks based on minimum dominating sets. International Journal of Network Management (2008)
8. Raghunathan, V., Schurgers, C., Park, S., Srivastava, M.: Energy-aware wireless microsensor networks. IEEE Signal Processing Magazine **19**(2), 40–50 (2002)
9. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on pattern analysis and machine intelligence **22**(8), 888–905 (2000)
10. Wang, L., Kulkarni, S.: Sacrificing a little coverage can substantially increase network lifetime. Ad Hoc Networks **6**(8), 1281–1300 (2008)
11. Wang, W., Srinivasan, V., Chua, K., Wang, B.: Energy-efficient coverage for target detection in wireless sensor networks. In: Proceedings of the 6th international conference on Information processing in sensor networks, pp. 313–322. ACM New York, NY, USA (2007)

12. Wang, X., Xing, G., Zhang, Y., Lu, C., Pless, R., Gill, C.: Integrated coverage and connectivity configuration in wireless sensor networks. In: Proceedings of the 1st international conference on Embedded networked sensor systems, pp. 28–39. ACM New York, NY, USA (2003)
13. Yen, L., Yu, C., Cheng, Y.: Expected k-coverage in wireless sensor networks. Ad Hoc Networks **4**(5), 636–650 (2006)