Title	Trading Sensing Coverage For An Extended Network Lifetime
Authors(s)	Lim, Jong Chern, Bleakley, Chris J.
Publication date	2010-10-15
Publication information	Lim, Jong Chern, and Chris J. Bleakley. "Trading Sensing Coverage For An Extended Network Lifetime." IEEE, October 15, 2010. https://doi.org/10.1109/WMNC.2010.5678753.
Conference details	3rd Joint IFIP Wireless and Mobile Networking Conference (WMNC), Budapest, Hungary, 13 - 15 October 2010
Publisher	IEEE
Item record/more information	http://hdl.handle.net/10197/7074
Publisher's statement	© © 2010 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Publisher's version (DOI)	10.1109/WMNC.2010.5678753

Trading Sensing Coverage For An Extended Network Lifetime

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

C.J. Bleakley
Complex & Adaptive Systems Laboratory,
School of Computer Science and Informatics,
University College Dublin,
Dublin, Ireland
Email: chris.bleakley@ucd.ie

***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 33%. 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 125% through the used of Ncut-GA.**

I. 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. This enables them to be 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 [1], 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 application which requires 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 the algorithm is able to detect x% of events, and 2) the duration till the first node dies.

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 node may be needed for other task as well. Furthermore we show that through the use of disjoint subsets the duration of which the coverage meets the users 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 and Greedy-MSC which is an algorithm proposed in [2]. Greedy is an algorithm which picks the best nodes to operate. Greedy-MSC is a coverage algorithm which uses disjoint subsets as well. The different between our proposed algorithm and Greedy-MSC is that Greedy-MSC does not make full use of all nodes within the network. They only build the best possible set of subsets and leave the remaining nodes to do nothing. This is a waste of resources. The algorithm presented herein ensures that all nodes participate 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.

II. 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 [3]. 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 [3] 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 [4] 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 [5]. 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 [4] is an algorithm which can provide different levels of coverage for the user. Similar to [5], 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 [6] 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 [7] and [8] 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 [8] 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) [9] [10] and Multiple Set Covers (MSC) [2]. In [9] and [10] 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 [10], 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 [9] 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 [9] 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 [2] in order to form the DSC.

Unlike DSC, MSC allows a node to join multiple subsets. In [2], 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. However, in the MSC algorithm, allowing some nodes to participate in multiple subsets means that certain nodes will die earlier. This may significantly impact coverage. In this paper, in order to compare results with Greedy-MSC we set the lifetime cost for a node to participate in a subset to 100%, thus allowing a node to only participate in one subset.

The algorithm proposed herein, is different from other centralized algorithms because it makes full use of every node. The centralized algorithms used in [9] and [2] build subsets which have full coverage. However they do not guarantee the full participation of every node within the network. The algorithm presented herein also allows trading-off coverage for an increase in the number of subsets which in turn increases network lifetime.

III. Overview of Approach

The proposed algorithm Ncut-GA is compared with two algorithms (Greedy and Greedy-MSC). In this research the underlying assumptions are:

- node locations are known
- two-tier network is used [3]
- the user defines node coverage as a disk of radius r
- lifetime L increases linearly as the number of subsets increase. For instance having 2 subsets compared to 1 will double the networks lifetime.

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. The 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.

The difference between Greedy and Greedy-MSC is that the Greedy algorithm ends once the first subset with full coverage is found. When a node within the operating subset dies the algorithm is used again. If a subset with full coverage can't be found both Greedy algorithms will pick the best n nodes to activate. The size of n is equal to the size of the last previous found subset which had full coverage.

Figure 2 shows how the proposed algorithm Ncut works. The Normalized Cut (N-cut) [11] 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.

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

$$D = \sqrt{(x_{t_i} - x_{S_u})^2 + (y_{t_i} - y_{S_u})^2} \tag{1}$$

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

$$P_{C_i} = \frac{N_{t_i}}{N_t} \times 100 \tag{2}$$

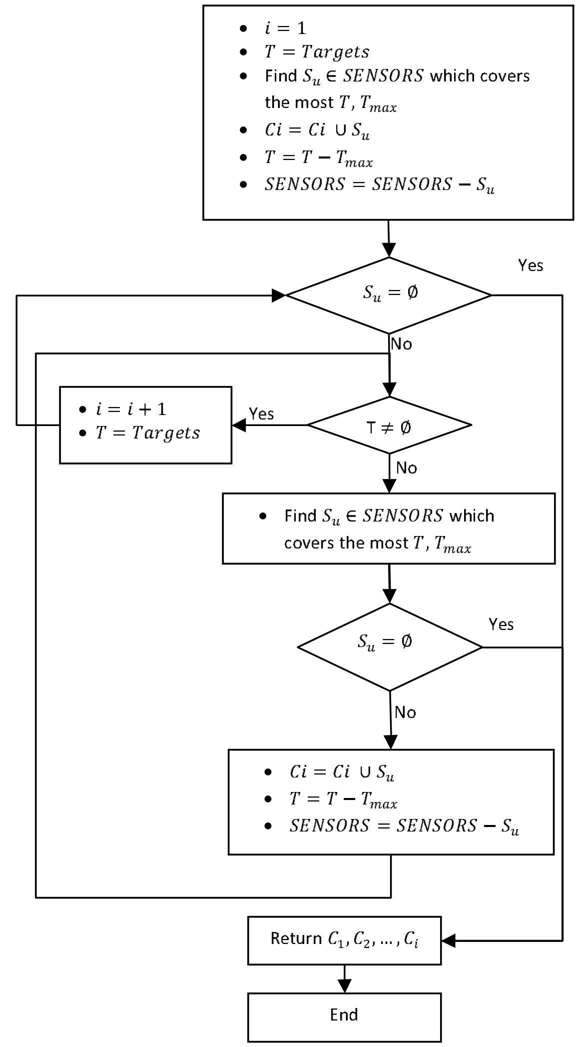

Fig. 1. Greedy Algorithm

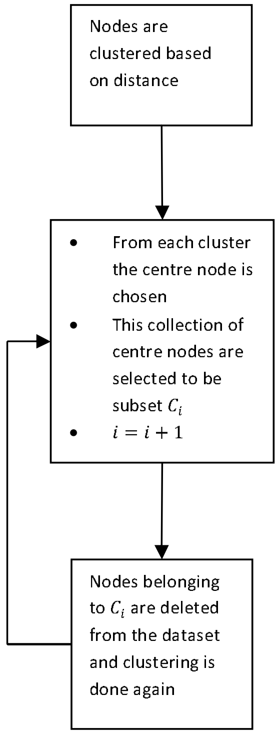

Fig. 2. Clustering Algorithm (Ncut)

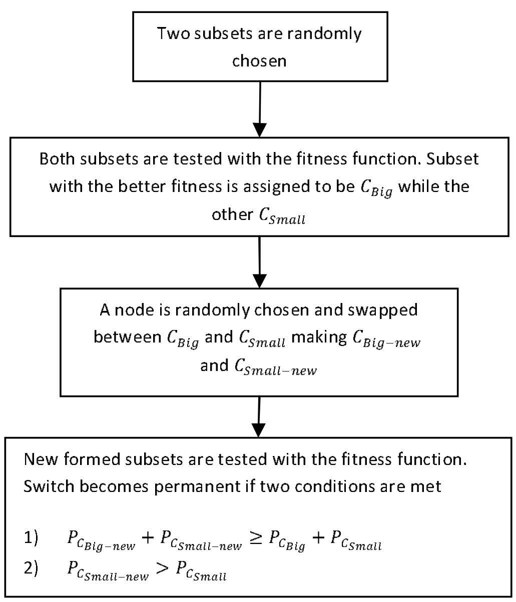

Fig. 3. Genetic Algorithm (GA)

IV. Results And Discussion

The algorithms were tested using Matlab and were applied to a random deployment of 100 nodes over a $10 \times 10m$ area. A virtual grid with spacing of 0.25m between each point was used as the Targets. Four hundred events were randomly simulated each day within the $10 \times 10m$ 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 [12].

An equation to calculate the coverage area given the lengths of the area, the number of nodes and the sensing radius of a node is 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:

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

Given a users coverage requirement this equation is used to estimate the number of nodes needed. For a sensing radius of 2m, and a coverage requirement of 90% it is calculated that the number of nodes needed is 20 thus we use Ncut-GA to divide the 100 node network into 5 different subsets. Figure 4 shows the performance of Greedy, Greedy-MSC, Ncut-GA and a Default Network. In the Default Network, every node is switched on. Results show that in terms of maintaining a 90% coverage Ncut-GA performs the best. There isn't much of a performance difference between Greedy and Greedy-MSC. Figure 5 shows that in terms of the length of time till the first node dies, Ncut-GA performs the best as well. Through the use of subsets, Greedy-MSC performs better than Greedy, however as the number of subsets formed is not maximized it can not match the performance of Ncut-GA. Table I summarizes the performance results of Greedy, Greedy-MSC, and Ncut-GA when compared with the Default Network implementation.

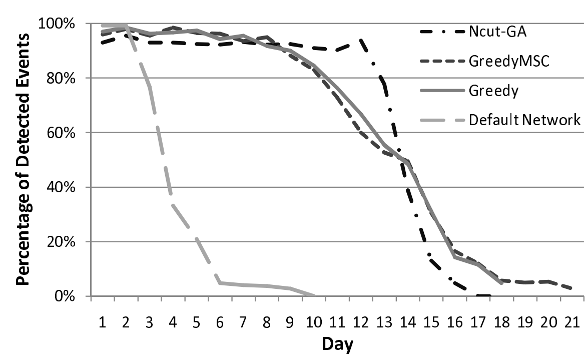

Fig. 4. Coverage Performance of Ncut-GA, Greedy, Greedy-MSC and Default Network (5 Subsets)

TABLE I
Improvement in Performance by Ncut-GA, Greedy, Greedy-MSC With Respect to Default Network (5 Subsets)

Method	Duration Till 1st Node Dies	Duration That Coverage is Within User Defined limit
Ncut-GA	350%	500%
Greedy-MSC	100%	300%
Greedy	0%	350%

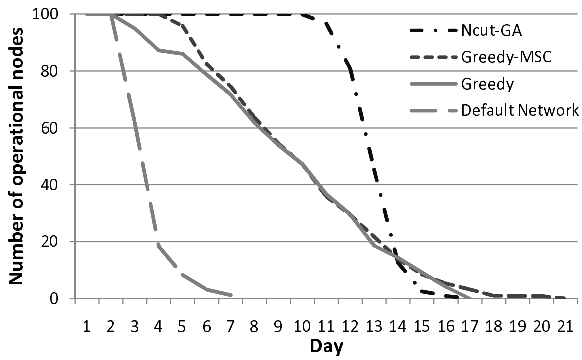

Fig. 5. Lifetime Performance of Ncut-GA, Greedy, Greedy-MSC and Default Network (5 Subsets)

Next the performance of Ncut-GA is evaluated when the sensing radius of a node is 3m. Using equation 3 and a 90% coverage requirement, it is determined that 10 nodes is needed. Ncut-GA is used to divide the network into 10 subsets. In Figure 6 we see that Ncut-GA performs the longest above the 90% coverage threshold. Similar to results seen when the radius is 2m, there is no performance difference between Greedy and Greedy-MSC. Figure 7 shows the number of nodes still in operation throughout the duration of the simulation. From the figure we see that Ncut-GA performs the longest till the first node dies. Table II summarizes the performance results of Greedy, Greedy-MSC, and Ncut-GA when compared with the Default Network implementation.

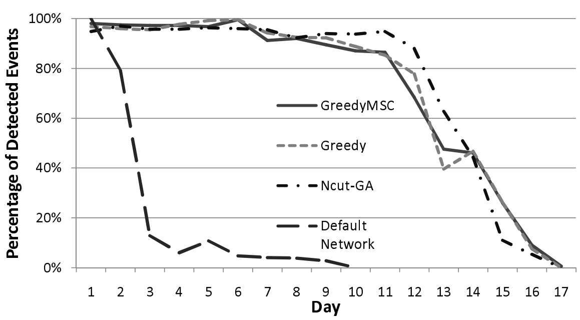

Fig. 6. Coverage Performance of Ncut-GA, Greedy and Greedy-MSC

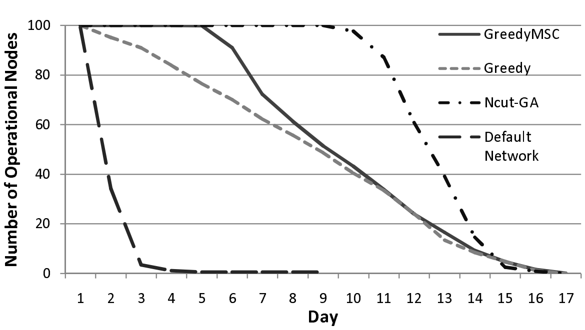

Fig. 7. Lifetime Performance of Ncut-GA, Greedy and Greedy-MSC

TABLE II
IMPROVEMENT IN PERFORMANCE BY NCUT-GA, GREEDY, GREEDY-MSC WITH RESPECT TO DEFAULT NETWORK (10 SUBSETS)

Method	Duration Till 1st Node Dies	Duration That Coverage is Within User Defined limit
Ncut-GA	800%	1000%
Greedy-MSC	400%	800%
Greedy	0%	800%

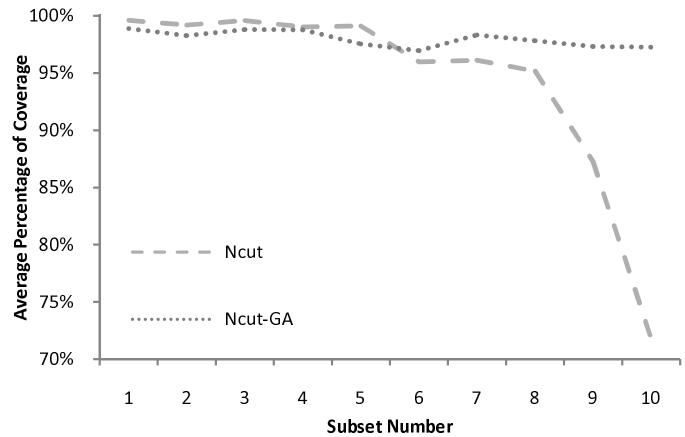

Fig. 8. Average Percentage of Coverage vs. Subset Number (10 Subsets)

As mentioned the GA is used to improve the initial subsets which were found by Ncut. Figure 8 shows the performance gain achieved in terms of percentage of coverage (Equation 2) 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.

Figure 9 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 Eq. 2. The performance increase gained through the use of the Genetic Algorithm begins to taper off around the 200th iteration. In the algorithm proposed, the genetic algorithm is ended when there is no improvement in performance after 50 iterations.

V. CONCLUSION

In this paper, we proposed a disjoint sub-setting algorithm which can tradeoff some coverage for an extended lifetime. Using Equation 3 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 disjoint subsets found by Ncut-GA is evaluated using two different performance metrics as lifetime. In terms of the length of time it can operate before the 1st

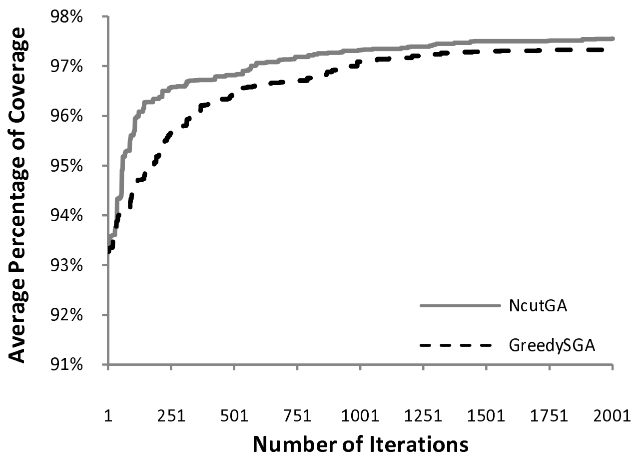

Fig. 9. Percentage of Coverage vs. Number of Iterations (10 Subsets)

node dies Ncut-GA is able to better the performance of the next best method Greedy-MSC by up to 125%. In terms of the duration Ncut-GA can perform within the user's coverage requirement it outperforms the rest of the algorithms by 33%.

Acknowledgment

This research was funded by Enterprise Ireland under project code CFTD/07/IT/303.

References

[1] V. Raghunathan, C. Schurgers, S. Park, and M. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 40–50, 2002.

[2] M. Cardei, M. Thai, Y. Li, and W. Wu, "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.

[3] O. Gnawali, K. Jang, J. Paek, M. Vieira, R. Govindan, B. Greenstein, A. Joki, D. Estrin, and E. Kohler, "The tenet architecture for tiered sensor networks," in *Proceedings of the 4th international conference on Embedded networked sensor systems*. ACM New York, NY, USA, 2006, pp. 153–166.

[4] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*. ACM New York, NY, USA, 2003, pp. 28–39.

[5] N. Georganas, "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*. ACM New York, NY, USA, 2002, pp. 32–41.

[6] L. Wang and S. Kulkarni, "Sacrificing a little coverage can substantially increase network lifetime," *Ad Hoc Networks*, vol. 6, no. 8, pp. 1281–1300, 2008.

[7] E. Bulut and I. Korpeoglu, "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.

[8] B. Pazand and A. Datta, "An energy-efficient node-scheduling scheme for wireless sensor networks based on minimum dominating sets," *International Journal of Network Management*, 2008.

[9] W. Wang, V. Srinivasan, K. Chua, and B. Wang, "Energy-efficient coverage for target detection in wireless sensor networks," in *Proceedings of the 6th international conference on Information processing in sensor networks*. ACM New York, NY, USA, 2007, pp. 313–322.

[10] M. Cardei and D. Du, "Improving wireless sensor network lifetime through power aware organization," *Wireless Networks*, vol. 11, no. 3, pp. 333–340, 2005.

[11] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[12] H. Gupta, V. Navda, S. Das, and V. Chowdhary, "Efficient gathering of correlated data in sensor networks," *ACM Trans. Sen. Netw.*, vol. 4, no. 1, pp. 1–31, 2008.

[13] L. Yen, C. Yu, and Y. Cheng, "Expected k-coverage in wireless sensor networks," *Ad Hoc Networks*, vol. 4, no. 5, pp. 636–650, 2006.