Power Efficient Monitoring Schedules in Sensor Networks

P. Berman G. Calinescu* A. Preethy† C. Shah† A. Zelikovsky†

***Abstract*— Optimizing the energy consumption in wireless sensor networks has recently become the most important performance objective. Since the battery life of the sensor nodes are directly related to the lifetime of the network, research community is aggressively addressing to energy optimization techniques.**

We assume the sensor network model in which (1) the energy consumption due to communication is proportional to the energy consumption for monitoring the environment (e.g. when each sensor directly communicates with the base) and (2) sensors can be scheduled to interchange idle and active modes. Given monitoring regions, battery life and energy consumption rate for each sensor, we formulate the problem of maximizing sensor network lifetime, i.e., time during which the monitored area is (partially or fully) covered.

Our contributions also include (1) an efficient data structure to represent the monitored area with at most n^2 points guaranteeing the full coverage which is superior to the previously used approach based on grid points, (2) efficient provably good algorithms for sensor monitoring schedule maximizing the total lifetime, (3) $(1+\ln(1-q)^{-1})$-approximation algorithm for the case when a q-portion of the monitored area is required to cover, e.g., for the 90% area coverage our schedule guarantees to be at most 3.3 times shorter than the optimum, (4) practical enhancement of the PTAS for packing linear programs by Garg and Könemann (5) extensive experimental study of the proposed algorithms showing significant advantage over existing methods in quality, scalability (it needs seconds to handle thousands of sensors) and flexibility.

I. Introduction

Wireless sensor networks have been the focus of considerable research during the past few years. The research issues currently addressed in wireless sensor networks are hardware constraints, communication and routing issues, data management problems, and software engineering principles. One of the most important issue apart from the above mentioned ones is energy optimization in wireless sensor networks.

Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA 16802-6106, E-mail: berman@cse.psu.edu

*Department of Computer Science, Illinois Institute of Technology, Chicago, IL 60616. E-mail: calinesc@cs.iit.edu.

‡Department of Computer Science, Georgia State University, Atlanta, GA 30303. E-mail: apreethy@cs.gsu.edu., chintan@alla.cs.gsu.edu, alexz@cs.gsu.edu.

A sensor network is composed of a large number of sensor nodes that are densely deployed either inside the environment or close to it. The position of sensor nodes need not be engineered or predetermined. This allows random deployment in inaccessible terrains or hazardous environments. Some of the most important application areas of sensor networks include military, natural calamities, health, and home. When compared to traditional ad hoc networks, the most noticeable point about sensor networks is that, they are limited in power, computational capacities, and memory. Hence optimizing the energy consumption in wireless sensor networks has recently become the most important performance objective. Since the battery life of the sensor nodes are directly related to the lifetime of the network, research community is aggressively addressing to energy optimization techniques.

The wireless sensor node, being a microelectronic device, can only be equipped with a limited power source. In some application scenarios, replenishment of power resources might be impossible. Sensor node lifetime, therefore shows a strong dependence on battery lifetime. In a multi hop ad hoc network, each node plays the dual role of data gatherer and data router. The malfunctioning of a few nodes can cause significant topological changes that might require rerouting of packets and reorganization of the network. Hence, power consumption and power management take on additional importance. It is for these reasons that researchers are currently focusing on the design of power aware protocols and algorithms for sensor networks.

The main task of a sensor node in a sensor network is to monitor events, perform quick local data processing, and then transmit the data. Power consumption can hence be divided into three domains: sensing, communication, and data processing. As noted in [1], the energy density of batteries has only doubled every 5 to 20 years, depending on the particular chemistry, and prolonged refinement of any chemistry yields diminishing returns. This shows that power management will be as critical in future sensor networks‘ as it now.

Our model of a sensor network is close to one described in [2], [3]. We assume that sensors are sprayed over the region R which is required to monitor, and each sensor p_i has its own *monitored region* R_i which it *covers*, i.e., p_i can collect the trustful data from R_i without help of any other sensor. We do not need to make any assumptions on the monitored region of each sensor, but for the clarity of presentation we will use disks of the same radius r. We assume that each sensor p_i has also a certain initial energy supply b_i which will be measured in time during which p_i can collect information from R_i. As mentioned above, we assume that the energy spent by sensors for aggregation and transmission collected data is proportional to the active time.

We also assume that the set of sensors is largely exceeds the amount necessary to monitor the required region R. Therefore, it is possible to turn some sensors in the sleeping mode while setting predefined time for internal clocks when they should wake up and start or continue monitoring. Note that number of times that sensors change their mode from sleeping to active has negligible effect on the energy consumption. The main constraint is that the monitored region R should be completely (or partially with specified portion of R) covered at any moment by active sensors. This assumption agrees with [4], where an OS-directed power management technique to improve the power efficiency of sensor nodes is proposed.

Below we give a formal definition of the energy preserving scheduling problem.

A set of sensors C covering R will be called *sensor cover*. Then a *monitoring schedule* is the set of pairs $(C_1,t_1),\ldots,(C_k,t_k)$, where C_i is a sensor cover and t_i is time during which C_i is active.

Maximum Sensor Network Life problem. Given a monitored region R, a set of sensors $p_1,\ldots,p_n$ and monitored region R_i and energy supply b_i for each sensor, find a monitoring schedule $(C_1,t_1),\ldots,(C_k,t_k)$ with the maximum length $t_1+\ldots+t_k$, such that for any sensor p_i the total active time does not exceed b_i.

Note that this formulation has never been clearly stated in the previous literature. In [2] the problem has been reduced to so called *disjoint set cover problem*. There it is assumed that any sensor cannot participate in different sensor covers, i.e., it can be only once change its mode from sleeping to active. Then all sensor covers should be disjoint.

Besides Slijepcevic and Potkonjak [2], Cardei and Du [5] also discuss the construction of disjoint set covers with the goal of extending the lifetime of wireless sensor networks. A similar problem but in a different model has been studied by Zussman and Segall [6]. They assume that the most of energy

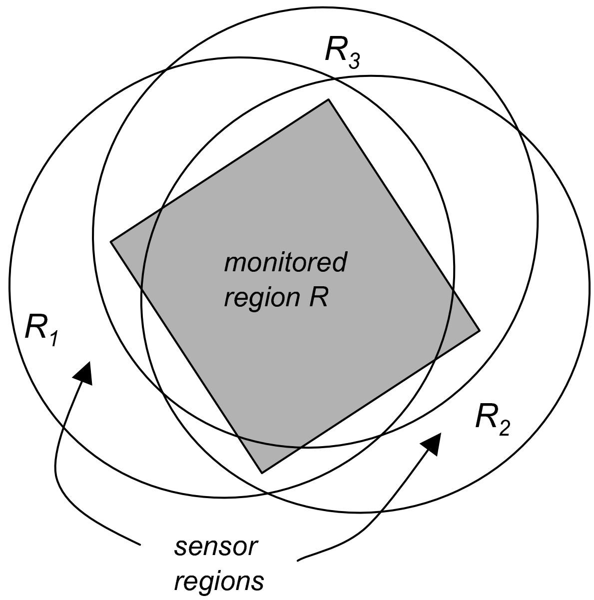

Fig. 1. 3 sensors with the disc monitored regions R_1, R_2 and R_3, respectively. The dark square in the center is the monitored region R. Any two sensors can cover R but any single one does not cover R.

consumption of wireless networks comes from routing the traffic, rather than monitoring. They look for the best traffic flow routes for a given set of traffic demands using concurrent flow approaches [7]. Marks et. al. [8] also consider lifetime maximization problems; however their published conference version contains neither theoretical analysis of the complexity of the problem nor experimental results validating the chosen heuristics.

Clearly, possibility of multiple mode change is easy to achieve and below we give an example illustrating the advantage of such possibility see Figure 1. In this example there are 3 sensors with disc monitored regions which are supposed to monitor a dark square region R inside. Assume that each sensor has 2 units of energy supply. Then there exist only a single disjoint sensor cover which can last for 2 units of time. On the other hand, the schedule $(\{p_1,p_2\},1)$, $(\{p_2,p_3\},1)$, $(\{p_3,p_1\},1)$ is clearly feasible and lasts for 3 units of time.

In order to solve this problem we use the *primal-dual* approach. This approach requires to solve the following dual problem.

Minimum Weight Sensor Cover problem. Given a monitored region R, a set of sensors $p_1,\ldots,p_n$ and monitored region R_i and the weight w_i for each sensor, find sensor cover with the minimum total weight.

In section IV we explain Garg-Könemann algorithm [9] for solving packing linear programs, which allows to solve the Maximum Sensor Network Life problem with almost the same guarantee with which one can solve the dual problem, i.e., Minimum Weight Sensor Cover problem. We also give a practical enhancement of their approach resulted in fast improvement of the quality of obtained solutions.

In section II we give an efficient data structure for representation of the monitored area with at most n^2 points guaranteeing the full coverage which is superior to the previously used approach based on grid points [2]. This data structure allows efficiently reduce the Minimum Weight Sensor Cover problem to the standard weighted set cover problem. This problem cannot be efficiently solved with guarantee better than $O(\log k)$, where k is the size of the largest set and it is not known to have any better algorithm for the case of planar sets. Anyway, our primal-dual solution is the first algorithm guaranteeing solution which is at most $O(logn)$ worse than the optimum, where n is the number of sensors.

In many cases, it can be required to cover sufficiently large portion of the monitored region R rather than the entire R. In section III we formulate the respective problem when only $q \cdot area(R)$ for a given $q \in [0,1]$. We then give an approximation algorithm which finds the solution at most $(1+\ln(1-q)^{-1})$ times heavier than the optimum. For instance, when $q = 0.9$, i.e., we wish to cover 90% of the region R, this implies 3.3-approximation algorithm for the Minimum Weight Sensor 0.9-Cover problem as well as the Maximum Sensor Network Life problem.

We have implemented all the suggested algorithms and present our results in section V. They show robustness of our approach give the trade-off between the sensor network lifetime and the required portion q of the monitored region to be covered.

II. Efficient Data Structure Representing Sensor Coverage

In [2] a grid data structure is used to express sensor node coverage over an area. They establish a set of grid points that form a $g \times g$-array to discretize the area. Then they determine for each point the subset of covering sensors disks. Next they partition all grid points into *fields*, where a field is defined as a subset of grid points covered by the same set of sensors. Similar idea has been suggested in [1]. The main advantage of this coverage model is ease of implementation if the covering area does not have to be delineated very precisely. If there are few points in the grid, then the area to be covered is not well defined. On the other hand, if the grid is very fine, then the calculation overhead for this model becomes quite large. Another challenging computation task is partitioning into fields since potentially $O(2^{n\times n})$ possible fields may exist only fields which are present should be extracted.

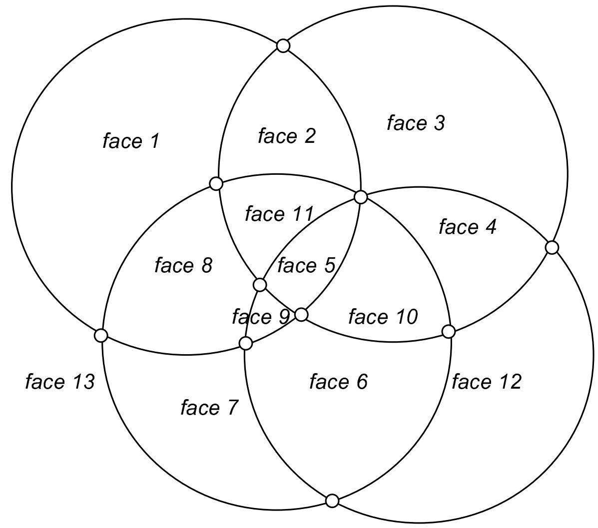

Fig. 2. The data structure for four sensors with disk monitored regions. The planar graph $P = (V,E)$ with vertices corresponding to points of intersections and edges connecting adjacent points. There are 10 vertices, 21 edges and 13 faces.

In this paper we present a superior data structure which overcomes the above disadvantages. The monitored region is represented by a planar graph $P = (V,E)$ with vertices V corresponding to the points of intersections of boundaries of all sensor's coverage regions (e.g., circles) and edges E connect pairs of adjacent intersection points along the boundaries of sensor circles (see Figure 2).

Theorem 1: Let m be the number of points of intersection of boundaries of the monitored regions of all sensors, then the number of faces in the graph P is at most $m+2$.

Proof: The proof is based on Euler's formula for planar graphs. Since $P = (V,E)$ is a planar graph, $|V|-|E|+|F| = 2$, where V, E and F are the sets of vertices, edges and faces of P, respectively. If several boundaries intersect at the same point, then by slightly changing boundaries, one can increase the number of faces (see Figure 3). Thus, for counting purposes, we can assume that each point can be intersection of boundaries of at most two monitored regions. Then each vertex of the graph P has degree 4. If we sum up degrees of all vertices, then we count each edge twice, therefore, $|E| = 2|V|$. Thus, $|V|-2|V|+|F| = 2$ and the number of faces equals $|F| = |V|+2$. ■

It is easy to see that the *faces* of the graph P, i.e., the parts of the plane bounded by edges, are regions covered by the same set of sensor disks. Therefore, if we would identify all the faces of the graph P we would effectively enumerate all fields and moreover, this will be a complete and accurate representation of *all* fields since we do not use grid points and do not rely on an assumption that we have sufficient amount of grid points to nail each face.
One can efficiently find all faces since the number of such faces is comparatively small. Indeed, the following simple fact bounds the number of faces.

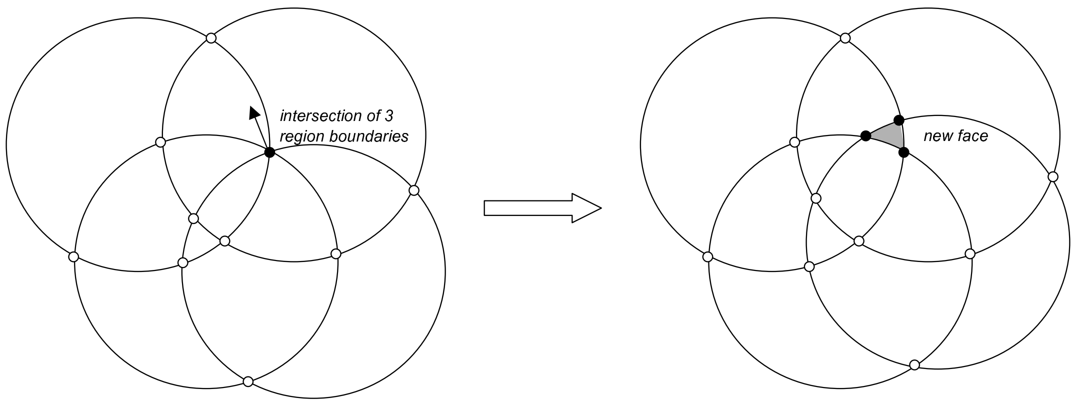

Fig. 3. The left example contains a triple intersection (the vertex is filled). On the right, there is a modified example with one circle slightly moved up. As a result a new dotted face appears.

If the monitored regions are convex, then any two boundaries can intersect at most twice, i.e., there are at most $n(n-1)$ intersection points.

Corollary 2: Given n sensors each with convex monitored region, the number of faces of the graph P is at most $n(n-1)+2$.

Below we describe an efficient implementation of our data structure when each monitored region is a disk (or in general a convex region) (see Figure 4). The graph P is considered to be directed with each edge represented by two opposite arcs and each arc belongs to the boundary of exactly one face. For each vertex all outgoing arcs are sorted in a counterclockwise order. The faces are identified by walking along the arcs, if we come to the node using arc e, then we should follow the arc next to the next arc which opposite to e. Finally, determining which sensors cover which faces is accomplished by measuring the distances of the face vertices to the sensors.

If monitored regions are not convex then we suggest to partition them into convex subregions and apply the planar graph data structure afterwards. This may happened in case of obstacles when collecting visual information (see Figure 5). If the monitored regions are given implicitly, then finding intersection points may be a challenging task and we may need to reuse the grid point representation. In this case, finding fields can be done efficiently using hash functions.

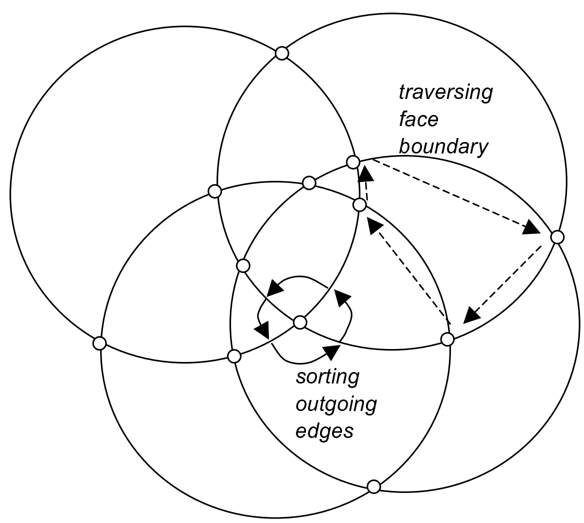

Fig. 4. Solid arcs show how the edges outgoing from the filled vertex are traversed, the dotted arcs show the order in which a face boundary is traversed.

III. Partial Sensor Coverage and the Set q-Cover Problem

As discussed above, for finding the next sensor cover while assembling a monitoring schedule, it is necessary to find the minimum *weighted* sensor cover, where weights $w(p_i)$ on sensors are induced by usage

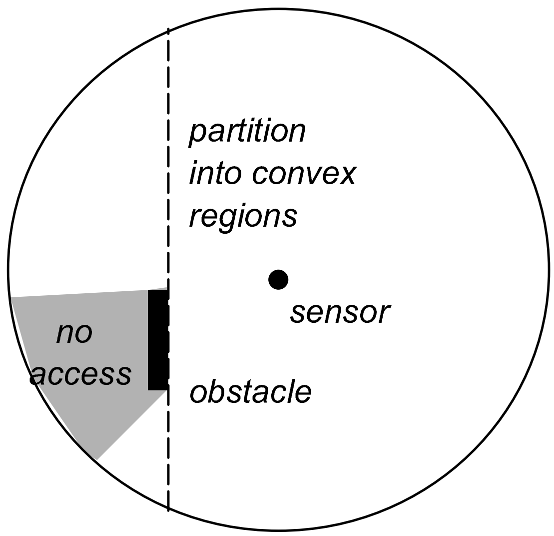

Fig. 5. An example of a sensor with nonconvex monitored region caused by a solid obstacle. The region is partitioned by a dashed line into the convex regions.

of these sensors in already accepted covers. In this section we consider the problem of finding the minimum weighted sensor set partially covering the monitored area.

Partial q-Coverage problem. Given a constant $q \in [0,1]$, the monitored region R with area M and a set of sensors *Sensors* find subset $p_1, p_2, \dots, p_t$ of *Sensors* such that

$$\sum w(p_i) \to \min$$

with constraint,

$$Area\left(\bigcup_{i=0}^{t} S_i\right) \geq qM,$$

where M is the total monitored area and S_i is the monitored region of sensor p_i.

Note that we have reduced in section II the sensor covering problem to the set cover problem where the elements are the faces and the sets are the sets of faces covered by a single sensor. Similarly, the Partial q-Coverage problem is equivalent to the weighted set cover problem, in which we need to cover q factor of the total cost of elements. Let consider the following equivalent abstract formulation.

Set q-Cover problem. Given a finite set X of elements each having cost[1] $cost : X \to R^+$ and family of $\mathcal{F}$ of subsets $S_i \subseteq X$, each having weight $w : S_i \to R^+$, and $q \in [0,1]$, find minimum weight subfamily $\mathcal{C} \subseteq \mathcal{F}$ covering subsets of X, such that total cost of elements covered by sets in $\mathcal{C}$ is at least $q \cdot cost(X)$, where $cost(X) = \sum_{x \in X} x$.

[1]The cost of an element corresponds to the area of a face.

Each subset also has cost, $cost : S_i \to R^+$ which is total cost of elements in X which are covered by set S_i after selecting sets $S_1, S_2, \dots, S_{i-1}$.

We consider the following greedy algorithm for Set q-Cover problem. Iteratively, we choose the set $S_i \in \mathcal{C}$, with minimum $\left(\frac{w(S_i)}{cost(S_i)}\right)$, where $cost(S_i) = \sum_{x \in S_i} cost(x)$. The greedy algorithm stops when the area covered equals or exceeds $q \cdot cost(X)$.

Theorem 3: The approximation ratio of the greedy algorithm for Set q-Cover problem is $1 + \ln \frac{1}{1-q}$

Proof: Let X_i be the set of elements of X which are uncovered after selecting set $S_1, S_2, \dots, S_i$, and cost of X_i, $cost : S_i \to R^+$ is total cost of elements of X_i. The $cost(X_0)$ is cost of elements of X before selecting any set S_i, so that $cost(X_0) = cost(X)$ and,

$$cost(X_i) = cost(X_{i-1}) - cost(S_i) \tag{1}$$

Let $S_1^*, S_2^*, \dots, S_k^*$ be sets selected in optimal solution, so that,

$$\begin{aligned} \frac{OPT}{cost(S_i)} &\geq \frac{\sum w(S_j^*)}{\sum cost(S_j^*)} \\ &\geq min\left(\frac{w(S_j)}{cost(S_j)}\right) \\ &\geq \frac{w(S_i)}{cost(S_i)} \end{aligned} \tag{2}$$

From (1),

$$cost(X_i) \leq cost(X_{i-1})\left(1 - \frac{w(S_i)}{OPT}\right)$$

Let m be the maximum number such that remaining cost $cost(X_m)$ is at least $(1-q)cost(X)$. So that,

$$cost(X_{m+1}) < (1-q)cost(X) \leq cost(X_m) \tag{3}$$

From inequality $\ln(1+x) \leq x$,

$$\begin{aligned} \ln\frac{cost(X_0)}{cost(X_m)} &\geq -\ln\prod\left(1 - \frac{w(S_i)}{OPT}\right) \\ &= -\sum \ln\left(1 - \frac{w(S_i)}{OPT}\right) \\ &\geq \frac{\sum w(S_i)}{OPT} \end{aligned} \tag{4}$$

The cost of solution by Greedy method is equal to

$\sum_{i=0}^{m+1} w(S_i)$, so using (3),(4) approximation ratio is,

$$\begin{aligned} APR &= \frac{\sum_{i=0}^{m} w(S_i) + (w(S_{m+1}))}{OPT} \\ &= \frac{\sum_{i=0}^{m} w(S_i)}{OPT} + \frac{w(S_{m+1})}{OPT} \\ &\leq \ln \frac{cost(X_0)}{cost(X_m)} + 1 \\ &\leq \ln \frac{1}{(1-q)} + 1 \end{aligned}$$

Because, from (2),

$$\frac{w(S_{m+1})}{OPT} \leq \frac{cost(S_{i+1})}{cost(X_{i+1})} \leq 1$$

■

IV. Sensor Network Life Problem

For the Sensor Network Lifetime problem is packing LP problem. After finding different set covers instances of sensors which satisfies sensor network constraint, we have to maximize the life by assigning time for every sensor covers. After obtaining family of sensor covers $SC = \{c_1, \ldots c_m\}$, we need to find the time variable t_j for each sensor cover c_j. We can create matrix C_{ij} with rows i=1, … ,m representing each sensor, and columns j=1, … ,m representing each cover.

$$Maximize : \sum_{j=1}^{m} t_j$$

$$Subject\ to \sum_{j=1}^{m} C_{ij} t_j \leq b_i$$

here b_i is lifetime of sensor i and

$$C_{ij} = \begin{cases} 0 & \text{if sensor i is not in set cover j} \\ 1 & \text{if sensor i is in disk cover j} \end{cases}$$

The linear program above is a packing LP. In general, a packing LP is defined as

$$\max\{c^T x | Ax \leq b, x \geq 0\} \qquad (5)$$

where A, b, and c have positive entries; we denote the dimensions of A as mxn. In our case the number of columns of A is prohibitively large (exponential in number of sensors) and we will use the $(1+\varepsilon)$-approximation Garg-Könemann algorithm [9]. The algorithm assumes that the LP is implicitly given by a vector $b \in R^m$ and an algorithm which finds the column of A minimizing so-called length. The *length* of column j with respect to LP in Equation (5) is defined as $length_y(j) = \frac{\Sigma_i A(i,j) y(i)}{c(j)}$ for any positive vector y.

We cannot directly apply the Garg-Könemann algorithm because, as we will see later, the problem of finding the minimum length column is NP-Hard in our case, and we can only approximate the minimum length column. Fortunately, it is not difficult to see that when the Garg-Könemann $(1+\varepsilon)$-approximation algorithm uses f-approximation minimum length columns it gives an $(1+\varepsilon)f$-approximation solution to the packing LP (5) [10][2]. The Garg-Könemann algorithm with f-approximate columns is presented in Figure 6.

Input: A vector $b \in R^m$, $\varepsilon > 0$, and an f-approximation algorithm F for the problem of finding the minimum length column $A_{q(y)}$ of a packing LP $\{\max\ c^T x | Ax \leq b, x \geq 0\}$
Output: A set of columns $\{A^j\}_{j=1}^k$ each supplied with the value of the corresponding variable x^j, such that $(x^1, \ldots, x^k)$ correspond to all non-zero variables in a near-optimal feasible solution of the packing LP $\{\max\ c^T x | Ax \leq b, x \geq 0\}$

---

(1) Initialize: $\delta = (1+\varepsilon)((1+\varepsilon)m)^{-1/\varepsilon}$, for $i = 1, \ldots, m$ $y(i) \leftarrow \frac{\delta}{b(i)}$, $D \leftarrow m\delta$, $j = 0$
(2) While $D < 1$
  Find the column A_q using the f-approximation F.
  Compute p, the index of the row with the minimum $\frac{b(i)}{A_q(i)}$
  $j \leftarrow j+1$, $x^j \leftarrow \frac{b(p)}{A_q(p)}$, $A^j \leftarrow A_q$
  For $i = 1, \ldots, m$, $y(i) \leftarrow y(i)\left(1 + \varepsilon \frac{b(p)}{A_q(p)} / \frac{b(i)}{A_q(i)}\right)$, $D \leftarrow b^T y$.
(3) Output $\{(A^j, \frac{x^j}{\log_{1+\varepsilon} \frac{1+\varepsilon}{\delta}})\}_{j=1}^k$

Fig. 6. The Garg-Könemann Algorithm with f-approximate minimum length columns

When applied to our LP (5), it is easy to see that the Garg-Könemann algorithm requires an (approximation) algorithm F solving the Minimum Weight Sensor Cover problem with weights proportional to the elements of vector y, i.e., for each node $i = 1, \ldots, m$, $w(i) = 1/y_i$. This implies the following general result.

Theorem 4: The Maximum Network Lifetime problem can be approximated within a factor of $(1+\varepsilon)f$, for any $\varepsilon > 0$, using Algorithm on Figure 6 with the algorithm F being the f-approximation algorithm for the Minimum Weight Sensor Cover problem.

Using the result from section II, we obtain the following

Corollary 5: The Maximum Network Lifetime problem can be approximated within a factor of $(1+\varepsilon)(1+2\ln n)$, for any $\varepsilon > 0$, using Algorithm on Figure 6 with the algorithm F being the greedy $(1+2\ln n)$-approximation algorithm for the Minimum Weight Sensor Cover problem.

Using the result from section III, we obtain the following

[2]Although this complexity aspect has not been published anywhere in literature, it involves only a trivial modification of [9] and will appear in its journal version [10].

Corollary 6: The Maximum Network Lifetime problem when q-portion of the monitored region R is required to cover can be approximated within a factor of $(1+\varepsilon)(1+\ln(1-q)^{-1})$, for any $\varepsilon > 0$, using Algorithm on Figure 6 with the algorithm F being the greedy $(1+\ln(1-q)^{-1})$-approximation algorithm for the Minimum Weight Sensor q-Cover problem.

V. Experimental Study

All algorithms were implemented in C++. The heuristics were compiled using `gpp` with `-O2` optimization, and run on a Sun workstation Ultra-60. The experiments were run on randomly generated testcases.
For experiments in table (I) and (II), we have taken sensor area $1000X1000$, monitored area as $800X800$ and each sensor node has 10 batteries.

In Table I, we have taken ε as 0.5 for Garg and Könemann algorithm. This table shows the trade-off between the lifetime of sensor network and the q-portion of the monitored region R which is required to cover. We should also mention (not in the table) that the runtime of the both Garg-Könemann and CPLEX are increasing when the portion q is decreasing since the lesser q the more freedom of sensor cover we can get.

In Table II, we have taken q as 0.9. After getting sensor covers from Garg and Könemann solution, we can find the optimal schedule by assigning the best times for each sensor cover by CPLEX, with constraint to satisfy battery requirement of Sensor Nodes and to maximize the total life time. Note that Garg-Könemann solution does not guarantee that there is a tight energy constraint, i.e., there is a sensor which completely exhausts its energy supply. The ”Real” solution is obtained from Garg-Könemann solution by finding the tightest energy constraint and making it tight by scaling up the timespans for each sensor cover. Note that CPLEX improves significantly over the Garg-Könemann (sometimes twice) and the ”Real” time is always between these two lifetimes.

References

[1] R. Hahn and H. Reichl, “Batteries and power supplies for wearable and ubiquitous computing,” in *Proc. 3rd Int. Symp. Wearable Computers*, 1999.

[2] S. Slijepcevic and M. Potkonjak, “Power efficient organization of wireless sensor networks,” in *Proc. IEEE International Conference on Communications (ICC)*, 2001, pp. 472–476.

[3] D. Tian and N. D. Georganas, “A coverage-preserving node scheduling scheme for large wireless sensor networks,” in *Proc. ACM WSNA02*, 2002.

[4] A. Sinha and A. Chandrakasan, “Dynamic power management in wireless networks,” *IEEE Design & Test of Computers*, pp. 62–74, 2001.

[5] D.-Z. D. M. Cardei, “Improving wireless sensor network lifetime through power aware organization,” *Submitted to ACM Wireless Networks.*

[6] G. Zussman and A. Segall, “Energy efficient routing in ad hoc disaster recovery networks,” in *IEEE INFOCOM’03*, 2003.

[7] F. Leighton and S.Rao, “Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms,” *Journal of the ACM*, vol. 6, pp. 787–832, 1999.

[8] R. J. M. II, A. K. Das, M. El-Sharkawi, P. Arabshahi, and A. Gray, “Maximizing lifetime in energy constrained wireless sens array using team optimization of cooperating systems,” in *Proc. IEEE World Conference on Computational Inteligence 2002*, 2002.

[9] N. Garg and J. Könemann, “Faster and simpler algorithms for multicommodity flow and other fractional packing problems,” in *Proceedings of FOCS*, 1997.

[10] J. Konneman, “Personal communication.”

No. of Sensors	Range	q	Lifetime	
			GK	CPLEX
100	100	1.00	7.53	14.00
100	100	0.90	10.26	15.00
100	100	0.80	11.02	15.00
100	100	0.50	25.49	45.00
100	200	1.00	28.32	52.00
100	200	0.90	53.81	90.00
100	200	0.80	58.67	107.00
100	200	0.50	105.60	212.00
100	300	1.00	53.00	109.00
100	300	0.90	105.20	223.00
100	300	0.80	127.45	272.00
100	300	0.50	203.92	418.00
500	100	1.00	33.07	49.00
1000	100	0.90	148.86	283.00

TABLE I

LIFETIME OF SENSOR NETWORK FOR DIFFERENT COVERAGE OF MONITORED AREA, RANGE IS RADIUS OF DISC WHICH IS COVERED BY SENSOR NODE, Q IS SHOWING HOW MUCH PERCENT OF MONITORED AREA IS COVERED, GK LIFETIME IS TIME GIVEN BY GARG AND KÖNEMANN, CPLEX LIFETIME IS TIME GIVEN BY CPLEX FOR DISC COVER INSTANCES GIVEN BY GARG AND KÖNEMANN ALGORITHM

No. of Sensors	Range	ε	GK		Real		CPLEX	
			Life time	Runtime (Sec)	Life time	Runtime (Sec)	Life time	Runtime (Sec)
100	100	0.50	15.78	1.24	22.94	0.07	30.00	0.02
100	100	0.10	27.15	2.69	28.92	2.44	30.00	0.12
100	100	0.05	28.54	7.32	29.41	9.95	30.00	0.56
100	200	0.50	45.32	6.63	70.00	0.21	83.00	0.06
100	200	0.10	73.75	25.19	78.72	6.63	92.00	0.89
100	200	0.05	36.67	33.91	37.73	12.97	43.20	0.41
100	300	0.50	108.03	14.85	178.00	0.48	223.00	0.09
200	100	0.50	28.08	9.75	46.47	0.28	51.00	0.11
200	100	0.10	48.93	22.45	53.05	10.05	56.00	6.90
200	200	0.50	101.30	43.10	167.65	1.03	170.00	0.04
200	200	0.10	156.99	262.91	169.87	32.19	170.00	0.50
200	300	0.50	222.50	141.93	391.25	2.26	409.00	0.13
500	100	0.50	73.50	152.93	126.32	2.28	129.00	0.62
1000	100	0.50	148.86	1179.63	255.71	9.62	283.00	33.43

TABLE II

COMPARISON OF LIFETIME AND RUNTIME OF GK, REAL, LOOP AND CPLEX FOR DIFFERENT SENSOR NODES AND RANGES