A Distributed Algorithm for Approximate Mobile Sensor Coverage

Esther Ezra* Jiemin Zeng† Jie Gao†

Abstract

We consider the problem of covering a domain by mobile sensors and the design of an efficient schedule that reduces unnecessary sensor overlap and energy consumption. The problem is motivated by emerging participatory sensing applications as well as other coverage problems involving mobile nodes. The problem of minimizing the total energy consumption while maintaining the same level of coverage guarantee is NP-hard. We develop distributed algorithms achieving a constant approximation factor when sensors have unit disk sensing ranges, and a $(1+\varepsilon)$ approximation factor when the sensors also have constant bounded density. For all these algorithms the communication cost is asymptotically bounded by the cost of simply maintaining the direct neighborhood of each sensor. The constant approximation distributed algorithm can be generalized for the k-coverage problem, when each point of interest has to be covered by at least k sensors.

1 Introduction

Motivation. The past couple of years have witnessed the wide adoption of smartphones. Although smartphones are mainly designed for user communication and social interactions, they also provide an opportunistic platform for communities to sense and contribute sensory information to form a body of knowledge. This has been termed as *participatory sensing* [3] or *crowdsourcing* applications. A recent study using anonymous cell phone records discovered that human trajectories show a high degree of temporal and spatial regularity [13]. The study also confirms, not surprisingly, that individuals return to a few highly frequented locations, such as home or work and that there is a lot of overlaps in different individuals' trajectories. Therefore it is expected that there will be ample redundant sensors populating certain 'hot spots' and there is an opportunity to selectively turn on such sensors to reserve battery power.

Problem definition. In this paper we study how to efficiently schedule mobile sensors for monitoring the environment. We represent a sensor by a point (which we also refer to as "node") that is moving around in a pre-specified region Σ (e.g., a polygon) in the plane. We assume two different types of sensing requirements. In the *continuous coverage* model, the entire domain Σ needs to be covered at all time. In the *discrete coverage* model, we assume that a discrete set $\mathcal{L}$ of landmarks inside Σ need to be covered at all time.

Each sensor has a fixed sensing range, and thus we assume its monitoring region is a unit disk (which is a common assumption in the literature). Since the disks may overlap, we consider scheduling sensors to minimize the total energy consumption. Assume that sensor i, kept on, will consume energy at a rate of w_i and it is kept on for a total of t_i time. Then the total energy usage $\sum_i w_i t_i$ is the weighted sum of the time that sensors are turned on. When the weights are uniform (i.e., all nodes consume power at the same rate), the objective becomes the total time that all sensors are kept on.

The research question at hand is to design a clever schedule of the sensors such that the total energy usage is minimized and the coverage quality does not degrade – i.e., what could be covered by the original collection of sensors at any snapshot is still covered. Notice that this problem is related to the problem of removing redundant sensors for static sensor coverage. In our setting the mobility of the sensors makes this problem more challenging – one main difference is the addition of the temporal dimension in the sense that we may selectively turn on and off any sensor to reserve energy.

Our results. We start by introducing a framework to analyze the complexity of mobile coverage. Although the nodes move continuously, their coverage only changes at some discrete points of time, which we call "critical events". To enable rigorous theoretical analysis on the number of such critical events, we make a number of additional natural assumptions. We take Σ to be the unit square and assume that the critical events are tracked via a standard HELLO beacon broadcast. Also, without loss of generality, we assume that all the sensors, at any snapshot, collectively cover the entire domain Σ or the entire set of landmarks[1].

We then describe our algorithms for finding approximate mobile coverage solution. We first present a static distributed algorithm. This algorithm achieves a constant approximation factor, where the main observation

*Courant Institute, New York University, esther@courant.nyu.edu

†Dept. of Computer Science, Stony Brook University, jiezeng, jgao@cs.stonybrook.edu

[1] We describe our assumptions and model in more detail in Section 3, however, we give a brief description here as well in order to report our results.

is to use a grid partitioning of the input sensors into squares of side length $r/2$, where r is the sensing range (i.e., the radius of the disks). For all the sensors whose range intersect the same square, we calculate a constant approximate solution for the local version of a geometric set cover problem. The solution is taken as the union of all the local solutions. We show that this results in a constant approximation factor of the global optimum.

Next, we show that the above algorithm for the static case can be extended to the mobile case such that we only need to update the solution for a local problem whenever a critical event occurs. This is achieved due to our assumption that the critical events are tracked via HELLO beacon broadcast.

We also describe the distributed implementation of a polynomial-time approximation scheme (PTAS) for the case where each sensor has only a constant number of neighbors in its communication range. This is by adapting an existing centralized algorithm (the so-called "shifting" technique [15]) with the critical insight that the core structure is similar to our grid based solution—see below.

We also show that the distributed algorithm can be generalized (with only uniform weights though) to handle *k-coverage* with a constant approximation factor, that is, each point of interest is covered by at least k sensors [25, 18]. The k-coverage problem is a natural extension to the standard coverage problem by improving the robustness and accuracy of the coverage solution.

2 Related Work

Sensor coverage. The problem of *sensor coverage* in the static setting has been extensively studied. We only have space to review the most relevant results to our work, and refer the reader to the survey paper [22] for more details. In particular, our work belongs to the category of using the boolean disk coverage model for area coverage (covering the entire region) or point coverage (covering discrete targets).

For area coverage, one of the most well studied problems is to determine the optimal placement pattern for the infinite plane with minimum sensor density, when sensors can be placed anywhere in the plane. It is shown by Kershner that the triangle lattice pattern is the optimal pattern if the sensors have unit disk sensing regions [17]. Bai *et al.* [1] considered both optimal density and network connectivity. They provided a strip-based placement pattern and proved its asymptotic optimality for achieving both complete coverage and 1-connectivity.

Both the area coverage and point coverage problems can be generalized to k-coverage, where each target or each point in the monitored area must be covered by at least k sensors [25, 18].

Our work is also closely related to the problem of sensor activity scheduling, which is to schedule nodes to be activated alternatively such that the network operation time can be prolonged and area coverage requirement is still met. Various optimization objectives have been used for this problem, ranging from ensuring the area coverage ratio, minimizing the number of active sensors, and prolonging network lifetime [20, 16, 24, 8]. A number of local tests have been developed to check for redundant nodes and put them to sleep while still maintaining full coverage(see, e.g., [8, 24]. The list of reference is too long to survey here and we refer the readers to the survey papers [19, 23].

Despite these previous studies, we are not aware of any previous work which considered mobile sensors, and thus our work is seminal in this topic.

Geometric set cover. The *geometric set-cover* problem is an abstractization of the sensor coverage problem. In a typical such setting, we are given a set $X \subseteq \Re^d$ of points (which is either discrete or continuous) and a set $\mathcal{R}$ of simply-shaped regions in $\Re^d$ (e.g., halfspaces, balls, simplices, cylinders, etc.). The goal is to compute a smallest set of regions that altogether cover X.

When $X = \Re^d$, or when X is a continuous subset in $\Re^d$, we refer to this setup of the problem as the *continuous model*, and when X is a finite point set, this setup of the problem is referred to as the *discrete model*.

The geometric set-cover problem is NP-Hard, under both discrete and continuous models, even for very simple geometric settings in $\mathbb{R}^2$, e.g., when $\mathcal{R}$ is a set of unit disks or unit squares [12, 14]. Therefore attention has mostly focused on developing polynomial time approximation algorithms. The well-known greedy algorithm, which always selects the set that maximizes the residual coverage, yields a $O(\log n)$-approximation in polynomial time for the smallest set-cover [21, Chapter 2], and the known lower-bound results suggest that this is the (asymptotically) best approximation factor one can hope to achieve in polynomial time for arbitrary (that is, not necessarily geometric) settings [11]. However, by exploiting the underlying geometry, one can obtain polynomial-time algorithms with better approximation factors for various geometric set-cover problems. These algorithms employ and adapt a wide range of novel techniques, including variants of the greedy algorithm, dynamic programming, LP-relaxation, and "ε-nets". It is beyond the scope of this paper to give a comprehensive review of the known results. We only mention a few results that are directly relevant to our problem, and refer the reader to [2, 4, 10, 15] and the references therein for further details. Specifically, Clarkson and Varadarajan [7] show a constant approximation factor (achievable in expected polynomial time) for arbitrary disks with uniform weights. A recent result by Chan *et al.* [5] is a constant approximation factor (achievable in expected polynomial time) for the *weighted* set-cover problem of points and arbitrary disks in the plane.

3 Preliminaries

One of the main challenges in analyzing mobile coverage problems is to properly factor the continuity of the motion. We next describe how to model the mobile coverage problem in the time-space domain and how to track *critical events*.

The model. We consider a domain Σ to be monitored by mobile sensors; Σ is typically of a simple shape, say, a square or a disk. The sensors are represented by a set $\mathcal{D} = \{D_1, \ldots, D_n\}$ of n unit disks in the plane, where each disk D_i is assigned a weight w_i representing its energy consumption at all times, for $i = 1, \ldots, n$. In what follows, the weights are assumed to be arbitrary, unless stated otherwise.

For each sensor, we denote its sensing range by r (i.e., the radius of the disk in our model), and assume that it has a communication range $3r$. In this case, full coverage of Σ implies network connectivity. We note that in practice the sensing ranges are much smaller than the diameter ρ of the domain Σ, we thus assume $r \ll \rho$ (we also note that $\rho \leq r$ implies that Σ can be covered by a small number of disks, which is a scenario of less interest in theory). Applying a common assumption in mobile networks, we assume that the sensors periodically send Hello beacons to detect and maintain direct communicating neighbors.

Critical Events. Each sensor moves along an arbitrary trajectory. We denote the position c_i of the center of D_i at time t, for $i = 1, \ldots, n$, by $c_i := c_i(t) = (X_i(t), Y_i(t))$, and $D_i := D_i(t)$ denotes D_i at time t. Let $\mathcal{U}(t) := \bigcup_{i=1}^{n} D_i(t)$ be the *union* of the disks at time t (that is, all regions in the plane that are covered by at least one disk at time t). Without loss of generality, we assume that time ranges from 0 to 1 in the mobile coverage problem.

Let us initialize the process with a subset $\mathcal{S}$ of the disks in $\mathcal{D}$ whose union covers the domain Σ at time $t = 0$. Since the union $\mathcal{U}(t)$ of all disks covers the domain Σ at all times $0 \leq t \leq 1$, such a subset cover exists, and let us denote its union in time $t \geq 0$ by $\mathcal{U}'(t) := \bigcup_{D \in \mathcal{S}} D(t)$. We note that the topological structure of the cover $\mathcal{U}'(t)$ changes only at some *critical events*. Each of these events is characterized by a triple of disks that meet at a point p. Such an event may correspond to the appearance of a new connected component in the complement of the union of $\mathcal{U}'(t)$ (that is, a new "hole" in the union appears) or to the disappearance of an existing component (that is, an existing hole is sealed). In the first case we need to add a new disk to $\mathcal{S}$ to restore coverage, and in the latter case we may remove an existing disk from $\mathcal{S}$, to save energy. See Figure 1 for an illustration.

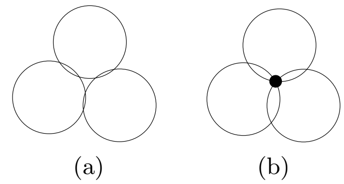

Figure 1. (a) A triple of disks with an empty intersection, just before they meet, and (b) their position at the moment they meet in a common point (depicted by the black bullet).

Therefore, the optimal solution to the kinetic sensor coverage problem will only change at a critical event, and our assumption on the Hello beacon broadcast implies that each sensor can locally detect the critical event when its sensing region just starts or stops overlapping with another one's sensing region. Similarly, three sensors can all detect locally when their sensing regions just start or stop having a common intersection.

4 Distributed Approximation Algorithms

4.1 The General Framework

In what follows, we first consider the discrete coverage requirement when a set of landmarks to be covered is given, and is denoted by L. We assume, without loss of generality, that they are covered by the sensors (otherwise, an uncovered point in L can be removed from further consideration). Later on, we show how to choose L carefully so that it guarantees a full coverage in the continuous coverage model.

The static problem. Our static algorithm is a variant of the set-cover algorithm of Hochbaum and Maass [15], although they refer to the scenario where the disk set is unrestricted (that is, this set is continuous and thus one can choose any arbitrary disk to participate in the coverage), whereas in our scenario this set is *given*.

We place a grid Γ over Σ of side length $r/2$. We say that a cell is *empty* if it does not contain a point of L, otherwise, it is *non-empty*. Our goal is to locally construct a small cover in each non-empty cell of Γ, and then combine all these covers to form the global coverage. Due to our special structure (that is, all disks have radius r) and the existence of a machinery to handle each cell locally, we will be able to show that our coverage is not any larger than the optimal coverage up to a constant factor.

It is easy to verify that each disk $D \in \mathcal{D}$ must fully contain at least one cell of Γ, moreover, it meets at most 25 cells of Γ. See Figure 2 for an illustration. Let us now consider a fixed non-empty cell $\tau \in \Gamma$, with a landmark point

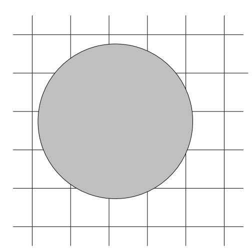

Figure 2. A unit disk placed on a grid of side length 1/2.

$p \in L$ inside. By the assumption that all points in L are covered by $\mathcal{D}$, p must have a disk $D \in \mathcal{D}$ with $p \in D$. In particular, D must intersect τ (it either overlaps τ or fully contains it). We now collect all points $p \in L$ that lie inside τ and the disks intersecting τ; let L_τ, $\mathcal{D}_\tau$ be these resulting subsets. We next construct a set cover for L_τ with the disks in $\mathcal{D}_\tau$ using the algorithm of Chan *et al.* [4]. As noted in Section 2 this algorithm produces a set cover whose size is only within a constant factor of the smallest set cover. Let $\mathcal{S}_\tau \subseteq \mathcal{D}_\tau$ be the set cover just produced. We then report the set

$$\mathcal{S} := \bigcup_{\tau \in \Gamma, \tau \cap L \neq \emptyset} \mathcal{S}_\tau .$$

We next show:

Lemma 1. *Let* OPT *be the smallest set cover for* L *and* $\mathcal{D}$. *Then* $|\mathcal{S}| = O(|\mathrm{OPT}|)$.

Proof. Let $\tau \in \Gamma$ be a fixed non-empty cell. We next observe that the optimal set cover OPT^*_τ for L_τ, $\mathcal{D}_\tau$ is at least as good as the entire optimal set cover OPT restricted to τ (as OPT^*_τ is the best coverage for L_τ, $\mathcal{D}_\tau$ whatsoever). That is, $|\mathrm{OPT}^*_\tau| \le |\mathrm{OPT}_\tau|$, where OPT_τ is the subset of disks in OPT meeting τ.

By the properties of the approximation algorithm of Chan *et al.* [5] it follows that $|\mathcal{S}_\tau| \le C \cdot |\mathrm{OPT}^*_\tau|$, where $C > 0$ is an absolute constant. On the other hand, we claim that $\sum_{\tau \in \Gamma, \tau \cap L \neq \emptyset} |\mathrm{OPT}_\tau| \le 25|\mathrm{OPT}|$, since, as observed above, each disk of D can meet at most 25 cells of the grid, and thus the multiplicity of a fixed disk in the optimal solution OPT cannot exceed this constant. Combining the above inequalities we obtain:

$$|\mathcal{S}| \le \sum_{\tau \in \Gamma, \tau \cap L \neq \emptyset} |\mathcal{S}_\tau| \le C \sum_{\tau \in \Gamma, \tau \cap L \neq \emptyset} |\mathrm{OPT}^*_\tau| \le$$

$$C \sum_{\tau \in \Gamma, \tau \cap L \neq \emptyset} |\mathrm{OPT}_\tau| \le 25C|\mathrm{OPT}|,$$

from which we conclude $|\mathcal{S}| = O(|\mathrm{OPT}|)$, as asserted. $\square$

In order to make the algorithm distributed, we observe that since the communication range of each sensor is $3r$, any two disks that meet the same cell τ can directly communicate with each other. Indeed, all disks (of radius r) that meet (a non-empty cell) τ must have their centers within distance at most r from τ. We now observe that the centers c, c' of two distinct such disks D, D' are located within distance at most $2r + r/\sqrt{2} < 3r$ from each other. This can be verified by placing c, c' within distance r from two opposite corners of τ and the fact that the distance between these two corners is $r/\sqrt{2}$. Having this property at hand, the communication graph refined to the corresponding nodes (whose disks meet τ) forms a clique, and hence all disks meeting τ can "nominate" (say, by selecting the one with smallest ID) one such disk D^* to hold all information about the local setting to our set cover problem, that is, L_τ, $\mathcal{D}_\tau$, and then the computation for the approximated set cover (as described above) can be done within D^*.

The kinetic problem. At the initial time t_0 we apply the static algorithm presented above in order to find an approximation for the set cover. Then for each node (that is, a sensor), we keep track of the state of all possible triples with that node, that is, whether they have an empty or non-empty intersection (recall that our model supports that). Then, when a critical event is detected (and so we also have the triple it involves), the local solution within the grid cell τ on which the critical event occurred will be re-computed using the algorithm for the static case (*only within* τ). We compute τ by locating in the grid Γ the cell containing the common intersection of the disks at the corresponding critical event.

Running time and communication cost. For the static problem, computing the approximate set-cover locally within each cell τ is done in expected polynomial time [5]. We recall that each disk D communicates only with other disks whose centers lie in its communication range, in other words, each sensor exchanges messages only with its *neighbors* in the communication network. Recall that we do so independently in each cell $\tau \in \Gamma$ that D meets, and that there are at most 25 such cells, and thus the overall number of exchanged messages that D involves is proportional to its degree in the communication network. Thus the total number of exchanged messages is proportional to the sum of these degrees (over all nodes representing the disks D), which is just $O(|E|)$, where E is the set of links in the network. This bound is no more than the number of messages needed for each node to discover its neighborhood (up to a constant factor), which is a necessary routine for almost all networks.

Concerning the kinetic version, in each round imposed by the periodic beacons, each disk D exchanges messages in its neighborhood as described above for the static problem. Thus the total number of messages exchanged is $O(T|E|)$, where T is the number of rounds. Here too, we piggyback the communication cost on the cost of maintaining neighborhoods, and thus the algorithm does not incur extra cost. We have thus shown:

Theorem 2. *Given a network of mobile sensors with an* HELLO *beacon model as above, where each of the sensors has a unit sensing radius* r *and a communication radius* $3r$, *and a set of landmarks* L *(confined to a unit-square domain* Σ*) to be monitored, one can compute a coverage for* L *at each time* t, *whose overall weight is at most a constant factor from the optimum. The* *overall number of messages exchanged from each sensor is proportional to its neighborhood size at any time t.*

4.2 Sparse networks.

A natural case that we study is where the communication degree in the network is constant for each node. This scenario arises in the context of participatory sensing, as one cannot condense many participants (e.g., people who carry smartphones) at the same location. We first note that when applying our previous algorithm in this case, the total number of messages exchanged from and to each node is only $O(1)$ (at each round), and the overall number of messages exchanged in the network is only linear in the number of nodes (at each round). Nevertheless, we show below that for this setting, one can achieve a better approximation factor.

We apply the polynomial approximation scheme (PTAS) of Hochbam and Mass [15], based on a *shifting technique*. A crucial property in their algorithm is the fact that each point in the plane is covered by at most $O(1)$ disks, which corresponds to our scenario, as for each disk there are only $O(1)$ other disks that it meets. Thus, in particular, a "deep" point (a point that is covered by arbitrarily many disks) implies that there is a large clique in the network, and thus the communication degree (of the nodes in the clique) must be large as well, which contradicts our assumption. Thus all points are "shallow" (i.e., have constant depth).

In this algorithm we fix a small error parameter $\varepsilon > 0$, and then set the side length of the grid Γ to be $\Delta := O(r/\varepsilon)$ (recall that r is the sensing radius of the disks). A key observation of the algorithm in [15] is the fact that due to the constant-depth property, within each cell τ of the grid the *optimal* solution can be computed in polynomial time (where the degree of the polynomial depends on ε). The algorithm in [15] then shows that if we place the origin of this grid at a *random* point in $[0, \Delta]^2$, find the optimal set-cover OPT^*_τ locally in each cell τ, and then collect all these local solutions to form the actual set-cover $\mathcal{S}$, we obtain that $|\mathcal{S}|$ is at most $(1 + \varepsilon)|\mathrm{OPT}|$ (on expectation), where OPT is the optimal solution for the whole setting. That is, the main difference between this algorithm and the previous one given for an arbitrary communication network, is the fact that here (i) we randomly shift the grid, and (ii) we compute the actual optimal solution within each grid cell rather than an approximate solution. These two facts eventually leads to the PTAS in [15], which is an improvement over the constant approximation factor we obtained earlier.

Here too we can make the algorithm distributed using a similar approach as in the general framework, however, we observe that the communication graph confined to the nodes, whose corresponding disks meet a fixed cell τ, is not necessarily connected. Nevertheless, we claim that the coverages constructed w.r.t. each connected component must be pairwise disjoint, and then an optimal coverage is just the disjoint union of the respective optimal coverages computed for each connected component. Indeed, two sensors in different connected components must lie within distance at least $3r$ from each other, and thus if we draw a disk of radius r centered in each node of these components, the two resulting structures (each of which is a union of disks that contain an optimal coverage) remain disjoint. Thus in each such connected component we can nominate a disk to compute the optimal set-cover.

In the kinetic problem, we track the critical events and make the local computation in the grid at each such event, this is done almost verbatim as in the general framework.

Concerning the communication cost, in this solution we need to expand the neighborhood of each node by $O(1/\varepsilon^2)$. Specifically, the fact that each node has only $O(1)$ neighbors in the network, and, on the other hand, each grid cell τ is now of side length $O(r/\varepsilon)$, implies that τ may be intersected by only $O(1/\varepsilon^2)$ disks (we omit the easy computation, which follows from a straightforward packing argument), and thus each disk may exchange messages with that many disks in τ. On the other hand, consider the connected components formed by the disks covering the same grid cell τ. Each connected component has a diameter of at most $O(1/\varepsilon)$. Thus each node searching in its $O(1/\varepsilon)$-hop neighborhood can find all potential nodes in its connected component. Since each disk appears in only $O(1)$ cells (assuming ε is sufficiently small, as otherwise we can resort to the previous algorithm), the total number of exchanged messages involving D is $O(1/\varepsilon^2)$, and this bound is $O(n/\varepsilon^2)$ over all nodes. We have thus shown:

Theorem 3. *Given a network of mobile sensors of constant communication degree and a set of landmarks L as in Theorem 2, one can compute a coverage of L at each time t, whose weight is within a factor of $(1 + \varepsilon)$ from the optimum, for any $\varepsilon > 0$. The overall number of messages exchanged from each sensor is $O(1/\varepsilon^2)$ at any time t.*

4.3 The continuous coverage problem

We now aim to cover the entire domain Σ at all times. Consider that one can produce an appropriate point set P by constructing the planar subdivision induced by the disks (also referred to as the "arrangement of the disks"), with the property that P is covered if and only if Σ is covered; such constructions are standard in the theory of computational geometry [9]. This computation involves *in each round* the construction of the planar subdivision induced by the disks.

Since Σ is covered at all times, the underlying communication graph is connected. Moreover, this graph remains connected in any refinement to a subset of nodes, whose corresponding disks meet a fixed cell τ. Thus all disks meeting a cell τ can nominate one disk D^* to locally compute the subdivision within τ. Then the entire subdivision is obtained by gluing together all these "pieces". Having this subdivision at hand, the computation of P is fairly standard (see, e.g., [9]). At time t_0 we compute P with respect to the initial disk locations, and then apply the static algorithm with this set of landmarks. For each critical event, before we update our set cover, we only re-compute the portion of P that intersects with the cell we are updating (this is sufficient as the combinatorial structure of the coverage does not change elsewhere).

Using similar arguments as above, the communication overhead, at any time t, is proportional to the sum of the neighborhood sizes over all nodes.

4.4 The k-Coverage Problem

The *k-coverage problem* is a special case of the so-called *set multi-cover* problem, and in our scenarios this implies that we require each point to be covered by at least k disks, or, more specifically, each point should be monitored by at least k sensors, where $k > 0$ is an integer parameter (and thus the case $k = 1$ is just the standard set cover problem).

In order to solve this problem efficiently, we use a similar grid construction as the one for the original problem, and instead of applying the set-cover algorithm of Chan *et al.* [5], we apply the set multi-cover algorithm of Chekuri *et al.* [6]. This algorithm can be applied only when the weights are uniform, and computes in expected polynomial time a k-coverage whose size is only within a constant factor from the optimum. Plugging this into our machinery, we obtain a similar bound as in Lemma 1.

Theorem 4. *Given the setting of Theorem 2 (while assuming uniform weights), and an integer parameter $k > 0$, one can compute a k-coverage for L at each time t, whose size is at most a constant factor from the optimum. The overall number of messages exchanged from each sensor, at any time t, is proportional to its neighborhood size.*

5 Concluding Remarks

In this paper we studied the problem of efficiently constructing coverage by mobile sensors in order to reduce energy consumption, we are not aware of any previous works on this topic. Our approximation algorithms are based on the shifting technique technique of Hochbaum and Maass [15], nevertheless, this machinery has never been subject before to a distributed computing setting, which, as our work shows, is an important and useful machinery for both theory and practice. We hope that this observation will be useful for existing and future participatory sensing applications.

Acknowledgments. Esther Ezra would like to acknowledge the support of NSF under grant CCF-12-16689. Jiemin Zeng and Jie Gao would like to acknowledge the support of NSF through CNS-1116881, CNS-1217823 and DMS-1221339.

References

[1] X. Bai, S. Kumar, D. Xuan, Z. Yun, and T. H. Lai. Deploying wireless sensors to achieve both coverage and connectivity. In *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, pages 131–142, 2006.

[2] H. Brönnimann and M. T. Goodrich. Almost optimal set covers in finite VC dimensions. *Discrete Comput. Geom*, 14:463–479, 1995.

[3] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and N. Srivanstava. Participatory sensing. In *Proceedings of the World Sensor Web Workshop, ACM SENSYS*, 2006.

[4] T. M. Chan. Polynomial-time approximation schemes for packing and piercing fat objects. *J. Algorithms*, 46:178–189, 2003.

[5] T. M. Chan, E. Grant, J. Könemann, and M. Sharpe. Weighted capacitated, priority, and geometric set cover via improved quasi-uniform sampling. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 1576–1585, 2012.

[6] C. Chekuri, K. L. Clarkson, and S. Har-Peled. On the multi-cover problem in geometric settings. In *Proc. 25th Annu. Sympos. Comput. Geom*, pages 341–350, 2009.

[7] K. L. Clarkson and K. Varadarajan. Improved approximation algorithms for geometric set cover. *Discrete Comput. Geom*, 37:43–58, 2007.

[8] B. Cărbunar, A. Grama, J. Vitek, and O. Cărbunar. Redundancy and coverage detection in sensor networks. *ACM Trans. Sen. Netw*, 2(1):94–128, February 2006.

[9] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin, 3rd edition, 2008.

[10] G. Even, D. Rawitz, and S. Shahar. Hitting sets when the VC-dimension is small. *Inform. Process. Letts*, 95:358–362, 2005.

[11] U. Feige. A threshold of $\ln n$ for approximating set cover. *J. ACM*, 45:634–652, 1998.

[12] R. Fowler, M. Paterson, and S. Tanimoto. Optimal packing and covering in the plane are NP-complete. *Inform. Process. Letts*, 12:133–137, 1981.

[13] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, June 2008.

[14] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2. of Algorithms and Combinatorics, Springer, 1988.

[15] D. S. Hochbaum and W. Maass. Approimation schemes for covering and packing problems in image processing and VLSI. *J. ACM*, 32:130–136, 1985.

[16] C.-F. Huang, Y.-C. Tseng, and H.-L. Wu. Distributed protocols for ensuring both coverage and connectivity of a wireless sensor network. *ACM Trans. Sen. Netw*, 3(1), March 2007.

[17] R. Kershner. The number of circles covering a set. *American Journal of Mathematics*, 61(3):665–671, 1939.

[18] S. Kumar, T. H. Lai, and J. Balogh. On k-coverage in a mostly sleeping sensor network. In *Proceedings of the 10th annual international conference on Mobile computing and networking*, MobiCom '04, pages 144–158, 2004.

[19] I. Stojmenovic and J. Wu. Broadcasting and activity scheduling in ad hoc networks. *Mobile Ad Hoc Networking*, pages 205–229, 2004.

[20] D. Tian and N. D. Georganas. A node scheduling scheme for energy conservation in large wireless sensor networks. 3(2):271–290, 2003.

[21] V. V. Vazirani. *Approximation Algorithms*. Springer-Verlag, New York, 2001.

[22] B. Wang. Coverage problems in sensor networks: A survey. *ACM Comput. Surv*, 43(4):32:1–32:53, October 2011.

[23] B. Wang, K. C. Chua, V. Srinivasan, and W. Wang. Scheduling sensor activity for point information coverage in wireless sensor networks. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2006 4th International Symposium on*, pages 1–8, April 2006.

[24] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration for energy conservation in sensor networks. *ACM Trans. Sen. Netw.*, 1(1):36–72, Aug. 2005.

[25] Z. Zhou, S. Das, and H. Gupta. Connected k-coverage problem in sensor networks. In *In Proceedings of the International Conference on Computer Communications and Networks (IC3N)*, pages 373–378, 2004.