Robust Sensor Placements at Informative and Communication-efficient Locations

Andreas Krause† Carlos Guestrin* Anupam Gupta*
Jon Kleinberg#

August 2010
CMU-ML-10-108

Robust Sensor Placements at Informative and Communication-efficient Locations

Andreas Krause[†] **Carlos Guestrin**[*] **Anupam Gupta**[*]
Jon Kleinberg[#]

August 2010
CMU-ML-10-108

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

When monitoring spatial phenomena with wireless sensor networks, selecting the best sensor placements is a fundamental task. Not only should the sensors be informative, but they should also be able to communicate efficiently. In this paper, we present a data-driven approach that addresses the three central aspects of this problem: measuring the predictive quality of a set of hypothetical sensor locations, predicting the communication cost involved with these placements, and designing an algorithm with provable quality guarantees that optimizes the NP-hard tradeoff. Specifically, we use data from a pilot deployment to build non-parametric probabilistic models called *Gaussian Processes* (GPs) both for the spatial phenomena of interest and for the spatial variability of link qualities, which allows us to estimate predictive power and communication cost of unsensed locations. Surprisingly, uncertainty in the representation of link qualities plays an important role in estimating communication costs. Using these models, we present a novel, polynomial-time, data-driven algorithm, pSPIEL, which selects Sensor Placements at Informative and communication-Efficient Locations. Our approach exploit two important properties of this problem: *submodularity*, formalizing the intuition that adding a node to a small deployment can help more than adding it to a large deployment; and *locality*, under which nodes that are far from each other provide *almost* independent information. Exploiting these properties, we prove strong approximation guarantees for our approach. We also show how our placements can be made robust against changes in the environment, and how pSPIEL can be used to plan informative paths for exploration using mobile robots. We provide extensive experimental validation of this practical approach on several real-world placement problems, and built a complete system implementation on 46 Tmote Sky motes, demonstrating significant advantages over existing methods.

[†] Computing and Mathematical Sciences Department, California Institute of Technology, Pasadena, CA, USA
[*] School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA
[#] Department of Computer Science, Cornell University, Ithaca, NY, USA.

Keywords: Sensor networks, communication cost, link quality, spatial monitoring, sensor placement, approximation algorithms, Gaussian Processes

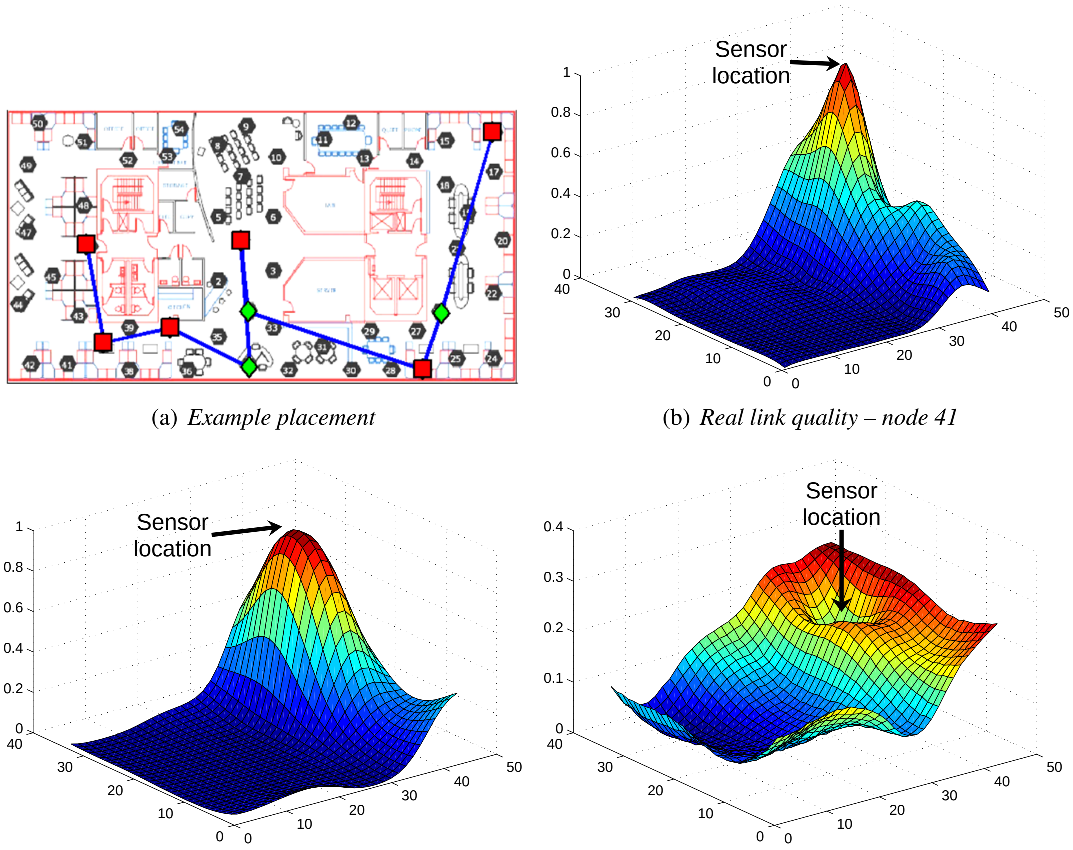
1 Introduction

Networks of small, wireless sensors are becoming increasingly popular for monitoring spatial phenomena, such as the temperature distribution in a building (Deshpande et al., 2004). Since only a limited number of sensors can be placed, it is important to deploy them at most informative locations. Moreover, the use of wireless communication to collect data leads to additional challenges. Poor link qualities, which can arise if sensors are too far apart, or due to obstacles such as walls or radiation from appliances, can cause message loss and hence require a large number of retransmissions in order to collect the data effectively. Such retransmissions drastically consume battery power, and hence decrease the overall deployment lifetime of the sensor network. This suggests that communication cost is a fundamental constraint which must be taken into account when placing wireless sensors.

Existing work on sensor placement under communication constraints (Gupta et al., 2003; Kar and Banerjee, 2003; Funke et al., 2004) has considered the problem mainly from a geometric perspective: Sensors have a fixed *sensing region*, such as a disc with a certain radius, and can only communicate with other sensors which are at most a specified distance apart. These assumptions are problematic for two reasons. Firstly, the notion of a *sensing region* implies that sensors can perfectly observe everything within the region, but nothing outside, which is unrealistic: e.g., the temperature can be highly correlated in some areas of a building but very uncorrelated in others (*c.f.*, Figure 2(a)). Moreover, sensor readings are usually noisy, and one wants to make predictions utilizing the measurements of multiple sensors, making it unrealistic to assume that a single sensor is entirely responsible for a given sensing region. Secondly, the assumption that two sensors at fixed locations can either perfectly communicate (i.e., they are "connected") or not communicate at all (and are "disconnected") is unreasonable, as it does not take into account variabilities in the link quality due to moving obstacles (e.g., doors), interference with other radio transmissions, and packet loss due to reflections (Cerpa et al., 2005). Figure 1(b) shows link quality estimates (package transmission probabilities) between a fixed sensor location (sensor 41) and other sensor locations in the sensor network deployment at Intel Research, Berkeley, as shown in Figure 1(a).

In order to avoid the *sensing region* assumption, previous work (*c.f.*, Cressie, 1991) established *probabilistic models* as an appropriate framework for predicting sensing quality by modeling correlation between sensor locations. Krause et al. (2007) present a method for selecting informative sensor placements based on the *mutual information* criterion. They show that this criterion, originally proposed by Caselton and Zidek (1984), leads to intuitive placements with superior prediction accuracy when compared to existing methods. Furthermore, they provide an efficient algorithm for computing near-optimal placements with strong theoretical performance guarantees. However, this algorithm does not take communication costs into account.

In this paper, we address the general (and much harder) problem of selecting sensor placements that are simultaneously informative, and achieve low communication cost. Note that this problem cannot be solved merely by first finding the most informative locations, and then connecting them up with the least cost—indeed, it is easy to construct examples where such a two-phase strategy performs very poorly. We also avoid the *connectedness* assumption (sensors are "connected" iff they can perfectly communicate): In this paper, we use the *expected number of retransmissions* (De Couto et al., 2003) as a cost metric on the communication between two sensors. This cost metric



(a) *Example placement*

(b) *Real link quality – node 41*

(c) *GP link quality – node 41*

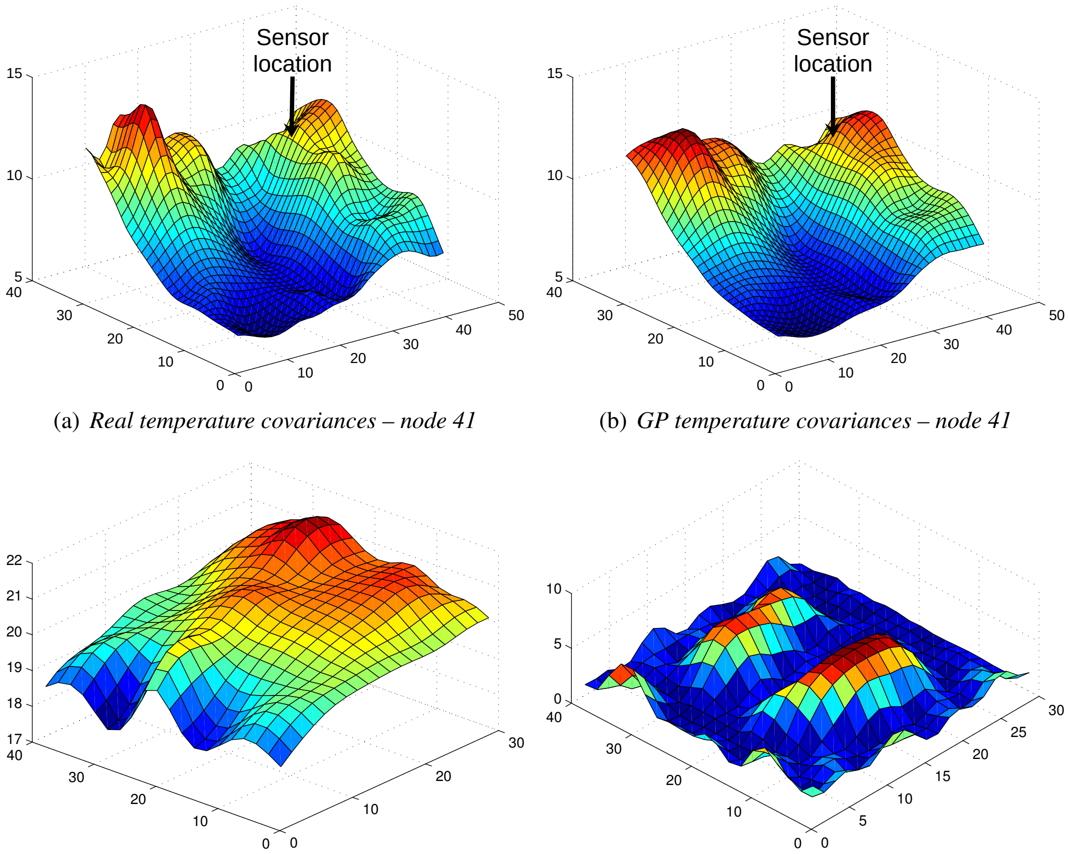
(d) *Link quality var. - node 41*

Figure 1: (a) Indoor deployment of 54 nodes and an example placement of six sensors (squares) and three relay nodes (diamonds); (b) measured transmission link qualities for node 41; (c) GP fit of link quality for node 41 and (d) shows variance of this GP estimate.

directly translates to the deployment lifetime of the wireless sensor network. We propose to use the probabilistic framework of *Gaussian Processes* (Rasmussen and Williams, 2006) not only to model the monitored phenomena, but also to predict communication costs.

Balancing informativeness of sensor placements with the need to communicate efficiently can be formalized as a novel discrete optimization problem. We present a novel algorithm for this placement problem in wireless sensor networks; the algorithm selects sensor placements achieving a specified amount of certainty, with approximately minimal communication cost. Our algorithm centers around a new technique that we call the *modular approximation graph*. In addition to allowing us to obtain sensor placements with efficient communication, we show how this technique can be generalized, e.g., to plan informative paths for robots.

When using probabilistic models for sensor placement, it is possible that an optimized sensor placement can become uninformative if the environment changes. In building monitoring for example, building usage can change, leading to fluctuations in light and temperature patterns. To address



(a) *Real temperature covariances – node 41*

(b) *GP temperature covariances – node 41*

(c) *GP prediction of temperature surface*

(d) *Variance of GP temperature surface*

Figure 2: (a) Measured temperature covariance between node 41 and other nodes in the deployment; (b) predicted covariance using non-stationary GP; (c) predicted temperatures for sensor readings taken at noon on February 28th 2004, and (d) shows the variance of this prediction.

this challenge, we show how our sensor placements can be made robust against such environmental changes.

In summary, our main contributions are:

- A unified method for learning a probabilistic model of the underlying phenomenon and for the expected communication cost between any two locations from a small, short-term initial deployment. These models, based on *Gaussian Processes*, allow us to avoid strong assumptions previously made in the literature.

- A novel and efficient algorithm for Sensor Placements at Informative and cost-Effective Locations (pSPIEL). Exploiting the concept of *submodularity*, this algorithm is guaranteed to provide near-optimal placements for this hard problem.

- An extension to our algorithm that allows us to obtain placements that are robust against changes in the environment.

- A complete solution for collecting data, learning models, optimizing and analyzing sensor placements, realized on Tmote Sky motes, which combines all our proposed methods.
- Extensive evaluations of our proposed methods on temperature and light prediction tasks, using data from real-world sensor network deployments, as well as on a precipitation prediction task in the Pacific Northwest.

2 Problem statement

In this section, we briefly introduce the two fundamental quantities involved in optimizing sensor placements. A *sensor placement* is a finite subset of locations $\mathcal{A}$ from a ground set of possible sensor locations $\mathcal{V}$. Any possible placement is assigned a *sensing quality* $F(\mathcal{A}) \geq 0$, and a *communication cost* $\mathrm{c}(\mathcal{A}) \geq 0$, where the functions F and c will be defined presently. We will use a temperature prediction task as a running example: In this example, our goal is to deploy a network of wireless sensors in a building in order to monitor the temperature field, e.g., to actuate the air conditioning or heating system. Here, the sensing quality refers to our temperature prediction accuracy, and the communication cost depends on how efficiently the sensors communicate with each other. More generally, we investigate the problem of solving optimization problems of the form

$$\min_{\mathcal{A}\subseteq\mathcal{V}} \mathrm{c}(\mathcal{A}) \text{ subject to } F(\mathcal{A}) \geq Q, \tag{1}$$

for some *quota* $Q > 0$, which denotes the required amount of certainty achieved by any sensor placement. This optimization problem aims at finding the minimum cost placement that provides a specified amount of certainty Q, and is called the *covering problem*. We also address the dual problem of solving

$$\max_{\mathcal{A}\subseteq\mathcal{V}} F(\mathcal{A}) \text{ subject to } \mathrm{c}(\mathcal{A}) \leq B, \tag{2}$$

for some *budget* $B > 0$. This optimization problem aims at finding the most informative placement subject to a budget on the communication cost, and is called the *maximization problem*. In practice, one would often want to specify a particular location $s \in \mathcal{V}$ that must be contained in the solution $\mathcal{A}$. This requirement arises, for example, if the deployed network needs to be connected to a base station that is positioned at a fixed location s. We call the optimization problem (1) (resp. (2)) that includes such an additional constraint a *rooted* covering (resp. *rooted* maximization) problem. In this paper, we present efficient approximation algorithms for both the covering and maximization problems, in both the rooted and unrooted formulations.

2.1 What is sensing quality?

In order to quantify how informative a sensor placement is, we have to establish a notion of uncertainty. We associate a random variable $\mathcal{X}_s \in \mathcal{X}_\mathcal{V}$ with each location $s \in \mathcal{V}$ of interest; for a subset $\mathcal{A} \subseteq \mathcal{V}$, let $\mathcal{X}_\mathcal{A}$ denote the set of random variables associated with the locations $\mathcal{A}$. In our temperature measurement example, $\mathcal{V} \subset \mathbb{R}^2$ describes the subset of coordinates in the building where sensors can be placed. Our probabilistic model will describe a joint probability distribution

$P(\mathcal{X}_\mathcal{V})$ over all these random variables. In order to make predictions at a location s, we will consider conditional distributions $P(\mathcal{X}_s = x_s \mid \mathcal{X}_\mathcal{A} = \mathbf{x}_\mathcal{A})$, where we condition on all observations $\mathbf{x}_\mathcal{A}$ made by all sensors $\mathcal{A}$ in our placement. To illustrate this concept, Figure 2(c) shows the predicted temperature field given the measurements of the 54 sensors we deployed, and Figure 2(d) shows the variance in this distribution.

We use the conditional entropy of these distributions,

$$H(\mathcal{X}_s \mid \mathcal{X}_\mathcal{A}) = -\int_{x_s, \mathbf{x}_\mathcal{A}} P(x_s, \mathbf{x}_\mathcal{A}) \log P(x_s \mid \mathbf{x}_\mathcal{A}) dx_s d\mathbf{x}_\mathcal{A}$$

to assess the uncertainty in predicting $\mathcal{X}_s$. Intuitively, this quantity expresses how "peaked" the conditional distribution of $\mathcal{X}_s$ given $\mathcal{X}_\mathcal{A}$ is around the most likely value, averaging over all possible observations $\mathcal{X}_\mathcal{A} = \mathbf{x}_\mathcal{A}$ the placed sensors can make. To quantify how informative a sensor placement $\mathcal{A}$ is, we use the criterion of *mutual information*:

$$F(\mathcal{A}) = I(\mathcal{X}_\mathcal{A}; \mathcal{X}_{\mathcal{V}\setminus\mathcal{A}}) = H(\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}}) - H(\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}} \mid \mathcal{X}_\mathcal{A}). \tag{3}$$

This criterion, first proposed by Caselton and Zidek (1984), expresses the expected reduction of entropy of all locations $\mathcal{V} \setminus \mathcal{A}$ where we did not place sensors, after taking into account the measurements of our placed sensors. Krause et al. (2007) show that mutual information leads to intuitive placements with prediction accuracy superior to alternative approaches. Section 3 explains how we model and learn a joint distribution over all locations $\mathcal{V}$ and how to efficiently compute the mutual information. In addition to mutual information, other objective functions $F(\mathcal{A})$ can be used to measure the sensing quality (*c.f.*, Krause and Guestrin, 2007, and Section 5.1).

2.2 What is communication cost?

Since each transmission drains battery of the deployed sensors, we have to ensure that our sensor placements have reliable communication links, and the number of unnecessary retransmissions is minimized. If the probability for a successful transmission between two sensor locations s and t is $\theta_{s,t}$, the expected number of retransmissions is $1/\theta_{s,t}$. Since we have to predict the success probability between any two locations $s, t \in \mathcal{V}$, we will in general only have a distribution $P(\theta_{s,t})$ with density $p(\theta_{s,t})$ instead of a fixed value for $\theta_{s,t}$. Surprisingly, this uncertainty has a fundamental effect on the expected number of retransmissions. For a simple example, assume that with probability $\frac{1}{2}$ we predict that our transmission success rate is $\frac{3}{4}$, and with probability $\frac{1}{2}$, it is $\frac{1}{4}$. Then, the mean transmission rate would be $\frac{1}{2}$, leading us to assume that the expected number of retransmissions might be 2. In expectation over the success rate however, our expected number of retransmissions becomes $\frac{1}{2} \cdot 4 + \frac{1}{2} \cdot \frac{4}{3} = 2 + \frac{2}{3} > 2$. More generally, the expected number is

$$\mathrm{c}_{s,t} = \int_\theta \frac{1}{\theta_{s,t}} p(\theta_{s,t}) d\theta_{s,t}. \tag{4}$$

Using this formula, we can compute the expected number of retransmissions for any pair of locations. If $\mathcal{V}$ is finite, we can model all locations in $\mathcal{V}$ as nodes in a graph $\mathcal{G} = (\mathcal{V}, E)$, with the edges E labeled by their communication costs. We call this graph the *communication graph* of $\mathcal{V}$. For any sensor placement $\mathcal{A} \subseteq \mathcal{V}$, we define its cost $\mathrm{c}(\mathcal{A})$ by the minimum cost tree $\mathcal{T}$, $\mathcal{A} \subseteq \mathcal{T} \subseteq \mathcal{V}$, connecting all sensors $\mathcal{A}$ in the communication graph for $\mathcal{V}$. [1] This cost model applies to the common setting where all sensors obtain sensor measurements and send them to a base station. Finding this minimum cost tree $\mathcal{T}$ to evaluate the cost function $\mathrm{c}(\mathcal{A})$ is called the *Steiner tree* problem; an $\mathbf{NP}$-complete problem that has very good approximation algorithms (Vazirani, 2003). Our algorithm, pSPIEL, will however not just find an informative placement and then simply add relay nodes, since the resulting cost may be exorbitant. Instead, it *simultaneously* optimizes sensing quality and communication cost.

Note that if we threshold all link qualities at some specified cut-off point, and define the edge costs between two locations in the communication graph as 1 if the link quality is above the cut-off point, and infinite if the link quality is below the cut-off point, then the communication cost of a sensor placement is exactly (one less than) the number of placed sensors. Hence, in this special case, we can interpret the maximization problem (2) as the problem of finding the most informative connected sensor placement of at most $B + 1$ nodes.

2.3 Overview of our approach

Having established the notions of sensing quality and communication cost, we now present an outline of our proposed approach.

1. We collect sensor and link quality data from an initial deployment of sensors. From this data, we learn probabilistic models for the sensor data and the communication cost. Alternatively, we can use expert knowledge to design such models.

2. These models allow us to predict the sensing quality $F(\mathcal{A})$ and communication cost $\mathrm{c}(\mathcal{A})$ for any candidate placement $\mathcal{A} \subseteq \mathcal{V}$.

3. Using pSPIEL, our proposed algorithm, we then find highly informative placements which (approximately) minimize communication cost. We can approximately solve both the covering and maximization problems.

4. After deploying the sensors, we then possibly add sensors or redeploy the existing sensors, by restarting from Step 2), until we achieve a satisfactory placement. (This step is optional.)

Consider our temperature prediction example. Here, in step 1), we would place a set of motes throughout the building, based on geometrical or other intuitive criteria. After collecting training data consisting of temperature measurements and packet transmission logs, in step 2), we learn probabilistic models from the data. This process is explained in the following sections. Figure 2(c) and Figure 2(d) present examples of the mean and variance of our model learned during this step. As expected, the variance is high in areas where no sensors are located. In step 3), we would then explore the sensing quality tradeoff for different placements proposed by pSPIEL, and select an appropriate one. This placement automatically suggests if relay nodes should be deployed. After

[1] In general, the locations $\mathcal{A}$ may include distant sensors, requiring us to place *relay nodes*, which do not sense but only aid communication. It can occur that $\mathrm{c}(\{s,t\}) < \mathrm{c}_{s,t}$, i.e., it can be more cost effective to route messages from s to t via some other intermediate node than send them directly from s to t.

deployment, we can collect more data, and, if the placement is not satisfactory, iterate by repeating from step 2).

3 Predicting sensing quality

In order to achieve highly informative sensor placements, we have to be able to predict the uncertainty in sensor values at a location $s \in \mathcal{V}$, given the sensor values $\mathbf{x}_\mathcal{A}$ at some candidate placement $\mathcal{A}$. This is an extension of the well-known regression problem (*c.f.*, Guestrin et al., 2004), where we use the measured sensor data to predict values at locations where no sensors are placed. The difference is that in the placement problem, we must be able to predict not just sensor values at uninstrumented locations, but rather *probability distributions* over sensor values. *Gaussian Processes* are a powerful class of models for making such predictions. To introduce this concept, first consider the special case of the multivariate normal distribution over a set $\mathcal{X}_\mathcal{V}$ of random variables associated with n locations $\mathcal{V}$:

$$P(\mathcal{X}_\mathcal{V} = \mathbf{x}_\mathcal{V}) = \frac{1}{(2\pi)^{n/2}|\Sigma|} e^{-\frac{1}{2}(\mathbf{x}_\mathcal{V}-\mu)^T\Sigma^{-1}(\mathbf{x}_\mathcal{V}-\mu)}.$$

This model has been successfully used for example to model temperature distributions (Deshpande et al., 2004), where every location in $\mathcal{V}$ corresponds to one particular sensor placed in the building. The multivariate normal distribution is fully specified by providing a mean vector μ and a covariance matrix Σ. If we know the values of some of the sensors $\mathcal{A} \subseteq \mathcal{V}$, we find that for $s \in \mathcal{V} \setminus \mathcal{A}$ the conditional distribution $P(\mathcal{X}_s = x_s \mid \mathcal{X}_\mathcal{A} = \mathbf{x}_\mathcal{A})$ is a normal distribution, where mean $\mu_{s|\mathcal{A}}$ and variance $\sigma^2_{s|\mathcal{A}}$ are given by

$$\begin{aligned} \mu_{s|\mathcal{A}} &= \mu_s + \Sigma_{s\mathcal{A}}\Sigma^{-1}_{\mathcal{A}\mathcal{A}}(\mathbf{x}_\mathcal{A} - \mu_\mathcal{A}), && (5)\\ \sigma^2_{s|\mathcal{A}} &= \sigma^2_s - \Sigma_{s\mathcal{A}}\Sigma^{-1}_{\mathcal{A}\mathcal{A}}\Sigma_{\mathcal{A}s}. && (6) \end{aligned}$$

Hereby, $\Sigma_{s\mathcal{A}} = \Sigma^T_{\mathcal{A}s}$ is a row vector of the covariances of $\mathcal{X}_s$ with all variables in $\mathcal{X}_\mathcal{A}$. Similarly, $\Sigma_{\mathcal{A}\mathcal{A}}$ is the submatrix of Σ, only containing the entries relevant to $\mathcal{X}_\mathcal{A}$, and σ^2_s is the variance of $\mathcal{X}_s$. $\mu_\mathcal{A}$ and μ_s are the means of $\mathcal{X}_\mathcal{A}$ and $\mathcal{X}_s$ respectively. Hence the covariance matrix Σ and the mean vector μ contain all the information needed to compute the conditional distributions of $\mathcal{X}_s$ given $\mathcal{X}_\mathcal{A}$. The goal of an optimal placement will intuitively be to select the observations such that the posterior variance (6) for all variables becomes uniformly small. If we can make a set of T measurements $\mathbf{x}^{(1)}_\mathcal{V}, \ldots, \mathbf{x}^{(T)}_\mathcal{V}$ of all sensors $\mathcal{V}$, we can estimate Σ and μ, and use it to compute predictive distributions for any subsets of variables. However, in the sensor placement problem, we must reason about the predictive quality of locations where we do *not* yet have sensors, and thus need to compute predictive distributions, conditional on variables for which we do not have sample data.

Gaussian Processes are a solution for this dilemma. Technically, a Gaussian Process (GP) is a joint distribution over a (possibly infinite) set of random variables, such that the marginal distribution over any finite subset of variables is multivariate Gaussian. In our temperature measurement example, we would associate a random variable $\mathcal{X}(s)$ with each point s in the building, which can be modeled as a subset $\mathcal{V} \subset \mathbb{R}^2$. The GP $\mathcal{X}(\cdot)$, which we will refer to as the *sensor data process*, is fully specified by a *mean function* $\mathcal{M}(\cdot)$ and a symmetric positive definite *Kernel function* $\mathcal{K}(\cdot,\cdot)$, generalizing the mean vector and covariance matrix in the multivariate normal distribution: For any random variable $\mathcal{X}(s) \in \mathcal{X}$, $\mathcal{M}(s)$ will correspond to the mean of $\mathcal{X}(s)$, and for any two random variables $\mathcal{X}(s), \mathcal{X}(t) \in \mathcal{X}$, $\mathcal{K}(s,t)$ will be the covariance of $\mathcal{X}(s)$ and $\mathcal{X}(t)$. This implies, that for any finite subset $\mathcal{A} = \{s_1, s_2, \ldots, s_m\}$, $\mathcal{A} \subseteq \mathcal{V}$ of locations variables, the covariance matrix $\Sigma_{\mathcal{A}\mathcal{A}}$ of the variables $\mathcal{X}_{\mathcal{A}}$ is obtained by

$$\Sigma_{\mathcal{A}\mathcal{A}} = \begin{pmatrix} \mathcal{K}(s_1,s_1) & \mathcal{K}(s_1,s_2) & \ldots & \mathcal{K}(s_1,s_m) \\ \mathcal{K}(s_2,s_1) & \mathcal{K}(s_2,s_2) & \ldots & \mathcal{K}(s_2,s_m) \\ \vdots & \vdots & & \vdots \\ \mathcal{K}(s_m,s_1) & \mathcal{K}(s_m,s_2) & \ldots & \mathcal{K}(s_m,s_m) \end{pmatrix},$$

and its mean is $\mu_{\mathcal{A}} = (\mathcal{M}(s_1), \mathcal{M}(s_2), \ldots, \mathcal{M}(s_m))$. Using formulas (5) and (6), the problem of computing predictive distributions is reduced to finding the mean and covariance functions $\mathcal{M}$ and $\mathcal{K}$ for the phenomena of interest. In general, this is a difficult problem – we want to estimate these infinite objects from a finite amount of sample data. Consequently, often strongly limiting assumptions are made: It is assumed that the covariance of any two random variables is independent of their location (stationarity), or even only a function of their distance (isotropy). A kernel function often used is the Gaussian kernel

$$\mathcal{K}(s,t) = \exp\left(-\frac{\|s-t\|_2^2}{h^2}\right). \tag{7}$$

These isotropy or stationarity assumptions lead to similar problems as encountered in the approach using geometric sensing regions, as spatial inhomogeneities such as walls, windows, reflections etc. are not taken into account. These inhomogeneities are however dominantly encountered in real data sets, as indicated in Figure 2(a).

In this paper, we do *not* make these limiting assumptions. We use an approach to estimate nonstationarity proposed by Nott and Dunsmuir (2002). Their method estimates several stationary GPs with kernel functions as in (7), each providing a local description of the nonstationary process around a set of reference points. These reference points are chosen on a grid or near the likely sources of nonstationary behavior. The stationary GPs are combined into a nonstationary GP, whose covariance function interpolates the empirical covariance matrix estimated from the initial sensor deployment, and near the reference points behaves similarly to the corresponding stationary process. Figure 2(b) shows a learned nonstationary GP for our temperature data. We refer the reader to Nott and Dunsmuir (2002) for more details. Note that as non-parametric models, given enough sensor data, GPs can model very complex processes, including phenomena decaying as inverse polynomial laws, etc. Rasmussen and Williams (2006).

Once we have obtained estimates for the mean and covariance functions, we can use these functions to evaluate the mutual information criterion. In order to evaluate Equation (3), we need to compute conditional entropies $H(\mathcal{X}_s \mid \mathcal{X}_{\mathcal{A}})$, which involve integrals over all possible assignments to the placed sensors $\mathbf{x}_{\mathcal{A}}$. Fortunately, there is a closed form solution: We find that

$$H(\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}} \mid \mathcal{X}_{\mathcal{A}}) = \frac{1}{2}\log((2\pi e)^n \left|\Sigma_{\mathcal{V}\setminus\mathcal{A}|\mathcal{A}}\right|),$$

hence it only depends on the determinant of the predictive covariance matrix $\Sigma_{\mathcal{V}\setminus\mathcal{A}|\mathcal{A}}$. Hereby, $\Sigma_{\mathcal{V}\setminus\mathcal{A}|\mathcal{A}}$ can be inferred using Equation (6). Details on efficient computation are described, e.g., by Krause et al. (2007).

In addition to the mutual information criterion, our approach applies to a variety of other criteria, discussed in more detail in Section 5.1. Also note that in order to address phenomena that change over time, one can replace the spatial model $P(\mathcal{X}_\mathcal{V})$ with a *spatio-temporal* model. In this case, we associate with every location $s \in \mathcal{V}$ the set of all measurements that will be made at this location over time, similarly as in the approach by Meliou et al. (2007).

4 Predicting communication cost

As discussed in Section 2.2, an appropriate measure for communication cost is the expected number of retransmissions. If we have a probability distribution $P(\theta_{s,t})$ over transmission success probabilities $\theta_{s,t}$, Equation (4) can be used in a Bayesian approach to compute the expected number of retransmissions. The problem of determining such predictive distributions for transmission success probabilities is very similar to the problem of estimating predictive distributions for the sensor values as discussed in Section 3, suggesting the use of GPs for predicting link qualities. A closer look however shows several qualitative differences: When learning a model for sensor values, samples from the actual values can be obtained. In the link quality case however, we can only determine whether certain messages between nodes were successfully transmitted or not. Additionally, transmission success probabilities are constrained to be between 0 and 1. Fortunately, GPs can be extended to handle this case as well (Csato et al., 2000). In this *classification* setting, the predictions of the GP are transformed by the sigmoid, also called link function, $f(x) = \frac{1}{1+\exp(-x)}$. For large positive values of x, $f(x)$ is close to 1, for large negative values it is close to 0 and $f(0) = \frac{1}{2}$.

Since we want to predict link qualities for every *pair* of locations in $\mathcal{V}$, we define a random process $\Theta(s,t) = f(W(s,t))$, where $W(s,t)$ is a GP over $(s,t) \in \mathcal{V}^2$. We call $\Theta(s,t)$ the *link quality process*. This process can be learned the following way. In our initial deployment, we let each sensor broadcast a message once every epoch, containing its identification number. Each sensor also records, from which other sensors it has received messages this epoch. This leads to a collection of samples of the form $(s_{i,k}, s_{j,k}, \theta_k(s_i,s_j))_{i,j,k}$, where i,j range over the deployed sensors, k ranges over the epochs of data collection, and $\theta_k(s_i,s_j)$ is 1 if node i received the message from node j in epoch k, and 0 otherwise. We will interpret $\theta_k(s_i,s_j)$ as samples from the link quality process $\Theta(\cdot,\cdot)$. Using these samples, we want to compute predictive distributions similar to those described in Equations (5) and (6). Unfortunately, in the classification setting, the predictive distributions cannot be computed in closed form anymore, but one can resort to approximate techniques (*c.f.*, Csato et al., 2000). Using these techniques, we infer the link qualities by modeling the underlying GP $W(s,t)$. Intuitively, the binary observations will be converted to "hallucinated" observations of $W(s,t)$, such that $\Theta(s,t) = f(W(s,t))$ will correspond to the empirical transmission probabilities between locations s and t. We now can use Equations (5) and (6) to compute the predictive distributions $W(s,t)$ for *any* pair of locations $(s,t) \in \mathcal{V}^2$. Applying the sigmoid transform will then result in a probability distribution over transmission success probabilities. In our implementation, instead of parameterizing $W(s,t)$ by pairs of coordinates, we use the parametrization $W(t-s,s)$. The

```
Input: Locations $\mathcal{C} \subseteq \mathcal{V}$
Output: Greedy sequence $g_1, g_2, \ldots, g_{|\mathcal{C}|}, \mathcal{C}_i = \{g_1, \ldots, g_i\}$
begin
    $\mathcal{C}_0 \leftarrow \emptyset$;
    for $j$ = $1$ to $|\mathcal{C}|$ do
        $g_j \leftarrow \operatorname*{argmax}_{g \in \mathcal{C} \setminus \mathcal{C}_{j-1}} F(\mathcal{C}_{j-1} \cup \{g\}) - F(\mathcal{C}_{j-1})$;
        $\mathcal{C}_j \leftarrow \mathcal{C}_{j-1} \cup g_j$;
    end
end
```

Algorithm 1: Greedy algorithm for maximizing mutual information.

first component of this parametrization is the displacement the successful or unsuccessful message has traveled, and the second component is the actual set of physical coordinates of the transmitting sensor. This parametrization tends to exhibit better generalization behavior, since the distance to the receiver (component 1) is the dominating feature, when compared to the spatial variation in link quality. Figure 1(c) shows an example of the predicted link qualities using a GP for our indoors deployment, Figure 1(d) shows the variance in this estimate.

What is left to do is to compute the expected number of retransmissions, as described in formula (4). Assuming the predictive distribution for $W(s,t)$ is normal with mean μ and variance σ^2, we compute $\int \frac{1}{f(x)} \mathcal{N}(x; \mu, \sigma^2) dx = 1 + \exp(-\mu + \sigma^2)$, where $\mathcal{N}(\cdot; \mu, \sigma^2)$ is the normal density with mean μ and variance σ^2. Hence we have a closed form solution for this integral. If $\sigma^2 = 0$, we simply retain that the expected number of retransmissions is the inverse of the transmission success probability. If σ^2 is very large however, the expected number of retransmission drastically increases. This implies that even if we predict the transmission success probability to be reasonably high, e.g., $2/3$, if we do not have enough samples to back up this prediction and hence our predictive variance σ^2 is very large, we necessarily have to expect the worst for the number of retransmissions. So, using this GP model, we may determine that it is better to select a link with success probability $1/3$, about which we are very certain, to a link with a higher success probability, but about which we are very uncertain. Enabling this tradeoff is a great strength of using GPs for predicting communication costs. Note that instead of using GPs, any other method for quantifying communication cost between arbitrary pairs of locations can be used in our approach as well.

5 Problem structure in sensor placement optimization

We now address the covering and maximization problems described in Section 2. We will consider a discretization of the space into finitely many points $\mathcal{V}$, e.g., points lying on a grid. For each pair of locations in $\mathcal{V}$, we define the edge cost as the expected number of retransmissions required to send a message between these nodes (since link qualities are asymmetric, we use the worse direction as the cost). The set of edges that have finite cost is denoted by E. The challenge in solving the optimization problems (1) and (2) is that the search space—the possible subsets $\mathcal{A} \subseteq \mathcal{V}$—is exponential; more concretely, the problem is easily seen to be $\mathbf{NP}$-hard as a corollary to the hardness of the unconstrained optimization problem (Krause et al., 2007; Kar and Banerjee, 2003). Given this, we seek an efficient approximation algorithm with strong performance guarantees. In Section 6, we present such an algorithm. The key to finding good approximate solutions is understanding and exploiting problem structure.

Intuitively, we expect that in many cases, the sensor placement problem satisfies the following diminishing returns property: The more sensors already placed, the less the addition of a new sensor helps us. This intuition is formalized by the concept of *submodularity*: A set function F defined on subsets of $\mathcal{V}$ is called *submodular* (*c.f.*, Nemhauser et al., 1978), if

$$F(\mathcal{A} \cup \{s\}) - F(\mathcal{A}) \geq F(\mathcal{B} \cup \{s\}) - F(\mathcal{B}), \tag{8}$$

for all $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}$ and $s \in \mathcal{V} \setminus \mathcal{B}$. The function F is called *monotonic* if $F(\mathcal{A}) \leq F(\mathcal{B})$ for all $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{V}$. Note that the rate at which diminishing returns occurs can vary across the sensing domain. For example, in our temperature prediction example, there could be two types of rooms in the builing: large rooms where the temperature varies smoothly, and thus a small number of measurements would allow accurate predictions; and small rooms, where temperature fluctuates more rapidly due to external influences such as outside temperature, different appliances etc. If the temperature were, for example, represented by a probabilistic model, one may imagine that the correlations are more far reaching in the large rooms, and narrower in the smaller rooms. Thus, in the large rooms, the first measurement provides high sensing quality, and the incremental benefits decrease quickly afterwards. In the smaller rooms, the first measurement provides low utility (as it helps predicting only for a small area), but the next measurements provide significant additional information, thus diminishing returns occurs later. There are certainly sensing problems that are not submodular (e.g., Krause and Guestrin (2009)), where a strong increase in sensing quality can be achieved only by placing multiple sensors. As we show in Section 5.1, however, many practical sensing quality functions are provably submodular.

In addition to *submodularity*, the sensing quality exhibits another important *locality* property: Sensors which are very far apart are approximately independent. This implies that if we consider placing a subset of sensors $\mathcal{A}_1$ in one area of the building, and $\mathcal{A}_2$ in another area, then $F(\mathcal{A}_1 \cup \mathcal{A}_2) \approx F(\mathcal{A}_1) + F(\mathcal{A}_2)$. More formally, we say two sets $\mathcal{A}_1$ and $\mathcal{A}_2$ have distance

$$d(\mathcal{A}_1, \mathcal{A}_2) = \min_{s \in \mathcal{A}_1, t \in \mathcal{A}_2} c(\{s, t\}).$$

We will abstract out the locality property to assume that there are constants $r > 0$ and $0 < \gamma \leq 1$, such that for any subsets of nodes $\mathcal{A}_1$ and $\mathcal{A}_2$ such that $d(\mathcal{A}_1, \mathcal{A}_2) \geq r$ it holds that $F(\mathcal{A}_1 \cup \mathcal{A}_2) \geq F(\mathcal{A}_1) + \gamma F(\mathcal{A}_2)$. Such a submodular function F will be called (r, γ)-*local*. Note that even phenomena that appear to be non-local can often be modeled using local objective functions. Consider our temperature prediction example. External influences such as outside temperature can induce correlation between all sensing locations. However, often, such external influences can be modeled by subtracting an appropriately chosen mean function (which, e.g., models the mean temperature at different locations during different times of the day) from the GP (Rasmussen and Williams, 2006). In this case, the correlations between the deviations from the mean function (and therefore the sensing quality function) are typically local.

5.1 Examples of (r, γ)-local submodular functions

There are several important examples of monotonic, submodular and (r, γ)-local objective functions:

Geometric coverage Suppose that with each location $s \in \mathcal{V}$, we associate a sensing region $B_s \subseteq \mathcal{V}$. Then the function

$$F(\mathcal{A}) = \left| \bigcup_{s \in \mathcal{A}} B_s \right|$$

measures the size of the region covered by placing sensors at locations $\mathcal{A}$. In this case, F is monotonic, submodular and $(r, 1)$-local, where $r = 2 \max_s \max_{i,j \in B_s} d(i, j)$ is twice the maximum diameter of the sensing regions. This objective function F captures the commonly used disk model.

Probabilistic detections Suppose, we want to place sensors for event detection (e.g., detecting fires in buildings), and that a sensor placed at a location s can detect events at distance d with probability $0 \leq \varphi_s(d) \leq 1$, where φ_s is a monotonically decreasing function. Further suppose that we place sensors at locations $\mathcal{A}$. Then, if each sensor detects independently of the others, the probability of detecting an event happening at location $t \in \mathcal{V}$ is

$$F_t(\mathcal{A}) = 1 - \prod_{s \in \mathcal{A}} (1 - \varphi_s(d(s, t))).$$

Now let $r = 2\varphi^{-1}(1/2)$, i.e., the distance at which detection happens with probability $1/2$. Then F_t is a monotonic, submodular and $(r, 1/2)$-local. Now suppose we have a distribution $Q(t)$ over the possible outbreak locations. Then the expected detection performance

$$F(\mathcal{A}) = \sum_t Q(t) F_t(\mathcal{A})$$

is, as a convex combination of monotonic, submodular and $(r, 1/2)$-local objectives also monotonic, submodular and $(r, 1/2)$-local.

Mutual information Suppose $P(\mathcal{X}_\mathcal{V})$ is a GP with compact kernel, i.e., there is a constant r such that $\mathcal{K}(s, t) = 0$ whenever $d(s, t) \geq r/2$. Then the mutual information $F(\mathcal{A}) = H(\mathcal{X}_{\mathcal{V} \setminus \mathcal{A}}) - H(\mathcal{X}_{\mathcal{V} \setminus \mathcal{A}} \mid \mathcal{X}_\mathcal{A})$ is $(r, 1)$-local. Even if the kernel is not compact, mutual information empirically exhibits (r, γ)-locality (*c.f.*, Figure 6(d)).

Expected mean squared prediction error Another possible choice for the sensing quality function is the expected reduction in mean squared prediction error (MSE):

$$F(\mathcal{A}) = \sum_{s \in \mathcal{V}} \int P(\mathbf{x}_\mathcal{A}) [\mathrm{Var}(\mathcal{X}_s) - \mathrm{Var}(\mathcal{X}_s \mid \mathbf{x}_\mathcal{A})] d\mathbf{x}_\mathcal{A},$$

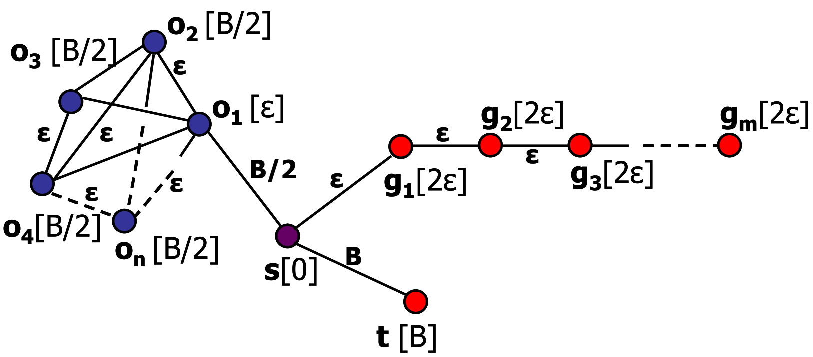


Figure 3: Example demonstrating the poor performance of the greedy algorithm.

where

$$\mathrm{Var}(\mathcal{X}_s \mid \mathbf{x}_\mathcal{A}) = \mathbb{E}[(\mathcal{X}_s - \mathbb{E}(\mathcal{X}_s \mid \mathbf{x}_\mathcal{A}))^2 \mid \mathbf{x}_\mathcal{A}]$$

is the predictive variance at location $\mathcal{X}_s$ after observing $\mathcal{X}_\mathcal{A} = \mathbf{x}_\mathcal{A}$. $F(\mathcal{A})$ is always monotonic. Under the same conditions as for the mutual information criterion $F(\mathcal{A})$ is $(r, \gamma)-$local. Under some additional conditions[2] on the covariance function $\mathcal{K}$, the criterion is also submodular.

5.2 The greedy algorithm

With any such monotonic submodular set function F, we can associate the following greedy algorithm: Suppose we would like to find the set of k locations maximizing the sensing quality $F(\mathcal{A})$, irrespective of the cost $\mathrm{c}(\mathcal{A})$. The greedy algorithm starts with the empty set, and at each iteration add to the current set $\mathcal{A}'$ the element s which maximizes the *greedy improvement* $F(\mathcal{A}' \cup \{s\}) - F(\mathcal{A}')$, and continue until $\mathcal{A}'$ has the specified size of k elements. Perhaps surprisingly, if $\mathcal{A}_G$ is the set selected by the greedy algorithm (with $|\mathcal{A}_G| = k$) and if F is monotonic submodular with $F(\emptyset) = 0$, then

$$F(\mathcal{A}_G) \geq (1 - 1/e) \max_{\mathcal{A}:|\mathcal{A}|=k} F(\mathcal{A}),$$

i.e., $\mathcal{A}_G$ is at most a constant factor $(1 - 1/e)$ worse than the optimal solution (Nemhauser et al., 1978). Krause et al. (2007) prove that the mutual information criterion is submodular and *approximately* monotonic: For any $\varepsilon > 0$, if we choose the discretization fine enough (polynomially-large in $1/\varepsilon$), then the solution obtained by the greedy algorithm is at most $(1-1/e)OPT - \varepsilon$. Algorithm 1 presents the greedy algorithm for mutual information; for details we refer the reader to Krause et al. (2007).

Unfortunately, the near-optimality of the greedy algorithm only holds when we do not take communication cost into account, and does not generalize to the covering and maximization problems (1) and (2) which we study in this paper. For an illustration, consider Figure 3. In this illustration, we consider an additive (a special case of a submodular) sensing quality function F defined on the ground set $\mathcal{V} = \{s, t, o_1, \ldots, o_n, g_1, \ldots, g_m\}$, where the sensing quality of a selected set $\mathcal{A}$ of nodes is the sum of the values in squared brackets associated with the selected nodes (e.g.,

[2] Under *conditional suppressor-freeness*, *c.f.*, Das and Kempe (2008) for details.

$F(\{s, o_1, g_1\}) = 3\varepsilon$). Consider the setting where we want to solve the maximization problem with root s and budget B. The optimal solution would be to choose the set $\mathcal{A}^* = \{s, o_1, \ldots, o_{B/(2\varepsilon)}\}$, with value $F(\mathcal{A}^*) = (B/\varepsilon - 1)(B/2) + \varepsilon$. The simple greedy algorithm that ignores cost would first pick t, and hence immediately run out of budget, returning set $\mathcal{A}_G = \{s, t\}$ with total sensing quality B. A greedy algorithm that takes the cost into account greedily selecting the element

$$s^* = \operatorname*{argmax}_{s \in \mathcal{V} \setminus \mathcal{A}} \frac{F(\mathcal{A} \cup \{s\} - F(\mathcal{A})}{\mathrm{c}(\mathcal{A} \cup \{s\}) - \mathrm{c}(\mathcal{A})}$$

and hence optimizing the benefit/cost ratio of the chosen element s^* would select the set $\mathcal{A}_{GCB} = \{s, g_1, \ldots, g_{B/\varepsilon}\}$ with total value $2B$. Hence, as $\varepsilon \to 0$, the greedy algorithm performs arbitrarily worse than the optimal solution. In Section 7 we show that this poor performance of the greedy algorithm actually occurs in practice.

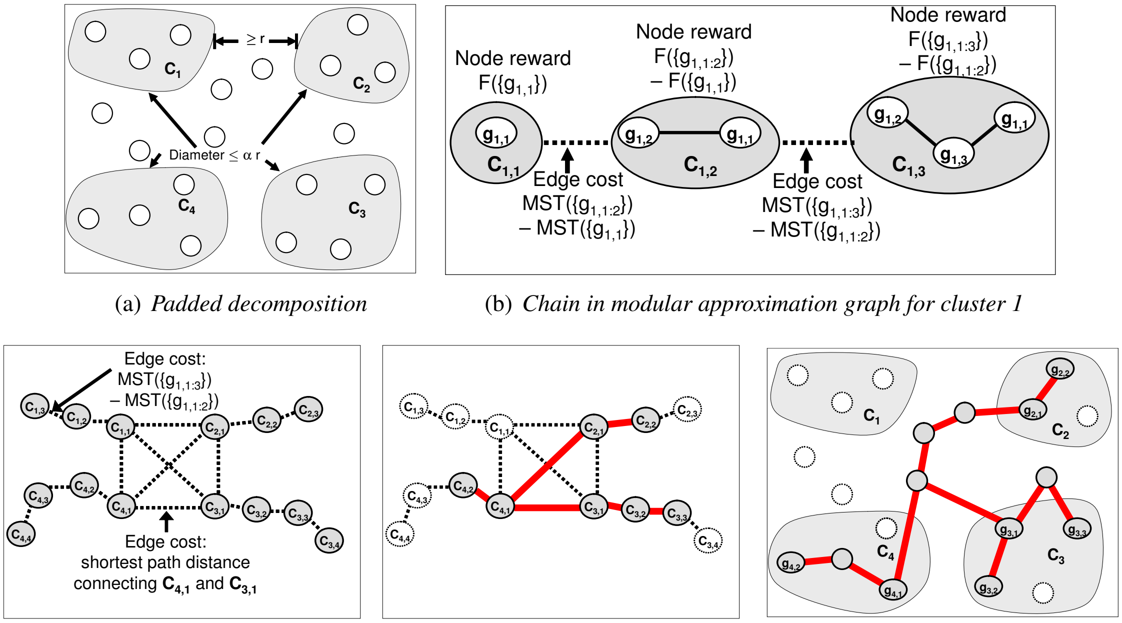
6 Approximation algorithm

In this section, we propose an efficient approximation algorithm for selecting Padded Sensor Placements at Informative and cost-Effective Locations (pSPIEL). Our algorithm assumes that the sensing quality function is (r, γ)-local submodular (as discussed in Section 5). As input, it is given the discretization $\mathcal{V}$ of the sensing domain, the sensing quality function $F(\mathcal{A})$ (for example, based on the mutual information criterion applied to a GP) and a communication graph $\mathcal{G} = (\mathcal{V}, E)$, where the edges indicate the communication cost (e.g., based on the expected number of retransmissions) between any two possible sensing locations. Before presenting our results and performance guarantees, here is an overview of our algorithm.

1. We randomly select a decomposition of the possible locations $\mathcal{V}$ into *small* clusters using Algorithm 2 (*c.f.*, Figure 4(a), Section 6.1, Gupta et al. (2003)). Nodes close to the "boundary" of their clusters are stripped away and hence the remaining clusters are "well-separated". (We prove that not too many nodes are stripped away). The well-separatedness and the locality property of F ensure the clusters are approximately independent, and hence very informative. Since the clusters are small, we are not concerned about communication cost within the clusters.

2. We use the greedy algorithm (Algorithm 1) within each cluster i to get an order $g_{i,1}, g_{i,2}, \ldots g_{i,n_i}$ on the n_i nodes in cluster i. We call $z_i = g_{i,1}$ the *center* of cluster i. Create a chain for this cluster by connecting the vertices in this order, with suitably chosen costs for each edge $(g_{i,j}, g_{i,j+1})$, as in Figure 4(b). The submodularity of F ensures that the first k nodes in this chain are almost as informative as the best subset of k nodes in the cluster (Krause et al., 2007).

3. Create a "modular approximation graph" $\mathcal{G}'$ from $\mathcal{G}$ by taking all these chains, and creating a fully connected graph on the cluster centers $z_1, z_2, \ldots, z_m$, the first nodes of each chain. The edge costs $(z_i, z_{i'})$ correspond to the shortest path distances between z_i and $z_{i'}$, as in Figure 4(c).


≥ r
C_1
C_2
C_3
C_4
Diameter ≤ α r
Node reward F({$g_{1,1}$})
Node reward F({$g_{1,1:2}$}) – F({$g_{1,1}$})
Node reward F({$g_{1,1:3}$}) – F({$g_{1,1:2}$})
$g_{1,1}$
$C_{1,1}$
$g_{1,2}$
$C_{1,2}$
$g_{1,3}$
$C_{1,3}$
Edge cost MST({$g_{1,1:2}$}) – MST({$g_{1,1}$})
Edge cost MST({$g_{1,1:3}$}) – MST({$g_{1,1:2}$})
Edge cost: MST({$g_{1,1:3}$}) – MST({$g_{1,1:2}$})
$C_{1,1}$
$C_{2,1}$
$C_{2,2}$
$C_{2,3}$
$C_{4,1}$
$C_{4,2}$
$C_{4,3}$
$C_{4,4}$
$C_{3,1}$
$C_{3,2}$
$C_{3,3}$
$C_{3,4}$
Edge cost: shortest path distance connecting $C_{4,1}$ and $C_{3,1}$
$g_{2,1}$
$g_{2,2}$
$g_{3,1}$
$g_{3,2}$
$g_{3,3}$
$g_{4,1}$
$g_{4,2}$

(a) *Padded decomposition*

(b) *Chain in modular approximation graph for cluster 1*

(c) *Modular approximation graph*

(d) *Quota-MST solution on MAG*

(e) *Final solution*

Figure 4: Illustration of our algorithm: (a) presents a padded decomposition into four clusters; (b) displays the chain in the modular approximation graph associated with cluster 1; (c) shows the modular approximation graph with chains induced by greedy algorithm and the complete "core"; (d) the solution of the Quota-MST problem on the modular approximation graph; and (e) is the final solution after expanding the Quota-MST edges representing shortest paths.

4. We now need to decide how to distribute the desired quota to the clusters. Hence, we approximately solve the Quota-MST problem (for the covering version) or the Budget-MST problem (for the maximization problem) on $\mathcal{G}'$ (Garg, 2005; Johnson et al., 2000) (Figure 4(d)).

5. Expand the chosen edges of $\mathcal{G}'$ in terms of the shortest paths they represent in $\mathcal{G}$, as in Figure 4(e).

Suppose $n = |\mathcal{V}|$ is the number of nodes in $\mathcal{V}$, and $\mathcal{A}^*$ denotes the optimal set (for the covering or maximization problem), with cost ℓ^*. Finally, let $\dim(\mathcal{V}, E)$ be the *doubling dimension* of the data, which is constant for many graphs (and for costs that can be embedded in low-dimensional spaces), and is $\mathcal{O}(\log n)$ for arbitrary graphs (*c.f.*, Gupta et al., 2003). We prove the following guarantee:

Theorem 1. *Let $\mathcal{G} = (\mathcal{V}, E)$ be a graph and a F be a (r,γ)-local monotone submodular function on $\mathcal{V}$. Suppose $\mathcal{G}$ contains a tree $\mathcal{T}^*$ with cost ℓ^*, spanning a set $\mathcal{A}^*$. Then* pSPIEL *can find a tree $\mathcal{T}$ with cost $\mathcal{O}(r\dim(\mathcal{V},E))\times\ell^*$, spanning a set $\mathcal{A}$ with expected sensing quality $F(\mathcal{A})\geq\Omega(\gamma)\times F(\mathcal{A}^*)$. The algorithm is randomized and runs in polynomial-time.* □

In other words, Theorem 1 shows that we can solve the covering and maximization problems (1) and (2) to provide a sensor placement for which the communication cost is at most a small factor (at worst logarithmic) larger, and for which the sensing quality is at most a constant factor worse

```
Input: Graph $(\mathcal{V}, E)$, shortest path distance $d(\cdot,\cdot)$, $r > 0$, $\alpha \geq 64\,\mathrm{dim}(\mathcal{V}, E)$
Output: $(\alpha, r)$-padded decomposition $\mathcal{C} = \{\mathcal{C}_u : u \in \mathcal{U}\}$
begin
    repeat
        $\mathcal{C} \leftarrow \emptyset$; $r' \leftarrow \frac{\alpha r}{4}$; $\mathcal{U} \leftarrow$ {a random element in $\mathcal{V}$};
        while $\exists\, v \in \mathcal{V} : \forall u \in \mathcal{U}\ d(u,v) > r'$ do $\mathcal{U} \leftarrow \mathcal{U} \cup \{v\}$;
        $\pi \leftarrow$ random permutation on $\mathcal{U}$;
        $R \leftarrow$ uniform at random in $(r', 2r']$;
        foreach $u \in \mathcal{U}$ according to $\pi$ do
            $\mathcal{C}_u \leftarrow \{v \in \mathcal{V} : d(u,v) < R,$ and $\forall u' \in$
            $\mathcal{U}$ appearing earlier that $u$ in $\pi$ , $d(u',v) \geq R\}$;
        end
    until at least $\frac{1}{2}$ nodes r-padded ;
end
```

Algorithm 2: Algorithm for computing padded decompositions.

than the optimal solution. The proof can be found in the Appendix. While the actual guarantee of our algorithm holds in expectation, running the algorithm a small (polynomial) number of times will lead to appropriate solutions with arbitrarily high probability. Details on this procedure can be found in Section 8. In the rest of this section, we flesh out the details of the algorithm, giving more technical insight and intuition about the performance of our approach.

6.1 Padded decompositions

To exploit the locality property, we would like to decompose our space into "well-separated" clusters; loosely, an r-padded decomposition is a way to do this so that most vertices of $\mathcal{V}$ lie in clusters $\mathcal{C}_i$ that are at least r apart. Intuitively, *padded decompositions* allow us to split the original placement problem into approximately independent placement problems, one for each cluster $\mathcal{C}_i$. This padding and the locality property of the objective function F guarantee that, if we compute selections $\mathcal{A}_1$,..., $\mathcal{A}_m$ for each of the m clusters separately, then it holds that $F(\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_m) \geq \gamma \sum_i F(\mathcal{A}_i)$, i.e., we only lose a constant factor. An example is presented in Figure 4(a).

If we put all nodes into a single cluster, we obtain a padded decomposition that is not very useful. To exploit our locality property, we want clusters of size about r that are at least r apart. It is difficult to obtain separated clusters of size exactly r, but padded decompositions exist for arbitrary graphs for cluster sizes a constant α larger, where α is $\Omega(\mathrm{dim}(\mathcal{V}, E))$ (Gupta et al., 2003). We want small clusters, since we can then ignore communication cost within each cluster.

Formally, an (α, r)-padded decomposition is a probability distribution over partitions of $\mathcal{V}$ into *clusters* $\mathcal{C}_1, \ldots, \mathcal{C}_m$, such that:

(i) Every cluster $\mathcal{C}_i$ in the partition is guaranteed to have bounded diameter, i.e., $\mathrm{diam}(\mathcal{C}_i) \leq \alpha r$.

(ii) Each node $s \in \mathcal{V}$ is r-padded in the partition with probability at least ρ. (A node s is *r-padded* if all nodes t at distance at most r from s are contained in the same cluster as s.)

The parameter ρ can be chosen as a constant (in our implementation, $\rho = \frac{1}{2}$). In this paper, we use the term padded decomposition to refer both to the distribution, as well as samples from the distribution, which can be obtained efficiently using Algorithm 2 (Gupta et al., 2003). In pSPIEL, for a fixed value of the locality parameter r, we gradually increase α, stopping when we achieve a partition, in which at least half the nodes are r-padded. This rejection sampling is the only randomized part of our algorithm, and, in expectation, the number of required samples is polynomial.

Our algorithm strips away nodes that are not r-padded, suggesting a risk of missing informative locations. The following Lemma proves that we will not lose significant information in expectation.

Lemma 2. *Consider a submodular function $F(\cdot)$ on a ground set $\mathcal{V}$, a set $\mathcal{B} \subseteq \mathcal{V}$, and a probability distribution over subsets $\mathcal{A}$ of $\mathcal{B}$ with the property that, for some constant ρ, we have $\Pr[v \in \mathcal{A}] \geq \rho$ for all $v \in \mathcal{B}$. Then $\mathbb{E}[F(\mathcal{A})] \geq \rho F(\mathcal{B})$.* □

The proof of this Lemma appears in the Appendix. Let $\mathcal{A}^*$ be the optimal solution for the covering or maximization problem, and let $\mathcal{A}^*_r$ denote a subset of nodes in $\mathcal{A}^*$ that are r-padded. Lemma 2 proves that, in expectation, the information provided by $\mathcal{A}^*_r$ is at most a constant factor ρ worse than $\mathcal{A}^*$. Since the cost of collecting data from $\mathcal{A}^*_r$ is no larger than that of $\mathcal{A}^*$, this lemma shows that our padded decomposition preserves near-optimal solutions.

6.2 The greedy algorithm

After having sampled a padded decomposition, we run the greedy algorithm as presented in Algorithm 1 on the r-padded nodes in each cluster $\mathcal{C}_i$, with k set to n_i, the number of padded elements in cluster $\mathcal{C}_i$. Let us label the nodes as $g_{i,1}$, $g_{i,2}$, …, g_{in_i} in the order they are chosen by the greedy algorithm, and let $\mathcal{C}_{i,j} = \{g_{i,1}, \dots, g_{i,j}\}$ denote the greedy set after iteration j. From Krause et al. (2007) we know that each set $\mathcal{C}_{i,j}$ is at most a factor $(1 - 1/e)$ worse than the optimal set of j padded elements in that cluster. Furthermore, from (r, γ)-locality and using the fact that the nodes are r-padded, we can prove that

$$F(\mathcal{C}_{1,j_1} \cup \dots \cup \mathcal{C}_{m,j_m}) \geq \gamma \textstyle\sum_{k=1}^m F(\mathcal{C}_{k,j_k}) \geq \gamma \left(1 - \frac{1}{e}\right) \sum_{k=1}^m F(\mathcal{C}^*_{k,j_k})$$

for any collection of indices $j_1, \dots, j_m$, where $\mathcal{C}^*_{k,j_k}$ denotes the optimal selection of j_k nodes within cluster k.

6.3 The modular approximation graph $\mathcal{G}'$

In step 3), pSPIEL creates the auxiliary *modular approximation graph* (MAG) $\mathcal{G}'$ from $\mathcal{G}$. Intuitively, this MAG will approximate $\mathcal{G}$, such that running the Quota-MST algorithm on it will decide how many nodes should be picked from each cluster. The nodes of $\mathcal{G}'$ are the greedy sets $\mathcal{C}_{i,j}$. The greedy sets for cluster i are arranged in a chain with edge $e_{i,j}$ connecting $\mathcal{C}_{i,j}$ and $\mathcal{C}_{i,j+1}$ for every i and j. For a set of nodes $\mathcal{B}$, if $\mathrm{c}_{MST}(\mathcal{B})$ is the cost of a minimum spanning tree (MST) connecting the nodes in $\mathcal{B}$ by their shortest paths, the weight of $e_{i,j}$ in $\mathcal{G}'$ is the difference in costs of the MSTs of $\mathcal{C}_{i,j}$ and $\mathcal{C}_{i,j+1}$ (or 0 if this difference becomes negative), i.e., $\mathrm{c}(e_{i,j}) = \max\left[\mathrm{c}_{MST}(\mathcal{C}_{i,j+1}) - \mathrm{c}_{MST}(\mathcal{C}_{i,j}), 0\right]$. We also associate a "reward" $\mathrm{reward}(\mathcal{C}_{i,j}) = F(\mathcal{C}_{i,j}) - F(\mathcal{C}_{i,j-1})$ with each node, where $F(\mathcal{C}_{i,0}) \triangleq 0$. Note that, by telescopic sum, the total reward of the first k elements in chain i is $F(\mathcal{C}_{i,k})$, and the total cost of the edges connecting them is $\mathrm{c}_{MST}(\mathcal{C}_{i,k})$, which is at most 2 times the the cost of a minimum Steiner tree connecting the nodes in $\mathcal{C}_{i,k}$ in the original graph $\mathcal{G}$. By property (i) of the padded decomposition, $\mathrm{c}_{MST}(\mathcal{C}_{i,k}) \leq \alpha\ r\ k$. By associating these rewards with each node, we define a *modular* set function F' on $\mathcal{G}'$, such that for a set $\mathcal{B}$ of nodes in $\mathcal{G}'$, its value $F'(\mathcal{B})$ is the sum of the rewards of all elements in $\mathcal{B}$. Figure 4(b) presents an example of a chain associated with cluster 1 in Figure 4(a). Additionally, we connect every pair of nodes $\mathcal{C}_{i,1}, \mathcal{C}_{j,1}$ with an edge with cost being the shortest path distance between $g_{i,1}$ and $g_{j,1}$ in $\mathcal{G}$. This fully connected subgraph is called the *core* of $\mathcal{G}'$. Figure 4(c) presents the modular approximation graph associated with the padded decomposition of Figure 4(a).

6.4 Solving the covering and maximization problems in $\mathcal{G}'$

The modular approximation graph $\mathcal{G}'$ reduces the problem of optimizing a submodular set function in $\mathcal{G}$ to one of optimizing a *modular* set function F' (where the value of a set is the sum of rewards of its elements) in $\mathcal{G}'$ to minimize communication costs. This is a well studied problem, and constant factor approximation algorithms have been found for the covering and maximization problems. The (rooted) *Quota-MST* problem asks for a minimum weight tree $\mathcal{T}$ (with a specified root), in which the sum of rewards exceeds the specified quota. Conversely, the *Budget-MST* problem desires a tree of maximum reward, subject to the constraint that the sum of edge costs is bounded by a budget. The best known approximation factors for these problems is 2 for rooted Quota-MST (Garg, 2005), and $3 + \varepsilon$ (for any $\varepsilon > 0$) for unrooted Budget-MST (Levin, 2004). We can use these algorithms to get an approximate solution for the covering and maximization problems in $\mathcal{G}'$. From Section 6.3, we know that it suffices to decide which chains to connect, and how deep to descend into each chain; any such choice will give a subtree of $\mathcal{G}'$. To find this tree, we consider all $\mathcal{C}_{i,1}$ for each i as possible roots, and choose the best tree as an approximate solution. (For the Budget-MST problem, we only have an unrooted algorithm, but we can use the structure of our modular approximation graph to get an approximately optimal solution.) Figure 4(d) illustrates such a Quota-MST solution.

6.5 Transferring the solution from $\mathcal{G}'$ back to $\mathcal{G}$

The Quota- or Budget-MST algorithms select a tree $\mathcal{T}'$ in $\mathcal{G}'$, which is at most a constant factor worse than the optimal such tree. We use this solution $\mathcal{T}'$ obtained for $\mathcal{G}'$ to select a tree $\mathcal{T} \subseteq \mathcal{G}$: For every cluster i, if $\mathcal{C}_{i,j} \in \mathcal{T}'$ we mark $g_{i,1}, \ldots, g_{i,j}$ in $\mathcal{G}$. We then select $\mathcal{T}$ to be an approximately optimal Steiner tree connecting all marked nodes in $\mathcal{G}$, obtained, e.g., by computing an MST for the fully connected graph over all marked vertices, where the cost of an edge between s and t is the shortest path distance between these nodes in $\mathcal{G}$. This tree $\mathcal{T}$ is the approximate solution promised in Theorem 1. (Figure 4(e) presents the expansion of the Quota-MST from Figure 4(d).)

6.6 Additional implementation details

pSPIEL relies heavily on the monotonic submodularity and locality assumptions. In practice, since we may not know the constants r and γ, we run the algorithm multiple times with different choice for r. Since the algorithm is randomized, we repeat it several times to achieve a good solution with high probability. Finally, since we do not know γ, we cannot directly specify the desired quota when solving the covering problem. To alleviate all these intricacies, we use the following strategy to select a good placement: For a fixed number of iterations, randomly sample an r between 0 and the diameter of $\mathcal{G}$. Also sample a quota Q between 0 and $Q_{\max}$, the maximum submodular function value achieved by the unconstrained greedy algorithm. Run pSPIEL with these parameters r and Q, and record the actual placement, as well as the communication cost and sensing quality achieved by the proposed placement. After N iterations, these values result in a cost-benefit curve, which can be used to identify a good cost-benefit tradeoff as done in Section 7.

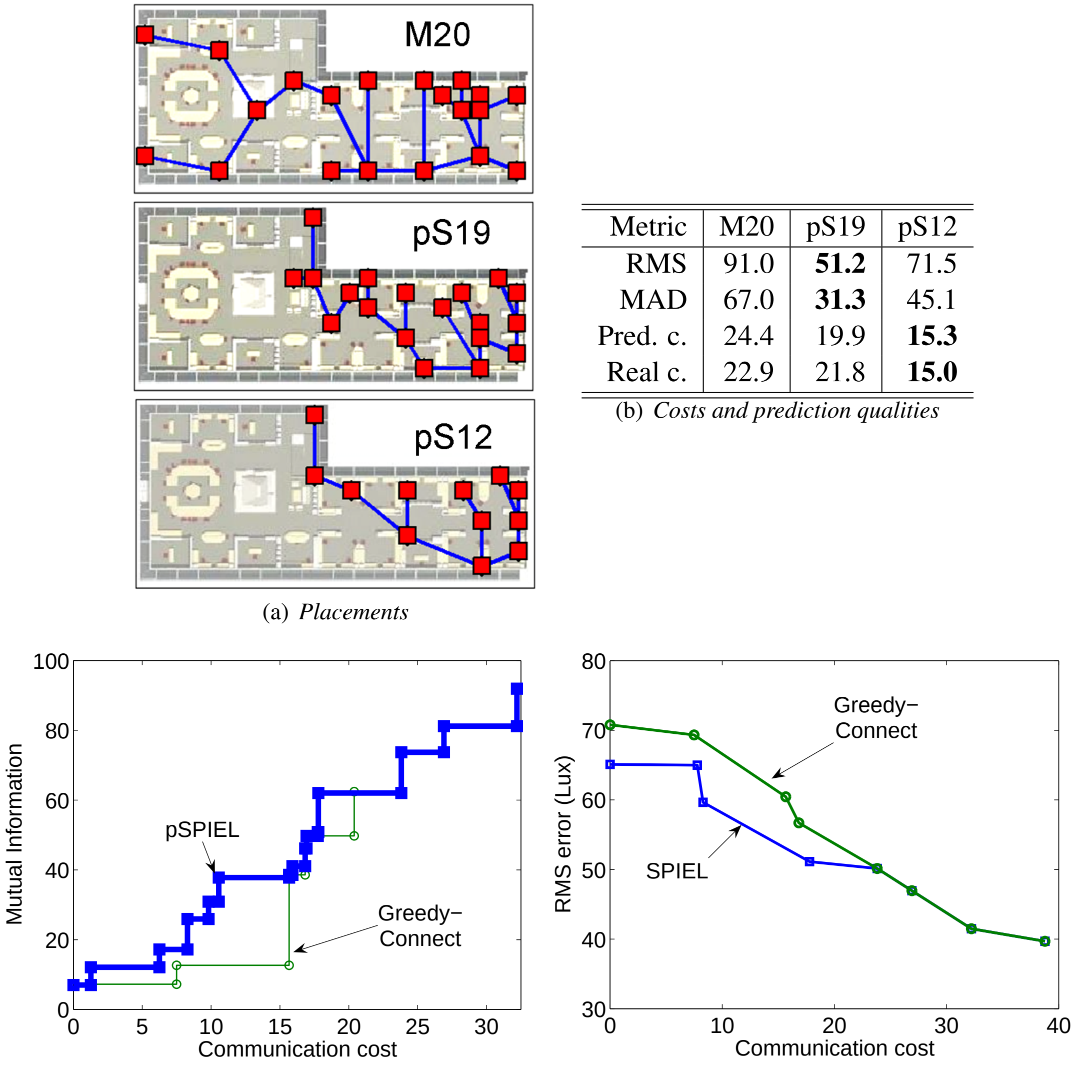
Also, note that a key step of pSPIEL is to run the greedy algorithm in each cluster. Using the technique of *lazy evaluations*, originally proposed by Minoux (1978) and applied to mutual information by Krause et al. (2007), this step can often be drastically sped up.

7 Experiments

In order to evaluate our method, we computed sensor placements for three real-world problems: Indoor illumination measurement, the temperature prediction task as described in our running example, and the prediction of precipitation in the United States' Pacific Northwest.

System implementation We developed a complete system implementation of our sensor placement approach, based on Tmote Sky motes. The data collection from the pilot deployment is based on the TinyOS SurgeTelos application, which we extended to collect link quality information. Once per epoch, every sensor sends out a broadcast message containing its unique identifier. Upon receipt of these messages, every sensor will compile a bitstring, indicating from which neighbor it has heard in the current epoch. This transmission log information will then be transmitted, along with the current sensor readings, via multi-hop routing to the base station. After enough data has been collected, we learn GP models for sensing quality and communication cost, which are subsequently used by the pSPIEL algorithm. Our implementation of pSPIEL uses a heuristically improved approximate k-MST algorithm as described by Johnson et al. (2000). Using pSPIEL, we generate multiple placements and plot them in a trade-off curve as described in Section 6.6. We then identify an appropriate trade-off by selecting good placements from this trade-off curve.

Proof-of-concept study As a proof-of-concept experiment, we deployed a network of 46 Tmote Sky motes in the Intelligent Workplace at CMU. As a baseline deployment, we selected 20 locations (M20) that seemed to capture the overall variation in light intensity. After collecting the total solar radiation data for 20 hours, we learned GP models, and used pSPIEL to propose a placement of 19 motes (pS19). Figure 5(a) shows the 20 and 19 motes deployments. After deploying the competing placements, we collected data for 6 hours starting at 12 PM and compared the prediction accuracy for all placements, on validation data from 41 evenly distributed motes. Figure 5(b) presents the results. Interestingly, the proposed placement (pS19) drastically reduces the prediction error by about 50%. This reduction can be explained by the fact that there are two components in lighting: natural and artificial. Our baseline deployment placed sensors spread throughout the environment, and in many intuitive locations near the windows. On the other hand, pSPIEL decided not to explore the large western area, a part of the lab that was not occupied during the night, and thus



(a) *Placements*

Metric	M20	pS19	pS12
RMS	91.0	**51.2**	71.5
MAD	67.0	**31.3**	45.1
Pred. c.	24.4	19.9	**15.3**
Real c.	22.9	21.8	**15.0**

(b) *Costs and prediction qualities*

(c) *Cost-benefit for light data*

(d) *RMS error for light data*

Figure 5: Experimental results. (a) shows the expert placement (M20) as well as two placements proposed by pSPIEL, (pS19) and (pS12). (b) presents root-mean-squares (RMS) and mean-absolute-deviation (MAD) prediction errors for the manual placement and two placements from pSPIEL. (c) compares the cost-benefit tradeoff curves for the light data GP on a 187 points grid. (d) compares the root-mean-squares error for the light data.

had little fluctuation with artificial lighting. Focusing on the eastern part, pSPIEL was able to make sufficiently good natural light predictions throughout the lab, and better focus of the sources of variation in artificial light. We repeated the evaluation for a 12 motes subsample (pS12, Figure 5(a)), also proposed by pSPIEL, which still provides better prediction than the manual placement of 20 nodes (M20), and significantly lower communication cost. We also compared the predicted communication cost using the GPs with the measured communication cost. Figure 5(b) shows that the prediction matches well to the measurement. Figs. 5(c) and 5(d) show that pSPIEL outperforms the Greedy heuristic explained below, both in the sensing quality and communication cost tradeoff and in predictive RMS error.

Indoor temperature measurements In our second set of experiments, we used an existing deployment (*c.f.*, Figure 1(a)) of 52 wireless sensor motes to learn a model for predicting temperature and communication cost in a building. After learning the GP models from five days of data, we used pSPIEL to propose improved sensor placements. We compared pSPIEL to three heuristics, and—for small problems—with the optimal algorithm which exhaustively searches through all possible deployments. The first heuristic, *Greedy-Connect*, runs the unconstrained greedy algorithm (Algorithm 1), and then connects the selected sensors using a Steiner tree approximation. The second heuristic, *Distance-weighted Greedy*, is inspired by an algorithm that provides near-optimal solutions to the Quota-MST problem (Awerbuch et al., 1999). This heuristic initially starts with all nodes in separate clusters, and iteratively merges – using the shortest path – clusters maximizing the following greedy criterion:

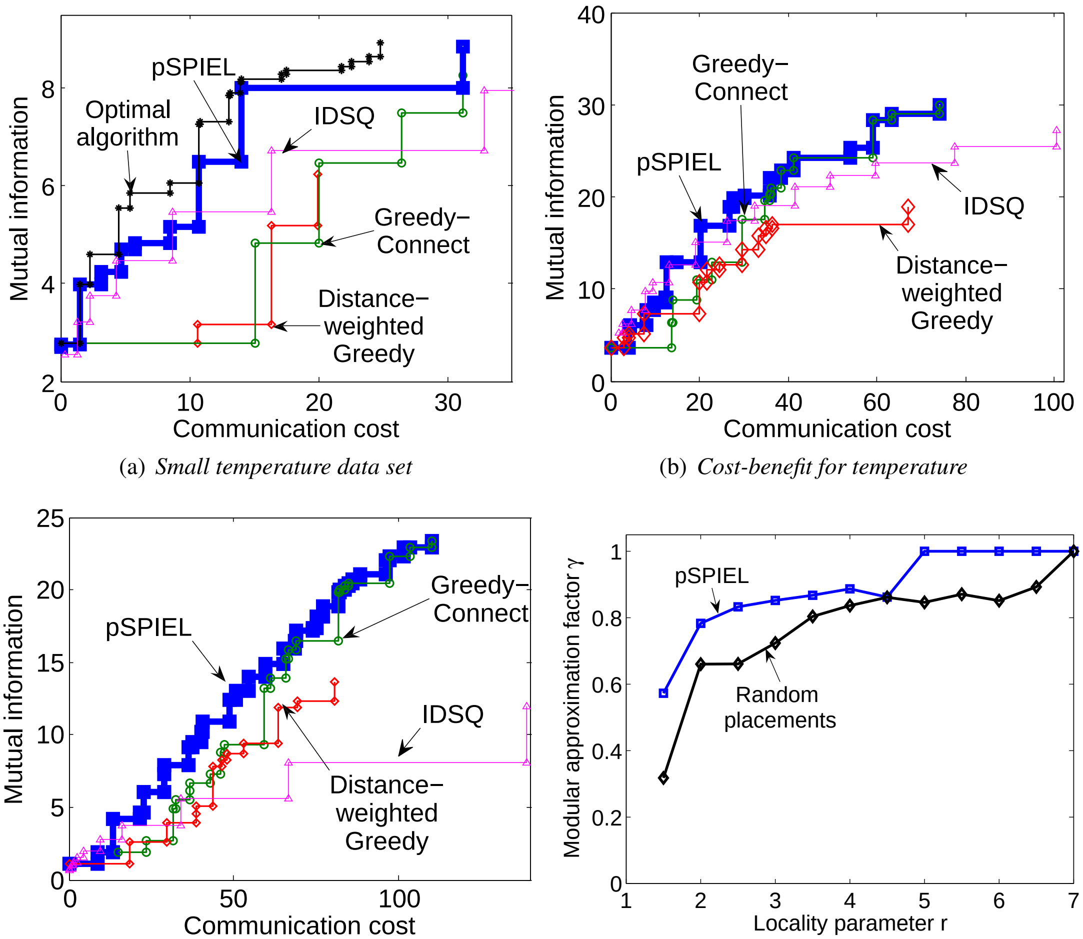
$$\text{gain}(\mathcal{C}_1, \mathcal{C}_2) = \frac{\min_{i \in 1,2}(F(\mathcal{C}_1 \cup \mathcal{C}_2) - F(\mathcal{C}_i))}{\text{dist}(\mathcal{C}_1, \mathcal{C}_2)}.$$

The intuition for this greedy rule is that it tries to maximize the benefit-cost ratio for merging two clusters. Since it works near-optimally in the modular case, we would hope it performs well in the submodular case also. The algorithm stops after sufficiently large components are generated (*c.f.*, Awerbuch et al., 1999). We also compare against the *Information Driven Sensor Querying (IDSQ)* approach (Zhao and Guibas, 2004). In IDSQ, a leader node s_1 is elected that is able to sense the monitored phenomenon. Subsequently, sensors s_j are greedily selected for communication with s_1, to maximize a linear combination of incremental utility and communication cost to the leader node,

$$s_j \in \underset{s}{\operatorname{argmax}}\, \alpha[F(\{s_1, \dots, s_{j-1}, s\}) - F(\{s_1, \dots, s_{j-1}\})] - (1-\alpha)\,\mathrm{c}(\{s_1, s\}).$$

Hereby, α is a parameter varying between 0 and 1 controlling the cost-benefit tradeoff. Selection stops when no positive net-benefit can be achieved. Since all nodes can sense the phenomenon, we elect the leader node s_1 at random and report expected cost and sensing quality over 10 random trials.

Figure 6(a) compares the performance of pSPIEL with the other algorithms on a small problem with only 16 candidate locations. We used the empirical covariance and link qualities measured from 16 selected sensors. In this small problem, we could explicitly compute the optimal solution by exhaustive search. Figure 6(a) indicates that the performance of pSPIEL is significantly closer to the optimal solution than any of the other approaches. Figure 6(b) presents a comparison of the algorithms for selecting placements on a 10×10 grid. We used our GP models to predict the covariance and communication cost for this discretization. From Figure 6(b) we can see that for very low quotas (less than 25% of the maximum), the algorithms performed very similarly. Also, for very large quotas (greater than 80%), pSPIEL does not significantly outperform not *Greedy-Connect*, since, when the environment is densely covered, communication is not an issue. In fact, if the information quota requires a very dense deployments, the padded decomposition tends to strip away many nodes, leading pSPIEL to increase the locality constant r, until r is large enough to include



(a) *Small temperature data set*

(b) *Cost-benefit for temperature*

(c) *Cost-benefit for precipitation*

(d) (r, γ) *for temperature data*

Figure 6: Experimental results. (a) compares trade-off curves for a small subset of the temperature data. (b) shows tradeoff curves for the temperature GPs on a 10x10 grid. (c) compares tradeoffs for precipitation data from 167 weather stations. (d) compares the locality parameter r and the loss γ incurred by the modular approximation for the temperature GPs.

all nodes are in a single cluster. In this case, pSPIEL essentially reverts back to the *Greedy-Connect* algorithm. In the important region between 25% and 80% however, pSPIEL clearly outperforms the heuristics. Our results also indicate that in this region the steepest drop in out-of-sample root mean squares (RMS) prediction accuracy occurs. This region corresponds to placements of approximately $10-20$ sensors, an appropriate number for the target deployment Figure 1(a).

In order to study the effect of the locality parameter r, we generated padded decompositions for increasing values of r. For random subsets of the padded nodes, and for placements from pSPIEL, we then compared the modular approximation, i.e., the sum of the local objective values per cluster, with the mutual information for the entire set of selected nodes. As r increases to values close to 2, the approximation factor γ drastically increases from .3 to .7 and then flattens as r encompasses the the entire graph $\mathcal{G}$. This suggests that the value $r = 2$ is an appropriate choice for the locality parameter, since it only incurs a small approximation loss, but guarantees small diameters of the padded clusters, thereby keeping communication cost small. For placements proposed by pSPIEL, the approximation factor is even better.

Precipitation data In our third application, our goal was to place sensors for predicting precipitation in the Pacific North-West. Our data set consisted of daily precipitation data collected from 167 regions during the years 1949–1994 (Widmann and Bretherton, 1999). We followed the preprocessing from Krause et al. (2007). Since we did not have communication costs for this data set, we assumed that the link quality decayed as the inverse square of the distance, based on physical considerations. Figure 6(c) compares the sensing quality – communication cost tradeoff curves for selecting placements from all 167 locations. pSPIEL outperforms the other approaches up to very large quotas.

8 Robust Sensor Placements

In Section 7, we have seen that optimized placements can lead to much higher prediction accuracy and lower communication cost as compared to manual placements. However, such optimization can lead to negative effects if the model that the optimization is based on changes. For example, in our proof-of-concept study, it is conceivable that the building usage patterns change, and the western part of the building becomes occupied. In this case, the optimized placement will fail to capture important variations in light distribution. Intuitively, the manual placement (M20) should be able to capture such change of the environment better, as the sensors are more uniformly spread out. This intuitive assessment comes from our prior assumptions that, since lights spreads uniformly over space, a regularly-spaced distributed placement should be quite informative.

How can we place sensors that perform well both according to the current state of the world, as well as to possible future changes? One possibility is to require the sensor placement to perform well both according to our prior assumptions (i.e., favoring uniform placements) and to the data we collected so far. We can formalize this idea by defining two separate sensing quality functions, F_1 and F_2. $F_1(\mathcal{A})$ measures the informativeness of placement $\mathcal{A}$ under the isotropic prior. F_2 measures the informativeness according to the collected data, as described before. Assuming a priori that the phenomenon will always uniformly spread in the environment, we could choose F_1 to be the mutual information of an isotropic Gaussian process, as in Equation (7). Optimizing according to F_1 only would lead to sensor placements that are (asymptotically) distributed on a regular grid, and we would hence expect such placements to be robust against changes in the environment. F_2 is the mutual information according to the complex, data-dependent, nonstationary Gaussian process as considered in the earlier parts of this paper. Optimizing for F_2 would lead to placements that exploit the correlations in the data collected from the pilot deployment. Based on these two objective functions, we would then like to find a placement $\mathcal{A}$ that jointly optimizes F_1 and F_2, i.e., which is both robust, and exploits correlations estimated from data.

More generally, we would like to solve the robust optimization problem

$$\min_{\mathcal{A}} \mathrm{c}(\mathcal{A}) \text{ such that for all } i, F_i(\mathcal{A}) \geq Q, \tag{9}$$

where $F_1, \dots, F_m$ is a collection of monotonic submodular functions. Note that, unlike problem (1), in problem (9) there are now multiple submodular constraints $F_1(\mathcal{A}) \geq Q, \dots, F_m(\mathcal{A}) \geq Q$. The following key idea allows us to reduce problem (9) to problem (1): For each function F_i, define a new truncated objective function

$$\widehat{F}_{i,Q}(\mathcal{A}) = \min\{F_i(\mathcal{A}), Q\}.$$

It holds that whenever F_i is monotonic and submodular, $\widehat{F}_{i,Q}$ is monotonic and submodular as well (Fujito, 2000). As a nonnegative linear combination of monotonic submodular functions, the function

$$\overline{F}_Q(\mathcal{A}) = \frac{1}{m}\sum_i \widehat{F}_{i,Q}(\mathcal{A})$$

is monotonic submodular as well. Furthermore, it holds that $\overline{F}_Q(\mathcal{A}) = Q$ if and only if $F_i(\mathcal{A}) \geq Q$ for all i. Hence, instead of problem (9), we can equivalently solve[3]

$$\min_{\mathcal{A}} \mathrm{c}(\mathcal{A}) \text{ such that } \overline{F}_Q(\mathcal{A}) \geq Q,$$

which is an instance of problem (1). However, we cannot readily apply pSPIEL to this problem, since it is only guaranteed to return a solution such that, in expectation, $\overline{F}_Q(\mathcal{A}) \geq \beta Q$, for $\beta = (1-1/e)\gamma/2$ (from Theorem 1). Unfortunately, $\overline{F}_Q(\mathcal{A}) \geq \beta Q$ does not guarantee that $F_i(\mathcal{A}) \geq \beta Q$ for each i.

However, we can nevertheless use pSPIEL to solve problem (9). We first present an overview of our algorithm, and then discuss the details in Section 8.1.

1. We first convert pSPIEL into an algorithm, for which Theorem 1 holds not just in expectation, but with high probability. For arbitrary $\delta > 0$, this new algorithm will be guaranteed to provide, with probability at least $1 - \delta$, a solution $\mathcal{A}$ such that $\overline{F}_Q(\mathcal{A}) \geq \beta Q/2$.
2. Call $Q_{togo} = Q - \overline{F}_Q(\mathcal{A})$ the remaining quota that still needs to be covered. When applying pSPIEL once, $Q_{togo} \leq Q(1 - \beta/2)$. We show how we can iteratively apply pSPIEL to a modified objective function to obtain larger and larger sets $\mathcal{A}$ such that after k iterations $Q_{togo} \leq Q(1 - \beta/2)^k$. Hence, after logarithmically many iterations, $Q_{togo} \leq \varepsilon/m$. This implies that $F_i(\mathcal{A}) \geq Q(1 - \varepsilon)$ for all i.

8.1 Algorithm details

We first need to turn pSPIEL into an algorithm that solves problem (1) with high probability $1 - \delta$. Let M be an upper bound on the optimal value. Then each run of pSPIEL returns a solution $\mathcal{A}_i$ with $0 \leq F(\mathcal{A}_i) \leq M$.

Lemma 3. *We need*

$$N = \left\lceil \frac{1}{2}\left(\frac{2M}{Q\beta}\right)^2 \log\frac{1}{\delta}\right\rceil$$

samples to guarantee a sample $\mathcal{A}_i$ with $\overline{F}_Q(\mathcal{A}_i) \geq \frac{\beta Q}{2}$ with probability $1 - \delta$.

[3] A similar construction was used by Krause et al. (2007) to develop the SATURATE algorithm for robust optimization of submodular functions.

In order to guarantee that $\min_i F_i(\mathcal{A}_i) \geq (1-\varepsilon)Q$, we follow the following strategy: Let $F^{(1)} = \overline{F}_Q$. We invoke random sampling of pSPIEL applied to $F^{(1)}$ until we obtain a solution $\mathcal{A}_1$ such that $F^{(1)}(\mathcal{A}_1) \geq \frac{\beta Q}{2}$. We then define a new monotonic submodular function, $F^{(2)}(\mathcal{A}) = \overline{F}_Q(\mathcal{A} \cup \mathcal{A}_1) - \overline{F}_Q(\mathcal{A}_1)$. $F^{(2)}$ is called a *residual submodular function.* We then repeatedly run pSPIEL to obtain a solution $\mathcal{A}_2$ such that $F^{(2)}(\mathcal{A}_2) \geq \frac{\beta(Q-F_1(\mathcal{A}_1))}{2}$. After k steps, we define the function

$$F^{(k+1)}(\mathcal{A}) = \overline{F}_Q(\mathcal{A} \cup \mathcal{A}_1 \cup \cdots \cup \mathcal{A}_k) - \overline{F}_Q(\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_k),$$

and use pSPIEL to obtain a solution such that

$$F^{(k+1)}(\mathcal{A}_{k+1}) \geq \frac{\beta(Q - \overline{F}_Q(\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_k))}{2}.$$

Note that after k steps, it holds that $(Q - \overline{F}_Q(\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_k) \leq Q(1 - \frac{1}{2}\beta)^k$, and hence after

$$k = \left\lceil \frac{\log \frac{m}{\varepsilon}}{\log \frac{2}{2-\beta}} \right\rceil$$

iterations it holds that

$$\overline{F}_q(\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_k) \geq Q(1 - \frac{\varepsilon}{m}),$$

and hence $F_i(\mathcal{A}_1 \cup \cdots \cup \mathcal{A}_k) \geq Q(1-\varepsilon)$. We choose δ small enough to apply the union bound over all k trials. Algorithm 3 presents pseudo-code for our approach.

We summarize our analysis in the following Theorem:

Theorem 4. *Given a graph $\mathcal{G} = (\mathcal{V}, E)$, constants ε, δ, Q and (r,γ)-local monotone submodular functions $F_1, \ldots, F_m$ bounded above by M, we can find a tree $\mathcal{T}$ with cost*

$$\mathrm{c}(\mathcal{T}) = \mathcal{O}(r \dim(\mathcal{V}, E)) \times \left\lceil \frac{\log \frac{m}{\varepsilon}}{\log \frac{2}{2-\Omega(\gamma)}} \right\rceil \times \ell^*,$$

spanning a set $\mathcal{A}$ with $F_i(\mathcal{A}) \geq Q(1-\varepsilon)$ for all i. The algorithm is randomized and runs in expected time polynomial in the size of the problem instance and polynomial in $\frac{M}{Q}$. □

Hence, for an arbitrary $\varepsilon > 0$ we can, in expected polynomial time, find a sensor placement with sensing quality $F_i(\mathcal{A}) \geq Q(1-\varepsilon)$. The cost of the solution $\mathrm{c}(\mathcal{T})$ grows logarithmically in $\frac{m}{\varepsilon}$ which depends on the number m of objective functions.

8.2 Experiments

We use our robust version of pSPIEL to make the sensor placement in our proof-of-concept study more robust. We choose $F_1(\mathcal{A})$ as the mutual information obtained in an isotropic Gaussian process with kernel (7) and fixed bandwidth h. As F_2, we choose the nonstationary GP learned from the pilot deployment, as in Section 7.

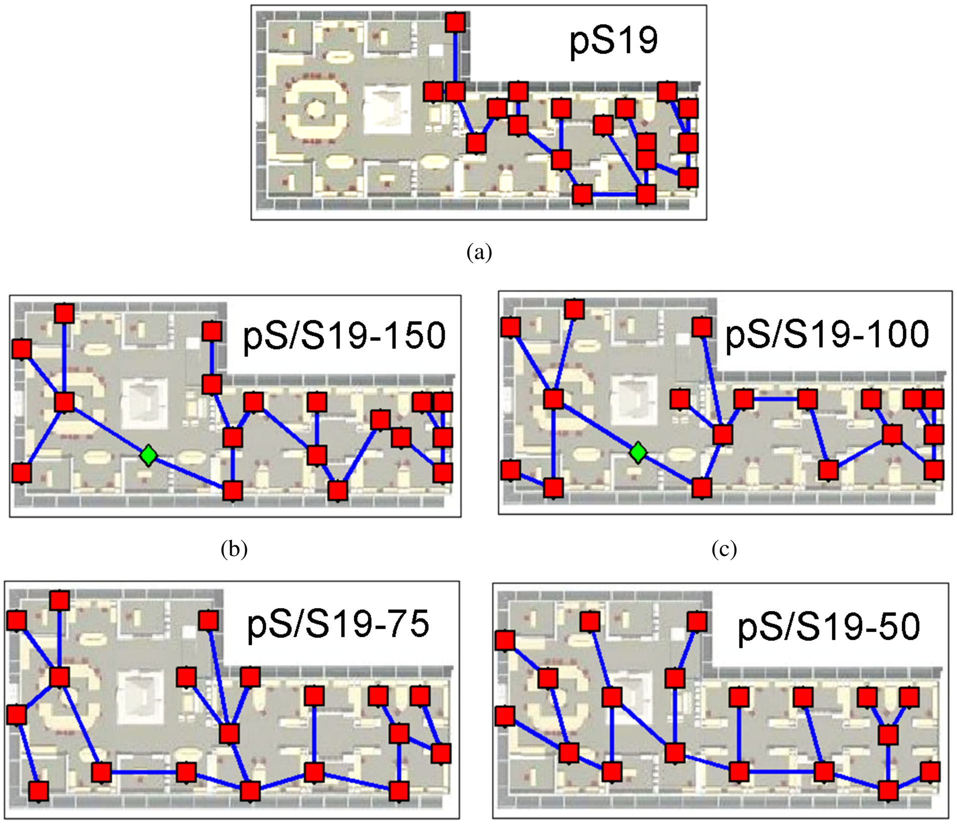
In order to model F_1 using an isotropic GP (modeling the uniform spreading of light), we need to specify the bandwidth parameter h in (7). This bandwidth parameter encodes our smoothness

```
Input: Graph $(\mathcal{V}, E)$, $F_1, \dots, F_m$, $Q$, $M$, $\beta$, $\varepsilon$, $\delta$
Output: Placement $\mathcal{A}$ such that $F_i(\mathcal{A}) \geq Q$ for all $i$ with probability $1 - \delta$.
begin
    $\overline{F}_Q(\mathcal{A}) \leftarrow \frac{1}{m} \sum_{i=1}^{m} \min\{F_i(\mathcal{A}), Q\}$;
    $F^{(1)} \leftarrow \overline{F}_Q$;
    $\mathcal{A} \leftarrow \emptyset$;
    $k \leftarrow 0$;
    while $\overline{F}_Q(\mathcal{A}) \leq Q(1 - \varepsilon/m)$ do
        $k \leftarrow k + 1$;
        $F^{(k)}(\mathcal{A}') \leftarrow \overline{F}_Q(\mathcal{A}' \cup \mathcal{A}) - \overline{F}_Q(\mathcal{A})$;
        $Q_{togo} \leftarrow Q - \overline{F}_Q(\mathcal{A})$;
        repeat
            $\mathcal{A}' \leftarrow pSPIEL(F^{(k)}, Q_{togo})$
        until $F^{(k)}(\mathcal{A}') \geq Q_{togo}\beta/2$ ;
        $\mathcal{A} \leftarrow \mathcal{A} \cup \mathcal{A}'$;
    end
end
```

Algorithm 3: Algorithm for robust sensor placements.

assumptions about the world. The smaller h, the quicker the assumed correlation decays with distance, and hence the more rough we assume the phenomenon to be. In addition, the smaller h, the more sensors we need in order to obtain a high level of mutual information. This means, that for small values of h, the uninformed sensing quality F_1 dominates the optimization. For large values of h however, the fewer sensors we need to obtain high sensing quality F_1, and hence F_2 dominates the objective value. Hence, by varying h, we can vary the amount of robustness. Figure 7 shows different sensor placements obtained by choosing an increasing bandwidth h. For small bandwidths, the placements are basically uniformly spread out over the space. For high bandwidths, the robust places resemble the non-robust placement pS19 (from Section 7).

We also compare the manual placement (M20) and the non-robust pSPIEL placement (pS19) of Section 7 with the robust solutions. For each robust placement $\mathcal{A}$, we compute $\min_i F_i(\mathcal{A})$, which measures how well the placement performs with respect to both the data-driven model F_2 and the uniform prior F_1. The placement in Figure 7(b) maximizes this score over all the robust placements, indicating that Figure 7(b) is a good compromise between the data-driven model and prior assumptions. Figure 8 compares the manual placement (M20) with both optimized placements. Note that while the robust placement obtains higher RMS error and communication cost than the non-robust placement (pS19), it still performs drastically better than the manual placement (M20). Also note that the robustness scores $\min_i F_i(\mathcal{A})$ of both the robust and the manual placement are higher than for the non-robust placement (pS19).



(a)

(b)

(c)

(d)

(e)

Figure 7: Experimental results. (a) shows the non-robust placement. (b-e) show placements optimized using Algorithm 3 where F_2 is the mutual information according to the pilot deployment, and F_1 is the mutual information w.r.t. an isotropic GP with different bandwidths: pS/S19-x refers to the result when using a bandwidth proportional to x. Notice that with decreasing bandwidths, the placements become more and more regularly-spaced.

9 Modular approximation for other combinatorial problems: Informative Path Planning

The key idea behind pSPIEL was to reduce the problem of maximizing sensing quality, a submodular function, to the problem of maximizing a *modular* function on the Modular Approximation Graph, which we can solve using existing combinatorial algorithms for modular functions. This idea of reducing a local-submodular optimization to a modular optimization is quite powerful. For example, we can use the same algorithmic idea to solve other local-submodular combinatorial optimization problems. In the following, we will discuss one such example: Applying pSPIEL for informative path planning.

Consider the setting where, instead of deploying a wireless sensor network, we use a robot to collect observations. In this case, we also want to identify locations $\mathcal{A}$ of high sensing quality, but the robot needs to travel between successive sensing location. We can model this setting by specifying a

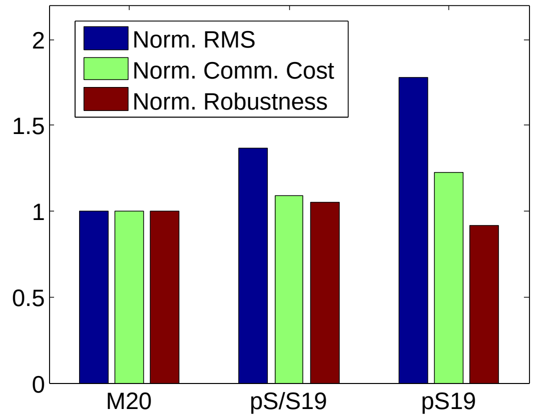


Figure 8: Experimental results comparing the manual placement with 20 sensors (left) to the robust (center) and non-robust (right) placements obtained using pSPIEL. For each placement, three bars are shown, measuring improvement in RMS error, communication cost and robustness compared to the manual solution as baseline (which is normalized to 1). Higher values are better. Both the robust and non-robust placements outperform the manual deployment in communication cost and prediction error. The robust placement also obtains higher robustness score than the manual deployment. The robustness score of the non-robust placement is even lower than that of the manual placement.

graph $\mathcal{G} = (\mathcal{V}, E)$ over the sensing locations, and our goal will be to find an informative path $\mathcal{P} = (a_1, \ldots, a_k)$ spanning the locations $\mathcal{A} \subseteq \mathcal{P}$. In this setting, rather than modeling the communication cost, the cost $\mathrm{c}(\mathcal{P})$ is the length of the path $\mathcal{P}$, i.e.,

$$\mathrm{c}(\mathcal{P}) = \sum_{i=1}^{k-1} \mathrm{c}(\{a_i, a_{i+1}\}).$$

Similarly, $\mathrm{c}(\mathcal{A})$ is the cost of the shortest path spanning nodes $\mathcal{A}$. Using this modified notion of cost, we designate specific nodes $s, t \in \mathcal{V}$ as starting and ending locations, i.e., require that $a_1 = s$ and $a_k = t$, and solve

$$\max_{\mathcal{P}} F(\mathcal{P}) \text{ subject to } \mathrm{c}(\mathcal{P}) \leq B \text{ and } \mathcal{P} \text{ is an } s - t \text{ path in } \mathcal{G} \tag{10}$$

For modular functions F, this problem is known as the $s - t$ *orienteering* problem (*c.f.*, Chekuri et al., 2008). For the more general case where F is submodular, so far for this problem only an algorithm with quasipolynomial running time was proposed by Chekuri and Pal (2005), which has been extended and used for informative path planning by Singh et al. (2007).

Using an approximate algorithm for $s - t$ orienteering on the modular approximation graph, we obtain the first polynomial time approximation algorithm for (r, γ)-local submodular orienteering.

Theorem 5. *Given a graph $\mathcal{G} = (\mathcal{V}, E)$, $s, t \in \mathcal{V}$ and an (r, γ)-local monotone submodular function F,* pSPIEL *will find an $s - t$ path $\mathcal{P}$ with cost $\mathcal{O}(r \dim(\mathcal{V}, E)) \times \ell^*$, spanning a set $\mathcal{A}$ with expected* *sensing quality* $F(\mathcal{A}) \geq \Omega(\gamma) \times F(\mathcal{A}^*)$. *The algorithm is randomized and runs in polynomial-time.* □

Singh et al. (2007) proved that any κ-approximate algorithm for submodular orienteering can be extended to an efficient $\kappa+1$-approximate algorithm for the more complex problem planning *multiple* paths (for multiple robots). This result can immediately be used to extend pSPIEL to the setting of multiple robot informative path planning.

10 Related work

In this section, we relate our approach to work in several areas.

10.1 Sensor placement to monitor spatial phenomena

The problem of selecting observations for monitoring spatial phenomena has been investigated extensively in geostatistics (*c.f.*, Cressie (1991) for an overview), and more generally (Bayesian) experimental design (*c.f.*, Chaloner and Verdinelli, 1995). Heuristics for actively selecting observations in GPs in order to achieve high mutual information have been proposed by Caselton and Zidek (1984). Submodularity has been used to analyze algorithms for placing a fixed set of sensors (Krause et al., 2007). These approaches however do not consider communication cost as done in this paper. The problem of optimally placing a small number of relay nodes to facilitate communication between sensors of a deployed network has been studied by a number of researchers (Ergen and Varaiya, 2006; Lloyd and Xue, 2007; Cheng et al., 2008). However, these approaches do not jointly optimize over the sensor placement and the deployment of relay nodes as considered in this paper.

10.2 Sensor placement under communication constraints

Existing work on sensor placement under communication constraints (Gupta et al., 2003; Kar and Banerjee, 2003; Funke et al., 2004) has considered the problem mainly from a geometric perspective: Sensors have a fixed *sensing region*, such as a disc with a certain radius, and can only communicate with other sensors that are at most a specified distance apart. In addition, it is assumed that two sensors at fixed locations can either perfectly communicate or not communicate at all. As argued in Section 1 these assumptions are problematic. Sensor selection considering both the value of information together with the cost of acquiring the information in the context of sensor networks was first formalized by Zhao et al. (2002). Their Information Driven Sensor Querying (IDSQ) approach greedily trades off sensing quality and communication cost. While their approach flexibly accommodates different sensing quality and communication cost functions, their optimization algorithm does not provide any performance guarantees. Bian et al. (2006) describe an approach for selecting sensors with submodular and supermodular utility functions, trading off utility and cost. However, their approach requires that sensors are able to send “fractional” amounts of information. While this fractional selection can be realistic in some applications, it cannot be used to optimize sensor placements: In sensor placement, a location is either selected or not selected. The first version of this paper (Krause et al., 2006) was, to the best of our knowledge, the first approach to near-optimally place sensor networks under realistic models for both the monitored phenomenon and the wireless link quality. In contrast to previous approaches, pSPIEL applies for all (r, γ)-local submodular sensing quality functions. The present version is significantly extended, providing more details as well as new empirical and theoretical results (Sections 8 and 9).

10.3 Statistical models for modeling link quality

Often, it is assumed that a transmitting node has perfect connectivity with all nodes inside a given radius and zero connectivity with nodes outside that disk (Cerpa et al., 2005; Bai et al., 2006). However, depending on how the disk radius is chosen, such disk models may not capture actual communication patterns in one particular network. In order to allow more flexibility, Cerpa et al. (2005) consider a data-driven, probabilistic link model. Like the regular disk model, it assumes connectivity is a function only of geometric distance between motes but unlike that model, it can predict a real-valued connectivity value, that is, a probability of packet reception that is not zero or one. However, their isotropic approach does not adapt to a specific environment (containing obstacles like walls, doors, furniture, etc.). Cerpa et al. (2005) also study the impact of temporal autocorrelation on routing decisions. Incorporating such temporal aspects into sensor placement optimization is an interesting avenue for further research.

In order to account for more complex behavior, physical models like radio propagation or path loss equations were obtained from real-data and describes the signal quality fall off away from a transmitting sensor (Friis, 1946; Rappaport, 2000; Zuniga and Krishnamachari, 2007). These equations can model complex communication behaviors with parameters encoding for the number of walls, the construction materials of the clutter, multipath signal effects, and microwave interference (Rappaport, 2000; Morrow, 2004). Unfortunately, this deployment-specific information can be as hard to model and obtain as the packet transmission data needed for data-driven approaches.

Our link quality model described in Section 2.2 allows to both model complex, environment dependent behavior, and is completely data driven (i.e., no deployment-specific information needs to be manually supplied). After the first version of our paper was published (Krause et al., 2006), Ertin (2007) proposed an approach for learning Gaussian Process models for link quality estimation, explicitly taking into account that fact that sensor measurements are lost (censored). Note that our pSPIEL approach can use such alternative approaches for estimating link quality as well.

10.4 Related work on submodular optimization

Problem (1) for an arbitrary integer valued monotonic submodular function F is called the polymatroid Steiner tree problem (Calinescu and Zelikovsky, 2005). Calinescu and Zelikovsky (2005) developed a polylogarithmic approximation algorithm for this problem. However, their approach does not exploit locality, and hence leads to approximation guarantees that are worse than those obtained by pSPIEL (which solves the problem for all (r, γ)-local submodular functions F) if locality is present. The submodular orienteering problem, i.e., the problem of finding a path of bounded length maximizing a submodular utility function, was first considered by Chekuri and Pal (2005), who developed an algorithm with quasipolynomial running time. While providing important theoretical insights, their approach does not scale to practical sensing problems. Singh et al. (2007) proposed a spatial decomposition approach as well as branch and bound techniques to significantly speed up the approach of Chekuri and Pal (2005). They also applied it to informative path planning in the context of environmental monitoring problems. However, their approach still has worst-case quasipolynomial running time. The approach presented in Section 9 is the first efficient (polynomial-time) algorithm for submodular orienteering, in the case where the objective function F is (r, γ)-local.

The robust sensor placement problem (9) was first studied by Krause et al. (2007) for the case of finding the best k sensor locations. In this paper, we extend their approach to more complex cost functions (such as communication cost).

11 Conclusions

We proposed a unified approach for robust placement of wireless sensor networks. Our approach uses Gaussian Processes, which can be chosen from expert knowledge or learned from an initial deployment. We propose to use GPs not only to model the monitored phenomena, but also for predicting communication costs. We presented a polynomial time algorithm – pSPIEL– selecting Sensor Placements at Informative and cost-Effective Locations. Our algorithm provides strong theoretical performance guarantees. Our algorithm is based on a new technique, the modular approximation graph, that is more general and can also be used, for example, to plan informative paths for robots. pSPIEL also applies more generally to arbitrary (r, γ)-submodular sensing quality functions, and any communication model where the cost of a sensor deployment can be formalized as the sum of edge costs connecting the sensors. We extended our pSPIEL approach to obtain sensor placements that are robust against changes in the environment. We built a complete implementation on Tmote Sky motes and extensively evaluated our approach on real-world placement problems. Our empirical evaluation shows that pSPIEL significantly outperforms existing methods.

Future work While our approach applies to a variety of sensor placement problems, there are several open questions that we leave as interesting directions for future work. For example, it would be interesting to investigate whether more general notions of communication cost can be incorporated. When taking into account interference between sensors, the communication cost does not only depend on pairwise distances between sensors, but on the density of sensors deployed in particular areas. Another interesting direction is the incorporation of temporal effects, using spatiotemporal models for the observed phenomenon. Lastly, an interesting algorithmic question is whether the assumption of locality can be relaxed.

Acknowledgements We would like to thank Adrian Perrig for providing us with motes and Vipul Singhvi for helping with the deployment. This work was supported by NSF Grant No. CNS-0509383, CNS-0625518, CNS-0932392, ANI-00331481, CCR-0120778, CCF-0448095, CCF-0729022, CCF-0325453, IIS-0329064, CNS-0403340, CCR-0122581, by the Office of Naval Research Grant N000140911044 and gifts from Intel Corporation and Microsoft Corporation. Andreas Krause was partly supported by a Microsoft Research Graduate Fellowship. Anupam Gupta and Carlos Guestrin were partly supported by Alfred P. Sloan Fellowships. Carlos Guestrin was also partly supported by an IBM Faculty Fellowship and an ONR Young Investigator Award. Jon Kleinberg was supported by a David and Lucile Packard Foundation Fellowship; work done in part while on sabbatical leave at CMU.

References

Awerbuch, B., Azar, Y., Blum, A., and Vempala, S. 1999. New approximation guarantees for minimum-weight k-trees and prize-collecting salesmen. *SIAM J. Computing 28*, 254–262.

Bai, X., Kumar, S., Yun, Z., Xuan, D., and Lai, T. H. 2006. Deploying wireless sensors to achieve both coverage and connectivity. In *Proc. of the ACM International Symposium on Mobile Ad Hoc Networks (MobiHoc)*. Florence, Italy.

Bian, F., Kempe, D., and Govindan, R. 2006. Utility based sensor selection. In *IPSN ’06: Proceedings of the fifth international conference on Information processing in sensor networks*.

Calinescu, G. and Zelikovsky, A. 2005. The polymatroid steiner tree problems. *Journal of Combinatorial Optimization 3*, 281–294.

Caselton, W. F. and Zidek, J. V. 1984. Optimal monitoring network designs. *Statistics and Probability Letters 2,* 4, 223–227.

Cerpa, A., Wong, J. L., Kuang, L., Potkonjak, M., and Estrin, D. 2005. Statistical model of lossy links in wireless sensor networks. In *Proc. of ACM/IEEE International Conference on Information Processing in Sensor Networks*.

Cerpa, A., Wong, J. L., Potkonjak, M., and Estrin, D. 2005. Temporal properties of low power wireless links: modeling and implications on multi-hop routing. In *MobiHoc ’05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*. ACM, New York, NY, USA, 414–425.

Chaloner, K. and Verdinelli, I. 1995. Bayesian experimental design: A review. *Statistical Science 10,* 3 (Aug.), 273–304.

Chekuri, C., Korula, N., and Pal, M. 2008. Improved algorithms for orienteering and related problems. In *Proc. of ACM-SIAM Symposium on Discrete Algorithms*.

Chekuri, C. and Pal, M. 2005. A recursive greedy algorithm for walks in directed graphs. In *Annual IEEE Symposium on. Foundations of Computer Science*. 245–253.

Cheng, X., Du, D.-Z., Wang, L., and Xu, B. 2008. Relay sensor placement in wireless sensor networks. *Wireless Networks 14,* 3 (June), 347–355.

Cressie, N. A. 1991. *Statistics for Spatial Data*. Wiley.

CSATO, L., FOKUE, E., OPPER, M., SCHOTTKY, B., AND WINTHER, O. 2000. Efficient approaches to gaussian process classification. In *Advances in Neural Information Processing Systems*.

DAS, A. AND KEMPE, D. 2008. Algorithms for subset selection in linear regression. In *Proc. of ACM Symposium on Theory of Computing*.

DE COUTO, D. S. J., AGUAYO, D., BICKET, J., AND MORRIS, R. 2003. A high-throughput path metric for multi-hop wireless routing. In *Proceedings of the 9th ACM International Conference on Mobile Computing and Networking (MobiCom '03)*. San Diego, California.

DESHPANDE, A., GUESTRIN, C., MADDEN, S., HELLERSTEIN, J., AND HONG, W. 2004. Model-driven data acquisition in sensor networks. In *Proc. of International Conference on Very Large Data Bases*.

ERGEN, S. C. AND VARAIYA, P. 2006. Optimal placement of relay nodes for energy efficiency in sensor networks. In *IEEE International Conference on Communications*.

ERTIN, E. 2007. Gaussian process models for censored sensor readings. In *IEEE/SP 14th Workshop on Statistical Signal Processing*.

FRIIS, H. 1946. A note on a simple transmission formula. *Proc IRE*.

FUJITO, T. 2000. Approximation algorithms for submodular set cover with applications. *TIEICE: IEICE Transactions on Communications/Electronics/Information and Systems*.

FUNKE, S., KESSELMAN, A., KUHN, F., LOTKER, Z., AND SEGAL, M. 2004. Improved approximation algorithms for connected sensor cover. In *Proc. of 3rd Int. Conf. on ADHOC Networks and Wireless (ADHOC-NOW)*.

GARG, N. 2005. Saving an epsilon: a 2-approximation for the k-mst problem in graphs. In *Proc. of ACM Symposium on Theory of Computing*.

GUESTRIN, C., BODIK, P., THIBAUX, R., PASKIN, M., AND MADDEN, S. 2004. Distributed regression: an efficient framework for modeling sensor network data. In *Proc. of ACM/IEEE International Conference on Information Processing in Sensor Networks*.

GUPTA, A., KRAUTHGAMER, R., AND LEE, J. R. 2003. Bounded geometries, fractals, and low-distortion embeddings. In *Proc. of Annual IEEE Symposium on. Foundations of Computer Science*.

GUPTA, H., DAS, S. R., AND GU, Q. 2003. Connected sensor cover: Self-organization of sensor networks for efficient query execution. In *Proc. of the ACM International Symposium on Mobile Ad Hoc Networks (MobiHoc)*.

JOHNSON, D. S., MINKOFF, M., AND PHILLIPS, S. 2000. The prize collecting steiner tree problem: theory and practice. In *Proc. of ACM-SIAM Symposium on Discrete Algorithms*.

KAR, K. AND BANERJEE, S. 2003. Node placement for connected coverage in sensor networks. In *Proc. of Intl. Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*.

KRAUSE, A. AND GUESTRIN, C. 2007. Near-optimal observation selection using submodular functions. In *Proc. of AAAI Conference on Artificial Intelligence Nectar track*.

KRAUSE, A. AND GUESTRIN, C. 2009. Optimal value of information in graphical models. *Journal of Artificial Intelligence Research 35*, 557–591.

KRAUSE, A., GUESTRIN, C., GUPTA, A., AND KLEINBERG, J. 2006. Near-optimal sensor placements: Maximizing information while minimizing communication cost. In *Proceedings of the Fifth International Symposium on Information Processing in Sensor Networks (IPSN)*.

KRAUSE, A., MCMAHAN, B., GUESTRIN, C., AND GUPTA, A. 2007. Selecting observations against adversarial objectives. In *Advances in Neural Information Processing Systems*. Vancouver, Canada.

KRAUSE, A., SINGH, A., AND GUESTRIN, C. 2007. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. In *Journal of Machine Learning Research*.

LEVIN, A. 2004. A better approximation algorithm for the budget prize collecting tree problem. *Ops. Res. Lett. 32*, 316–319.

LLOYD, E. L. AND XUE, G. 2007. Relay node placement in wireless sensor networks. *IEEE Transactions on Computers 56,* 1, 134–138.

MELIOU, A., KRAUSE, A., GUESTRIN, C., AND HELLERSTEIN, J. M. 2007. Nonmyopic informative path planning in spatio-temporal models. In *Proc. of AAAI Conference on Artificial Intelligence Nectar track*.

MINOUX, M. 1978. Accelerated greedy algorithms for maximizing submodular set functions. *Optimization Techniques, LNCS*, 234–243.

MORROW, R. 2004. *Wireless Network Coexistence*, 1 ed. McGraw-Hill Professional.

NEMHAUSER, G., WOLSEY, L., AND FISHER, M. 1978. An analysis of the approximations for maximizing submodular set functions. *Mathematical Programming 14*, 265–294.

NOTT, D. J. AND DUNSMUIR, W. T. M. 2002. Estimation of nonstationary spatial covariance structure. *Biometrika 89*, 819–829.

RAPPAPORT, T. 2000. *Wireless Communication: Principles and Practice*. Prentice Hall.

RASMUSSEN, C. E. AND WILLIAMS, C. K. 2006. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. The MIT Press.

SINGH, A., KRAUSE, A., GUESTRIN, C., KAISER, W. J., AND BATALIN, M. A. 2007. Efficient planning of informative paths for multiple robots. In *International Joint Conference on Artificial Intelligence (IJCAI)*. Hyderabad, India, 2204–2211.

VAZIRANI, V. V. 2003. *Approximation Algorithms*. Springer.

WIDMANN, M. AND BRETHERTON, C. S. 1999. 50 km resolution daily precipitation for the pacific northwest. http://www.jisao.washington.edu/data_sets/widmann/.

ZHAO, F. AND GUIBAS, L. 2004. *Wireless Sensor Networks: An Information Processing Approach.* Morgan Kaufmann.

ZHAO, F., SHIN, J., AND REICH, J. 2002. Information-driven dynamic sensor collaboration for tracking applications. *IEEE Signal Processing 19,* 2, 61–72.

ZUNIGA, M. AND KRISHNAMACHARI, B. 2007. An analysis of unreliability and asymmetry in low-power wireless links. In *ACM Transactions on Sensor Networks*.

APPENDIX

of Lemma 2. Given a collection of weights $\mathcal{P} = \{p_S : S \subseteq \mathcal{B}\}$, we write $E(\mathcal{P}) = \sum_{S \subseteq \mathcal{B}} p_S \cdot F(S)$. Note that $\mathbb{E}[F(A)] = E(\mathcal{P}_0)$ for $\mathcal{P}_0 = \{\Pr[A = S] : S \subseteq \mathcal{B}\}$.

Starting with the set of weights $\mathcal{P}_0$, we iteratively apply the following "uncrossing" procedure. As long as there is a pair of sets $S, T \subseteq \mathcal{B}$ such that neither of S or T is contained in the other, and p_S, $p_T > 0$, we subtract $x = \min(p_S, p_T)$ from both p_S and p_T, and we add x to both $p_{S \cap T}$ and $p_{S \cup T}$. Note the following properties of this procedure.

(i) The quantity $\sum_{S \subseteq \mathcal{B}} p_S$ remains constant over all iterations.

(ii) For each element $X \in \mathcal{B}$, the quantity $\sum_{S \subseteq \mathcal{B}: X \in S} p_S$ remains constant over all iterations,

(iii) The quantity $\sum_{S \subseteq \mathcal{B}} p_S |S|^2$ strictly increases every iteration.

(iv) By the submodularity of F, the quantity $E(\mathcal{P})$ is non-increasing over the iterations.

By (i) and (iii), this sequence of iterations, starting from $\mathcal{P}_0$, must terminate at a set of weights $\mathcal{P}^*$. At termination, the sets S on which $p_S > 0$ must be totally ordered with respect to inclusion, and by (ii) it follows that $p_{\mathcal{B}} \geq \rho$. Finally, by (iv), we have

$$\mathbb{E}[F(\mathcal{A})] = E(\mathcal{P}_0) \geq E(\mathcal{P}^*) \geq \rho F(\mathcal{B}), \tag{11}$$

as required. □

In order to prove Theorems 1 and 5, let us consider the subset $\mathcal{A}^*$ spanned by the optimal tree (or path), and let $\overline{\mathcal{A}^*} \subseteq \mathcal{A}^*$ denote its r-padded nodes with respect to a random partition drawn from the padded decomposition. (Recall that each node is r-padded with probability at least ρ.) Now Lemma 2 implies that $F(\overline{\mathcal{A}^*})$, the expected value of the nodes in A that are r-padded, is at least $\rho F(\mathcal{A}^*)$. The algorithm is based on the idea of trying to build a tree (a path) that recoups a reasonable fraction of this "padded value".

The following lemma will be useful in converting subtrees and paths of $\mathcal{G}'$ back to solutions of our original problem.

Proposition 6. *Given any subtree $\mathcal{T}'$ or path $\mathcal{P}'$ of $\mathcal{G}'$ spanning nodes $\mathcal{A}'$ with total weight W containing at least one cluster center, it is possible to find a subtree $\mathcal{T} \subseteq \mathcal{G}$ resp. path $\mathcal{P} \subseteq \mathcal{G}$ spanning the same vertices $\mathcal{A}'$, with a total length no more than $\ell(\mathcal{T}')$ resp. $\ell(\mathcal{P}')$, and with $F(\mathcal{A}') \geq \gamma W$.*

Proof. Each edge of $\mathcal{G}'$ (and hence of $\mathcal{T}'$) corresponds to some shortest path in $\mathcal{G}$, and we can add all these paths together to form a connected subgraph. Let $\mathcal{T}$ be any spanning tree of this subgraph; clearly, its length is no more than $\ell(\mathcal{T}')$. If $V_i \subseteq P_i$ is the subpath of P_i contained in $\mathcal{T}'$, then the total weight of these vertices $V(P_i')$ is exactly the total submodular value $F(V(P_i'))$, just by the definition the weights. Furthermore, since each pair of distinct paths are at distance at least r from each other, the locality property assures that the value of their union is at least γW. For paths, just observe that expanding edges of $\mathcal{P}'$ in $\mathcal{G}$ results in another path $\mathcal{P}$. □

Proposition 7. *If the graph $\mathcal{G}$ contains a subtree $\mathcal{T}^*$ spanning nodes $\mathcal{A}^*$, of length $\mathrm{c}(\mathcal{T}^*) = \ell^*$ and value $F(\mathcal{A}^*)$, then there is a subtree $\mathcal{T}'$ of the graph $\mathcal{G}'$ that has length at most*

$$\ell^* \times (\alpha(r+2)+2) \tag{12}$$

and whose expected sensing quality is at least

$$F(\mathcal{A}^*) \times (1-e^{-1}) \times \rho \tag{13}$$

Proof. Let a cluster $\mathcal{C}_i$ be called *occupied* if $\overline{\mathcal{A}^* \cap \mathcal{C}_i} \neq \emptyset$; w.l.o.g., let the $s+1$ clusters $\mathcal{C}_0, \mathcal{C}_2, \ldots, \mathcal{C}_s$ be occupied. Let $z_0, \ldots, z_s$ be the cluster centers (i.e., the first nodes picked by the greedy algorithm). We start building $\mathcal{T}'$ by adding a spanning tree on the centers of the clusters that are occupied.

The Cost. Let us bound the length of this center-spanning tree. Since $\mathcal{A}^*$ contains a point (say a_i) from each $\overline{\mathcal{C}_i}$, the padding condition ensures that the r-balls $B_r(a_i)$ must be disjoint, and hence the length of $\mathcal{T}^*$ is at least rs. Now, to attach a_i to z_i, we can add paths of length at most αr to $\mathcal{T}^*$; thus causing the resulting tree to have length $\ell^* + \alpha rs \leq (\alpha+1)\ell^*$. Since this is a Steiner tree on the centers, we can get a spanning tree of at most twice the cost; hence the cost of the edges connecting the spanning centers is at most

$$2(\alpha+1)\,\ell^*. \tag{14}$$

Now consider an occupied cluster $\mathcal{C}_i$, and let $|\overline{\mathcal{A}^* \cap \mathcal{C}_i}| = n_i$ be the number of padded nodes in $\mathcal{C}_i$. We now add to $\mathcal{T}'$ the subpath of P_i containing first n_i nodes $\{Z_i = G_{i,1}, G_{i,2}, \ldots, G_{i,n_i}\}$. Note that the length of edges added for cluster $\mathcal{C}_i$ is at most $\alpha r n_i$; summing over all occupied clusters gives a total length of $\alpha r \sum_i n_i \leq \alpha r |\mathcal{A}^*| \leq \alpha r \ell^*$, since each edge in $\mathcal{T}^*$ has at least unit length. Adding this to (14) proves the claim on the length of $\mathcal{T}'$.

The value. Finally, let us calculate the sensing quality value of the tree $\mathcal{T}'$: by the properties of the greedy algorithm used in the construction of $\mathcal{G}'$, the total *weight* of the set S_{in_i} added in cluster $\mathcal{C}_i$ is at least

$$(1-e^{-1})F(\overline{\mathcal{A}^* \cap \mathcal{C}_i}) \tag{15}$$

Summing this over occupied clusters, we get that the total value is at least $(1-e^{-1})F(\overline{\mathcal{A}^*})$, whose expected value is at least $(1-e^{-1})\rho F(\mathcal{A}^*)$. □

Proposition 8. *If the graph $\mathcal{G}$ contains an $s - t$ path $\mathcal{P}^*$ spanning nodes $\mathcal{A}^*$, of length $\mathrm{c}(\mathcal{P}^*) = \ell^*$ and value $F(\mathcal{A}^*)$, then there is an $s - t$ path $\mathcal{P}'$ of the graph $\mathcal{G}'$ that has length at most*

$$2 \times \ell^* \times (\alpha(r+2) + 6) \tag{16}$$

and whose expected sensing quality is at least

$$F(\mathcal{A}^*) \times (1 - e^{-1}) \times \rho \tag{17}$$

Proof. This result is an immediate corollary from Proposition 7, noting that the vertices s and t need to be connected to the center spanning tree by paths at most $2\ell^*$, and that the path $\mathcal{P}^*$ is a tree, and hence there exists a subtree $\mathcal{T}'$ in $\mathcal{G}'$ of length at most $\ell^* \times (\alpha(r+2) + 4)$. This tree is readily converted into a path (e.g., by traversal) of length at most twice the cost of the tree, i.e., of at most $2 \times \ell^* \times (\alpha(r+2) + 6)$. □

Combining these results, we now prove a slightly more detailed statement of Theorems 1 and 5:

Theorem 9. ***Trees:*** *For the covering problem* (1)*,* pSPIEL *will find a solution $\mathcal{T}$, with cost at most*

$$\kappa_{Quota}\,\ell^*\,(\alpha(r+2) + 2) \tag{18}$$

and whose expected sensing quality is at least

$$(1 - e^{-1})\,\gamma\rho F(\mathcal{A}^*), \tag{19}$$

where ℓ^* is the sensing quality of the optimum tree $\mathcal{A}^*$. For the maximization problem (2)*,* pSPIEL *will find a solution $\mathcal{T}$ with cost at most*

$$\ell^*\,(\alpha(r+2) + 2) \tag{20}$$

and whose expected sensing quality is at least

$$\kappa_{Budget}^{-1}(1 - e^{-1})\,\gamma\rho F(\mathcal{A}^*), \tag{21}$$

where κ_{Quota} and κ_{Budget} denote the approximation guarantees for approximately solving Quota- and Budget-MST problems (currently, $\kappa_{Quota} = 2$ and $\kappa_{Budget} = 3 + \varepsilon$, for $\varepsilon > 0$, are the best known such guarantees Garg (2005); Johnson et al. (2000)).

Paths: *For the path planning problem* (10)*,* pSPIEL *will find a solution $\mathcal{P}$, with cost at most*

$$\kappa_{Orient}\,2\ell^*\,(\alpha(r+2) + 6) \tag{22}$$

and whose expected sensing quality is at least

$$(1 - e^{-1})\,\gamma\rho F(\mathcal{A}^*), \tag{23}$$

where ℓ^* is the sensing quality of the optimum tree $\mathcal{A}^*$. Hereby, κ_{Orient} is the approximation guarantee for approximately solving the Orienteering problem (currently, $\kappa_{Orient} = 3 + \varepsilon$ is the best known such guarantee (Chekuri et al., 2008)).

Proof. Proposition 7 proves the existence of a tree $\mathcal{T}'$ in the graph $\mathcal{G}'$, for which both cost and total weight are close to the optimal tree $\mathcal{T}$ in $\mathcal{G}$. The construction in the proof also guarantees that the tree $\mathcal{T}'$ contains at least one cluster center $G_{i,1}$ for some i (or is empty, in which case $\mathcal{T}$ is empty). Proposition 6 handles the transfer of the solution to the original graph $\mathcal{G}$. Hence, in order to solve the covering problem (1) or optimization problem (2) in $\mathcal{G}$, we need to solve the respective covering and maximization problem in the modular approximation graph $\mathcal{G}'$, rooted in one of the cluster centers. Any κ_{Quota} approximate algorithm to the Quota-MST problem can be used for the covering problem, using a quota of $Q = (1 - e^{-1})\rho\, F(\mathcal{A}^*)$. While for the unrooted version of the Budget-MST problem, there is a constant factor $\kappa_{Budget} = 3 + \varepsilon$ approximation algorithm, the best known approximation for rooted Budget-MST is $4 + \varepsilon$ by Chekuri et al. (2008). We can however exploit the structure of the MAG to still get an approximation guarantee and prove Theorem 1 for an improved guarantee of $3 + \varepsilon$. We simply need to prune all nodes in $\mathcal{G}'$ which are further than $B = \ell^* \left(\alpha(r+2)+2\right)$ away from the core of $\mathcal{G}'$, and then run the unrooted approximation algorithm Johnson et al. (2000) on $\mathcal{G}'$. If this algorithm, started with budget $B = \ell^* \left(\alpha(r+2)+2\right)$ selects nodes from sub-chain i, not including center $G_{i,1}$, we instead select the entire i-th chain. By construction, this procedure is guaranteed not to violate the budget, and the submodular function value can only increase.

The result for paths follows analogously, using Proposition 8 instead of Proposition 7, and applying the approximation algorithm for $s - t$ orienteering to the modular approximation graph. □

ML

MACHINE LEARNING
DEPARTMENT

Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

Carnegie Mellon.