



# Multi-objective optimization for sensor placement: An integrated combinatorial approach with reduced order model and Gaussian process

Zhaoyi Xu <sup>\*</sup>, Yanjie Guo, Joseph Homer Saleh

School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

## ARTICLE INFO

### Keywords:

Sensor placement  
Multi-objective combinatorial optimization  
Proper orthogonal decomposition  
Lazy greedy algorithm  
Branch and bound

## ABSTRACT

We develop a novel sensor placement method that maximizes monitoring performance while minimizing deployment cost. Our method integrates a reduced order model and multi-objective combinatorial optimization. We first decompose the spatio-temporal state field to be monitored by proper orthogonal decomposition (POD), and we use the Gaussian Process to model the uncertainty in each POD mode. Next, we develop a lazy greedy (LG)- $\epsilon$ -constraint optimization to derive the Pareto-optimal sensor configurations. We further design a branch and bound algorithm to calculate the global optimum and validate the correctness of select configurations on the LG-derived Pareto frontier. We evaluate and benchmark our algorithm in computational experiments based on the temperature dataset of the Berkeley Intel Lab. The computational results demonstrate that our algorithm places sensors at locations of large magnitude in the POD modes, and that our method achieves better state estimation accuracy and smaller reconstruction errors compared with alternative methods.

## 1. Introduction

Sensor network technology has significantly developed in the past decade due to its importance in smart homes [1–3], traffic and road safety [4–6], and Internet of Things [7–9]. A sensor network refers to a group of distributed sensors placed to monitor and supervise the state of a system or environment [10]. Research in this area focuses not only on developing novel hardware and algorithms [11] but also on optimizing sensor placement [12–14]. In this work, we develop a novel, computationally efficient sensor placement algorithm that provides high monitoring performance and state estimation accuracy, and it outperforms existing solutions to this problem. We demonstrate our sensor placement algorithm using the Berkeley Intel Lab thermal sensor placement experiments and dataset.

By its nature, the sensor placement problem is a min–max multi-objective optimization problem [15,16]. It aims to **minimize** sensor deployment cost while **maximizing** some measure of monitoring performance (details in Section 2). Research to date has leveraged different numerical algorithm [17–21], optimization [22–29], and machine learning [30–33] approaches to address this problem in different applications, in structural health monitoring and water distribution for example, and typically by using training and testing datasets from the system or environment to be monitored [34–38]. Hou et al. [39]

developed a sensor placement optimization algorithm based on the genetic algorithm heuristic and reported significant improvement in structural damage detection accuracy compared with traditional methods. Clark et al. [38] designed a genetic algorithm with cost constraint for sensor placement optimization, and they reported high computational efficiency and near-optimal results in several applications.

Three general challenges are recognized in current sensor placement optimizations. First, given the high-dimensional nature of a given spatio-temporal state field to be monitored, the model complexity is generally high, and exact algorithms such as the branch and bound (BnB) are computationally prohibitive. Heuristics such as greedy algorithms generally trade significant estimation accuracy for computational efficiency and result in unbounded local optima [39]. Convex relaxation is another popular approach to mitigate the computational burden in sensor placement optimization [40–42]. However, the optimization accuracy of the sensor placement can be compromised in the process to simplify and convexify the original problem. Second, an important problem in sensor placement optimization is the selection of the objective function to measure the monitoring performance. For example, in our work, we consider improving state estimation accuracy as the end-goal of the sensor placement and monitoring performance. The objective function is required to be a good proxy of this end-objective, and it

<sup>\*</sup> Corresponding author.

E-mail address: [zxu328@gatech.edu](mailto:zxu328@gatech.edu) (Z. Xu).

needs to have specific analytical features that can reduce computational cost without significantly compromising the estimation accuracy (details in Section 2.3). Third, by its nature, sensor placement is a multi-objective optimization problem, and it consists in minimizing sensor deployment cost and maximizing the monitoring performance. Other considerations can also be appended to this baseline, such as the reliability and resilience of the sensor network monitoring function.

In order to address these challenges in sensor placement, we integrate the following three analytical tools in our algorithm. First, we leverage a proper orthogonal decomposition (POD) reduced order model (ROM) to reduce the dimensionality of the spatio-temporal field data to mitigate our optimization model complexity. Second, we select the mutual information (MI) as our objective function to measure the sensors' monitoring performance. As will be discussed in Section 2.3, MI quantifies the 'amount of information' about one group of random variables obtained by observing another group, and consequently the optimization of MI can improve the state estimation accuracy. We validate this MI is a good proxy for the state estimation accuracy and compare it with alternative metrics, such as predictive variance [35,43], by our computational results in Section 5.4. Finally, we use a multi-objective combinatorial optimization approach with  $\epsilon$ -constraint algorithm to solve the original min–max sensor placement problem without any further simplifications. Our algorithm determines the Pareto frontier of sensor cost and monitoring performance.

The main contribution of this work is the development of a novel method and algorithm for sensor placement multi-objective combinatorial optimization which achieves the highest monitoring performance with the minimum deployment cost. More specifically, the main contribution is a novel method that integrates proper orthogonal decomposition (POD) with the LG- $\epsilon$ -constraint optimization algorithm for sensor placement. The use of POD lowers the dimensionality of the spatial–temporal data and improves the computational efficiency of the LG- $\epsilon$ -constraint. One important advantage of this LG- $\epsilon$ -constraint is that its result is robust and theoretically bounded.

The remainder of the article is organized as follows. The formal problem formulation of the sensor placement is provided in Section 2. The details of our ROM-based multi-objective combinatorial optimization are introduced in Section 3. The details of the Berkeley Intel Lab thermal sensor placement computational experiments are presented in Section 4. The computational results to evaluate and benchmark the performance of our algorithm against alternative sensor placement methods are discussed in Section 5. In Section 6, we conclude this work and indicate some fruitful venues for future work.

## 2. Problem formulation

In this section, we discuss the optimization statement to address the sensor placement problem. We also introduce the sensor deployment cost and the MI as the objective function of the monitoring performance.

### 2.1. Optimization statements

We consider two aspects of the sensor placement problem: the sensor network deployment cost, and the network monitoring performance. In real-world applications, the cost is either bounded by an available budget ( $B$ ) or is to be minimized. Similarly, the monitoring performance is either required to be larger than a lower baseline monitoring threshold or it is to be maximized. Given these considerations, we formulate the sensor placement statement as a **multi-objective optimization** problem as shown in Eq. (1), where  $X$  is the set of fixed candidate locations,  $S \subseteq X$  is a set of chosen sensor placement locations, and  $C(S)$  is the cost function of a sensor placement  $S$ ,  $F(S, X)$  is the MI function, which we use in this work as the measure of monitoring performance, and  $B \in R^+$  is the budget. The decision variable is the sensor placement set  $S$ .

$$\begin{cases} \max_S & [F(S, X), -C(S)] \\ s.t. & S \subseteq X \end{cases} \quad (1)$$

This statement reflects the nature of the problem to maximize the monitoring performance while minimizing the deployment cost. In this multi-objective optimization problem, we search for a Pareto optimal group of sensor placements given these two objectives [44]. This multi-objective optimization formulation and its Pareto frontier can be particularly helpful for decision-makers, in informing requirement-setting and budget constraints, for example, for the sensor network (e.g., in identifying knees in the Pareto frontier, or whether some performance threshold is prohibitive or a minor budget change can 'buy' us significant performance improvement).

### 2.2. Sensor deployment cost

Minimizing the sensor deployment cost is one of the two objectives in our placement problem. Here, we consider the cost function mapping a certain sensor placement ( $S$ ) to a positive real number cost such that  $C : 2^X \rightarrow R^+$ . We use the additive cost function in our sensor placement problem since it is widely used in real-world applications, and it requires fewer assumptions and hyperparameters compared with the sub-additive or super-additive cost functions [35]. For an element  $x \in X$ , we assign a cost  $c(x) \in R$  to each possible location and define the cost of a set  $C(S)$  as given by Eq. (2), where  $s \in S$  is an element in the sensor placement set  $S$ . For example, if we set the cost of a single sensor at location  $x$  as  $c(x) = 1$  for all  $s \in X$ , we obtain  $C(S) = |S|$ .

$$C(S) = \sum_{s \in S} c(s) \quad (2)$$

We assign a constant cost  $c$  to all possible locations  $x \in X$ , and as a consequence, the optimization of the additive cost function becomes equivalent to minimizing the sensor number. We therefore switch to minimizing the sensor number for cost optimization in the rest of this work.

### 2.3. Mutual information for measuring monitoring performance

In this subsection, we discuss the objective function chosen as a proxy measure of the sensor network monitoring performance. Recall that one of the end-objectives of sensor placement is to reduce the error in, and improve the accuracy of the state estimation at locations where sensors are not available (or more broadly of the entire state field). We propose that the objective function has to satisfy the following two requirements. First, the candidate objective function should be capable of reflecting this state estimation accuracy. Second, because the sensor placement is nondeterministic polynomial (NP)-complete [27,45,46], the heuristic methods used to address it, such as greedy and genetic algorithms [39], should be able to calculate the Pareto frontier in polynomial-time. Consequently, the candidate objective functions should have special properties, such as submodularity to guarantee the quality of the solutions obtained by heuristic algorithms.

To satisfy these two requirements, we select the MI function, shown in Eq. (3) [47], to measure the monitoring performance. In Eq. (3),  $Y$  stands for the (random) state variable,  $A$  and  $B$  are alternative subsets of the set of locations  $X$ , and  $a$  and  $b$  are elements in the subsets with  $a \in A$  and  $b \in B$ .  $p(Y_a, Y_b)$  is the joint distribution and  $p(Y_a)$  and  $p(Y_b)$  are the marginal distribution of state variable pair  $(Y_a, Y_b)$ . The joint and marginal distributions are calculated based on the GP model introduced in Section 3.3.

$$I(Y_B; Y_A) = \sum_{a \in A} \sum_{b \in B} p(Y_a, Y_b) \log \left( \frac{p(Y_a, Y_b)}{p(Y_a)p(Y_b)} \right) \quad (3)$$

In information theory, MI is a measure of the correlation between two groups of variables. More specifically, MI quantifies the 'amount of

information' about one group of random variables obtained by observing another group. Here, we take  $A = S$  as the sensor observing group and  $B = X \setminus S$  as the variable group of unobserved states. Intuitively, we maximize the MI objective function  $F(S, X) = I(Y_S; Y_{X|S})$  to optimize the 'amount of information' about the unobserved states conditioned on sensor measurements. The more 'information' regarding the unobserved states a sensor configuration can provide, the more accurate the state estimation will be. Consequently, the state estimation accuracy can be reflected in the MI function. Thus, the MI function satisfies the first requirement for candidate objective functions to reflect the monitoring performance; we will evaluate its effectiveness based on our computational results in [Section 5](#).

For the second requirement, as discussed in [\[35\]](#), the MI function for monitoring performance is submodular. This indicates a  $(1 - 1/e)$ -approximate optimal for a greedy heuristic for submodular function with  $F(\emptyset, X) = 0$  if  $F(\hat{A}, X)$  is non-decreasing [\[35\]](#). This approximation ratio means the greedy solution  $F(S_{\text{greedy}}, X)$  is no less than  $(1 - 1/e)F(S_{\text{opt}}, X)$ , where  $S_{\text{opt}}$  is the global optimal. According to Ref. [\[35\]](#), the MI is conditionally non-decreasing when the sensor number is smaller than a threshold,  $|S| \leq S_0 < |X|$ . We will use this non-decreasing threshold  $S_0$  in the Pareto frontier of the sensor number as  $1 \leq |S| \leq S_0$ . The details of our sensor placement optimization process are discussed in the next section.

In short, MI satisfies both requirements for candidate objective functions of monitoring performance, and it is used hereafter for this purpose in our optimization method. We simplify the notation of the MI function from  $F(S, X)$  to  $F(S)$  in the rest of this work because the grid of possible candidate locations  $X$  can be made as fine as needed, and it is fixed at the onset of the problem.

### 3. Proper orthogonal decomposition and multi-objective combinatorial optimization for sensor placement

In this section, we first provide a high-level overview of our sensor placement optimization method. We then discuss its analytical details.

#### 3.1. Overview of the sensor placement optimization

The overview of our sensor placement optimization is shown in [Fig. 1](#).

The objective of this algorithm is to calculate and validate a Pareto frontier of the sensor deployment cost and monitoring performance of a spatio-temporal field. The inputs to the algorithm are the training snapshots at different epochs of the field to be monitored. These can be obtained from real-world experiments (as discussed in section 4) or from computational simulations when available. One snapshot of the field contains the spatial information of the entire field at one point in time. In sections 4 and 5, we use a temperature field as an illustrative application of our method. The output of the algorithm is a validated Pareto frontier of the optimal sensor placement configurations for a range of sensor numbers. The algorithm consists of three phases: (1) POD model reduction; (2) LG- $\epsilon$ -constraint multi-objective combinatorial optimization; and (3) BnB validation. This third phase is not essential for our sensor placement method; it is appended to partially validate and assess the quality of the results obtained in Phase II as we discuss shortly.

In Phase I, the POD model reduction shown in the top panel in [Fig. 1](#), we use the POD ROM (details in [Section 3.2](#)) to decompose the state field into a group of modes ranked by their corresponding eigenvalues. We then use the GP technique to model the uncertainty of different POD modes (details in [Section 3.3](#)). We calculate a group of uncertainty covariance matrices of different POD modes based on the GP models, and we use these matrices for sensor placement optimization as discussed next.

In Phase II, the LG- $\epsilon$ -constraint multi-objective combinatorial optimization (MOCO) shown in the middle panel in [Fig. 1](#), we first initialize

the sensor placement ( $S$ ) with  $|S| = 1$  and optimization process by the LG- $\epsilon$ -constraint optimizer. We then leverage the uncertainty matrices of different POD modes from Phase I to calculate the weighted MI function (details in [Section 3.4](#)) for the sensor placement. Next, we examine the convergence of LG. If greedy convergence is not achieved, we leverage the LG optimizer to update the sensor placement configuration and start a new optimization loop. When greedy convergence is achieved for a given number of sensors, say  $|S| = m$ , we note the configuration that achieves the maximum monitoring performance as  $S_m^*$ . We then examine the non-decreasing characteristic of the weighted MI to guarantee the  $(1 - 1/e)$ -approximation ratio of the LG heuristic (details in [Section 3.4](#)). If the non-decreasing characteristic holds, we increase the sensor number ( $|S| = |S| + 1$ ) and start the next loop for the  $\epsilon$ -constraint optimization (details in [Section 3.5](#)). When the weighted MI decreases, we stop the process and output the Pareto frontier of the sensor number minimization versus weighted MI maximization. We output this Pareto frontier and its corresponding sensor placement configurations for the Phase III validation.

In Phase III, the BnB validation shown in the bottom panel in [Fig. 1](#), we assess the quality of the Pareto frontier calculated in Phase II by a BnB exact algorithm (details in [Section 3.6](#)). The LG algorithm provides the local optimal results with  $(1 - 1/e)$ -approximate rate. In order to assess the correctness of the LG heuristic approximation, we devise a BnB exact method. This BnB realizes the global optimization of sensor placement for a given number of sensors, and we compare this global optimal solution with the result from Phase II to assess the quality of LG optimization solution. However, using this exponentially complex BnB [\[48\]](#) for the entire Pareto frontier is computationally prohibitive<sup>1</sup>. We therefore focus on a few select points with special application interest, for example the knee point in the Pareto frontier or for a specific cost/sensor number. We first compute the entire Pareto optimal set with the LG heuristic algorithm and then select the points of interest for specific applications. Here, to demonstrate this partial validation approach, we use the knee point in the Pareto frontier [\[49\]](#) (details in Appendix A).

In the next subsections, we expand on this introductory material and discuss the analytical details of the main elements in our method.

#### 3.2. Proper orthogonal decomposition reduced order model

POD, also known as Karhunen-Loeve expansion, has been used for developing low-dimensional models of spatio-temporal state field [\[50\]](#). Given a set of time streaming data, say  $y(t)$ , POD finds a subspace  $y_r$  of fixed dimension  $r$ , such that the error in the projection onto the subspace is minimized. Here, we consider a temperature field and focus on capturing the dynamic modes of this unsteady field. We use for the temperature data  $y_{\text{pod}}(X, t) = T(X, t) - \bar{T}(X)$ , where  $y_{\text{pod}}(X, t)$ ,  $T(X, t)$ ,  $\bar{T}(X) \in \mathbb{R}^n$  and  $n = |X|$ .  $y_{\text{pod}}(X, t)$  removes the time average component,  $\bar{T}(X)$  from the temperature field  $T(X, t) - \bar{T}(X)$ . We drop  $X$  in  $y_{\text{pod}}(X, t)$  hereafter to simplify the notation since  $X$  is fixed.

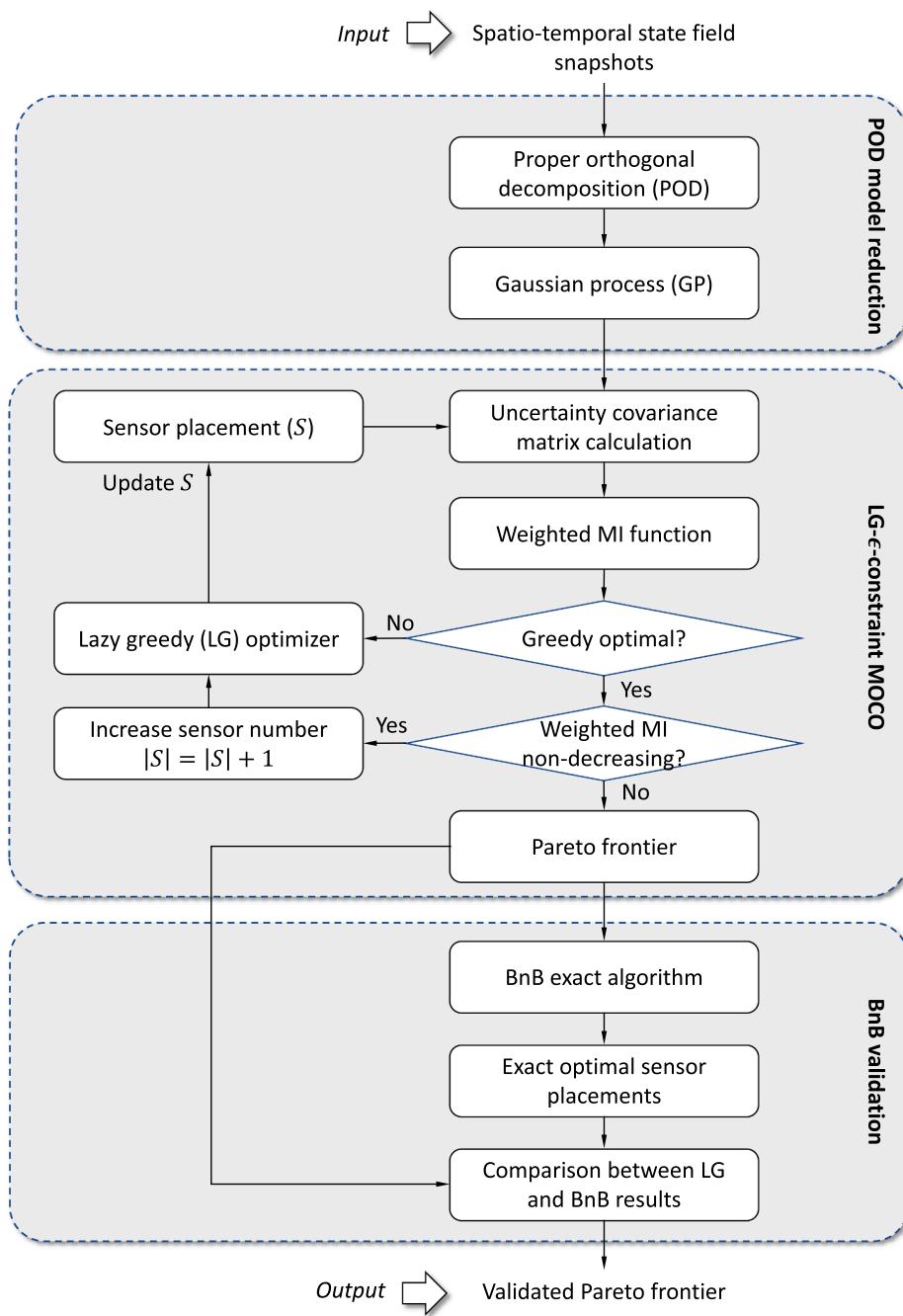
Our set of data,  $y_{\text{pod}}(t)$ , is in  $1 \leq t \leq T$  of a period of  $T$ . We seek a projection function  $P_r(y_{\text{pod}}(t)) : \mathbb{R}^n \rightarrow \mathbb{R}^r$  of fixed rank  $r$ , which minimizes the total error as calculated in Eq. [\(4\)](#).

$$\text{err} = \sum_{t=1}^T (y_{\text{pod}}(t) - P_r(Y_{\text{pod}}(t)))^2 \quad (4)$$

To solve this problem, we introduce the  $n \times n$  matrix as shown in Eq. [\(5\)](#).

$$R = \sum_{t=1}^T y_{\text{pod}}(t)y_{\text{pod}}^T(t) \quad (5)$$

<sup>1</sup> The computational experiments are conducted on a MATLAB platform on Windows 10 machine with AMD Ryzen 7 8 cores CPU processor and 31.9 GB system memory.



**Fig. 1.** Overall process of the sensor placement optimization method.

We then determine the eigenvalues and eigenvectors of  $R$ , given by Eq. (6).

$$R\phi_k = \lambda_k \phi_k, \lambda_1 \geq \dots \geq \lambda_n \geq 0 \quad (6)$$

Since  $R$  is symmetric, positive-semidefinite, as shown in Eq. (5), all the eigenvalues  $\lambda_k$  are real and non-negative, and the eigenvectors  $\phi_k$  are orthonormal. The main result of the POD is the optimal subspace of dimension  $r$  spanned by  $\{\phi_1, \dots, \phi_r\}$ , and the optimal projection function  $P_r(\hat{A} \cdot)$  is given by Eq. (7). The vectors  $\phi_k$  are called POD modes.

$$P_r(y_{pod}(t)) = \sum_{k=1}^r \phi_k \phi_k^T y_{pod}(t) \quad (7)$$

In POD applications dealing with velocity fields, the eigenvalue  $\lambda$  stands for the kinetic energy of the corresponding mode [51]. Here, we

decompose the temperature field and use the eigenvalue to stand for a pseudo ‘kinetic energy’ to rank the POD modes. The first few modes have larger eigenvalues and contain most of the ‘kinetic energy’ of the field. For more information on the POD method, the reader is referred to Ref. [52].

In the next subsection, we introduce the GP modelling to calculate the spatial uncertainty covariance of different POD modes for the weighted MI function.

### 3.3. Modelling sensor data using Gaussian process

We consider a set of discrete candidate sensor locations  $X$  with  $|X| = n$ . In order to estimate the state at one of these locations from the other sensor readings, we need the joint distribution over temperatures at these  $n$  locations. An effective approach is to assume that the tempera-

ture field has a (multivariate) Gaussian joint distribution [53] for a set of  $n$  random variables  $Y_X$  with a distribution shown in Eq. (8), where  $\mu_X$  is the mean vector and  $\Sigma_{XX}$  the covariance matrix.

$$P(Y_X = y_X) = \frac{1}{(2\pi)^{n/2} |\Sigma_{XX}|} \exp\left(-\frac{1}{2}(y_X - \mu_X)^T \Sigma_{XX}^{-1} (y_X - \mu_X)\right) \quad (8)$$

An important property of GP is that for every finite subset  $S \subseteq X$ , the joint distribution over the corresponding random variables  $Y_S$  is a Gaussian distribution. To specify this distribution, a GP is associated with a mean function  $M(\hat{A} \cdot)$ , and a symmetric positive-definite kernel function  $K(\hat{A} \cdot, \hat{A} \cdot)$ , called the covariance kernel. We apply a quadratic basis function for  $M(\hat{A} \cdot)$  and a squared exponential kernel for  $K(\hat{A} \cdot, \hat{A} \cdot)$  [36]. The hyperparameters of  $M(\hat{A} \cdot)$  and  $K(\hat{A} \cdot, \hat{A} \cdot)$  are determined by the maximum likelihood estimation (MLE) using the training data [54]. We use the trained kernel function  $K(\hat{A} \cdot, \hat{A} \cdot)$  to calculate the covariance matrix for different POD modes (labeled as  $\Sigma_i$ ).

In the next subsection, we introduce the calculation of the weighted MI for the sensor placement objective function.

### 3.4. Weighted mutual information function calculation

By using the POD model introduced in Section 3.2, we map the spatio-temporal field into a discrete set of POD modes. The POD modes are ranked by their eigenvalues and the corresponding ‘kinetic energy’. We calculate the uncertainty covariance matrices of POD modes ( $\Sigma_i$ ) from the GP modelling of different modes. To incorporate the POD modes into the MI function, we calculate a weighted MI based on  $\Sigma_i$ s as explained next.

For each mode, we calculate the MI (Eq. (3)) for a specific sensor placement  $S$ . We then calculate the weighted MI function as shown in Eq. (9), where  $n$  is the POD mode number,  $F(\hat{A} \cdot)$  is the weighted MI function,  $\lambda_i$  is the eigenvalue of the  $i$ -th POD mode, and  $F_i(\hat{A} \cdot)$  is the MI function corresponding to POD mode  $i$ .

$$F(S) = \sum_{i=1}^n \lambda_i F_i(S) \quad (9)$$

In order to use the LG heuristic with  $(1 - 1/e)$ -approximate rate properly, we prove the submodularity property of the weighted MI function in Appendix B.

### 3.5. LG- $\epsilon$ -constraint optimization

In this subsection, we introduce the LG- $\epsilon$ -constraint algorithm to solve our sensor placement problem. By using this LG- $\epsilon$ -constraint optimization, we derive a Pareto frontier of sensor configurations, each point on the frontier consists of an optimal sensor placement that maximizes the weighted MI function for a given number of sensors.

The  $\epsilon$ -constraint is a simple approach to multi-objective optimization for minimizing one objective, say  $f_i(x)$ , subject to the additional constraints that  $f_j(x) \leq \epsilon_j$  for all  $j \neq i$ , where  $\epsilon_j$  ranges from a lower bound to an upper bound of  $f_j$  [55]. In our application, the  $\epsilon$  constraint is the sensor number. We range  $\epsilon$  from its lower bound of 1 sensor until the upper bound is reached, given by the number of sensors for which the weighted MI function no longer exhibits a non-decreasing behavior. More details of the  $\epsilon$ -constraint MOO can be found in Ref. [55].

We incorporate a lazy greedy (LG) [56] algorithm with this  $\epsilon$ -constraint to form a LG- $\epsilon$ -constraint multi-objective combinatorial optimization method. In each  $\epsilon$ -constraint iteration, we use the LG local heuristic to solve the optimal weighted MI with the current sensor number and varying the sensor configuration. The LG combinatorial optimization was first proposed in Ref. [56]. It is a variant of the greedy heuristic to enhance the efficiency of the optimization process. We provide the pseudo-code of this LG in Appendix C. The LG heuristic is  $(1 - 1/e)$ -approximate guaranteed for the submodular weighted MI

function in our sensor placement problem [35].

The greedy algorithm is a simple and intuitive heuristic, which seeks to find a local optimal sensor location choice sequentially at each step given the weighted MI function. The LG serves as a more efficient greedy algorithm [56]. Intuitively, if a location  $x$  is selected for a sensor placement, nearby locations become significantly less desirable as their marginal contribution to the weighted MI will decrease as well. In this way, these locations will not be considered as possible optimal locations by the LG for several iterations. The lazy evolution avoids the unnecessary consideration of these points, and thus saves significant computational time. More details about the LG can be found in [56].

As shown in Phase II in Fig. 1, we use this LG- $\epsilon$ -constraint to calculate the Pareto frontier of the deployment cost versus weighted MI functions. We output this Pareto frontier to Phase III to (partially) validate its correctness, as discussed next.

### 3.6. Branch and bound global optimization

We devise a BnB exact algorithm to calculate the global optimal of the sensor placement configuration and validate the correctness of the LG heuristic solution. This BnB can calculate the global optimization of the weighted MI objective function and the corresponding sensor placement configuration.

BnB is an algorithm design paradigm for combinatorial optimization problems. Different from greedy heuristics, which solves the optimization problem sequentially, the BnB provides the global optimal by conducting a tree search. A BnB algorithm consists of a systematic enumeration of candidate solutions by means of state space search: the set of candidate solutions is thought of as forming a rooted tree with the full set at the root. The algorithm explores branches of this tree, which represent subsets of the solution set. Before enumerating the candidate solutions of a branch, the branch is checked against upper and lower estimated bounds on the optimal solution, and the corresponding node and branch are pruned if it is impossible to produce a better solution than the best one found so far. The reader is referred to Ref. [48] for more information about BnB.

To improve the efficiency and guarantee the global optimality of BnB, the calculations of the lower and upper bounds are particularly important in this algorithm. Given the maximization problem of MI, the lower bound is the best weighted MI solution we have obtained so far at a node in the candidate solution tree. The upper bound is the weighted MI value, which is larger or equal to all possible weighted MI result the current node can lead to. If the upper bound for the current node is smaller than the lower bound, it indicates the current node cannot lead to a solution that outperforms the lower bound. This node, therefore, is pruned to improve computational efficiency.

In our BnB, the lower bound is initialized by the LG solution. We update the lower bounds in iteration  $i$  of BnB by the optimal objective in the first  $i$  iterations.

For the upper bound, we devise a novel upper bound calculation. We use the LG heuristic method with an approximation rate of  $(1 - 1/e)$  to estimate the best solution the current node can lead to. At any alternative node in the tree, we define  $X_n$  as the set of locations where the sensor placement has been determined,  $S_n \subseteq X_n$  is the sensor placement set in  $X_n$ ,  $X_l = X \setminus X_n$  is a set of locations, which has not been selected. The upper bound is calculated as Eq. (10) where  $F(\hat{A} \cdot)$  is the weighted MI objective, and  $S_{LG} \subseteq X_l$  is the LG heuristic sensor placement solution for the undetermined locations  $X_l$ . Lower and upper bounds for the maximization problem

$$UB = F(S_n) + [1/(1 - 1/e)] \times F(S_{LG}) \quad (10)$$

A proof of the correctness of this upper bound update method (Eq. (10)) is provided in Appendix E.

We apply this BnB exact algorithm with this novel upper bound calculation (Eq. (10)) to the global optimal sensor placement at select

points in the Pareto frontier to (partially) validate and assess the quality of the LG results. The specific application and results of our method are discussed in the next sections.

#### 4. Experiments and dataset

To evaluate the performance of our proposed sensor placement optimization method, we design some computational experiments based on the temperature field dataset of the Berkeley Intel Lab [57]. There are available locations to place sensors in this lab, and we seek the optimal placement for varying sensor numbers. In this section, we introduce the details of the temperature dataset and discuss the metric used for capturing the state estimation accuracy.

##### 4.1. Berkeley Intel Lab temperature dataset

The spatial layout of the Berkeley Intel Lab is shown in Fig. 2.

This spatio-temporal dataset consists of time streaming temperature measurements from 54 nodes as the group of fixed candidate locations  $X$  shown in Fig. 2. The temperature dataset was collected between February 28th and April 5th in 2004 [57]. We select snapshots of the entire temperature field (54 locations) at 50 different time instances or epochs. The epochs are selected uniformly and sequentially over this time window. We use the first 30 snapshots for training and the last 20 snapshots for testing. Fig. 3 presents a few snapshots of the temperature field in the lab.

##### 4.2. State estimation accuracy metric

We determine the optimal sensor placement given this temperature field by leveraging our sensor placement algorithm and analyzing the training snapshots. We then examine the state estimation accuracy of the optimal sensor placement on the testing snapshots. To evaluate this accuracy, we first reconstruct the entire temperature field by the measurements from our sensor configurations. For example, given 20 sensors, we first use our sensor placement method to optimize the sensor configuration to maximize the weighted MI by analyzing the training data. We then use the measurements from these 20 sensors to estimate the temperatures at the remaining 34 nodes on the testing snapshots.

To accomplish this state estimation task, we use Gaussian Process regression (GPR) [36] to estimate the temperature at all locations  $X$

based on the measurements of the sensor placed at locations  $S$ . The estimated temperature state of the whole field,  $\mu_{X|S}(t)$  is shown in Eq. (11). We train the mean and kernel function ( $M(\hat{A} \cdot)$  and  $K(\hat{A} \cdot, \hat{A} \cdot)$ ) of the GPR model at every time step  $t$  based on the sensor measurements  $y_S(t)$  to produce accurate GPR estimation for the state field  $\mu_{X|S}(t)$ .

$$\mu_{X|S}(t) = \mu_X(t) + \Sigma_{XS} \Sigma_{SS}^{-1} (y_S(t) - \mu_S(t)) \quad (11)$$

To evaluate the accuracy of the state estimation of the testing dataset, we calculate the root mean square error (RMSE) comparing the mean state estimation ( $\mu_{X|S}(t)$ ) with the ground truth temperature measurements ( $T(X, t)$ ). This RMSE index is frequently used to measure the differences between the estimated value and the ground truth. We use it here to evaluate the estimation accuracy of our sensor placement for the entire temperature field compared with the ground truth in the testing set. The sensor placement testing RMSE is calculated by Eq. (12), where  $T_{\text{test}}$  denotes the testing period.

$$RMSE = \sqrt{\frac{\sum_{t \in T_{\text{test}}} \sum_{x \in X} (\mu_{x|S}(t) - T(x, t))^2}{\sum_{t \in T_{\text{test}}} \sum_{x \in X} 1}} \quad (12)$$

## 5. Results and discussion

In this section, we first present the POD result of the temperature field. We then discuss the Pareto frontier obtained by the LG- $\epsilon$ -constraint optimization, and we partially validate the LG heuristic result of the sensor placement by the BnB global optimization. Next, we calculate the testing RMSE error to benchmark the performance of our algorithm against the alternative sensor placements algorithms. Finally, we conduct a sensitivity analysis of the state estimation accuracy to different POD mode numbers or ‘kinetic energy’.

### 5.1. Proper orthogonal decomposition results of temperature field modes

We first use POD to decompose the temperature field for the training snapshots. The magnitudes of the eigenvalues and the corresponding ‘kinetic energy’ distribution of the different modes is shown in Fig. 4.

Fig. 4 shows the magnitude of eigenvalue and the corresponding mode ‘kinetic energy’ distribution. We calculate the sum of the eigenvalue of all modes as the overall POD ‘kinetic energy’ of the spatio-

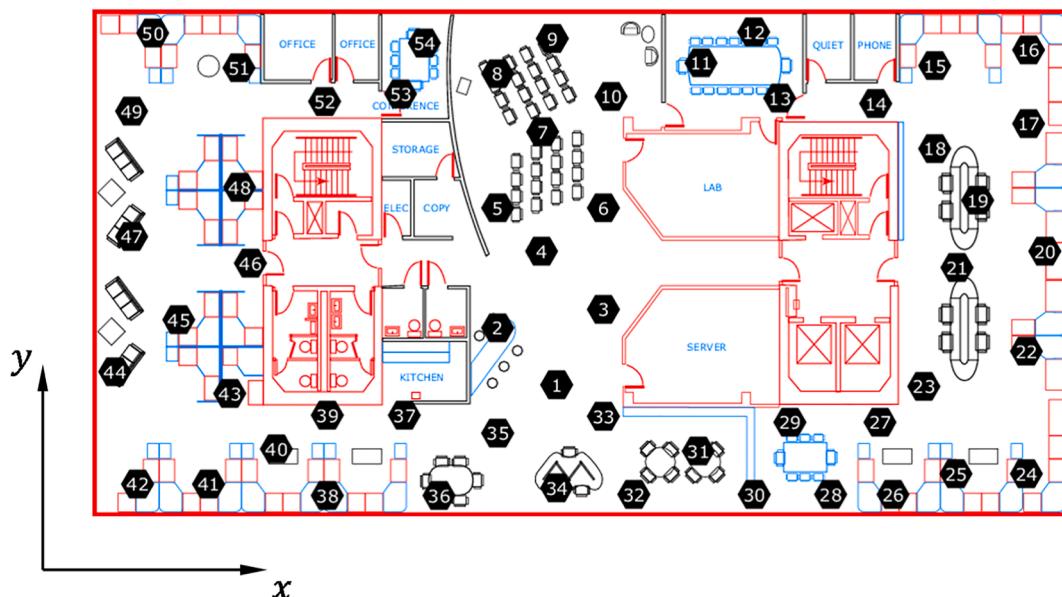
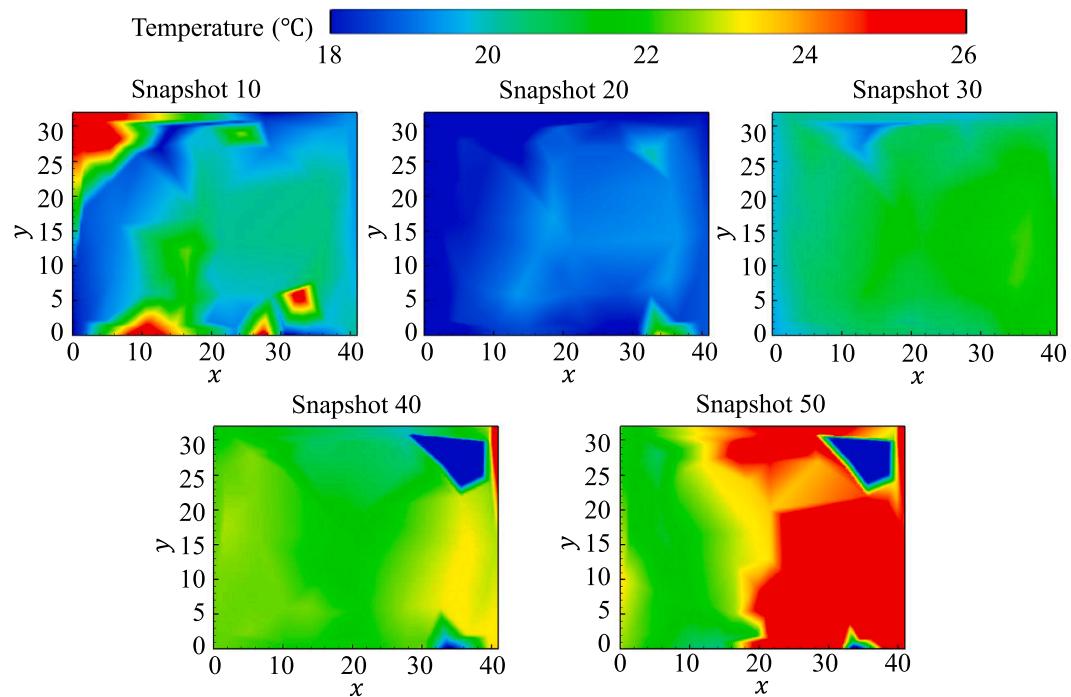
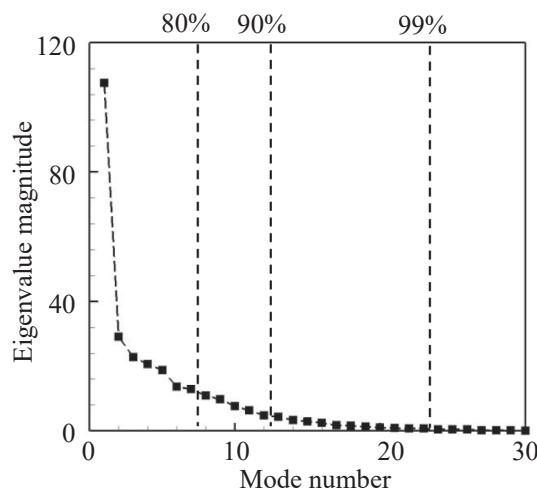


Fig. 2. The spatial layout of the Berkeley Intel Lab.



**Fig. 3.** The temperature profiles of the snapshots of the Berkeley Intel Lab dataset.



**Fig. 4.** The eigenvalue and ‘kinetic energy’ distribution of different POD modes.

temporal field. The first 7 modes contain 80%, the first 12 modes 90%, and the first 23 modes 99% of the total ‘kinetic energy’. We use 90% of the overall ‘kinetic energy’ and the first 12 modes in our computational experiment to calculate the weighted MI function, and we conduct a sensitivity analysis in subsection 5.5 to inform and assess the robustness of our results given this choice. The profiles of the first 12 POD modes are shown in Fig. 5.

The POD profile indicates the magnitude of local temperature changes in the training data of the Intel Berkeley Lab. The temperatures in the blue and red areas (large absolute mode magnitude in the contour) change significantly over time. The first mode shows the major pattern of this temperature field development. For example, there is a blue area in mode 1 around the bottom-right corner (top-left panel in Fig. 5), which indicates that the temperature there changes significantly over time. With the increase in mode number and the decrease of ‘kinetic energy’, the corresponding POD mode becomes less important to the

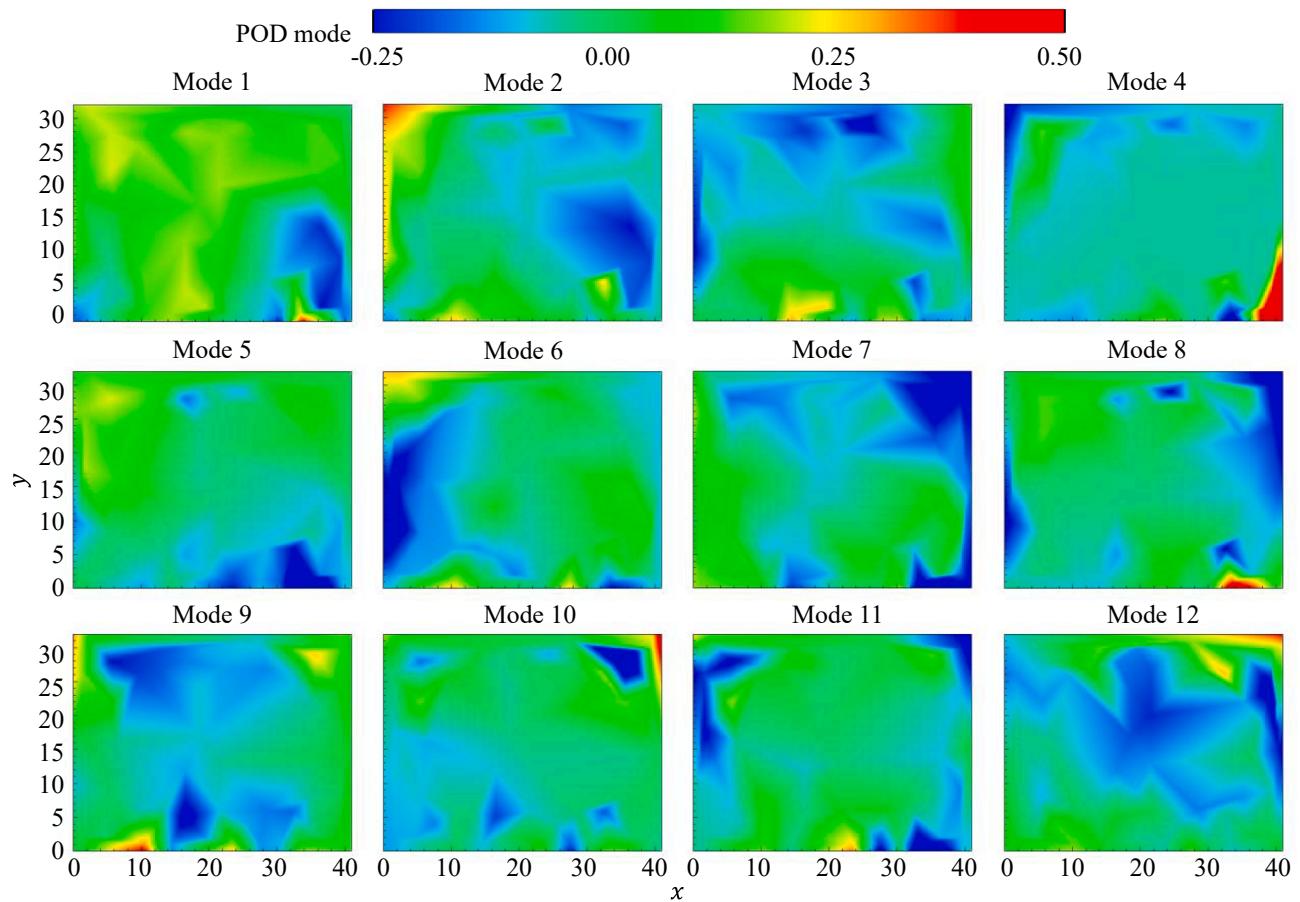
temperature field variations.

### 5.2. Multi-objective optimization result: The Pareto frontier

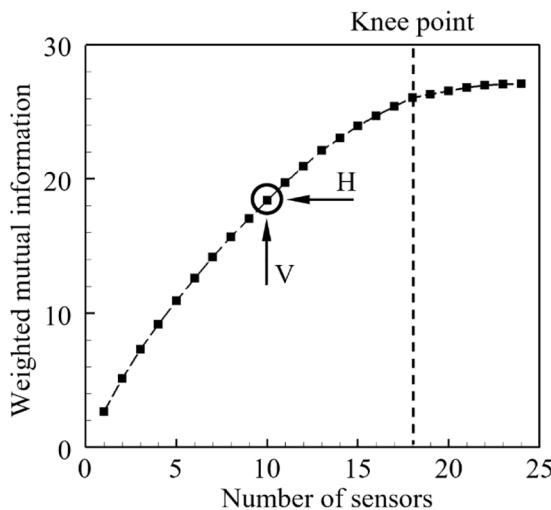
Next, we use our LG- $\epsilon$ -constraint optimization for the sensor placement task based on the POD results. The optimal weighted MI value versus sensor number, that is, the Pareto frontier of deployment cost and weighted MI function, is provided in Fig. 6. The non-decreasing characteristic of the weighted MI function holds until the number of sensors reaches 24.

The Pareto frontier shows the dominating configurations in the 2D space of weighted MI function and sensor numbers. This Pareto frontier provides the optimal sensor network configurations that achieve (1) the maximum weighted MI for a required sensor number, and (2) the minimum sensor number for a required environmental monitoring performance. Each point in the Pareto frontier corresponds to an optimal sensor configuration determined by the LG- $\epsilon$ -constraint algorithm.

For example, consider the point on the Pareto frontier with 10 sensors and weighted mutual information 18.4 as labeled by the circle in Fig. 6. This point indicates that (1) when the sensor number is limited to 10, the maximum weighted MI that any sensor placement can achieve is 18.4. The LG- $\epsilon$ -constraint algorithm also identifies the optimal placement that achieves this maximum weighted MI (V arrow in Fig. 6); (2) when the required weighted MI is 18.4, the minimum required sensor number to achieve this performance level is 10 (H arrow in Fig. 6). Next, we consider the computational cost associated with this algorithmic solution compared with an exhaustive search result. For the optimization of  $S_{10}^*$ , the brute force or exhaustive search algorithm would require examining  $\binom{54}{10} \cong 2.4 \times 10^{10}$  configurations and the calculation of the weighted MIs for all these configurations. We note that although the BnB is more computationally efficient than brute force method, it still exhibits exponential computational complexity. In sharp contrast, our LG- $\epsilon$ -constraint algorithm examines about  $54 \times 10 = 540$  configurations, and it exhibits polynomial computational complexity. As such the use of the LG heuristic is important to compute the entire Pareto frontier and the corresponding sensor configurations shown in Fig. 6. We will use the BnB to partially validate the Pareto frontier at special points of



**Fig. 5.** The profile of the first 12 POD modes with 90% of the ‘kinetic energy’



**Fig. 6.** The Pareto frontier of sensor number (cost) and weighted MI in the LG- $\epsilon$ -constraint optimization.

interest. We discuss next the details of a sensor configuration optimized by the LG- $\epsilon$ -constraint algorithm at the knee point of the Pareto frontier.

We calculate the knee point of the curve [49] for further analysis by the BnB exact algorithm. The knee point and optimal sensor number sets at  $|S| = 18$  in this curve. The optimal sensor placement  $S_{18}^*$  obtained by the LG heuristic with 18 sensors is shown in Fig. 7, and it provides a weighted  $MI_{LG} = 26.1$ .

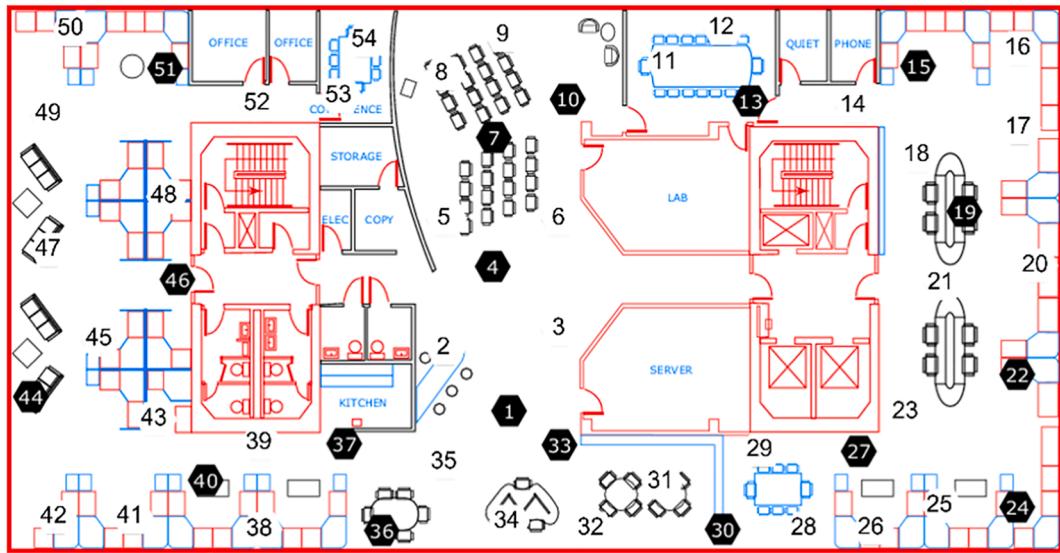
First, we note that our method arranges the configuration of sensors

with some level of spatial uniformity, that is, it distributes the sensors to cover the whole Berkeley Intel Lab. Second, the most important observation is that the sensors are placed at locations of large magnitude in the POD modes, especially in mode 1, which has the highest ‘kinetic energy’ as shown in Fig. 4. For example, our method places three sensors at node 22, 24, and 27, to supervise the temperature at the bottom-right area of the Lab, which has a large magnitude in the POD mode 1 distribution as shown in Fig. 5. This sensor placement configuration will result in a reduction of the state estimation error compared with alternative approaches, which we will discuss shortly. Next, we compare this result with that of the BnB global optimization.

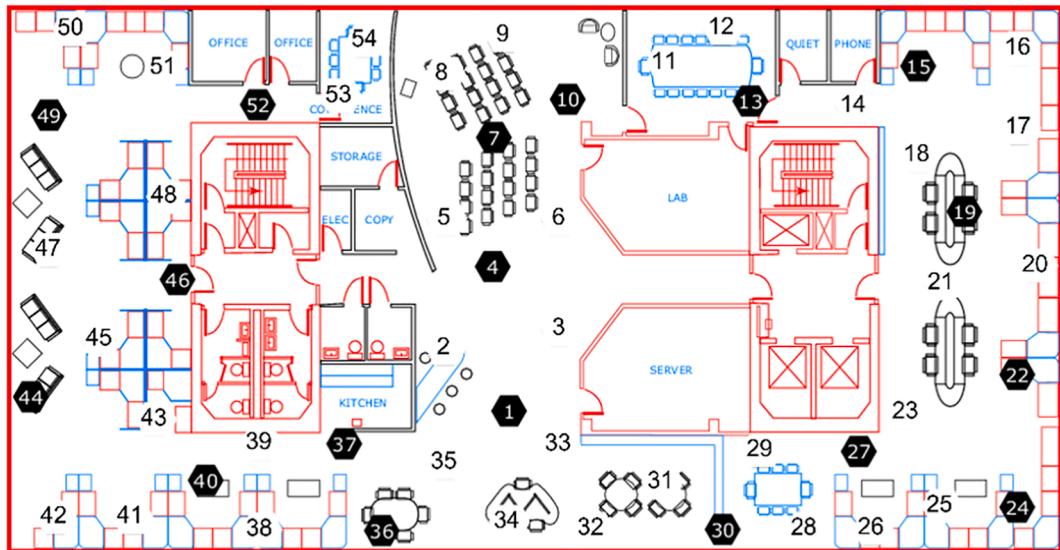
### 5.3. Branch and bound global optimal sensor placement

We leverage our BnB exact algorithm to calculate the global optimal weighted MI and sensor configuration at the knee point of the Pareto frontier  $|S| = 18$  to assess the quality of the previous LG result. The BnB global optimal configuration provides a weighted  $MI_{BnB} = 26.3$ , a minor improvement over that of the LG heuristic weighted  $MI_{LG} = 26.1$ . This result shows the LG result is no less than  $(1 - 1/e) \times 26.3 = 16.6$  and the  $(1 - 1/e)$  approximation is valid at this point in the Pareto frontier. The sensor placement layout result of BnB is shown in Fig. 8.

Comparing the sensor placement result of LG with that of BnB (Figs. 7 and 8), we find they distribute most of the sensors at the same nodes. The BnB moves two sensors from nodes 33 and 51 to nodes 49 and 52. Some discussion of this minor difference between these two optimal (global and local) configurations,  $S_{18,LG}^*$  and  $S_{18,BnB}^*$ , can be informative and is provided next. First, node 33 is in the immediate vicinity of node 1, and it does not provide much improvement to the overall state estimation accuracy when node 1 is selected (with both LG



**Fig. 7.** The sensor placement layout for the LG optimization result for sensor number 18 with the optimal sensor locations are indicated black hexagons.



**Fig. 8.** The sensor placement layout for the BnB global optimization result for sensor number 18 with the optimal sensor locations indicated by black hexagons.

and BnB). Second, the LG algorithm selects node 51 as the optimal choice at the top-left in Fig. 8, whereas the BnB selects the combination of two adjacent nodes, 49 and 52, in order to supervise temperature in that corner. Both these changes provide the marginal improvement in weighted MI of the BnB configuration over the LG,  $\Delta MI = 0.2$  or less than 1% relative change. Fig. 5 shows that the top-left area in the lab has a large magnitude in mode 2 and mode 4 (thin red and blue corners respectively), but small magnitude in mode 1. Given the result in Fig. 4, mode 2 and mode 4 have significantly smaller ‘kinetic energy’ than mode 1, which explains in part why the LG made the choice of a sensor placement at node 51, instead of the globally optimal choice of nodes 49 and 52. In other words, the signature of a marginal improvement in weighted MI was contained in mode 2 and 4, but its amplitude was small enough that it was discarded by the LG local heuristic.

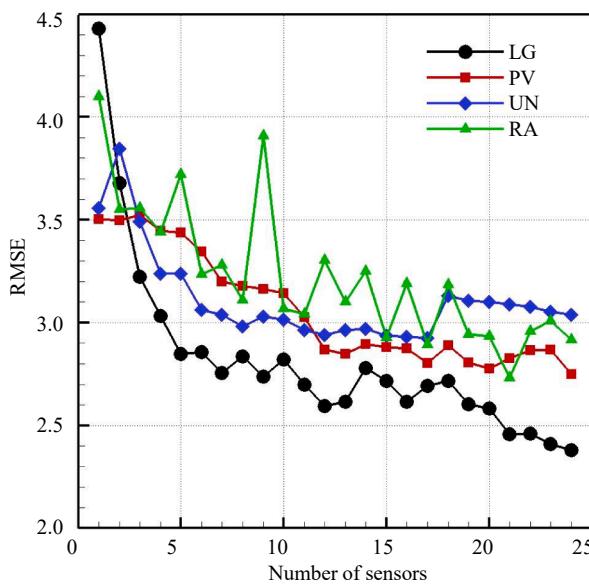
Overall, the overlap between the LG-optimal sensor configuration and the BnB-globally optimal configuration is significant, and the minor changes remain within the immediate vicinity of each other and lead to marginal improvement in weighted MI.

#### 5.4. Temperature state estimation RMSE

Having obtained an LG optimal sensor configuration, we are now ready for the ultimate test, the evaluation of the state estimation accuracy of our sensor placement method against alternative methods, namely the random placement method, the uniform placement method [58], and the predictive variance optimization method [35,59]. Details of these methods can be found in Appendix F. The state estimation accuracy is measured by the RMSE of the 20 testing temperature snapshots, as discussed in subsection 4.2. The RMSE results of different sensor placement methods are provided in Fig. 9.

Several observations can be made based on these results, the two most important are the following:

- First, our LG optimization provides consistent and significant improvement in terms of state estimation accuracy (testing RMSE) over the alternative methods. Notice, for example, that while the predictive variance optimization is the second-best placement method for sensor number larger than 12, it is outperformed by the



**Fig. 9.** RMSE results as a function of sensor numbers for different sensor placement optimization methods (LG- $\epsilon$ -constraint is labeled as LG, predictive variance as PV, uniform placement as UN, and random placement as RA).

uniform placement method and does not fair better than the random placement method when fewer sensors are considered. Its performance is not robust. Our LG method provides on average about 0.2 °C to 0.4 °C better estimation accuracy than the second-best alternative method. The relevance of this improvement is best assessed in light of the accuracy requirements of specific applications.

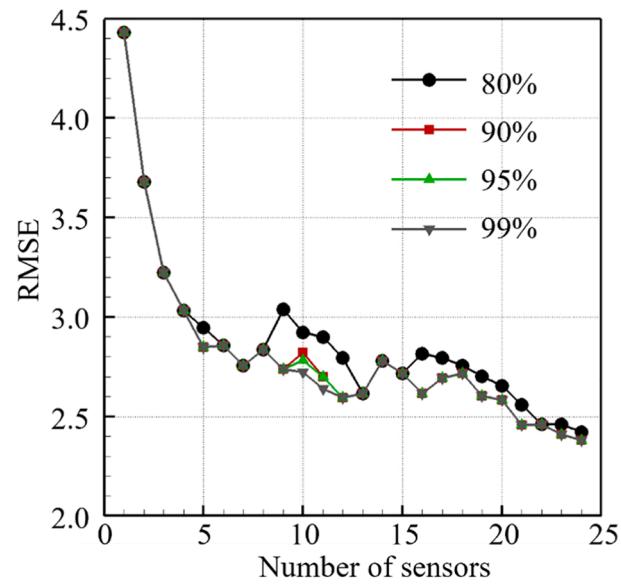
i. The overall trend in the testing RMSE of our LG optimization method decreases with increasing number of sensors. This corresponds to the improvement in weighted MI with sensor number shown in Fig. 6. As discussed in subsection 2.3, the maximization of the weighted MI can increase the ‘amount of information’ of the unobserved nodes conditioned on the sensor measurements, and consequently, it improves the state estimation accuracy. The result in Fig. 9 further supports the choice to use the weighted MI as a measure of the monitoring performance and to include it in our optimization method.

As a side note, the minor ‘oscillation’ in the RMSE of the LG curve in Fig. 9 (from sensor number 5 to 18) is due to the small discrepancies between the training and testing datasets, and the fact that the weighted MI is a proxy but not a direct measurement of the state estimation error.

##### 5.5. Sensitivity analysis of POD ‘kinetic energy’ threshold

To evaluate the effect of the POD ‘kinetic energy’ and the number of modes included in the sensor placement results, we conduct the following sensitivity analysis: we vary the POD ‘kinetic energy’ from 80% to 99% of the temperate field, and we examine the corresponding changes in RMSE of the LG- $\epsilon$ -constraint optimization for the testing dataset. The results are provided in Fig. 10.

Fig. 10 shows that while some RMSE improvements occur when accounting for 99% instead of 80% of the POD ‘kinetic energy’ (23 modes with the former instead of 7 with the latter), an almost perfect overlap is obtained with the 90%, 95%, and 99% RMSE curves. This result indicates that the state estimation accuracy is robust to the POD ‘kinetic energy’ in the LG optimization when 90% or more ‘kinetic energy’ is accounted for. In other words, 12 modes are sufficient for this particular application, and this sensitivity analysis can help inform the choice of the number of modes to include when our method is used in



**Fig. 10.** The testing RMSE sensitivity to the POD ‘kinetic energy’ from 80% to 99%

other applications.

## 6. Conclusion and future directions

In this work, we developed a novel sensor placement method that maximizes monitoring performance while minimizing the deployment cost. Our method integrates reduced order model (ROM) with Gaussian Process (GP) and multi-objective combinatorial optimization (MOCO). We first decomposed the spatio-temporal state field by proper orthogonal decomposition (POD). We used the GP to model the uncertainty of each POD mode. Next, we developed a LG- $\epsilon$ -constraint optimization to derive the Pareto-optimal sensor configurations based on the POD results. We devised a branch and bound exact algorithm for the sensor placement with a novel upper bound method to calculate the global optimal sensor placement and validate the correctness of the LG-derived Pareto frontier. We benchmarked our algorithm in the computational experiment based on the temperature dataset of the Berkeley Intel Lab. The computational results showed, for example, that our sensor placement algorithm locates sensors at locations with significant temperature change over time, and that our method achieves better state estimation accuracy and smaller RMSE and reconstruction error compared with alternative methods.

Our method and results should be considered in light of the following limitations, which point the way to several fruitful venues for future work. The first limitation is due to the fact that we used the weighted mutual information (MI) as an objective function. As shown in Fig. 6 of the Pareto frontier, the marginal improvement of the weighted MI decreases with the sensor number. The weighted MI reaches its peak at sensor number 24 in our application, and further increase of the sensor number would decrease the weighted MI value and break the  $(1 - 1/e)$  approximate rate of the LG solution. Consequently, the LG algorithm with weighted MI does not further guide the sensor placement when  $n > 24$ . In future work, we plan to extend the application of our method unconditionally to all sensor numbers up to  $|X|$ . The second limitation is related to the fact that the sensor placement results are contingent on the number of snapshots and epochs available of the spatio-temporal field to be monitored. We propose to conduct sensitivity analysis to help guide the selection of the number of epochs for our sensor placement method. Finally, our sensor placement method was validated using the Berkeley Intel Lab temperature dataset. We propose to conduct more extensive testing in the future and examine its applicability in other contexts, in

power distribution and other infrastructure systems for example, and for a host of environmental monitoring tasks.

#### CRediT authorship contribution statement

**Zhaoyi Xu:** Conceptualization, Methodology, Software, Validation, Writing – original draft. **Yanjie Guo:** Conceptualization, Writing – review & editing. **Joseph Homer Saleh:** Conceptualization, Investigation, Writing – review & editing, Supervision.

#### Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### Acknowledgments

This work was supported in part by a Space Technology Research Institute grant from NASA's Space Technology Research Grants Program.

## Appendix A. Knee point calculation in the Pareto frontier

In this appendix, we discuss the method to calculate the knee point of the Pareto frontier. We consider this knee point as the optimal solution in the Pareto set  $((x_1, F_1), (x_2, F_2), \dots, (x_n, F_n))$ .

First, we calculate the Menger curvature [60] of the curve for any point  $(x_i, F_i)$  ( $i = 2, 3, 4, \dots, n - 1$ ). The Menger curvature (MC) is calculated as Eq. (A.1).

$$\begin{aligned} MC(x_i) &= \frac{\sqrt{A - B^2}}{|pq||qr||rp|} \\ |pq| &= \sqrt{(x_i - x_{i-1})^2 + (F_i - F_{i-1})^2} \\ |qr| &= \sqrt{(x_{i+1} - x_i)^2 + (F_{i+1} - F_i)^2} \\ |rp| &= \sqrt{(x_{i+1} - x_{i-1})^2 + (F_{i+1} - F_{i-1})^2} \\ A &= 4|pq|^2|qr|^2 \\ B &= |pq|^2 + |qr|^2 - |rp|^2 \end{aligned} \quad (\text{A.1})$$

The knee point of this Pareto curve is at the maximization of the Menger curvature as Eq. (A.2).

$$KP = \operatorname{argmax}_{x_i} (MC(x_i), i = 2, 3, 4, \dots, n - 1) \quad (\text{A.2})$$

## Appendix B. The proof of the submodularity of the weighted mutual information function

According to Section 2.3 and Ref. [35], the  $(1 - 1/e)$  approximation ratio of the greedy solution holds when the weighted MI is submodular and non-decreasing. We prove the submodular and conditional non-decreasing properties of the weighted MI in the following steps.

Step I (proof of submodularity): For this weighted MI function in Eq. (9), because  $\lambda_i$  is positive, if all  $F_i(S)$  are submodule functions, the weighted sum,  $F(S)$  is a submodule function as well [61]. According to Ref. [35], the MI functions for different POD modes ( $F_i(S)$ ) are submodular. Consequently, the weighted MI function  $F(S)$  is submodular.

Step II (proof of non-decreasing): For alternative POD mode  $i$ , according to Section 2.3 and Ref. [35], when sensor number is smaller than a threshold,  $|S| \leq S_0^i$  the corresponding  $F_i(S)$  is non-decreasing. We calculate an overall threshold  $S_0$  by Eq. B. 1, where  $n$  is the POD modes number.

$$S_0 = \min_i \{S_0^1, S_0^2, \dots, S_0^n\} \quad (\text{B.1})$$

If  $|S| \leq S_0$ , we have  $|S| \leq S_0^i$  for alternative  $i$ . Then, the corresponding MI ( $F_i(S)$ ) is non-decreasing for all  $i$ , when  $|S| \leq S_0$ . Because  $\lambda_i$  is non-negative in Eq. (9), the weighted sum  $F(S)$  is non-decreasing.

Overall, we prove the submodularity and conditional non-decreasing (when  $|S| \leq S_0$ ) for the weighted MI objective function. With these two characteristics, the greedy heuristic solution to maximize weighted MI is  $(1 - 1/e)$ -approximate, when  $|S| \leq |S_0|$ . We guarantee  $|S| \leq |S_0|$  by checking the non-decreasing characteristic before increasing sensor number in Phase II of our MOCO as shown in Fig. 1.

## Appendix C. Pseudo-code of the lazy greedy algorithm

In this appendix, we provide the pseudo-code for the lazy greedy algorithm. The input of the function is the design space  $V$ , GP covariance matrix  $\Sigma_{VV}$ , sensor number limit  $k$ , and the MI function  $F$ . The output of the algorithm is the sensor placement  $S$ . The pseudo-code is shown in Algorithm C.1.

**Algorithm C.1:** Lazy greedy**Input:** design space  $V$ , GP covariance matrix  $\Sigma_{VV}$ , sensor number  $k$ , MI function  $F$ 


---

**Output:** Sensor placement  $S \subseteq V$

---

```

1   Function LazyGreedy( $V, \Sigma_{VV}, k, F$ ):
2      $S \leftarrow []$ ;
3     Foreach  $s \in V$  do
4        $\delta(s) \leftarrow 1000000$ ;
5       While  $|S| \leq k$  do
6         Foreach  $s \in V \setminus S$  do
7            $current(s) \leftarrow false$ ;
8           While true do
9              $s^* \leftarrow argmax_{s \in V \setminus S} \delta(s)$ ;
10            If  $current(s)$  then
11              break;
12             $\delta(s^*) \leftarrow F(S \cup s^*) - F(S)$ 
13             $current(s^*) \leftarrow true$ ;
14             $S \leftarrow S \cup s^*$ ;
15        Return  $S$ 

```

---

**Appendix D. Pseudo-code of the branch and bound global optimization**

Based on the  $(1 - 1/e)$ -approximation ratio characteristic of the Lazy greedy heuristic, we use lazy greedy to calculate the upper bound for the pruning to improve the efficiency of the algorithm. We develop a branch and bound global optimization as shown in Algorithm D.1. The input of the algorithm is the design space  $V$ , the GP covariance matrix  $\Sigma_{VV}$ , the sensor number  $k$ , MI function  $F$ , lower bound initialization  $lb$ , and the lower bound corresponding set  $S_{lb}$ . The lower bound is initialized by the Lazy greedy algorithm with the input of  $(V, \Sigma_{VV}, k, F)$ . The correctness of this BnB approach has been validated by a Brute force algorithm without the pruning by the lower bound as shown in Algorithm D.2.

**Algorithm D.1:** Branch and bound global optimization**Input:** design space  $V$ , GP covariance matrix  $\Sigma_{VV}$ , sensor number  $k$ , MI function  $F$ , lower bound  $lb$ , lower bound placement  $S_{lb}$ 


---

**Output:** Sensor placement  $S \subseteq V$

---

```

1   Function BnBGlobal( $V, \Sigma_{VV}, k, F, lb, S_{lb}$ ):
2      $List_{lb} \leftarrow [0, lb, S_{lb}, 0]$ ;
3      $List_{tree} \leftarrow [0, 0, [], k]$ ;
4     While  $List_{tree}^1 = []$  do
5        $currentNode \leftarrow List_{tree}[1]$ ;
6        $List_{tree}[1] = []$ ;
7        $[currentNum, currentR, currentSet, currentSenNum] \leftarrow currentNode$ ;
8       if  $currentNum < |V|$  do
9         if  $currentSenNum > 0$  do
10           $takeNum \leftarrow currentNum + 1$ ;
11           $takeSenNum \leftarrow currentSenNum - 1$ ;
12           $takeS \leftarrow [currentSet, x_{takeNum}]$ ;
13           $F_{take} \leftarrow F(takeS)$ ;
14          if  $takeSenNum > 0$  do
15             $V_{LG} \leftarrow V \setminus [x_1, x_2, \dots, x_{takeNum}]$ 
16             $S_{LG} \leftarrow LazyGreedy(V_{LG}, \Sigma_{V_{LG}V_{LG}}, takeSenNum, F)$  :
17            else do
18               $S_{LG} \leftarrow []$ ;
19               $takeUB \leftarrow F(takeS) + F(S_{LG}) \times (1/(1 - 1/e))$ ;
20              if  $takeUB > lb$  do
21                 $List_{tree} \leftarrow List_{tree} \cap [takeNum, F_{take}, takeS, takeSenNum]$ ;
22              if  $F_{take} > lb$  do
23                 $lb \leftarrow F_{take}$ ;
24                 $List_{lb} \leftarrow [takeNum, F_{take}, S_{take}, takeSenNum]$ ;
25                 $noNum \leftarrow currentNum + 1$ ;
26                 $noS \leftarrow currentSet$ ;
27                 $V_{LG} \leftarrow V \setminus [x_1, x_2, \dots, x_{noNum}]$ ;
28                 $S_{LG} \leftarrow LazyGreedy(V_{LG}, \Sigma_{V_{LG}V_{LG}}, currentSenNum, F)$  :
29                 $noUB \leftarrow F(noS) + F(S_{LG}) \times (1/(1 - 1/e))$ ;
30                if  $noUB > lb$  do
31                   $List_{tree} \leftarrow List_{tree} \cap [noNum, F(currentSet), noS, currentSenNum]$ ;
32                 $[-, -, S, -] \leftarrow List_{lb}$ ;
33        Return  $S$ 

```

---

**Algorithm D.2:** Brute-force global optimization

---

**Input:** design space  $V$ , GP covariance matrix  $\Sigma_{VV}$ , sensor number  $k$ , MI function  $F$ , lower bound  $lb$ , lower bound placement  $S_{lb}$ 


---

**Output:** Sensor placement  $S \subseteq V$

```

1   Function BnBGlobal( $V, \Sigma_{VV}, k, F, lb, S_{lb}$ ):
2      $List_{lb} \leftarrow [0, lb, S_{lb}, 0]$ ;
3      $List_{tree} \leftarrow [0, 0, [], k]$ ;
4     While  $List_{tree}! = []$  do
5       currentNode  $\leftarrow List_{tree}[1]$ ;
6        $List_{tree}[1] = []$ ;
7       [ $currentNum, currentR, currentSet, currentSenNum$ ]  $\leftarrow currentNode$ ;
8       if  $currentNum < |V|$  do
9         if  $currentSenNum > 0$  do
10          takeNum  $\leftarrow currentNum + 1$ ;
11          takeSenNum  $\leftarrow currentSenNum - 1$ ;
12          takeS  $\leftarrow [currentSet, x_{takeNum}]$ ;
13           $F_{take} \leftarrow F(takeS)$ ;
14           $List_{tree} \leftarrow List_{tree} \cap [takeNum, F_{take}, takeS, takeSenNum]$ ;
15          if  $F_{take} > lb$  do
16             $lb \leftarrow F_{take}$ ;
17             $List_{lb} \leftarrow [takeNum, F_{take}, S_{take}, takeSenNum]$ ;
18            noNum  $\leftarrow currentNum + 1$ ;
19            noS  $\leftarrow currentSet$ ;
20             $List_{tree} \leftarrow List_{tree} \cap [noNum, F(currentSet), noS, currentSenNum]$ ;
21             $[., ., S, .] \leftarrow List_{lb}$ ;
22        ReturnS

```

---

**Appendix E. The proof of the correctness of the upper bound calculation in branch and bound algorithm**

We suppose the optimal sensor placement conditioned on  $S_n$  is  $S_n^* = S_n \cup S_{opt}^{X_i}$ . Our objective is to prove that the best weighted MI of the sensor placement through the current node ( $F(S_n^*)$ ) is smaller than or equal to our upper bound  $F(S_n) + (1/(1 - 1/e)) \times F(S_{LG})$ . We prove this statement by the following steps.

Step I: According to the submodularity of the weighted MI function proved in Section 3.4, we can decompose  $F(S_n^*)$  by Eq. E. 1.

$$F(S_n^*) \leq F(S_n) + F(S_{opt}^{X_i}) \quad (\text{E1})$$

Step II: At first, we state that the weighted MI function is still non-decreasing at  $S_{LG}$  that  $|S_{LG}| \leq S_0$ , where  $S_0$  is the threshold value in Eq. B. 1. This statement is proved by Eq. E. 2, where  $|S_{knee}|$  is the sensor number at the knee point. Consequently, the non-decreasing and the  $(1 - 1/e)$ -approximation ratio of LG heuristic characteristics of  $F(\cdot)$  are valid at  $S_{LG}$ .

$$\begin{aligned} |S_{opt}^{X_i}| &\leq |S_n| + |S_{opt}^{X_i}| \\ &= |S_{knee}| \\ &\leq |S_0| \end{aligned} \quad (\text{E2})$$

According to the  $(1 - 1/e)$ -approximation ratio of LG heuristic to the weighted MI function, we have the inequality as Eq. E. 3.

$$F(S_{opt}^{X_i}) \leq (1/(1 - 1/e))F(S_{LG}) \quad (\text{E3})$$

Finally, we combine Eq. E. 1 and E. 3 and prove the correctness of the upper bound as Eq. E. 4.

$$\begin{aligned} F(S_{opt}^*) &\leq F(S_n) + F(S_{opt}^{X_i}) \\ &\leq F(S_n) + (1/(1 - 1/e))F(S_{LG}) \\ &= UB \end{aligned} \quad (\text{E4})$$

In addition, the correctness of our BnB and upper bound calculation has been proved computationally by the Brute force algorithm in Appendix D.

**Appendix F. Introductions of alternative sensor placements**

In this Appendix, we introduce the alternative method we compared. We compare the state estimation performance of our sensor placement with random placement, uniform placement [58], and a predictive variance method [35].

First, for the random placement, we generate 54 uniform random variables from (0,1) as the probability to choose the corresponding node. If we plan to select  $k$  sensors, we rank the probabilities and choose the top  $k$  with a larger probability for sensor location. This is a random placement method and used as a baseline to measure the effectiveness of the sensor placement.

Second, for the uniform method, we use a utility function ( $U(S)$ ), as shown in Eq. F.0.1, to measure the special uniformity of the sensor set. We then use the LG optimization to minimize the utility function for a uniform sensor distribution of the field.

$$U(S) = \min_{s_i, s_j \in S, i < j} |s_i - s_j| \quad (\text{F.1})$$

Finally, for the predictive variance placement, we use the predictive variance ( $u(x, S)$ ), as shown in Eq. F. 2, to measure the uncertainty of the spatial prediction. We use the LG to optimize this predictive variance. For more details of predictive sensor placement methods, the reader is referred to Ref. [35].

$$\begin{aligned} u(x, S) &= \sum_{s \in x} \operatorname{Var}(X_s) - \operatorname{Var}(X_s|x_S) \\ \operatorname{Var}(X_s) &= E[(X_s - E(X_s))^2] \\ \operatorname{Var}(X_s|x_S) &= E[(X_s - E(X_s|x_S))^2 | x_S] \end{aligned} \quad (\text{F.2})$$

## References

- [1] K. Islam, W. Shen, X. Wang, in: May. Security and privacy considerations for wireless sensor networks in smart home environments, IEEE, 2012, pp. 626–633.
- [2] Serna, M.Á., Sreenan, C.J. and Fedor, S., 2015, April. A visual programming framework for wireless sensor networks in smart home applications. In 2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP) (pp. 1-6). IEEE.
- [3] C.C. Castello, R.X. Chen, J. Fan, A. Davari, Context aware wireless sensor networks for smart home monitoring, *International Journal of Autonomous and Adaptive Communications Systems* 10 (2) (2013) 99–114.
- [4] A. Pascale, M. Nicoli, F. Deflorio, B. Dalla Chiara, U. Spagnolini, Wireless sensor networks for traffic management and road safety, *IET Intel. Transport Syst.* 6 (1) (2012) 67–77.
- [5] M. Bottero, B. Dalla Chiara, F.P. Deflorio, Wireless sensor networks for traffic monitoring in a logistic centre, *Transportation Research Part C: Emerging Technologies* 26 (2013) 99–124.
- [6] Y. Wen, J.L. Pan, J.F. Le, Survey on application of wireless sensor networks for traffic monitoring, in: In International Conference on Transportation Engineering 2007, 2007, pp. 2079–2084.
- [7] J. Duan, D. Gao, D. Yang, C.H. Foh, H.-H. Chen, An energy-aware trust derivation scheme with game theoretic approach in wireless sensor networks for IoT applications, *IEEE Internet Things J.* 1 (1) (2014) 58–69.
- [8] Q. Zhu, R. Wang, Q. Chen, Y. Liu, W. Qin, in: December. Iot gateway: Bridgingwireless sensor networks into internet of things, Ieee, 2010, pp. 347–352.
- [9] F. Wu, L. Xu, S. Kumari, X. Li, J. Shen, K.-K. Choo, M. Wazid, A.K. Das, An efficient authentication and key agreement scheme for multi-gateway wireless sensor networks in IoT deployment, *Journal of Network and Computer Applications* 89 (2017) 72–85.
- [10] S.L. Ullo, G.R. Sinha, Advances in Smart Environment Monitoring Systems Using IoT and Sensors, *Sensors* 20 (11) (2020) 3113.
- [11] G. Xing, R. Tan, B. Liu, J. Wang, X. Jia, C.W. Yi, September. Data fusion improves the coverage of wireless sensor networks, in: In Proceedings of the 15th annual international conference on Mobile computing and networking, 2009, pp. 157–168.
- [12] C. Yang, An adaptive sensor placement algorithm for structural health monitoring based on multi-objective iterative optimization using weight factor updating, *Mech. Syst. Sig. Process.* 151 (2021) 107363, <https://doi.org/10.1016/j.ymssp.2020.107363>.
- [13] C. Yang, A novel uncertainty-oriented regularization method for load identification, *Mech. Syst. Sig. Process.* 158 (2021) 107774, <https://doi.org/10.1016/j.ymssp.2021.107774>.
- [14] W. Ostachowicz, R. Soman, P. Malinowski, Optimization of sensor placement for structural health monitoring: A review, *Structural Health Monitoring* 18 (3) (2019) 963–988.
- [15] H.Z. Abidin, N.M. Din, N.A.M. Radzi, Z.I. Rizman, A review on sensor node placement techniques in wireless sensor networks, *International Journal on Advanced Science, Engineering and Information Technology* 7 (1) (2017) 190–197.
- [16] Du, D.Z. and Pardalos, P.M. eds., 2013. Minimax and applications (Vol. 4). Springer Science & Business Media.
- [17] Wei Lu, Runfa Wen, Jun Teng, Xiaoling Li, Chao Li, Data correlation analysis for optimal sensor placement using a bond energy algorithm, *Measurement* 91 (2016) 509–518.
- [18] Thaw Tar Thein Zan, Payal Gupta, Mengmeng Wang, Justin Dauwels, Abhisek Ukil, Multi-objective optimal sensor placement for low-pressure gas distribution networks, *IEEE Sens. J.* 18 (16) (2018) 6660–6668.
- [19] Yi Tan, Limao Zhang, Computational methodologies for optimal sensor placement in structural health monitoring: A review, *Structural Health Monitoring* 19 (4) (2020) 1287–1308.
- [20] Guang-Dong Zhou, Ting-Hua Yi, Mei-Xi Xie, Hong-Nan Li, Jun-Hong Xu, Optimal Wireless Sensor Placement in Structural Health Monitoring Emphasizing Information Effectiveness and Network Performance, *J. Aerosp. Eng.* 34 (2) (2021) 04020112, [https://doi.org/10.1061/\(ASCE\)AS.1943-5525.0001226](https://doi.org/10.1061/(ASCE)AS.1943-5525.0001226).
- [21] Guang-Dong Zhou, Ting-Hua Yi, Huan Zhang, Hong-Nan Li, Energy-aware wireless sensor placement in structural health monitoring using hybrid discrete firefly algorithm, *Structural Control and Health Monitoring* 22 (4) (2015) 648–666.
- [22] Gabriele D'Antona, Nima Seifnaragh, Analysis of the sensor placement for optimal temperature distribution reconstruction, *Measurement* 56 (2014) 58–69.
- [23] Kang He, Minping Jia, Conghu Liu, Zhuanzhe Zhao, Sensor layout optimization by integrating Bayesian approach to diagnose multi-station assembly processes, *Measurement* 146 (2019) 230–240.
- [24] Chengyu Hu, Liguo Dai, Xuesong Yan, Wenying Gong, Xiaobo Liu, Ling Wang, Modified NSGA-III for sensor placement in water distribution system, *Inf. Sci.* 509 (2020) 488–500.
- [25] Barthorpe, R.J. and Worden, K., 2009. Sensor placement optimization. *Encyclopedia of structural health monitoring*.
- [26] Padula, S.L. and Kincaid, R.K., 1999. Optimization strategies for sensor and actuator placement.
- [27] A. Krause, A. Singh, C. Guestrin, Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies, *Journal of Machine Learning Research* 9 (Feb) (2008) 235–284.
- [28] Chen Yang, Ke Liang, Xuepan Zhang, Strategy for sensor number determination and placement optimization with incomplete information based on interval possibility model and clustering avoidance distribution index, *Comput. Methods Appl. Mech. Eng.* 366 (2020) 113042, <https://doi.org/10.1016/j.cma.2020.113042>.
- [29] Chen Yang, Xinbin Hou, Shinan Chang, A synchronous placement and size-based multi-objective optimization method for heat dissipation design on antenna module of space solar power satellite, *Sustainable Energy Technol. Assess.* 45 (2021) 101183, <https://doi.org/10.1016/j.seta.2021.101183>.
- [30] Guang-Dong Zhou, Mei-Xi Xie, Ting-Hua Yi, Hong-Nan Li, Optimal wireless sensor network configuration for structural monitoring using automatic-learning firefly algorithm, *Adv. Struct. Eng.* 22 (4) (2019) 907–918.
- [31] R. Semaan, Optimal sensor placement using machine learning, *Comput. Fluids* 159 (2017) 167–176.
- [32] K. Kasper, L. Mathelin, H. Abou-Kandil, in: July. A machine learning approach for constrained sensor placement, IEEE, 2015, pp. 4479–4484.
- [33] X. Liu, S. Sun, X. Li, H. Qian, P. Zhou, Machine learning for noise sensor placement and full-chip voltage emergency detection, *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* 36 (3) (2016) 421–434.
- [34] F.Y.S. Lin, P.L. Chu, A near-optimal sensor placement algorithm to achieve complete coverage-discrimination in sensor networks, *IEEE Commun. Lett.* 9 (1) (2005) 43–45.
- [35] Krause, A., 2008. Optimizing sensing (Doctoral dissertation, PhD thesis, Carnegie Mellon University.(Section 5.7)).
- [36] Eric Schulz, Maarten Speekenbrink, Andreas Krause, A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions, *J. Math. Psychol.* 85 (2018) 1–16.
- [37] Brunton, B.W., Brunton, S.L., Proctor, J.L. and Kutz, J.N., 2013. Optimal sensor placement and enhanced sparsity for classification. *arXiv preprint arXiv: 1310.4217*.
- [38] Emily Clark, Travis Askham, Steven L. Brunton, J. Nathan Kutz, Greedy sensor placement with cost constraints, *IEEE Sens. J.* 19 (7) (2019) 2642–2656.
- [39] Rongrong Hou, Yong Xia, Qi Xia, Xiaoqing Zhou, Genetic algorithm based optimal sensor placement for L1-regularized damage detection, *Structural Control and Health Monitoring* 26 (1) (2019) e2274, <https://doi.org/10.1002/stc.v26.110.1002/stc.2274>.
- [40] Bartłomiej Blachowski, Andrzej Świercz, Mariusz Ostrowski, Piotr Tauzowski, Piotr Olaszek, Łukasz Jankowski, Convex relaxation for efficient sensor layout optimization in large-scale structures subjected to moving loads, *Comput.-Aided Civ. Infrastruct. Eng.* 35 (10) (2020) 1085–1100.
- [41] Pinaky Bhattacharyya, James Beck, Exploiting convexification for Bayesian optimal sensor placement by maximization of mutual information, *Structural Control and Health Monitoring* 27 (10) (2020), <https://doi.org/10.1002/stc.v27.10.1002/stc.2605>.
- [42] V. Kekatos, G.B. Giannakis, in: December. A convex relaxation approach to optimal placement of phasor measurement units, IEEE, 2011, pp. 145–148.
- [43] A. Damianou, N. Lawrence, April. Deep gaussian processes, in: In Artificial Intelligence and Statistics, 2013, pp. 207–215.

- [44] Yair Censor, Pareto optimality in multiobjective problems, *Appl. Math. Optim.* 4 (1) (1977) 41–59.
- [45] Chaofan Ma, Wei Liang, Meng Zheng, Delay constrained relay node placement in two-tiered wireless sensor networks: A set-covering-based algorithm, *Journal of Network and Computer Applications* 93 (2017) 76–90.
- [46] C. Ma, W. Liang, M. Zheng, in: May. Set-covering-based algorithm for delay constrained relay node placement in wireless sensor networks, IEEE, 2016, pp. 1–6.
- [47] S. Guiaşu, *Information theory with new applications*, McGraw-Hill Companies, 1977.
- [48] J. Clausen, Branch and bound algorithms-principles and examples, University of Copenhagen, Department of Computer Science, 1999, pp. 1–30.
- [49] Xingyi Zhang, Ye Tian, Yaochu Jin, A knee point-driven evolutionary algorithm for many-objective optimization, *IEEE Trans. Evol. Comput.* 19 (6) (2015) 761–776.
- [50] G. Berkooz, P. Holmes, J.L. Lumley, The proper orthogonal decomposition in the analysis of turbulent flows, *Annu. Rev. Fluid Mech.* 25 (1) (1993) 539–575.
- [51] J.C. Gabelle, J. Morchain, A. Liné, Kinetic energy transfer between first proper orthogonal decomposition modes in a mixing tank, *Chem. Eng. Technol.* 40 (5) (2017) 927–937.
- [52] A. Chatterjee, An introduction to the proper orthogonal decomposition, *Curr. Sci.* (2000) 808–817.
- [53] A. Deshpande, C. Guestrin, S.R. Madden, J.M. Hellerstein, W. Hong, August. Model-driven data acquisition in sensor networks, in: In Proceedings of the Thirtieth international conference on Very large data bases-Volume 30, 2004, pp. 588–599.
- [54] C.K. Williams, C.E. Rasmussen, Gaussian processes for machine learning Vol. 2, No. 3 (2006) p. 4).
- [55] Pike-Burke, C., 2019. Multi-objective optimization.
- [56] Michel Minoux, in: *Lecture Notes in Control and Information SciencesOptimization Techniques*, Springer-Verlag, Berlin/Heidelberg, 1978, pp. 234–243.
- [57] Madden, S., 2003. Intel Berkeley research lab data. USA: Intel Corporation, 2004 [2004-06-08]. <http://berkeley.intel-research.net/labdata.html>.
- [58] Ali Katanforoush, Mehrdad Shahshahani, Distributing points on the sphere, I. *Experimental Mathematics* 12 (2) (2003) 199–209.
- [59] K. Chaloner, I. Verdinelli, Bayesian experimental design: A review, *Statistical Science* (1995) 273–304.
- [60] J.C. Léger, Menger curvature and rectifiability, *Ann. Math.* 149 (3) (1999) 831–869.
- [61] Vladimir Kolmogorov, Minimizing a sum of submodular functions, *Discrete Appl. Math.* 160 (15) (2012) 2246–2258.