Approximate Solutions and Performance Bounds for the Sensor Placement Problem

Muhammad Uddin, Anthony Kuh, Aleksandar Kavcic
Dept. of Electrical Engineering
University of Hawaii
Honolulu, HI 96822
Email: uddin@hawaii.edu, kuh@hawaii.edu, kavcic@hawaii.edu

Toshihisa Tanaka
Dept. of Electrical and Electronic Engineering
Tokyo University of Agriculture and Technology
Tokyo, Japan
Email: tanakat@cc.tuat.ac.jp

***Abstract*—This paper considers the placement of m sensors at $n > m$ possible locations. Given noisy observations, knowledge of the state correlation matrix, and a mean square error criterion, the problem can be formulated as an integer programming problem. The solution for large m and n is infeasible, requiring us to look at approximate algorithms. Using properties of matrices, we come up with lower and upper bounds for the optimal solution performance. We also formulate a greedy algorithm and a dynamic programming algorithm that runs in polynomial time of m and n. Finally, we show through simulations that the greedy and dynamic programming algorithms very closely approximate the optimal solution. The sensor placement problem has many energy applications where we are often confronted with limited resources. Some examples include where to place environmental sensors for an area where there are large amounts of distributed solar PV and where to place grid monitors on an electrical distribution microgrid.**

I. Introduction

In applications ranging from the electrical power grid to the natural environment to biomedical monitoring there is a need to deploy sensors to monitor and estimate the behavior of different complex systems. Often the number of sensors that can be deployed is limited by cost and other constraints. This paper formulates the problem as an optimization problem that is solved by finding a solution to an integer programming problem.

There has been much previous work on the optimal placement of sensors. In Dhillon et. al. [3], [2] the optimal placement of sensors is considered where the probability of sensor detection depends upon distance with sensors placed on a two or three dimensional grid. Other research has considered sensor placement for the power grid by considering placement of Phase Measurement Units (PMUs) [1], [15]. In their research the problem is formulated as a state estimation problem with PMU placement depending on a key condition to make the system observable. In [12], a binary particle swarm optimization (BPSO) based method is used to minimize the required number of PMUs while maximizing measurement redundancy. In work by Krause et. al. [9] they model spatial phenomena as a Gaussian Process and consider placement of sensors again using optimal experimental design. The goal is to maximize mutual information and the problem again becomes a combinatorial optimization problem that is NP-complete [4]. This paper also discusses a greedy algorithm and shows that mutual information is submodular and is monotonically increasing for small number of sensors. Performance bounds are obtained for the greedy algorithm. Simpler reduced computation algorithms and robust algorithms are also considered. More recently in [10], PMU placement is considered in a different context where observability is assumed and the goal is to optimize experimental design using a different criterion. The solution involves solving an integer programming problem which is NP-complete [4]. However, an approximate greedy solution is found that gives good results and runs in polynomial time. Then the greedy algorithm is tied to submodular and monotonic functions where bounds can be obtained to the greedy algorithm in relationship to the optimal algorithm. An estimation-theoretic approach to the PMU placement problem is proposed in [11]; after posing system state estimation as a linear regression problem, a convex relaxation is developed to suboptimally solve the PMU placement problem.

We consider a static discrete optimization problem where we have n discrete node locations where we can deploy sensors and we have m sensors to place. We assume that variables are jointly Gaussian and the objective is to minimize the sum of the mean squared estimation error at all node locations. This has some close ties to [10], [9] and has applications to energy problems. Examples include placement of meters such as Advanced Metering Infrastructure (AMI) on the distribution grid or to deploy environmental resource sensors where distributed PV solar panels are located.

The optimization criteria we consider is also known as the A- optimality design of experiments maximizing the trace of the inverse of the information matrix also considered in [9], [10], [11]. A key difference between this paper and the others is that here we consider a variety of computationally efficient approximation algorithms for the sensor placement problem and come up with analytical upper and lower bounds (depending on eigenvalues of the correlation matrix) for the cost function of the optimal sensor placement.

Section II gives a formulation of the problem. Since the optimal solution is computationally difficult to obtain for large n and m, we present some ad-hoc solutions and lower bounds to the optimal solution in Section III. These include the First Order Approximate Solution (FOAS), greedy algorithm, and

978-1-4673-0911-0/12/$31.00 ©2012 IEEE

an algorithm based on dynamic programming. Next, in Section IV we find an upper bound to the optimal solution in terms of the eigenvalues of the correlation matrix. In Section V we present some simulation results for randomly generated data and data generated from the IEEE 14-bus test system [16]. Finally, Section VI summarizes results of this paper and suggests directions for further work.

Notation: Upper case and lower case letters denote random variables and their realizations, respectively; underlined letters stand for vectors; boldface upper case letters denote matrices, and $\mathbf{I}$ denotes the identity matrix; $(\cdot)^T$ and $\mathbf{E}(\cdot)$ stand for transposition and expectation, respectively.

II. Problem Statement

We assume that the state vector is $\underline{X} \in \mathbb{R}^n$ and the observation vector is $\underline{Y} \in \mathbb{R}^m$ where $m \leq n$. Sensors are all identical and give noisy readings of the state, where the variance of each sensor reading is σ^2. The model is described by

$$\underline{Y} = \mathbf{C}(\underline{X} + \sigma \underline{N}) \tag{1}$$

where $\underline{N}$ is an iid Gaussian vector with mean $\underline{\mathbf{0}}$ and covariance matrix $\mathbf{\Lambda}_{\underline{N}} = \mathbf{I}_n$ with $\mathbf{I}_n$ being an identity matrix of dimension n and $\underline{X}$ is jointly Gaussian with mean $\underline{\mathbf{0}}$ and covariance matrix $\mathbf{\Lambda}_{\underline{X}}$. $\underline{X}$ and $\underline{N}$ are jointly Gaussian and independent. $\mathbf{C}$ is a binary matrix with orthonormal rows, where each row has one '1'. In other words, $\mathbf{C}$ is composed of m rows of the $n \times n$ identity matrix $\mathbf{I}_n$. The positions of ones in the matrix $\mathbf{C}$ denote the position of the sensors.

The minimum mean squared estimator is given by [7]

$$\hat{\underline{X}}(\underline{Y}) = \mathbf{E}(\underline{X}|\underline{Y}) = \mathbf{E}\left(\underline{X}\,\underline{Y}^T\right)\mathbf{E}\left(\underline{Y}\,\underline{Y}^T\right)^{-1}\underline{Y}. \tag{2}$$

The error is defined by $\underline{\mathcal{E}} = \underline{X} - \hat{\underline{X}}(\underline{Y})$ and the error covariance matrix is given by [7]

$$\mathbf{E}\left(\underline{\mathcal{E}}\,\underline{\mathcal{E}}^T\right) = \mathbf{\Lambda}_{\underline{X}} - \mathbf{E}\left(\underline{X}\,\underline{Y}^T\right)\mathbf{E}\left(\underline{Y}\,\underline{Y}^T\right)^{-1}\mathbf{E}\left(\underline{Y}\,\underline{X}^T\right), \tag{3}$$

where

$$\mathbf{E}\left(\underline{X}\,\underline{Y}^T\right) = \mathbf{\Lambda}_{\underline{X}}\mathbf{C}^T,$$

and

$$\mathbf{E}\left(\underline{Y}\,\underline{Y}^T\right) = \mathbf{C}\mathbf{\Lambda}_{\underline{X}}\mathbf{C}^T + \sigma^2\mathbf{I}_m.$$

Our task is to find the matrix $\mathbf{C} = \mathbf{C}^*$ that minimizes the total error $\operatorname{tr}\mathbf{E}\left(\underline{\mathcal{E}}\,\underline{\mathcal{E}}^T\right)$.

Definition. *Let $C^{[m\times n]}$ denote the* **set** *of all $m \times n$ matrices composed of m rows of the identity matrix $\mathbf{I}_n$.* ■

The optimization problem is then given by

$$\mathbf{C}^* = \underset{\mathbf{C}\in C^{[m\times n]}}{\arg\min}\operatorname{tr}\mathbf{E}\left(\underline{\mathcal{E}}\,\underline{\mathcal{E}}^T\right) = \underset{\mathbf{C}\in C^{[m\times n]}}{\arg\min}\,\mathbf{E}\left(\underline{\mathcal{E}}^T\underline{\mathcal{E}}\right). \tag{4}$$

Since the first term in (3) (i.e., $\mathbf{\Lambda}_{\underline{X}}$) does not depend on the choice of matrix $\mathbf{C}$, we can restate the optimization problem as an equivalent maximization problem using the following definition.

Definition. *Let the* **efficacy** *of matrix $\mathbf{C}$ be defined as*

$$J(\mathbf{C}) \triangleq \operatorname{tr}\left\{\mathbf{E}\left(\underline{X}\,\underline{Y}^T\right)\mathbf{E}\left(\underline{Y}\,\underline{Y}^T\right)^{-1}\mathbf{E}\left(\underline{Y}\,\underline{X}^T\right)\right\} \tag{5}$$

$$= \operatorname{tr}\left\{\left[\mathbf{C}(\mathbf{\Lambda}_{\underline{X}} + \sigma^2\mathbf{I})\mathbf{C}^T\right]^{-1}\mathbf{C}\mathbf{\Lambda}_{\underline{X}}^2\mathbf{C}^T\right\}. \tag{6}$$

where the final equality follows from the properties of the trace operator. ■

The optimization problem (4) is then equivalent to

$$\mathbf{C}^* = \underset{\mathbf{C}\in C^{[m\times n]}}{\arg\max}\, J(\mathbf{C}). \tag{7}$$

We can further rewrite the efficacy to take advantage of the eigenstructure of the underlying matrices.

Definition. *Let $\bar{\mathbf{C}}$ denote the* **complement** *of $\mathbf{C}$, with constraints $\bar{\mathbf{C}} \in C^{[(n-m)\times n]}$ and $\bar{\mathbf{C}}\mathbf{C}^T = \mathbf{0}$. [Note that $\bar{\mathbf{C}}$ may not be unique.]* ■

We perform the eigendecomposition of $\mathbf{C}\mathbf{\Lambda}_{\underline{X}}\mathbf{C}^T = \mathbf{U}_{(\mathbf{C})}\mathbf{D}_{(\mathbf{C})}\mathbf{U}_{(\mathbf{C})}^T$ where the columns of $\mathbf{U}_{(\mathbf{C})}$ are the eigenvectors of $\mathbf{C}\mathbf{\Lambda}_{\underline{X}}\mathbf{C}^T$ and $\mathbf{D}_{(\mathbf{C})}$ is a diagonal matrix whose diagonal entries are the eigenvalues of $\mathbf{C}\mathbf{\Lambda}_{\underline{X}}\mathbf{C}^T$. Let $\lambda_{(\mathbf{C}),1}, \cdots, \lambda_{(\mathbf{C}),m}$ be the eigenvalues of $\mathbf{C}\mathbf{\Lambda}_{\underline{X}}\mathbf{C}^T$. Then we can rewrite (6) as

$$J(\mathbf{C}) = \operatorname{tr}\left\{\mathbf{U}_{(\mathbf{C})}\left[\mathbf{D}_{(\mathbf{C})} + \sigma^2\mathbf{I}\right]^{-1}\mathbf{D}_{(\mathbf{C})}^2\mathbf{U}_{(\mathbf{C})}^T\right\}$$
$$+ \operatorname{tr}\left\{\bar{\mathbf{C}}\mathbf{\Lambda}_{\underline{X}}\mathbf{C}^T\mathbf{U}_{(\mathbf{C})}\left[\mathbf{D}_{(\mathbf{C})} + \sigma^2\mathbf{I}\right]^{-1}\mathbf{U}_{(\mathbf{C})}^T\mathbf{C}\mathbf{\Lambda}_{\underline{X}}\bar{\mathbf{C}}^T\right\}. \tag{8}$$

The first term in (8) is the trace of a diagonal matrix and it contributes to the efficacy by summing the diagonal terms $\lambda_{(\mathbf{C}),i}^2/(\lambda_{(\mathbf{C}),i} + \sigma^2)$. The second term accounts for the state correlations. It is the second term that is most difficult to deal with when attempting to solve (7). Intuitively, we want to pick the matrix $\mathbf{C}$ such that the second term contributes considerably to the efficacy, i.e., we want to place sensors in locations that are highly correlated to the remaining states.

We can restate (8) using inner products (i.e., correlations).

Definition. *Let $\underline{g}_i$ be the* **vector of inner products** *between the i-th eigenvector in $\mathbf{U}_{(\mathbf{C})}$ and the columns of $\mathbf{C}\mathbf{\Lambda}_{\underline{X}}\bar{\mathbf{C}}^T$, i.e.,*

$$\underline{g}_i = \left(\bar{\mathbf{C}}\mathbf{\Lambda}_{\underline{X}}\mathbf{C}^T\right)\left(\mathbf{U}_{(\mathbf{C})}\underline{e}_i^T\right),$$

where $\underline{e}_i$ is the i-th unit row vector and $\left(\mathbf{U}_{(\mathbf{C})}\underline{e}_i^T\right)$ is the i-th eigenvector in $\mathbf{U}_{(\mathbf{C})}$. ■

The efficacy in (8) now takes the form

$$J(\mathbf{C}) = \sum_{i=1}^{m}\frac{\lambda_{(\mathbf{C}),i}^2}{\lambda_{(\mathbf{C}),i} + \sigma^2} + \sum_{i=1}^{m}\frac{\underline{g}_i^T\,\underline{g}_i}{\lambda_{(\mathbf{C}),i} + \sigma^2}. \tag{9}$$

We can now readily interpret the second term in (9) as the contribution of the *energy* in the correlations (between sensor readings and the remaining states) to the efficacy. Clearly, we would like to find a matrix $\mathbf{C}$ so that the eigenvalues are large and the measurements are maximally correlated to the remaining states.

Problem (7) is an integer programming problem of choosing m rows of the identity matrix $\mathbf{I}_n$ that maximize the efficacy in

(9). This can be solved by an exhaustive search requiring testing all $\binom{n}{m}$ possible choices of m rows. Even for a moderately sized n and m this becomes computationally infeasible. The sensor placement problem is in fact NP-complete [4].

III. Ad-hoc solutions: Lower bounds on optimal efficacy

Since the optimization in (7) is difficult to perform, we resort to approximate solutions. Each approximate solution is in fact an ad-hoc solution because the exact solution requires an exhaustive search. If $\mathbf{C}$ is an ad-hoc solution to (7), then it provides a lower bound on the optimal efficacy $J(\mathbf{C}^*)$, i.e., $J(\mathbf{C}) \leq J(\mathbf{C}^*)$. Therefore, the search for good (suboptimal) solutions to (7) is equivalent to constructing tight lower bounds on $J(\mathbf{C}^*)$. Here we consider approximate solutions to the optimization problem requiring much fewer than $O\left(\binom{n}{m}\right)$ computations.

A. First order approximate solution (FOAS)

This is a trivial solution to consider. Let $J(\underline{e}_k)$ be the efficacy of the k-th unit row vector, i.e., the efficacy of the sensor placed at the location of the k-th state variable when $m = 1$. Then using (6) we have

$$J(\underline{e}_k) = \sum_{i=1}^{n} \frac{\left(\underline{e}_k \, \mathbf{\Lambda}_X \, \underline{e}_i^T\right)^2}{\underline{e}_k \, \mathbf{\Lambda}_X \, \underline{e}_k^T + \sigma^2}. \tag{10}$$

We rank the vectors $\underline{e}_k$ in descending order of their efficacies $J(\underline{e}_k)$. For any arbitrary m, we pick the m highest ranked vectors $\underline{e}_k$ and stack them to be the rows of the approximate solution $\mathbf{C}_{FO}$. Clearly we have $J(\mathbf{C}_{FO}) \leq J(\mathbf{C}^*)$.

B. Greedy solution

A greedy algorithm obtains a solution to (7) by making a sequence of choices [5]. At each step t, it assumes that t sensor locations are fixed, and makes a greedy choice where to place the $(t+1)$-st sensor. Let $\mathbf{C}_G$ denote the solution provided by the greedy algorithm. The algorithm can be described by the following.

Greedy Algorithm [5]

1) Initialization: Set iteration $t = 1$ and choose $\mathbf{C}_G^{(t)} = \underline{e}^*$ such that $\underline{e}^* = \arg\max_{\underline{e} \in \mathbb{C}^{[1\times n]}} J(\underline{e})$.
2) Find $\underline{e}^* = \arg\max_{\substack{\underline{e}\in\mathbb{C}^{[1\times n]} \\ : \mathbf{C}_G^{(t)} \underline{e}^T = \mathbf{0}}} J\left(\begin{bmatrix} \mathbf{C}_G^{(t)} \\ \underline{e} \end{bmatrix}\right)$.
3) Set $\mathbf{C}_G^{(t+1)} = \begin{bmatrix} \mathbf{C}_G^{(t)} \\ \underline{e}^* \end{bmatrix}$.
4) Increment: $t \leftarrow t + 1$.
5) if $t = m$ set $\mathbf{C}_G = \mathbf{C}_G^{(t)}$ and stop, else go to 2. ■

Note that the greedy solution may not be optimal for even $m = 2$, but it has complexity $O(mn)$ which is much smaller than $O\left(\binom{n}{m}\right)$ required to find the optimal solution $\mathbf{C}^*$.

C. n-path greedy solution

We propose the n-path greedy method to compute n candidate solutions, where each of the n candidate solutions $\mathbf{C}_k^{(m)}$ is attained by starting the greedy algorithm using each of the unit row vectors $\underline{e}_k$. In the end, we choose the n-path greedy solution $\mathbf{C}_{nG}$ to be the best of the n different candidate solutions.

$$\mathbf{C}_{nG} = \underset{\mathbf{C} \in \left\{\mathbf{C}_1^{(m)}, \mathbf{C}_2^{(m)}, \cdots, \mathbf{C}_n^{(m)}\right\}}{\arg\max} J(\mathbf{C}).$$

The n-path greedy algorithm runs in polynomial time. It has complexity $O(mn^2)$ which is larger than the $O(mn)$ complexity of the plain greedy algorithm in Section III-B, but the n-path greedy algorithm performs better than the plain greedy algorithm, thus giving a tighter lower bound $J(\mathbf{C}_{nG})$ on the optimal efficacy $J(\mathbf{C}^*)$, i.e., $J(\mathbf{C}_G) \leq J(\mathbf{C}_{nG}) \leq J(\mathbf{C}^*)$.

D. Dynamic programming solution

We propose the dynamic algorithm to solve the optimization problem (7) by dividing the problem into smaller subproblems [5], [8]. We use a *bottom-up* approach to rank the subproblems in terms of their problem sizes, smallest first. We save the intermediate solutions of the subproblems in a table and later use them to solve larger subproblems. For our optimization problem, we define a subproblem of size (number of sensors) t as finding the best candidate solution of size $t-1$ for a newly added sensor in a fixed location. Therefore, for any arbitrary number of sensors t, we have n subproblems of size t. Let $\mathbf{C}_j^{(t)}$ be the solution to a subproblem of size t, where $t \in \{1, 2, \cdots, m\}$ is the number of sensors and $j \in \{1, 2, \cdots, n\}$ is the index of the subproblem. The goal of the dynamic algorithm is to append the best existing solution $\mathbf{C}_j^{(t-1)}$ to a fixed $\underline{e}_k$ and thus construct a solution for a subproblem of size t. In other words, the dynamic programming algorithm is the *backtraced* version of the n-path greedy algorithm.

Let $\mathbf{C}_{DP}$ denote the approximate solution to (7) computed by the dynamic programming algorithm, in terms of the solutions of the subproblems as

$$\mathbf{C}_{DP} = \underset{\mathbf{C} \in \left\{\mathbf{C}_1^{(m)}, \mathbf{C}_2^{(m)}, \cdots, \mathbf{C}_n^{(m)}\right\}}{\arg\max} J(\mathbf{C}),$$

where $\left\{\mathbf{C}_1^{(m)}, \mathbf{C}_2^{(m)}, \cdots, \mathbf{C}_n^{(m)}\right\}$ is the set of the solutions to the subproblems of size m. The following procedure implements the dynamic algorithm.

Dynamic Programming Algorithm

1) Initialization: Set iteration $t = 1$ and inital matrices $\mathbf{C}_1^{(1)} = \underline{e}_1$, $\mathbf{C}_2^{(1)} = \underline{e}_2$, $\cdots$, $\mathbf{C}_n^{(1)} = \underline{e}_n$.
2) Set $k \leftarrow 1$.
3) Find $j^* = \arg\max_{j : \mathbf{C}_j^{(t)} \underline{e}_k^T = \mathbf{0}} J\left(\begin{bmatrix} \mathbf{C}_j^{(t)} \\ \underline{e}_k \end{bmatrix}\right)$.
4) $\mathbf{C}_k^{(t+1)} = \begin{bmatrix} \mathbf{C}_{j^*}^{(t)} \\ \underline{e}_k \end{bmatrix}$.
5) $k \leftarrow k + 1$.
6) if $k = n$, continue to step 7, else go to step 3.
7) $t \leftarrow t + 1$.
8) if $t = m$, set $\mathbf{C}_{DP} = \arg\max_{\mathbf{C} \in \left\{\mathbf{C}_1^{(t)}, \cdots, \mathbf{C}_n^{(t)}\right\}} J(\mathbf{C})$ and stop, else go to 2. ■

The dynamic programming algorithm has the same complexity $O(mn^2)$ as the n-path greedy algorithm, and performs better than the plain greedy approximation, i.e., $J(\mathbf{C}_G) \leq J(\mathbf{C}_{DP}) \leq J(\mathbf{C}^*)$. However, we cannot provide an a-priori

comparison between $J(\mathbf{C}_{nG})$ and $J(\mathbf{C}_{DP})$ without explicitly computing both values.

IV. Upper bound on optimal efficacy

It is clear from the previous section that there exists numerous ways of obtaining a lower bound on the optimal efficacy $J(\mathbf{C}^*)$. However, to evaluate the performance of these lower bounds we want to obtain a numerically computable upper bound for the difference $J(\mathbf{C}^*) - J(\mathbf{C})$. One way to achieve this goal is to find a numerically computable upper bound, say $\bar{J}$, on the optimal $J(\mathbf{C}^*)$ such that

$$J(\mathbf{C}^*) - J(\mathbf{C}) \leq \bar{J} - J(\mathbf{C}).$$

Hence, we devote this section to finding an upper bound $\bar{J}$ on the optimal efficacy $J(\mathbf{C}^*)$ by relaxing conditions on $\mathbf{C}$.

Definition. *For $m \leq n$, let $\mathcal{F}^{[m\times n]}$ be the* **set** *of all $m \times n$ matrices with rank m.* ■

Similar to (7) we formulate an optimization problem with the relaxed constraint as

$$\begin{aligned}\mathbf{F}^* &= \arg\max_{\mathbf{F}\in\mathcal{F}^{[m\times n]}} J(\mathbf{F}) \\ &= \arg\max_{\mathbf{F}\in\mathcal{F}^{[m\times n]}} \operatorname{tr}\left\{\left[\mathbf{F}(\mathbf{\Lambda}_{\underline{X}} + \sigma^2\mathbf{I})\mathbf{F}^T\right]^{-1}\mathbf{F}\mathbf{\Lambda}_{\underline{X}}^2\mathbf{F}^T\right\}.\end{aligned} \tag{11}$$

Since $C^{[m\times n]} \subseteq \mathcal{F}^{[m\times n]}$, we have $\bar{J} = J(\mathbf{F}^*) \geq J(\mathbf{C}^*)$. The following theorem provides a method for computing $J(\mathbf{F}^*)$ in closed form.

Theorem 1 (Upper bound)**.** *If $\mathbf{\Lambda}_{\underline{X}}$ is a non-negative definite matrix with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_n \geq 0$, then*

$$J(\mathbf{F}^*) = \sum_{j=1}^{m} \frac{\lambda_j^2}{\lambda_j + \sigma^2}.$$

Proof: From (11) we have

$$J(\mathbf{F}) = \operatorname{tr}\left\{\left[\mathbf{F}(\mathbf{\Lambda}_{\underline{X}} + \sigma^2\mathbf{I})\mathbf{F}^T\right]^{-1}\mathbf{F}\mathbf{\Lambda}_{\underline{X}}^2\mathbf{F}^T\right\}. \tag{12}$$

Let the eigendecomposition of $\mathbf{\Lambda}_{\underline{X}}$ be given by

$$\mathbf{\Lambda}_{\underline{X}} = \mathbf{U}\mathbf{D}\mathbf{U}^T, \tag{13}$$

where $\mathbf{U}$ contains the eigenvectors of $\mathbf{\Lambda}_{\underline{X}}$ as columns and $\mathbf{D}$ is a diagonal matrix with diagonal entries $\lambda_1 \geq \cdots \geq \lambda_n \geq 0$. Since

$$\mathbf{G} = \mathbf{F}(\mathbf{\Lambda}_{\underline{X}} + \sigma^2\mathbf{I})\mathbf{F}^T \tag{14}$$

is symmetric positive definite, there exists an $m \times m$ invertible matrix $\mathbf{G}^{1/2}$ such that

$$\mathbf{G} = \mathbf{G}^{1/2}(\mathbf{G}^{T/2}). \tag{15}$$

Define $\mathbf{B}$ of size $m \times n$ as

$$\mathbf{B} = \mathbf{G}^{-1/2}\mathbf{F}\mathbf{U}(\mathbf{D} + \sigma^2\mathbf{I})^{1/2}. \tag{16}$$

Combining (14) - (16) we have

$$\mathbf{B}\mathbf{B}^T = \mathbf{I}_m, \tag{17}$$

which means that the rows in $\mathbf{B}$ are orthonormal. From (16) we also have

$$\mathbf{F}\mathbf{U} = \mathbf{G}^{1/2}\mathbf{B}(\mathbf{D} + \sigma^2\mathbf{I})^{-1/2}, \tag{18}$$

and the efficacy now reads

$$\begin{aligned}J(\mathbf{F}) &= \operatorname{tr}\left[\mathbf{G}^{-1}\mathbf{F}\mathbf{\Lambda}_{\underline{X}}^2\mathbf{F}^T\right] \\ &= \operatorname{tr}\left[\mathbf{G}^{-1}\mathbf{F}\mathbf{U}\mathbf{D}^2\mathbf{U}^T\mathbf{F}^T\right] \\ &= \operatorname{tr}\left[\mathbf{G}^{-1}\mathbf{G}^{1/2}\mathbf{B}(\mathbf{D} + \sigma^2\mathbf{I})^{-1/2}\mathbf{D}^2(\mathbf{D} + \sigma^2\mathbf{I})^{-1/2}\mathbf{B}^T(\mathbf{G}^{T/2})\right] \\ &= \operatorname{tr}\left[\mathbf{B}\mathbf{\Sigma}\mathbf{B}^T\right],\end{aligned} \tag{19}$$

where

$$\mathbf{\Sigma} = \mathbf{D}^2(\mathbf{D} + \sigma^2\mathbf{I})^{-1} = \begin{pmatrix} \frac{\lambda_1^2}{\lambda_1+\sigma^2} & & 0 \\ & \ddots & \\ 0 & & \frac{\lambda_n^2}{\lambda_n+\sigma^2} \end{pmatrix}. \tag{20}$$

Under the constraint $\mathbf{B}\mathbf{B}^T = \mathbf{I}_m$, the efficacy in (19) is maximized when $\mathbf{B}$ consists of the eigenvectors of $\mathbf{\Sigma}$. Since $\mathbf{\Sigma}$ is diagonal, eigenvectors of $\mathbf{\Sigma}$ are the natural basis vectors (unit column vectors) $\underline{e}_k^T$. Thus, $J(\mathbf{F}^*)$ becomes

$$J(\mathbf{F}^*) = \sum_{j=1}^{m} \frac{\lambda_j^2}{\lambda_j + \sigma^2}. \tag{21}$$

■

Corollary 2 (Special case)**.** *If $\sigma^2 = 0$, then*

$$J(\mathbf{F}^*) = \sum_{j=1}^{m} \lambda_j.$$

■

Corollary 3 (Lower bound)**.**

$$J(\mathbf{C}^*) \geq \sum_{j=n-m+1}^{n} \frac{\lambda_j^2}{\lambda_j + \sigma^2}.$$

■

V. Simulation results

We considered different simulation scenarios to evaluate the performances of the approximate solutions and the upper bound. In Section V-A we considered realizations of the covariance matrix $\mathbf{\Lambda}_{\underline{X}}$ generated at random for different system sizes $n = 10$, 20, and 50. Next we used the standard IEEE 14-bus test system [16] in Section V-B to simulate the performance of our proposed algorithms.

A. Simulations with data generated at random

First we consider the case where $n = 10$. We generated 100 realizations of $\mathbf{\Lambda}_{\underline{X}}$ at random; σ^2 was kept constant for all realizations of $\mathbf{\Lambda}_{\underline{X}}$. We compared the efficacies of FOAS, optimal, greedy, n-path greedy, and dynamic programming solutions and the upper bound for each of the 100 realizations. Fig. 1 shows the average efficacies and upper bound, averaged

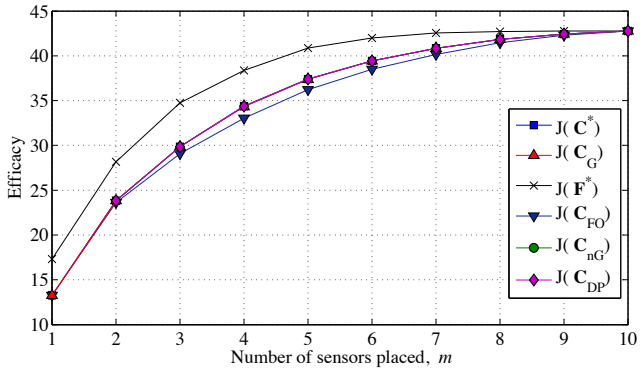

Fig. 1. $n = 10$: average efficacies of ad-hoc solutions and the upper bound $J(\mathbf{F}^*)$ compared to the average optimal efficacy $J(\mathbf{C}^*)$

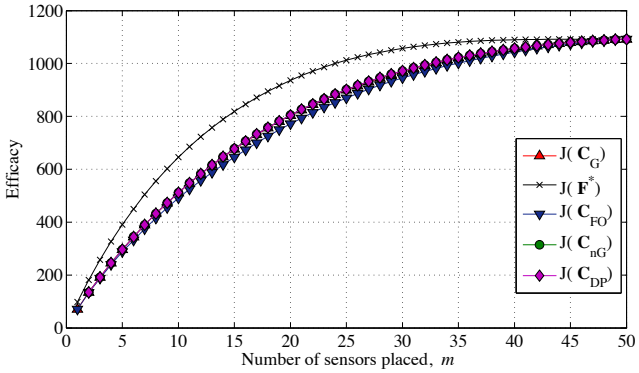

Fig. 3. $n = 50$: average efficacies of ad-hoc solutions and the upper bound $J(\mathbf{F}^*)$

over the 100 realizations. From Fig. 1 we note that the FOAS gets reasonably close to the optimal solution for each value of m. The upper bound based on Section IV is not tight. The greedy solution, the n-Path greedy solution, and the dynamic programming solution are indistinguishable from the optimal solution.

We then considered the case for $n = 20$. From Fig. 2 we note that the FOAS gets reasonably close to the optimal solution for

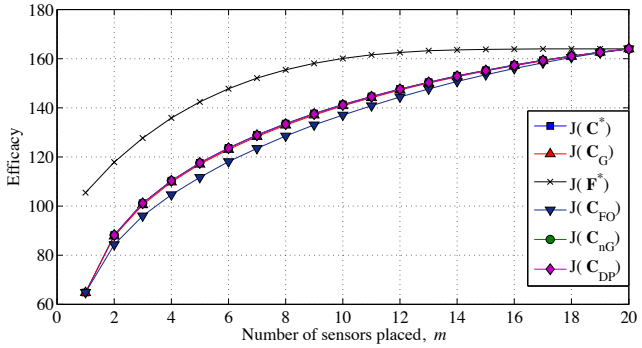

Fig. 2. $n = 20$: average efficacies of ad-hoc solutions and the upper bound $J(\mathbf{F}^*)$ compared to the average optimal efficacy $J(\mathbf{C}^*)$

each value of m. Again, the upper bound based on Section IV is not tight and actually worse than in the $n = 10$ case. Here the greedy solution performs just slightly worse than the optimal solution, the n-path greedy, and the dynamic programming algorithm.

We then considered the case for $n = 50$. For this case we could not compute the optimal solution because of the prohibitive complexity when $n = 50$. The FOAS again gets reasonably close to other approximate solutions, as shown in Fig. 3. The upper bound here is tighter than the upper bounds for the $n = 10$ and $n = 20$ cases. Here the greedy solution, n-Path greedy solution, and dynamic programming solution are almost identical.

From these simulation results there is not much improvement over the greedy solution by using the dynamic programming solution. Even the FOAS gets reasonably close to the optimal solution. The upper bound $J(\mathbf{F}^*)$ provides a bound on how close the approximate (ad-hoc) solutions are to the (unattainable) optimal solution.

B. IEEE 14-bus test system

We next apply the proposed sensor placement algorithms to the IEEE 14-bus test system [16]. The bus diagram of the test

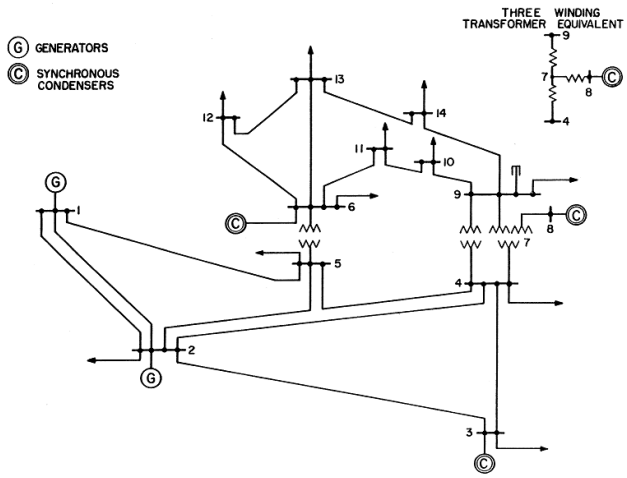

Fig. 4. IEEE 14-bus test system [16]

system is shown in Fig. 4. For simplicity, we only consider the voltage amplitudes at each bus as the states for our simulations. Without loss of generality, we assume that the conventional measurements of the test bus system are available to us. We further assume that a sensor is always placed at the swing bus [6] and therefore, the swing bus is not considered in our sensor placement algorithms.

We used the state estimator in the MATPOWER software [17] to construct the covariance matrix $\mathbf{\Lambda}_{\underline{X}}$ for the test bus system. In the state estimation process, the standard deviations for voltage magnitude, bus power injections, and line power flows measurements are 0.01, 0.015, and 0.02, respectively, in accordance with the setups in [17]. The simulations results for

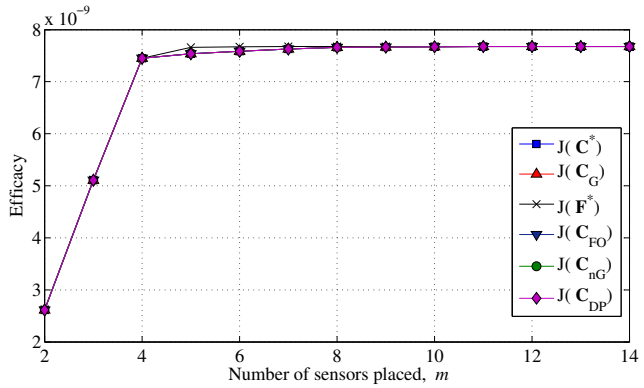

Fig. 5. IEEE 14-bus test case: efficacies of ad-hoc solutions and the upper bound $J(\mathbf{F}^*)$ compared to the optimal efficacy $J(\mathbf{C}^*)$

the proposed algorithms are shown in Fig. 5. It is observed that all of the proposed solutions achieve the optimal solution for this case. The upper bound is also tight. Fig. 5 further reveals that $m = 4$ sensors achieve roughly the same efficacy as $m = 14$ sensors. Therefore, a system with $m = 4$ sensors suffices for the purpose of monitoring the IEEE 14-bus test system if the mean squared error is the cost function.

VI. Summary and Further Directions

This paper considers the optimal placement of m sensors among n locations given states and observations are jointly Gaussian. The optimal solution involves solving an integer programming problem and is NP-complete [4]. We came up with lower and upper bounds for the optimal solution that are based on the eigenvalues of the correlation matrix. Several approximate solutions are considered with the greedy algorithm being simple and having good performance.

There are many further directions for this research. We would like to come up with tighter upper bounds for the optimal performance. Through simulations we showed that many of the approximation algorithms perform quite well when compared to the optimal algorithm. It would be interesting to come up with tighter theoretical lower bounds. Perhaps we can find bounds dependent on the structure of the correlation matrix, $\mathbf{\Lambda}_{\underline{X}}$.

We also want to consider particular applications such as the electrical power grid at the distribution level which has specific local and hierarchical correlations that can be represented by graphs that for the most part are radial. In these cases perhaps distributed algorithms can be used to find good approximate solutions to the sensor placement problem. This paper considered a static estimation problem. An interesting consideration would be to consider sensor placement for a dynamic state space model.

Acknowledgment

This work was supported in part by NSF grants ECCS-098344, 1029081, DOE grant DE-OE0000394, and the University of Hawaii REIS project.

References

[1] J.Chen and A. Abur, "Placement of PMUs to enable bad data detection in state estimation," *IEEE Trans. on Power Systems*, vol. 21, no. 4, pp. 1608-1614, Nov. 2006.

[2] S. S. Dhillon and K. Chakrabarty, "Sensor placement for effective coverage and surveillance in distributed sensor networks," *Proc. IEEE Wireless Commun. and Networking Conf.*, vol. 3, pp. 1609-1615, 2003.

[3] S. S. Dhillon, K. Chakrabarty, and S. S. Iyengar, "Sensor placement for grid coverage under imprecise detections," *Proc. 5th Int. Conf. Information Fusion*, vol. 2, pp. 1581-1587, 2002.

[4] D. J. Brueni and L. S. Heath, "The PMU placement problem," *SIAM Journal on Discrete Mathematics*, vol. 19, no. 3, pp. 744-761, Dec. 2005.

[5] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. Cambridge, MA: MIT Press, 2009.

[6] A. G. Phadke and J. S. Thorp, *Synchronized Phasor Measurements and Their Applications*, Springer, 2008.

[7] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Englewood Cliffs, NJ: PTR Prentice Hall, 1993.

[8] G. D. Forney, "The Viterbi algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268-278, Mar. 1973.

[9] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in Gaussian processes: theory, efficient algorithms, and empirical studies," *Journal of Machine Learning Research*, vol. 9, pp. 235-284, 2008.

[10] Q. Li, R. Negi, and M. Ilic, "Phasor measurement units placement for power system state estimation: a greedy approach," *Proc. IEEE PES General Meeting*, Detroit, MI, Jul. 2011.

[11] V. Kekatos and G. B. Giannakis, "A Convex Relaxation Approach to Optimal Placement of Phasor Measurement Units," *4th IEEE Intl. Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, San Juan, Puerto Rico, Dec. 2011.

[12] S. Chakrabarti, G. K. Venayagamoorthy, and E. Kyriakides, "PMU Placement for Power System Observability using Binary Particle Swarm Optimization," *Proc. Australasian Universities Power Engineering Conf.*, Sydney, Australia, Dec. 2008.

[13] G. Nemhauser, L. Wolsey, and M. Fisher, "An analysis of the approximations for maximizing submodular set functions," *Mathematical Programming*, vol 14, pp. 265-294, 1978.

[14] M. Asprou and E. Kyriakides, "Optimal PMU Placement for Improving Hybrid State Estimator Accuracy," *Proc. IEEE PowerTech*, Trondheim, Norway, Jun. 2011.

[15] B. Xu and A. Abur. "Observability analysis and measurement placement for system with PMUs" *Proc. IEEE PES Power Systems Conf. Expo*, New York, NY, Oct. 2004, pp. 943-946.

[16] Power systems test case archive. University of Washington. [Online]. Available: http://www.ee.washington.edu/research/pstca/

[17] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, "MATPOWER: steady-state operations, planning and analysis tools for power systems research and education," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12-19, Feb. 2011.