Sensor Network Design for Inverse Problems

Dariusz Uciński

***Abstract*— The aim of this work is to expose an optimal node activation algorithm in sensor networks whose measurements are supposed to be used to estimate unknown parameters of the underlying process model in the form of a partial differential equation. By partitioning the observation horizon into a finite number of consecutive intervals, the problem is set up to select nodes which will be active over each interval while the others will remain dormant such that a general convex design criterion defined on the Fisher information matrix associated with the estimated parameters is minimized. The search for the optimal solution is performed using the branch-and-bound method in which an efficient technique is employed to produce a lower bound to the minimum of the objective function.**

I. INTRODUCTION

Over the past few years the interest in sensor networks have continued to grow towards *distributed parameter systems* (DPSs), i.e., systems whose appropriate mathematical modeling yields partial differential equations (PDEs), see [1], [2]. The importance of sensor planning for such systems has been recognized in many application domains prior to the invention of sensor networks, see [3], [4], [5]. The operation and control of such systems usually requires precise information on the parameters which condition the accuracy of the underlying mathematical model, but that information is only available through a limited number of possibly expensive sensors. Most techniques, however, rely on exhaustive search over a predefined set of candidates and the combinatorial nature of the design problem is taken into account very occasionally. Needless to say that this approach, which is feasible for a relatively small number of possible locations, soon becomes useless as the number of possible location candidates increases. It goes without saying that these problems are compounded by the passage to scanning observations.

The aim of the investigations undertaken here was to establish a practical approach to selection of activated sensors over consecutive time stages which, while being independent of a particular model of the dynamic DPS in question, would be versatile enough to cope with practical sensor networks consisting of a large number of nodes. The presented idea was already studied by [6] in quite a similar vein, but for stationary sensors. Extensions to the scanning sensor setting are not that straightforward, however, especially due to the requirement that the number of active sensors over each time stage be always the same. The main contribution of this paper consists in the development of a simple, yet powerful, computational scheme to obtain lower bounds to the optimal values of the design criterion for the restricted problems.

This work was supported by the Ministry of Science and Higher Education under grant No. N N519 2971 33.

D. Uciński is with Institute of Control and Computation Engineering, University of Zielona Góra, Zielona Góra, Poland d.ucinski@issi.uz.zgora.pl

II. Ψ-OPTIMAL SENSOR ACTIVATION STRATEGIES

Consider a bounded spatial domain $\Omega \subset \mathbb{R}^d$ with sufficiently smooth boundary Γ, and a bounded time interval $T = (0, t_f]$. Assume that a DPS is given whose scalar state at a spatial point $\boldsymbol{x} \in \bar{\Omega} \subset \mathbb{R}^d$ and time instant $t \in \bar{T}$ is denoted by $y(\boldsymbol{x}, t; \boldsymbol{\theta})$, where $\boldsymbol{\theta} \in \mathbb{R}^m$ stands for a vector of unknown parameters which must be estimated using observations of the state.

Let $\boldsymbol{x}^i$, $i = 1, \ldots, I$ be given spatial positions of sensor network nodes. We form an arbitrary partition of the time interval T by choosing points $t_0 < t_1 < \cdots < t_K = t_f$ defining subintervals $T_k = (t_{k-1}, t_k]$, $k = 1, \ldots, K$ called *scanning stages*. It is assumed here that, over each time interval T_k, the state y is constantly observed by n active sensors selected from among the I possible. Introducing for each location $\boldsymbol{x}^i$ variables v_{ki}, $k = 1, \ldots, K$ which take the value 1 or 0 depending on whether or not a sensor located at $\boldsymbol{x}^i$ is active over the time interval T_k, respectively, we can represent the observations obtained from network nodes as follows:

$$z_{ki}(t) = v_{ki}\big[y(\boldsymbol{x}^i, t; \boldsymbol{\theta}) + \varepsilon(\boldsymbol{x}^i, t)\big], \quad i = 1, \ldots, I \tag{1}$$

for $t \in T_k$, $k = 1, \ldots, K$, where $\varepsilon(\boldsymbol{x}^i, t)$ denotes the measurement noise. It is customary to assume that the noise is zero-mean, Gaussian, spatial uncorrelated and white.

Inevitably, the accuracy of the resulting least-squares estimates of $\boldsymbol{\theta}$ depends on the selection of gauged sites. A quantitative measure Ψ of the 'goodness' of particular sensor configurations is customarily based on the concept of the *Fisher Information Matrix* (FIM) which is widely used in optimum experimental design theory for lumped systems [7], [8], [9]. In our setting, the FIM is given by

$$\boldsymbol{M}(\boldsymbol{v}) = \sum_{k=1}^{K} \sum_{i=1}^{I} v_{ki} \boldsymbol{M}_{ki}, \tag{2}$$

where

$$\boldsymbol{v} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_K), \qquad \boldsymbol{v}_k = (v_{k1}, \ldots, v_{kI}), \quad k = 1, \ldots, K, \tag{3}$$

$$\boldsymbol{M}_{ki} = \int_{T_k} \boldsymbol{g}(\boldsymbol{x}^i, t) \boldsymbol{g}(\boldsymbol{x}^i, t)^{\mathsf{T}} \, \mathrm{d}t. \tag{4}$$

Here

$$\boldsymbol{g}(\boldsymbol{x}, t) = \left[\frac{\partial y(\boldsymbol{x}, t; \boldsymbol{\vartheta})}{\partial \vartheta_1}, \quad \ldots, \quad \frac{\partial y(\boldsymbol{x}, t; \boldsymbol{\vartheta})}{\partial \vartheta_m} \right]^{\mathsf{T}}_{\boldsymbol{\vartheta} = \boldsymbol{\theta}^0} \tag{5}$$

ISBN 978-963-311-370-7

stands for the so-called *sensitivity vector*, $\boldsymbol{\theta}^0$ being a prior estimate to the unknown parameter vector $\boldsymbol{\theta}$ [5], [4], [10], [11]. It is straightforward to verify that the $m \times m$ matrices $\boldsymbol{M}_{ki}$ are nonnegative definite and, therefore, so is $\boldsymbol{M}(\boldsymbol{v})$.

As for a specific form of Ψ, various options exist [8], [7], [9], but the most popular criterion, called the D-optimality criterion, is

$$\Psi(M) = -\log\det(M). \tag{6}$$

The resulting D-optimum sensor configuration leads to the minimum volume of the uncertainty ellipsoid for the estimates. Another popular choice is the A-optimality criterion

$$\Psi(M) = \mathrm{tr}(M^{-1}). \tag{7}$$

Its use suppresses the average variance of the estimates.

In what follows, we shall assume that Ψ is convex over the cone of nonnegative definite matrices.

Our design problem is thus formulated as follows:
Problem P: Find a sequence

$$\boldsymbol{v} = (v_{11}, \ldots, v_{1I}, \ldots, v_{K1}, \ldots, v_{KI}) \in \mathbb{R}^{KI}$$

to minimize

$$\mathcal{P}(\boldsymbol{v}) = \Psi\big(\boldsymbol{M}(\boldsymbol{v})\big) \tag{8}$$

subject to the constraints

$$\sum_{i=1}^{I} v_{ki} = n, \qquad k = 1, \ldots, K, \tag{9}$$

$$v_{ki} \in \big\{0, 1\big\}, \quad i = 1, \ldots, I, \quad k = 1, \ldots, K. \tag{10}$$

III. BRANCH-AND-BOUND SCHEME

A. Overview

The branch-and-bound (BB) constitutes a general algorithmic technique for finding optimal solutions of various optimization problems, especially combinatorial. When implementing it for Problem P, we will use the symbol E to denote the set $\big\{1, \ldots, K\big\} \times \big\{1, \ldots, I\big\}$ of all possible pairs (k, i) of the indices identifying a scanning stage and a sensor location. We shall partition the feasible set

$$V = \Bigg\{\boldsymbol{v} = (v_{11}, \ldots, v_{1I}, \ldots, v_{K1}, \ldots, v_{KI}) \,\Big|\, \sum_{i=1}^{I} v_{ki} = n,\ k = 1, \ldots, K,\ v_{ki} = 0 \text{ or } 1,\ \forall (k,i) \in E\Bigg\}, \tag{11}$$

into subsets

$$V(E_0, E_1) = \big\{\boldsymbol{v} \in V \mid v_{ki} = 0,\ \forall (k,i) \in E_0,\ v_{ki} = 1,\ \forall (k,i) \in E_1\big\}, \tag{12}$$

where E_0 and E_1 are disjoint subsets of E. Consequently, $V(E_0, E_1)$ is the subset of V such that sensors are active at the stages and locations with indices in E_1, sensors are dormant at the stages and locations with indices in E_0, and sensors may be active or dormant at the remaining stages and locations.

Each subset $V(E_0, E_1)$ is identified with a node in the BB tree. The key assumption in the BB method is that for every nonterminal node $V(E_0, E_1)$, i.e., the node for which $E_0 \cup E_1 \neq E$, there is an algorithm that determines a lower bound $\underline{\mathcal{P}}(E_0, E_1)$ to the minimum design criterion over $V(E_0, E_1)$, i.e.,

$$\underline{\mathcal{P}}(E_0, E_1) \leq \min_{\boldsymbol{v} \in V(E_0, E_1)} \mathcal{P}(\boldsymbol{v}), \tag{13}$$

and a feasible solution $\bar{\boldsymbol{v}} \in V$ for which $\mathcal{P}(\bar{\boldsymbol{v}})$ can serve as an upper bound to the minimum design criterion over V. We may compute $\underline{\mathcal{P}}(E_0, E_1)$ by solving the following relaxed problem:

Problem $R(E_0, E_1)$: Find a sequence $\underline{\boldsymbol{v}}$ to minimize (8) subject to the constraints

$$\sum_{i=1}^{I} v_{ki} = n, \qquad k = 1, \ldots, K, \tag{14}$$

$$v_{ki} = 0, \quad (k,i) \in E_0, \tag{15}$$

$$v_{ki} = 1, \quad (k,i) \in E_1, \tag{16}$$

$$0 \leq v_{ki} \leq 1, \quad (k,i) \in E \setminus (E_0 \cup E_1). \tag{17}$$

In Problem $\mathrm{R}(E_0, E_1)$ all 0–1 constraints on the variables v_{ki} are relaxed by allowing them to take any value in the interval $[0, 1]$, except that the variables v_{ki}, $(k,i) \in E_0 \cup E_1$ are fixed at either 0 or 1. A simple and efficient method for its solution is given in Section IV. As a result of its application, we set $\underline{\mathcal{P}}(E_0, E_1) = \mathcal{P}(\underline{\boldsymbol{v}})$.

As for $\bar{\boldsymbol{v}}$, we can specify it as the best feasible solution (i.e., an element of V) found so far. If no solution has been found yet, we can either set the upper bound to ∞, or use an initial guess about the optimal solution (experience provides evidence that the latter choice leads to much more rapid convergence).

B. Branching Rule and BB Algorithm

The result of solving Problem $\mathrm{R}(E_0, E_1)$ can serve as a basis to construct a branching rule for the binary BB tree. We adopt here the approach in which the tree node/subset $V(E_0, E_1)$ is expanded (i.e., partitioned) by first picking out all fractional values from among the values of the relaxed variables, and then rounding to 0 and 1 a value which is the most distant from both 0 and 1. Specifically, we apply the following steps:

(i) Determine

$$(k_\star, i_\star) = \arg \min_{(k,i) \in E \setminus (E_0 \cup E_1)} |v_{ki} - 0.5|. \tag{18}$$

(In case there are several minimizers, randomly pick one of them.)

(ii) Partition $V(E_0, E_1)$ into $V(E_0 \cup \big\{(k_\star, i_\star)\big\}, E_1)$ and $V(E_0, E_1 \cup \big\{(k_\star, i_\star)\big\})$ whereby two descendants of the node in question are defined.

A recursive application of the branching rule starts from the root of the BB tree, which corresponds to the trivial subset $V(\emptyset, \emptyset) = V$ and the fully relaxed problem. Each node of the BB tree corresponds to a continuous relaxed

problem, $\mathrm{R}(E_0, E_1)$, while each edge corresponds to fixing one relaxed variable at 0 or 1.

The above scheme has to be complemented with a search strategy to incrementally explore all the nodes of the BB tree. Here we use a common depth-first technique. A recursive version of the resulting depth-first branch-and-bound is implemented in Algorithm 1. The operators involved in this implementation are as follows:

- SINGULARITY-TEST(E_0, E_1) returns true only if expansion of the current node will result in a singular FIM, see [6] for details.
- RELAXED-SOLUTION(E_0, E_1) returns a solution to Problem $\mathrm{R}(E_0, E_1)$.
- PSI-FIM$(\boldsymbol{v})$ evaluates the design criterion Ψ at the FIM corresponding to $\boldsymbol{v}$.
- INTEGRAL-TEST$(\boldsymbol{v})$ returns true only if the current solution $\boldsymbol{v}$ is integral.
- INDEX-BRANCH$(\boldsymbol{v})$ returns the pair of indices defined by (18).

Algorithm 1 A recursive version of the depth-first branch-and-bound method. It uses two global variables, *UPPER* and *v_best*, which are respectively the minimal value of the design criterion over feasible solutions found so far and the solution at which it is attained.

procedure RECURSIVE-DFBB(E_0, E_1)
  if $E_0 \cup E_1 = E$ **then** ▷ Deepest level of the BB tree has been attained
    psi_v ← PSI-FIM(*v*(E_0, E_1))
    if *psi_v* < *UPPER* **then**
      v_best ← *v*(E_0, E_1)
      UPPER ← *psi_v*
    end if
    return
  end if
  if SINGULARITY-TEST(E_0, E_1) **then**
    return ▷ Only singular FIMs can be expected
  end if
  v_relaxed ← RELAXED-SOLUTION(E_0, E_1)
  psi_relaxed ← PSI-FIM(*v_relaxed*)
  if *psi_relaxed* ≥ *UPPER* **then**
    return ▷ Pruning
  else if INTEGRAL-TEST(*v_relaxed*) **then**
    v_best ← *v_relaxed*
    UPPER ← *psi_relaxed*
    return ▷ Relaxed solution is integral
  else
    $(k_\star, i_\star)$ ← INDEX-BRANCH(*v_relaxed*) ▷ Partition into two descendants
    RECURSIVE-DFBB$(E_0 \cup \{(k_\star, i_\star)\}, E_1)$
    RECURSIVE-DFBB$(E_0, E_1 \cup \{(k_\star, i_\star)\})$
  end if
end procedure

IV. SIMPLICIAL DECOMPOSITION FOR THE RELAXED PROBLEM

For notational convenience, for each $k = 1, \ldots, K$ we define the sets

$$\mathcal{I}_1(k) = \left\{ i \,\middle|\, 1 \le i \le I \text{ and } (k,i) \in E_1 \right\}, \tag{19}$$

$$\mathcal{I}_0(k) = \left\{ i \,\middle|\, 1 \le i \le I \text{ and } (k,i) \in E_0 \right\}, \tag{20}$$

$$\mathcal{K} = \left\{ k \,\middle|\, 1 \le k \le K \text{ and } |\mathcal{I}_1(k) \cup \mathcal{I}_0(k)| < I \right\}. \tag{21}$$

Thus, at Stage k, sets $\mathcal{I}_1(k)$ and $\mathcal{I}_0(k)$ consist of the indices of sites for which the associated variables v_{ki} are fixed at 1 and 0, respectively, and hence cannot be qualified as relaxed, while $\mathcal{K}$ contains the indices of the stages at which there are relaxed variables.

Consider any bijection ς from $\{1, \ldots, L\}$ to $\mathcal{K}$, where $L = |\mathcal{K}|$. For each $l = 1, \ldots, L$, setting

$$r_l = n - |\mathcal{I}_1(\varsigma(l))|, \tag{22}$$

$$q_l = I - |\mathcal{I}_1(\varsigma(l)) \cup \mathcal{I}_0(\varsigma(l))|, \tag{23}$$

we can then construct a bijection ϱ_l from $\{1, \ldots, q_l\}$ to $I \setminus (\mathcal{I}_1(\varsigma(l)) \cup \mathcal{I}_0(\varsigma(l)))$. Thus, q_l stands for the number of relaxed variables at Stage $\varsigma(l)$ and r_l signifies the number of sensors which still remain to be activated at the same stage.

Accordingly, we can replace the relaxed variables v_{ki}, $(k,i) \in E \setminus (E_0 \cup E_1)$ by w_{lj} such that $w_{lj} = v_{\varsigma(l),\varrho_l(j)}$, $j = 1, \ldots, q_l$ and $l = 1, \ldots, L$. This leads to the following formulation:

Problem $R'(E_0, E_1)$*:* Find a sequence

$$\boldsymbol{w} = (w_{1,1}, \ldots, w_{1,q_1}, \ldots, w_{L,1}, \ldots, w_{L,q_L}) \in \mathbb{R}^f \tag{24}$$

where $f = \sum_{l=1}^{L} q_l$, to minimize

$$\mathcal{Q}(\boldsymbol{w}) = \Psi\big(\boldsymbol{G}(\boldsymbol{w})\big) \tag{25}$$

subject to the constraints

$$\sum_{j=1}^{q_l} w_{lj} = r_l, \quad l = 1, \ldots, L, \tag{26}$$

$$0 \le w_{lj} \le 1, \quad j = 1, \ldots, q_l, \quad l = 1, \ldots, L, \tag{27}$$

where

$$\boldsymbol{G}(\boldsymbol{w}) = \boldsymbol{A} + \sum_{l=1}^{L} \sum_{j=1}^{q_l} w_{lj} \boldsymbol{S}_{lj}, \qquad \boldsymbol{A} = \sum_{k=1}^{K} \sum_{i \in \mathcal{I}_1(k)} \boldsymbol{M}_{ki}, \tag{28}$$

$$\boldsymbol{S}_{lj} = \boldsymbol{M}_{\varsigma(l),\varrho_l(j)}, \quad j = 1, \ldots, q_l, \quad l = 1, \ldots, L. \tag{29}$$

(Note that at Stage $\varsigma(l)$ we have that $|\mathcal{I}_1(\varsigma(l))|$ sensors have already been activated at locations $\boldsymbol{x}^i$, $i \in \mathcal{I}_1(\varsigma(l))$, and thus a decision about the activation of r_l remaining sensors has to be made.)

In the sequel, W will stand for the set of all vectors $\boldsymbol{w}$ of the form (24) satisfying (26) and (27). Note that it forms a polygon in $\mathbb{R}^f$.

Simplicial Decomposition (SD) constitutes an important class of methods for solving large-scale continuous problems

in mathematical programming with convex feasible sets [12], [13]. In the original framework, where a convex objective function is to be minimized over a bounded polyhedron, it iterates by alternately solving a linear programming subproblem (the so-called *column generation problem*) which generates an extreme point of the polyhedron, and a nonlinear *restricted master problem* (RMP) which finds the minimum of the objective function over the convex hull (a simplex) of previously defined extreme points.

Problem $\mathrm{R}'(E_0, E_1)$ is perfectly suited for the application of the SD scheme. In this case, it boils down to Algorithm 2. Here $\nabla\mathcal{Q}(\boldsymbol{w})$ signifies the gradient of $\mathcal{Q}$ at $\boldsymbol{w}$, and it is easy to check that

$$\frac{\partial\mathcal{Q}(\boldsymbol{w})}{\partial w_{lj}} = \operatorname{tr}\left(\boldsymbol{S}_{lj}\,\frac{\partial\Psi(\boldsymbol{M})}{\partial\boldsymbol{M}}\bigg|_{\boldsymbol{M}=\boldsymbol{G}(\boldsymbol{w})}\right) \tag{30}$$

Algorithm 2 Algorithm model for simplicial decomposition.

Step 0: (Initialization)
Set

$$\boldsymbol{w}^{(0)} = (\underbrace{r_1/q_1,\ldots,r_1/q_1}_{q_1\text{ times}},\ldots,\underbrace{r_L/q_L,\ldots,r_L/q_L}_{q_L\text{ times}}).$$

and $Z^{(0)} = \{\boldsymbol{w}^{(0)}\}$. Select $0 < \epsilon \ll 1$, a parameter used in the stopping rule, and set $\tau = 0$.

Step 1: (Solution of the column generation subproblem)
Determine

$$\boldsymbol{z} = \arg\min_{\boldsymbol{w}\in W} \nabla\mathcal{Q}(\boldsymbol{w}^{(\tau)})^{\mathsf{T}}(\boldsymbol{w}-\boldsymbol{w}^{(\tau)}). \tag{31}$$

Step 2: (Termination check)
If $\nabla\mathcal{Q}(\boldsymbol{w}^{(\tau)})^{\mathsf{T}}(\boldsymbol{z}-\boldsymbol{w}^{(\tau)}) \geq -\epsilon$, then STOP and $\boldsymbol{w}^{(\tau)}$ is optimal. Otherwise, set $Z^{(\tau+1)} = Z^{(\tau)} \cup \{\boldsymbol{z}\}$.

Step 3: (Solution of the restricted master problem)
Find

$$\boldsymbol{w}^{(\tau+1)} = \arg\min_{\boldsymbol{w}\in\operatorname{co}(Z^{(\tau+1)})} \mathcal{Q}(\boldsymbol{w}) \tag{32}$$

and purge $Z^{(\tau+1)}$ of all extreme points with zero weight in the expression of $\boldsymbol{w}^{(\tau+1)}$ as a convex combination of elements in $Z^{(\tau+1)}$. Increment τ by one and go back to Step 1.

Since we deal with minimization of a convex function $\mathcal{Q}$ over a bounded polyhedral set W, the convergence of Algorithm 2 in a finite number of RMP steps is automatically guaranteed [12, p. 221]. Observe that Step 3 implements the *column dropping rule* [13], according to which any extreme point with zero weight in the expression of $w^{(\tau)}$ as a convex combination of elements in $Z^{(\tau)}$ is removed. This rule makes the number of elements in successive sets $Z^{(\tau)}$ reasonably low.

The SD algorithm may be viewed as a form of modular nonlinear programming, provided that one has an effective computer code for solving the restricted master problem, as well as access to a code which can take advantage of the linearity of the column generation subproblem [14]. The former issue can be addressed using the fast procedure outlined in [15], where a new procedure for finding the projection of the gradient of the objective function onto a simplicial cone was discussed. In turn, the latter issue can be easily settled, as in the linear programming problem of Step 1 the feasible region W is defined by L equality contraints (26) and $\sum_{l=1}^{L} q_l$ bound constraints (27). Note, however, that the positive definiteness of the matrix $\boldsymbol{G}^{-1}(\boldsymbol{w}^{(\tau)})$ and the nonnegative definiteness of $\boldsymbol{S}_{lj}$ taken in conjunction with (30) imply that $\partial\mathcal{Q}(\boldsymbol{w}^{(\tau)})/\partial w_{lj} \leq 0$. Hence it is easily seen that the column generation subproblem has an explicit solution where for each $l = 1,\ldots,L$ the relaxed weights w_{lj} corresponding to r_l lowest values selected from among $\partial\mathcal{Q}(\boldsymbol{w}^{(\tau)})/\partial w_{lj}$, $j = 1,\ldots,q_l$ are set to their maximal allowable values, the others beeing zero. Consequently, no specialized linear programming code is needed. This greatly simplifies the implementation and constitutes a clear advantage of the presented approach.

V. Numerical example

Consider the diffusion equation

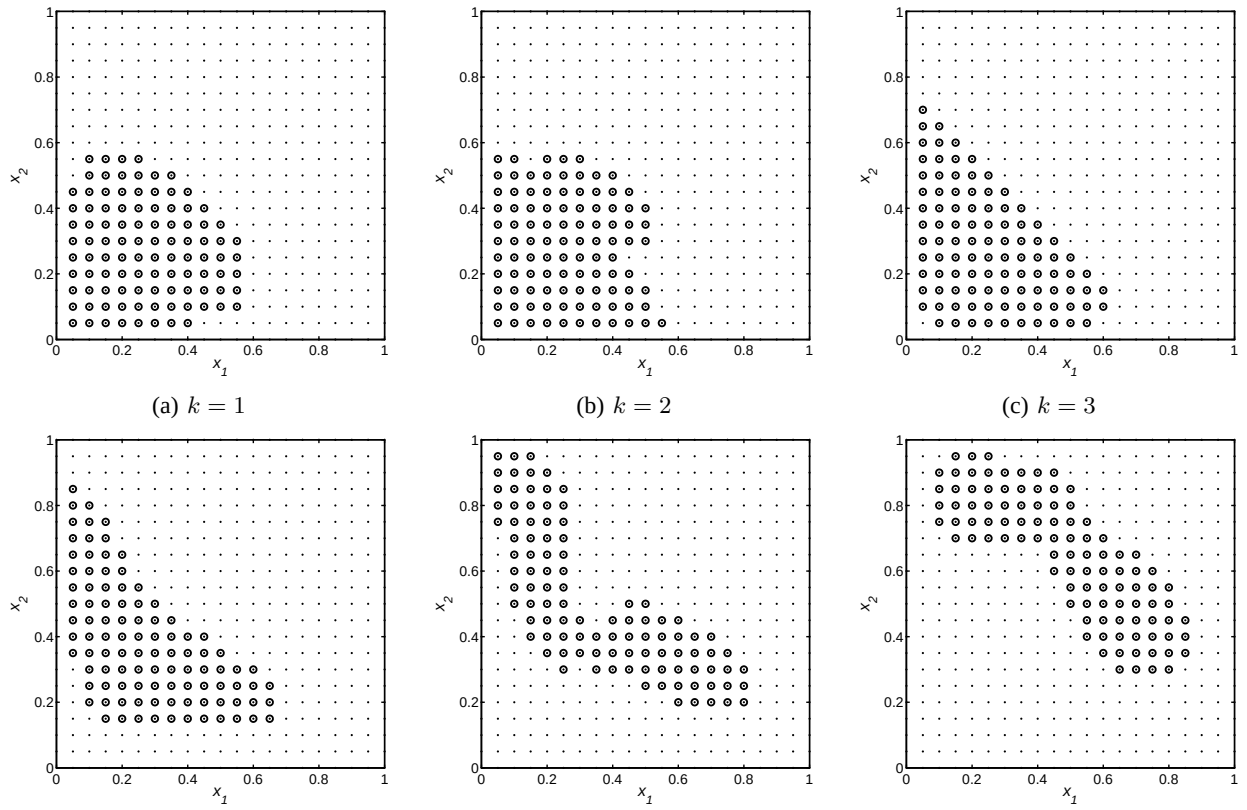
$$\begin{aligned}\frac{\partial y(\boldsymbol{x},t)}{\partial t} &= \frac{\partial}{\partial x_1}\left(\kappa(\boldsymbol{x};\boldsymbol{\theta})\frac{\partial y(\boldsymbol{x},t)}{\partial x_1}\right)\\ &\quad+\frac{\partial}{\partial x_2}\left(\kappa(\boldsymbol{x};\boldsymbol{\theta})\frac{\partial y(\boldsymbol{x},t)}{\partial x_2}\right)+u(\boldsymbol{x},t),\\ &\qquad \boldsymbol{x}\in\Omega=(0,1)\times(0,1),\quad t\in(0,1)\end{aligned} \tag{33}$$

subject to the homogeneous initial and boundary conditions. The diffusion coefficient to be identified has the form $\kappa(\boldsymbol{x};\boldsymbol{\theta}) = \theta_1 + \theta_2 x_1 + \theta_3 x_2$. As regards the forcing term in our model, it has the form

$$u(\boldsymbol{x},t) = 20\exp\bigl(-50(x_1+x_2-2t)^2\bigr). \tag{34}$$

Our purpose is to estimate κ (i.e., the parameters θ_1, θ_2 and θ_3) as accurately as possible based on the measurements made by n scanning sensors. A uniform mesh of $I = 30\times 30 = 900$ points was thus assumed as the set of sites where sensor network nodes are placed (they are marked with dots in Fig. 1).

A-optimal positions of active sensors were determined for the case of $n = 100$ sensors to be activated. The observation horizon was decomposed into $K = 6$ stages of equal length, which yielded the size of the solutions spaces to $\binom{900}{100}^6 > 6\times 10^{809}$. The optimal solutions displayed in Fig. 1 were obtained in no more than five seconds on a low-cost PC (Intel Core(TM)2 Quad CPU Q9400@2.66GH, 8 GB RAM) running Linux Fedora 12 and MATLAB 7.1. The parameter required by Algorithm 2 was set as $\epsilon = 0.01$. This fine performance was possible owing to a good initial lower bound on the optimal solution. It was obtained by solving a fully relaxed problem and activating sensors which corresponded to the largest weights. Otherwise, the solutions obtained by literally following Algorithm 1 consumed about fifteen minutes of the CPU time, which is not bad, either.



(a) $k = 1$ (b) $k = 2$ (c) $k = 3$

(d) $k = 4$ (e) $k = 5$ (f) $k = 6$

Fig. 1. D-optimal configurations of active sensors at consecutive stages ($K = 6$, $I = 900$, $n = 100$).

VI. Concluding remarks

An alternative approach to select a best n-element subset of active sensors from among a given I-element set of all candidate sensors could be to employ an exchange algorithm, see, e.g., [5, p. 105]. Algorithms of this type begin with an n-point starting sensor configuration which then sequentially evolves through addition of new elements selected from among vacant sites and deletion of sites at which sensors have provisionally been planned to reside, in an effort to improve the value of the adopted design criterion. It goes without saying that such an approach outperforms the BB technique proposed here as far as the running time is concerned. On should note, however, that exchange algorithms are only capable of finding globally competitive solutions (i.e., nearly optimal ones), with an explicit trade of global optimality for speed. The approach presented here is superior in the sense that it always produces global minima and, what is more, does it within tolerable time.

References

[1] F. Zhao and L. J. Guibas, *Wireless Sensor Networks: An Information Processing Approach*. Amsterdam: Morgan Kaufmann Publishers, 2004.

[2] C. G. Cassandras and W. Li, "Sensor networks and cooperative control," *European Journal of Control*, vol. 11, no. 4–5, pp. 436–463, 2005.

[3] C. S. Kubrusly and H. Malebranche, "Sensors and controllers location in distributed systems — A survey," *Automatica*, vol. 21, no. 2, pp. 117–128, 1985.

[4] N.-Z. Sun, *Inverse Problems in Groundwater Modeling*, ser. Theory and Applications of Transport in Porous Media. Dordrecht, The Netherlands: Kluwer Academic Publishers, 1994.

[5] D. Uciński, *Optimal Measurement Methods for Distributed-Parameter System Identification*. Boca Raton, FL: CRC Press, 2005.

[6] D. Uciński and M. Patan, "D-optimal design of a monitoring network for parameter estimation of distributed systems," *Journal of Global Optimization*, vol. 39, pp. 291–322, 2007.

[7] V. V. Fedorov and P. Hackl, *Model-Oriented Design of Experiments*, ser. Lecture Notes in Statistics. New York: Springer-Verlag, 1997.

[8] É. Walter and L. Pronzato, *Identification of Parametric Models from Experimental Data*, ser. Communications and Control Engineering. Berlin: Springer-Verlag, 1997.

[9] A. C. Atkinson and A. N. Donev, *Optimum Experimental Designs*. Oxford: Clarendon Press, 1992.

[10] E. Rafajłowicz, "Design of experiments for eigenvalue identification in distributed-parameter systems," *International Journal of Control*, vol. 34, no. 6, pp. 1079–1094, 1981.

[11] ——, "Optimal experiment design for identification of linear distributed-parameter systems: Frequency domain approach," *IEEE Transactions on Automatic Control*, vol. 28, no. 7, pp. 806–808, Jul. 1983.

[12] B. von Hohenbalken, "Simplicial decomposition in nonlinear programming algorithms," *Mathematical Programming*, vol. 13, pp. 49–68, 1977.

[13] M. Patriksson, "Simplicial decomposition algorithms," in *Encyclopedia of Optimization*, C. A. Floudas and P. M. Pardalos, Eds. Dordrecht, The Netherlands: Kluwer Academic Publishers, 2001, vol. 5, pp. 205–212.

[14] D. W. Hearn, S. Lawphongpanich, and J. A. Ventura, "Restricted simplicial decomposition: Computation and extensions," *Mathematical Programming Study*, vol. 31, pp. 99–118, 1987.

[15] N. D. Botkin and J. Stoer, "Minimization of convex functions on the convex hull of a point set," *Mathematical Methods of Operations Research*, vol. 62, no. 2, pp. 167–185, 2005.