Proceedings of the 2007 American Control Conference
Marriott Marquis Hotel at Times Square
New York City, USA, July 11-13, 2007

Ensemble-Based Adaptive Targeting of Mobile Sensor Networks

Han-Lim Choi, Jonathan P. How, and James A. Hansen

Abstract— **This work presents an efficient algorithm for an observation targeting problem that is complicated by the combinatorial number of targeting choices. The approach explicitly incorporates an ensemble forecast to ensure that the measurements are chosen based on their expected improvement in the forecast at a separate verification time and location. The primary improvements in the efficiency are obtained by computing the impact of each possible measurement on the uncertainty reduction over this verification site backwards. In particular, the approach determines the impact of a series of fictitious observations taken at the verification site back on the search space (and time), which provides all of the information needed to optimize the set of measurements to take and significantly reduces the number of times that the computationally expensive ensemble updates must be performed. A computation time analysis and numerical performance simulations using the two-dimensional Lorenz-95 chaos model are presented to validate the computational advantage of the proposed algorithm over conventional search strategies.**

I. Introduction

One key aspect of a sensor network is determining when, where, and which type of sensors should be deployed in order to achieve the best information, which can be the uncertainty reduction of a region of interest [1]–[3], coverage area of the sensors, localization of a specified target [4]–[7], and communication cost savings [3,4]. This type of decision making addresses two technical issues: an effective way to define and compute the information reward, and a logical way to determine the best choice. These two are often conflicting each other in practice. Expensive reward computation may lead to settling for a simple selection strategy; conversely, the need for a sophisticated strategy might necessitate the use of approximate reward computation. This is caused by the combinatorial aspect of the selection problem as well as by the limited time and computational resources.

Many studies have been devoted to effectively evaluating the reward value. Zhao et al. [4] introduced the Mahalanobis distance characterized by the location and modality of a sensor in a dynamic query network to avoid complicated entropy computation. Ertin et al. [5] pointed out the equivalence of minimization of the conditional entropy to maximization of the mutual information in Bayesian filtering framework. Wang et al. [7] presented an entropy-based heuristics to approximate the computationally expensive mutual information in the target tracking task, while Guestrin et al. [1] proposed an approximation algorithm based on local kernels for maximizing mutual information in Gaussian processes, and extended it to a randomized algorithm for a graphical model [3]. With respect to the selection strategy, however, they all adopted a *greedy strategy* with sacrificing optimality, since the reward calculations are too complicated to consider more sophisticated selection strategies in practice. Williams et al. [6] proposed an approximate dynamic programming approach to find the *receding-horizon optimal* sensor scheduling solution for best information gain with communication power being restricted. However, they just considered a single moving target with linear dynamics.

This paper addresses a large-scale dynamic sensor network design within the context of a weather prediction problem. Complex characteristics of the weather dynamics such as being chaotic, uncertain, and of multiple time- and length-scales, leads to the necessity of a large sensor network. Expansion of the static observation network is limited by geographic aspects; thus, an adaptive sensor network incorporating mobile sensor platforms (e.g. UAVs) has become an attractive solution to construct effectively large networks. Palmer et al. [9] and Daescu et al. [10] located the potential error-growing sites based on the sensitivity information inherited in the dyanamics; meanwhile, Majumdar et al. [11] and Leutbecher [12] quantified the future forecast covariance change within the framework of approximation of extended Kalman filter. However, due to the enormous size of the system – millions of state variables and thousands of adaptive measurements [11], the selection strategy was very simple – for the most, two flight paths was *greedily* selected out of 49 predetermined paths in Ref. [11].

This work provides a much more efficient way to perform the targeting of a sensor network in a large space when the goal is to reduce the uncertainty in a specified verification space/time. The dynamic sensor targeting problem is formulated as a static sensor selection problem associated with the ensemble-based filtering [13,14]. Mutual information is introduced as a measure of uncertainty reduction, and computed under the Gaussian assumption. To address the computational challenge resulting from the expense of determining the impact of each measurement choice on the uncertainty reduction in the verification site, the *commutativity of mutual information* is exploited. This enables the contribution of each measurement choice to be computed by propagating information *backwards* from the verification space/time to the search space/time. This significantly reduces the required number of ensemble updates that is computationally intensive. Analytic estimates of the computation time for the proposed approach are given in comparison to a conventional forward approach. Numerical

H. -L. Choi, Research Assistant, Dept. of Aeronautics and Astronautics, MIT, Cambridge, MA 02139, USA, hanlimc@mit.edu
J. P. How, Professor, Dept. of Aeronautics and Astronautics, MIT, Cambridge, MA 02139, USA, jhow@mit.edu
J. A. Hansen, Marine Meteorology Division, Naval Research Laboratory, jim.hansen@nrlmry.navy.mil

1-4244-0989-6/07/$25.00 ©2007 IEEE.

simulations are also presented to validate the computation advantage of the suggested approach.

II. Preliminaries

A. Ensemble Square Root Filter

Ensemble-based forecast represents the nonlinear features of the weather system better, and mitigates the computational burden of linearizing the nonlinear dynamics and keeping track of a large covariance matrix [13,14]. In the ensemble square root filter (EnSRF), the state estimate and the estimation error covariance are represented by the ensemble mean and perturbation ensemble variance, respectively. The perturbation ensemble being defined as $\tilde{\mathbf{X}} \equiv \eta\left(\mathbf{X} - \bar{\mathbf{x}} \times \mathbf{1}^T\right) \in \mathbb{R}^{L_S \times L_E}$, the error covariance is approximated as $\mathbf{P} = \tilde{\mathbf{X}}\tilde{\mathbf{X}}^T/(L_E - 1)$, where $\mathbf{X}$ is the ensemble matrix, $\bar{\mathbf{x}}$ is the ensemble mean vector, L_S and L_E denote the number of state and the ensemble members, and η is the inflation factor to avoid underestimation of the covariance. The propagation step for the EnSRF corresponds to the integration

$$\mathbf{X}^f(t + \Delta t) = \int_t^{t+\Delta t} \dot{\mathbf{X}} dt \tag{1}$$

with the initial condition $\mathbf{X}(t) = \mathbf{X}^a(t)$, superscripts 'f' and 'a' denote the *forecast* and the *analysis*, respectively. The measurement update step for the EnSRF consists of the mean update and the perturbation update as:

$$\bar{\mathbf{x}}^a = \bar{\mathbf{x}}^f + \mathbf{K}(\mathbf{y} - \mathbf{H}\bar{\mathbf{x}}^f), \;\; \tilde{\mathbf{X}}^a = (\mathbf{I} - \mathbf{K}\mathbf{H})\tilde{\mathbf{X}}^f \tag{2}$$

where $\mathbf{y}$ and $\mathbf{H}$ are the measurement vector and the observation matrix, while $\mathbf{K}$ represents the Kalman gain determined by a nonlinear matrix equation of $\mathbf{X}$ [13]. In the sequential framework devised for efficient implementation, the ensemble update by the m-th observation is written as

$$\tilde{\mathbf{X}}^{m+1} = \tilde{\mathbf{X}}^m - \alpha_m \beta_m \mathbf{p}_i^m \tilde{\xi}_i^m \tag{3}$$

with $\alpha_m = (1 + \sqrt{\beta_m R_i})^{-1}, \beta_m = (\mathbf{P}_{ii}^m + R_i)^{-1}$, when i-th state is measured. $\mathbf{p}_i^m$, $\tilde{\xi}_i^m$, and $\mathbf{P}_{ii}^m$ are the i-th column, the i-th row, and (i,i) element of the prior perturbation matrix $\mathbf{P}^m$, respectively. α_m is the factor for compensating the mismatch between the serial update and the batch update, while $\beta_m \mathbf{p}_l^m$ is equivalent to the Kalman gain.

B. Entropy and Mutual Information

Entropy represents the amount of information hidden in a random variable (A), and is defined as $h(A) = -E[\log(p_A(a))]$ [15]. This definition can be easily extended to the case of a random vector. The joint entropy $h(A_1, A_2)$ and the conditional entropy $h(A_2|A_1)$ is related as $h(A_1, A_2) = h(A_1) + h(A_2|A_1)$. The mutual information – also called information gain – that will be employed as a metric of the uncertainty reduction of the forecast by the measurement is defined as:

$$I(A_1; A_2) \equiv h(A_1) - h(A_1|A_2). \tag{4}$$

Note that the mutual information is commutative [15]: $I(A_1; A_2) \equiv I(A_2; A_1) = h(A_2) - h(A_2|A_1)$. In case a random vector A is jointly Gaussian, its entropy is expressed as $h(A) = \frac{1}{2}\log\left[(2\pi e)^k |\mathrm{Cov}(A)|\right]$. This work computes the mutual information between the verification and measurement variables by exploiting the commutativity under the Gaussian assumption.

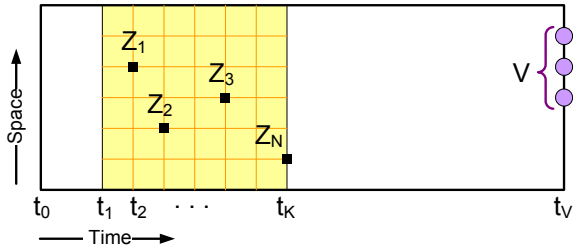

Fig. 1. Multiple targeting as a sensor placement problem

III. Problem Formulation

A. Sensor Selection Problem

Figure 1 illustrates the sensor targeting problem in the spatial-temporal gridspace. The objective of sensor targeting is to deploy n sensors in the search space/time (yellow region) in order to reduce the uncertainty in the verification region (purple region) at the verification time t_V. Without loss of generality, it assumed that each grid point is associated with a single state variable that can be directly measured. Denote the state variable at location i as X_i, and the measurement of X_i as Z_i, both of which are random variables. Also, define $Z \equiv \{Z_1, Z_2, \cdots, Z_N\}$ and $X \equiv \{X_1, X_2, \cdots, X_N\}$ the sets of all corresponding random variables over the entire search space of size N. Likewise, $V \equiv \{V_1, V_2, \cdots, V_M\}$ denotes the set of random variables representing states in the verification region at t_V, with M being the size of verification space. With slight abuse of notation, this paper does not distinguish a set of random variables from the random vector constituted by the corresponding random variables. Measurement is subject to Gaussian noise that is uncorrelated with any of the state variables as follows.

$$Z_i = X_i + N_i; \;\; N_i \sim \mathcal{N}(0, R_i), \;\; \forall \; i \in \mathbb{Z}_+ \cap [1, N], \tag{5}$$

$$\mathrm{Cov}(N_i, Y_j) = 0, \quad \forall \; Y_j \in X \cup V. \tag{6}$$

This work addresses the general selection problem of selecting n grid points from the search space that will give the greatest reduction in the uncertainty of V, which is written as:

General Selection Problem

$$\mathbf{i}^\star = \arg\max_{\mathbf{i}} I(V; Z_\mathbf{i}) = h(V) - h(V|Z_\mathbf{i}) \tag{7}$$

$$\text{where } \psi_{X \cup V}(x_1, \cdots, x_N, v_1, \cdots, v_M) = \text{given}. \tag{8}$$

$\mathbf{i} \triangleq \{i_1, i_2, \cdots, i_n\} \in \mathbb{S}_n \subseteq \mathbb{Z}_+^n \cap [1, N]$, and $Z_\mathbf{i} \triangleq (Z_{i_1}, Z_{i_2}, \cdots, Z_{i_n})$ where $\mathbb{S}_n$ is the set of n distinct grid points whose cardinality is $\binom{N}{n}$; $\psi_{X \cup V}(\cdot)$ denotes the sufficient statistics to compute the entropy of any set $Y \subset X \cup V$. The condition (8) might be a very stringent requirement in general, since only some of the first and the second moments of the required distribution are usually available in practice. In this work, the statistical ensembles of each random variable are given in advance; thus, one possible way

is empirically generating the approximate joint distribution out of ensembles, which, however, takes a great deal of computational resources.

In case all the random variables are jointly Gaussian one another, the entropy of the set can be evaluated from the covariance information. Then, the selection problem can be written as:

Forward Selection for Gaussian (FSG)

$$\mathbf{i}_F^\star = \arg\max_{\mathbf{i}} \frac{1}{2}\ln\left\{\frac{|\mathrm{Cov}(V)|}{|\mathrm{Cov}(V|Z_\mathbf{i})|}\right\} \tag{9}$$

$$\text{where } \mathrm{Cov}(Y_l, Y_m) = \text{given}, \quad \forall\, Y_l, Y_m \in X \cup V, \tag{10}$$

It is noted that the covariance of each pair of random variables can be easily estimated from the ensemble of each random variable. Although every quantity appearing in (9) can be computed from the given covariance information and measurement model, finding $\mathbf{i}_F^\star$ typically requires a brute force search over the entire candidate space $\mathbb{S}_n$; the selection process is subject to combinatorial explosion, when N and/or n are large. N is usually very large for the observation targeting problem for improving weather prediction. Moreover, computing the conditional covariance $\mathrm{Cov}(V|Z_\mathbf{i})$ and its determinant requires nontrivial computation time. In other words, combinatorial number of computations, each of which takes significant amount of time, are required to find the optimal solution for FSG formulation.

Given these computational issues, this paper proposes the following formulation of the selection problem:

Backward Selection for Gaussian (BSG)

$$\mathbf{i}_B^\star = \arg\max_{\mathbf{i}} I(Z_\mathbf{i}; V) = h(Z_\mathbf{i}) - h(Z_\mathbf{i}|V) \tag{11}$$

$$= \arg\max_{\mathbf{i}} \frac{1}{2}\ln\left\{\frac{|\mathrm{Cov}(X_\mathbf{i}) + R_\mathbf{i}|}{|\mathrm{Cov}(X_\mathbf{i}|V) + R_\mathbf{i}|}\right\} \tag{12}$$

$$\text{where } \mathrm{Cov}(Y_l, Y_m) = \text{given}, \quad \forall\, Y_l, Y_m \in X \cup V. \tag{13}$$

Note that this formulation gives the same solution as FSG under the jointly Gaussian assumption, since the mutual information is commutative.

Proposition 1 $\mathbf{i}_F^\star \equiv \mathbf{i}_B^\star$, since $I(V; Z_\mathbf{i}) \equiv I(Z_\mathbf{i}; V), \forall \mathbf{i} \in \mathbb{S}_n$.

BSG still determines $\mathbf{i}_B^\star$ relying on the brute force search; it is also subject to combinatorial explosion. Nevertheless, it has some preferable features compared with FSG: One thing is $\mathrm{Cov}(Z_\mathbf{i}|\cdot)$ can be expressed as simply $\mathrm{Cov}(X_\mathbf{i}|\cdot) + R_\mathbf{i}$, the first term of which is already embedded in the prior covariance structure. This type of simple relation does not exist for $\mathrm{Cov}(\cdot|Z_\mathbf{i})$ and $\mathrm{Cov}(\cdot|X_\mathbf{i})$, which appear in FSG. This simplicity was utilized in [6] for better formulation and efficient implementation.

Remark 1 In the conventional Kalman filter framework written in Joseph form [16], the conditional covariances in (9), and (12) are computed using

$$P_{X_\mathbf{i}\cup V|Z_\mathbf{i}} = (I - [K_{F_\mathbf{i}}\ \mathbf{0}])\, P_{X_\mathbf{i}\cup V}\, (I - [K_{F_\mathbf{i}}\ \mathbf{0}])^T + K_{F_\mathbf{i}} R_\mathbf{i} K_{F_\mathbf{i}}^T \tag{14}$$

$$P_{X_\mathbf{i}\cup V|V} = (I - [\mathbf{0}\ K_{B_\mathbf{i}}])\, P_{X_\mathbf{i}\cup V}\, (I - [\mathbf{0}\ K_{B_\mathbf{i}}])^T \tag{15}$$

where $K_{F_\mathbf{i}} = P_{VX_\mathbf{i}}\,[P_{X_\mathbf{i}} + R_\mathbf{i}]^{-1}$, and $K_{B_\mathbf{i}} = P_{X_\mathbf{i}V} P_V^{-1}$. The brute force searches for finding $\mathbf{i}_F^\star$ and $\mathbf{i}_B^\star$ will perform the above update equations $\binom{N}{n}$ times. Note, however, that $P_{X_\mathbf{i}|V}$ can be evaluated from the following update equation of a bigger size covariance matrix:

$$P_{X\cup V|V} = \left(I - \begin{bmatrix}\mathbf{0} & K_B\end{bmatrix}\right) P_{X\cup V} \left(I - \begin{bmatrix}\mathbf{0} & K_B\end{bmatrix}\right)^T \tag{16}$$

where $K_B = P_{XV} P_V^{-1}$. Then, computing $P_{X_\mathbf{i}|V}$ amounts to extracting the corresponding submatrix from $P_{X|V}$. Since the number of submatrices to extract is $\binom{N}{n}$, this process is also combinatorial. However, computing the determinant of this submatrix usually takes less time than updating the covariance as (14). Thus, it can be conjectured that BSG search will be computationally less expensive than FSG search. Detailed analytic comparisons of the computation time in the EnSRF framework are in section IV. ∎

BSG search becomes more preferable in case n is smaller than M, since it computes the matrix determinant of a smaller matrix than the FSG has to compute. This is usually the case for the UAV targeting problem in the context of numerical weather prediction. The size of verification region could be very large, while limited resources of UAV might keep us from selecting too many places to take measurement.

B. EnSRF-Based Targeting Problem

The sensor targeting problem concerns improving the weather forecast for the future time t_V broadcasted at another future time t_K ($< t_V$), by deploying additional observation networks from t_1 through t_K, where the known routine sensor networks are assumed to take measurement periodically (every 6 hours in practice) and the analysis ensemble at the current time t_0 is given. Since the actual value of the future measurement that affects the forecast error is not available at the decision time t_0, ensemble-based targeting considers the forecast uncertainty reduction in the *statistical* sense that can be quantified by prior covariance and the sensing noise variance. Also, to avoid computing the effect of the fixed observations repeatedly, the impact of the routine networks is processed in advance; the outcome of this process will be the prior information for the selection process.

To incorporate the effect of temporally-distributed observation, this work considers the noncausal covariance update described by the augmented forecast ensemble: $\mathcal{X}^f \equiv \left[\mathbf{X}_{t_1}^f; \cdots; \mathbf{X}_{t_K}^f; \mathbf{X}_{t_V}^f\right] \in \mathbb{R}^{(K+1)L_S \times L_E}$ where $\mathbf{X}_{t_k}^f$ is the forecast ensemble at t_k propagated from $\mathbf{X}^a(t_0)$. With an appropriate observation matrix $\mathcal{H}$ and Kalman gain $\mathcal{K}$, the ensemble update is written as $\tilde{\mathcal{X}}^a = (I - \mathcal{K}\mathcal{H})\tilde{\mathcal{X}}^f$ without ensemble mean update that depends on the actual value of the measurement. Note also that after incorporating all the fixed networks, only the submatrix of the perturbation ensemble associated with X and V is involved in the selection process.

IV. Computation Time Analysis

As described in Remark 1, the backward search can be faster than the forward search, if the covariance update is computationally expensive. This section details the computation time of both search schemes in the EnSRF-based

targeting framework. The estimated computation time of each scheme will be presented, with pointing out which part of computation makes the difference between the speed of FSG and BSG. The selection process is related to the four computation elements: 1) perturbation ensemble update, 2) covariance matrix computation from the ensemble, 3) determinant calculation of the covariance matrix, and 4) selection of the best candidate from the reward list. This section describes these four by introducing the following four atomic time units: δ_{L_E}, σ_{L_E}, τ_k, and θ_k.

δ_{L_E} represents the time for updating L_E ensemble members associated one state variable by one observation. In the sequential update framework, it will take approximately $pq\delta_{L_E}$ to update p states with q observations. Also, δ_{L_E} itself is proportional to L_E. σ_{L_E} is the time for computing inner product of two vectors with size L_E. Then, the multiplication of $p \times L_E$ matrix and its transpose on its right will take approximately $p^2\sigma_{L_E}$. τ_p denotes the time for calculating the determinant of $p \times p$ symmetric matrix; θ_p denotes the time for selecting the greatest element out of the list with length k. $\tau_p \propto p^3$ for large p, while $\theta_p \propto p$.

In FSG search, for every candidate consisting of n grid points, the ensemble members associated with $n+M$ variables $-$ the candidate points themselves and the verification variables $-$ are updated by n observations. Also, every ensemble update is followed by the covariance matrix construction and the determinant computation of that matrix, which take $(n+M)^2\sigma_{L_E}$, and τ_M, respectively. After all the mutual information has been computed, the best candidate is selected; which takes $\theta_{\binom{N}{n}}$. So the estimated time for FSG to compute the solution is

$$\hat{T}_F = \tbinom{N}{n}\Big[\underbrace{n(n+M)\delta_{L_E}}_{\text{ensemble update}} + \underbrace{(n+M)^2\sigma_{L_E}}_{\text{covariance comp.}} + \underbrace{\tau_M}_{\text{det.}}\Big] + \theta_{\binom{N}{n}}$$

For BSG, for each candidate measurement choice, its conditioned and unconditioned covariances are extracted from the overall covariance matrices corresponding to the entire search space. The ensemble update by V on all X and V takes $M(N+M)\delta_{L_E}$, while construction of two overall covariance matrices takes $2(N+M)^2\sigma_{L_E}$. Determinants of two $n \times n$ matrices are evaluated for each candidate. It is assumed that matrix extraction takes negligible time. Then, the estimated formula for the computation time of BSG is

$$\hat{T}_B = \underbrace{M(N+M)\delta_{L_E}}_{\text{ensemble update}} + 2\underbrace{(N+M)^2\sigma_{L_E}}_{\text{covariance comp.}} + 2\underbrace{\tbinom{N}{n}\tau_n}_{\text{det.}} + \theta_{\binom{N}{n}}$$

Note that $\hat{T}_B$ does not contain the combinatorial feature in the ensemble update and the covariance computation in contrast to $\hat{T}_F$. Even the combinatorial term in $\hat{T}_B$ is likely to be smaller than its counterpart in $\hat{T}_F$, since M is usually bigger than n. Thus, for a reasonable size of problem, BSG may find the optimal solution within acceptable time, while FSG cannot. However, increase of n will accelerate the explosion of $\hat{T}_B$, seeing that $\binom{N}{n}$ is exponential in n and τ_n is cubic in n.

V. Greedy Strategies

A. Algorithms

Computation time for the two full search methodologies given in section III exponentially grows with respect to the number of targeting points. One typical algorithm to avoid this exponential growth is the greedy algorithm that selects the best point one by one [17]. The greedy algorithm based on the forward selection formulation is stated as follows.

Forward Greedy Selection for Gaussian (FGSG)

$$\begin{aligned} i_k^{F\star} &= \arg\max_i \; h(V|Z_{\mathbf{i}_{k-1}^{F\star}}) - h(V|Z_i Z_{\mathbf{i}_{k-1}^{F\star}}) \\ &= \arg\max_i \frac{1}{2}\ln\left\{\frac{|\mathrm{Cov}(V|Z_{\mathbf{i}_{k-1}^{F\star}})|}{|\mathrm{Cov}(V|Z_i Z_{\mathbf{i}_{k-1}^{F\star}})|}\right\}, \forall k \geq 1 \end{aligned} \tag{17}$$

where $Z_{\mathbf{i}_k^{F\star}} \equiv (Z_{i_1^\star}, \cdots, Z_{i_k^{F\star}})$, and $Z_{\mathbf{i}_0^{F\star}} = \emptyset$. For every selection step, the decision of picking a single grid point is made conditioned on all the previous decisions. Since $\mathrm{Cov}(V|Z_{\mathbf{i}_k^{F\star}})$ is known for the k-th step, the required computation per selection step comprises N times of ensemble update followed by the determinant calculation of the updated covariance of V. Thus, the computation time increases as $\mathcal{O}(nN)$, which could be still big in case of large N. This work suggests a backward greedy selection algorithm that can be faster than FGSG.

Backward Greedy Selection for Gaussian (BGSG)

$$\begin{aligned} i_k^{B\star} &= \arg\max_i \; h(Z_i|Z_{\mathbf{i}_{k-1}^{B\star}}) - h(Z_i|VZ_{\mathbf{i}_{k-1}^{B\star}}) \\ &= \arg\max_i \frac{1}{2}\ln\left\{\frac{\mathrm{Var}(X_i|Z_{\mathbf{i}_{k-1}^{B\star}}) + R_i}{\mathrm{Var}(X_i|VZ_{\mathbf{i}_{k-1}^{B\star}}) + R_i}\right\} \end{aligned} \tag{18}$$

$\forall k \geq 1$, where $Z_{\mathbf{i}_k^{B\star}}$ is defined likewise. BGSG selects the site where the difference between the entropy conditioned on previous selections and that conditioned on the previous selections and V. Two aspects that characterize the computational benefits of this algorithm are: 1) BGSG is not involved in computation of the determinant of a large matrix, and 2) only two ensemble updates by $Z_{i_{k-1}^{B\star}}$ is needed for the k-th selection step, since $\mathrm{Var}(\cdot|Z_{\mathbf{i}_{k-2}^{B\star}})$ and $\mathrm{Var}(\cdot|VZ_{\mathbf{i}_{k-2}^{B\star}})$ are evaluated in advance. Note that BGSG gives the same solution as FGSG: $i_k^{F\star} = i_k^{B\star},\ \forall\ 1 \leq k \leq n$.

B. Computation Time

Computation time for the greedy algorithms can also be expressed using the atomic time units defined in section IV. In addition to time for computing the information of gain of each candidate point, FGSG needs additional ensemble update for computing the effect of the selected observation onto the rest of the search space. Thus, the estimated computation time is

$$\begin{aligned} \hat{T}_{FG} =& nN(1+M)\delta_{L_E} + (n-1)(N+M)\delta_{L_E} \\ &+ nN(1+M)^2\sigma_{L_E} + nN\tau_M + n\theta_N. \end{aligned} \tag{19}$$

Note that most terms are proportional to nN. In BGSG, the amount of computation needed for picking the rest of the points is smaller than that for picking the first point, since the impact of V is calculated only at the first selection

step. BGSG only computes the diagonal elements of the covariance matrix of $X \cup V$, which will take $(N+M)\sigma_{L_E}$. Then, the estimated computation time is

$$\begin{aligned}\hat{T}_{BG} =& M(N+M)\delta_{L_E} + 2(n-1)(N+M)\delta_{L_E} \\ &+ 2n(N+M)\sigma_{L_E} + n\theta_N.\end{aligned} \quad (20)$$

Recalling that $\mathcal{O}(1) \leq n \leq M \ll N$ for many problems, it is pointed out that $\hat{T}_{BG}$ becomes smaller than $\hat{T}_{FG}$ as n becomes larger. The δ_{L_E} term grows in $\mathcal{O}(nMN)$ for FGSG but $\mathcal{O}((M+2n)N)$ for BGSG; the σ_{L_E} terms grow in $\mathcal{O}(nNM^2)$ and $\mathcal{O}(2nN)$, respectively. Also, if M is very large, the τ_M in $\hat{T}_{FG}$ can be very big, since it increases in $\mathcal{O}(M^3)$. Therefore, BGSG will be much more beneficial than FGSG for large problems such as weather applications.

VI. Numerical Simulations

A. Lorenz-95 Model

The Lorenz-95 model [8] is an idealized chaotic model in which key aspects of weather dynamics are included, such as energy dissipation, advection, and external forcing. As such, it has been successfully implemented for the initial verification of the numerical weather prediction algorithms. [12] In this paper, the original one-dimensional model is extended to two-dimensions representing the mid-latitude region of the northern hemisphere. The system equations are

$$\begin{aligned}\dot{y}_{ij} =& (y_{i+1,j} - y_{i-2,j})\, y_{i-1,j} + \mu\,(y_{i,j+1} - y_{i,j-2})\, y_{i,j-1} \\ &- y_{ij} + F,\ (i = 1, \ldots, L_{on},\ j = 1, \ldots, L_{at}).\end{aligned} \quad (21)$$

where the subscript i denotes the west-to-eastern grid index, while j denotes the south-to-north grid index. The dynamics of the (i,j)-th grid point depends on its neighbors through the advection terms, on itself by the dissipation term, and on the external forcing ($F = 8$ in this work). There are $L_{on} = 36$ longitudinal and $L_{at} = 9$ latitudinal grid points. The dynamics in (21) are subject to cyclic boundary conditions in longitudinal direction ($y_{i+L_{on},j} = y_{i-L_{on},j} = y_{i,j}$) and to the constant advection condition($y_{i,0} = y_{i,-1} = y_{i,L_{at}+1} = 4$ in advection terms) in the latitudinal direction, to model the mid-latitude area as an annulus. Regarding to the time scale, 0.05 time units are are equivalent to real 6 hours.

B. Multiple Targeting

For numerical validation of the proposed backward scheme, the following multiple targeting scenarios are considered. The routine network with size 93 is assumed to be already deployed over the grid space (blue 'o' in figs. 2 and 3). The static network is dense in two portions of the grid space called lands, while it is sparse in the other two portions of the space called oceans. The routine network takes measurements every 0.05 time units. The leftmost part of the right land mass (consisting of 10 grid points depicted with red '□') is the verification region. The goal is to reduce the forecast uncertainty in the verification region 0.5 time units after the targeting time. The targeting time $t_K = 0.05$ with $K = 1$. The analysis ensemble at t_0 with size 1024 is obtained by running the EnSRF with routine observations for 500 time units. All the grid points in the left ocean are considered as a candidate point to locate additional measurement at t_1; therefore, $N = 108$. The measurement noise variance is 0.2^2 for routines and 0.02^2 for additional observations. With this setting, the targeting results, as the number of targeting points n being increased, are obtained by using the four algorithms: FSG/BSG/FGSG/BGSG.

It should be first pointed out that the backward algorithm gives the same solution as the forward algorithm for all the cases – not only the optimal locations but also the objective function values. This implies that the Gaussian assumption in computing the mutual information effectively holds. Figure 2 illustrates the optimal sensor locations(cyan '◇') by the exhaustive search strategy, while figure 3 shows the greedily selected sensor locations for $n = 1$, 3, and 5. The shaded contour in both figures represents $I(V; Z_i)$ for each grid point. It is noted that the optimal solution for $n = 5$ looks very different from the greedy solution, as it does not even select two dark points that it did select in case of $n = 3$. This is an exemplary case of representing suboptimality of the greedy strategy. Table I represents the mutual information value by different strategies for different n. It is seen that the performance gap between strategies becomes more distinctive as n increases; meanwhile, the gap between the optimal and the greedy one is smaller than that between the greedy and the naive solution that makes decision only based on $I(V; Z_i)$.

Tables II and III represent the actual and estimated computation time of each algorithm with respect to n. The atomic time units for computation time estimation are determined by the Monte Carlo simulation. Each atomic time unit has the value of: $\delta_{L_E} = 58.9$ μs, $\sigma_{L_E} = 2.6$ μs, $\tau_k = 0.7k^{0.6}$ μs, and $\theta_k = 7.6 \times 10^{-3}k$ μs. It is found that τ_k grows slower than k^3 for $k \leq 50$. Note in Table II that BSG is much faster and its rate of computation time growth is slower than FSG. Given that a real scenario is much bigger than this example, it is clear that FSG is not practical to implement for multiple targeting problems. The results confirm that the backward algorithm is much faster, and should be practical for selecting up to a few measurement locations. The superiority of backward scheme extends to the greedy case as well (Table III). Furthermore, note that the forward greedy algorithm actually takes longer than the backward full search algorithm when $n = 2$; this implies the optimal solution can be obtained by the backward scheme without sacrificing any computation resource at all. Finally, the estimated values of the computation time provides sufficient accuracy with less than 30% error, which must be small enough to indicate the tractability of the problem.

TABLE I
$I(V; Z_{\mathbf{i}^\star})$ WITH RESPECT TO THE STRATEGY

N	n	Optimal	Greedy	Naive
108	1	0.46	0.46	0.46
108	3	1.24	1.19	1.02
108	5	1.95	1.86	1.66

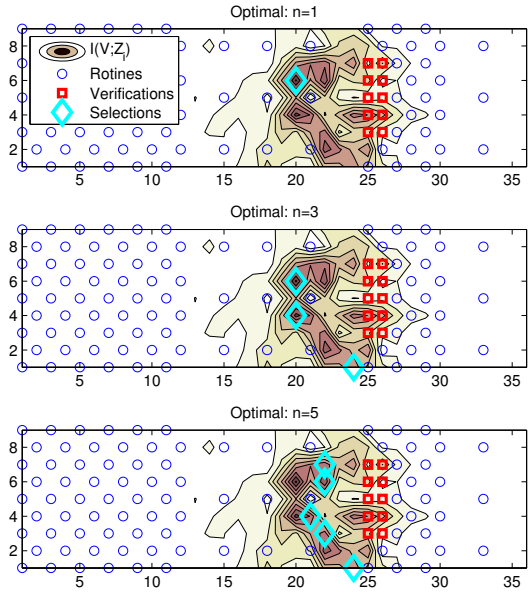

Fig. 2. Multiple targeting with full search algorithms

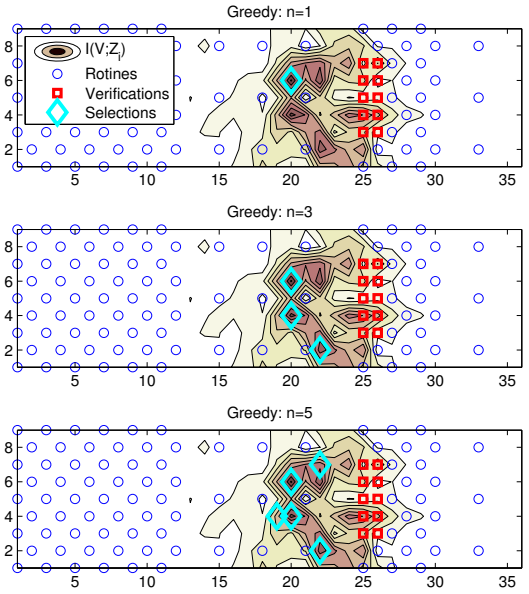

Fig. 3. Multiple targeting with greedy search algorithms

VII. Conclusions

This paper presented a backward formulation for the large scale sensor targeting incorporated with ensemble-based filtering, which enhanced the computational efficiency by reducing the number of times of ensemble updates. It was shown that the proposed approach provides the same solution as the conventional forward approach due to the commutativity of mutual information. Computation time analysis and numerical simulations using an idealized chaos model both supported the computational benefit of the proposed scheme. Future research will apply the proposed approach to more realistic weather models.

Acknowledgment

This work is funded by NSF CNS-0540331 as part of the DDDAS program with Dr. Frederica Darema as the overall program manager.

References

[1] A. Krause and C. Guestrin, "Near-Optimal Nonmyopic Value of Information in Graphical Methods," *21st Conference on Uncertainty in Artificial Intelligence*, Edinburgh, Jul. 2005.

[2] C. Guestrin, A. Krause, and A. Singh, "Near-Optimal Sensor Placements in Gaussican Processes," *22nd International Conference on Machine Learning*, Bonn, Aug. 2005.

[3] A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg, "Near-Optimal Sensor Placements: Maximizing Information while Minimizing Communication Cost," *5th International Conference on Information Processing in Sensor Networks*, Apr. 2006.

[4] F. Zhao, J. Shin, and J. Reich, "Information-Driven Dynamic Sensor Collaboration," *IEEE Signal Processing Magazine*, Vol.19, No.2, 2002.

[5] E. Ertin, J.W. Fisher, and L.C. Potter, "Maximum Mutual Information Principle for Dynamic Sensor Query Problems," *3rd International Conference on Information Processing in Sensor Networks*, 2003.

[6] J.L. Williams, J.W. Fisher III, and A.S. Willsky, "An Approximate Dynamic Programming Approach to a Communication Constrained Sensor Management," *8th Conference of Information Fusion*, 2005.

[7] H. Wang, K. Yao, G. Pottie, and D. Estrin, "Entropy-Based Sensor Selection Heuristic for Target Localization," *3rd International Conference on Information Processing in Sensor Networks*, Apr. 2004.

[8] E.N. Lorenz and K.A. Emanuel, "Optimal Sites for Supplementary Weather Observations: Simulation with a Small Model," *Journal of the Atmospheric Sciences*, Vol.55, No.3, pp.399-414, 1998.

[9] T.N. Palmer, R. Gelaro, J. Barkmeijer, and R. Buizza, "Singular Vectors, Metrics, and Adaptive Observations," *Journal of the Atmospheric Sciences*, Vol 55, No.4, pp.633-653, 1998.

[10] D.N. Daescu and I.M. Navon, "Adaptive Observations in the Context of 4D-Var Data Assimilation," *Meteorology and Atmospheric Physics*, Vol.85, pp.205-226, 2004.

[11] S.J. Majumdar, C.H. Bishop, B.J. Etherton, and Z. Toth, "Adaptive Sampling with the Ensemble Transform Kalman Filter: Part II Field Programming Implementation," *Monthly Weather Review*, Vol.130, No.3, pp.1356-1369, 2002.

[12] M. Leutbecher, "Adaptive Observations, the Hessian Metric and Singular Vectors," *ESMWF Seminar 2003*, 2003.

[13] J.S. Whitaker and H.M. Hamill, "Ensemble Data Assimilation without Perturbed Observations," *Monthly Weather Review*, Vol.130, No.7, pp.1913-1924, 2002.

[14] G. Evensen and P.J. van Leeuwen, "Assimilation of Altimeter Data for the Agulhas Current Using the Ensemble Kalman Filter with a Quasigeostrophic Model," *Monthly Weather Review*, Vol.123, No.1, pp.85-96, 1996.

[15] T.M. Cover and J.A. Thomas, *Elements of Information Theory*, Wiley Interscience, 1991.

[16] M.S. Grewal and A.P. Andrews, *Kalman Filtering: Theory and Practice Using MATLAB*, Wiley Interscience, 2001.

[17] W. Vazirani, *Approximation Algorithms*, Springer, 2001.

TABLE II

Computation time for full search algorithms

N	n	T_F (s)	T_B (s)	$\hat{T}_F$ (s)	$\hat{T}_B(s)$
108	1	0.13	0.13	0.10	0.14
108	2	8.17	0.16	10.4	0.15
108	3	344.5	0.72	561.4	0.61
108	4	–	18.8	5.7 hr	14.6
108	5	–	461.3	6.5 day	349.3
108	10	–	–	16000 yr	7.2 yr

TABLE III

Computation time for greedy search algorithms

N	n	T_{FG} (s)	T_{BG} (s)	$\hat{T}_{FG}$ (s)	$\hat{T}_{BG}(s)$
108	1	0.13	0.05	0.11	0.07
108	3	0.34	0.05	0.33	0.10
108	5	0.59	0.08	0.55	0.13
108	10	1.14	0.11	1.11	0.20