Prize-Collecting Data Fusion for Cost-Performance Tradeoff in Distributed Inference

Animashree Anandkumar*, Meng Wang*, Lang Tong*, and Ananthram Swami†

*ECE Dept., Cornell University, Ithaca, NY 14853, USA. Email: {aa332@,mw467@,ltong@ece.}cornell.edu
†Army Research Laboratory, Adelphi, MD 20783, USA. Email: a.swami@ieee.org

Abstract—A novel formulation for optimal sensor selection and in-network fusion for distributed inference known as the prize-collecting data fusion (PCDF) is proposed in terms of optimal tradeoff between the costs of aggregating the selected set of sensor measurements and the resulting inference performance at the fusion center. For i.i.d. measurements, PCDF reduces to the prize-collecting Steiner tree (PCST) with the single-letter Kullback-Leibler divergence as the penalty at each node, as the number of nodes goes to infinity. PCDF is then analyzed under a correlation model specified by a Markov random field (MRF) with a given dependency graph. For a special class of dependency graphs, a constrained version of the PCDF reduces to the PCST on an augmented graph. In this case, an approximation algorithm is given with the approximation ratio depending only on the number of profitable cliques in the dependency graph. Based on these results, two heuristics are proposed for node selection under general correlation structure, and their performance is studied via simulations.

Index Terms—Optimal Node Selection, Sensor Networks, In-network Aggregation, Detection, Prize-Collecting Steiner Tree.

I. INTRODUCTION

Consider a sensor network deployed in an area taking measurements for distributed inference. Here, a designated fusion center collects the sensor measurements and makes a final decision about the underlying signal field. The classical works on this topic are concerned with optimal inference rules [2], and the role of network constraints is not considered.

Sensor networks have many resource constraints, and it may not be feasible to route all the sensor measurements for inference. It is then crucial for the fusion center to select a set of sensor measurements based on the tradeoff between the routing costs, and the resulting inference performance at the fusion center. Intuitively, it is more economical to select nearby sensors with "informative" data for inference.

Efficient sensor selection for inference presents several challenges since optimization of cost-performance tradeoff is highly non-separable, where the costs (such as energy) of routing measurements and the resulting inference performance at the fusion center are intertwined in a complex way. On the other hand, a brute force approach of searching over all possible sensor subsets for selection is not feasible even for moderate-sized networks. Are there any heuristics for sensor selection with efficient cost-performance tradeoff? Is it possible to provide approximation guarantees for the heuristics with respect to the optimal solution? How do factors such as the correlation model and node topology affect the efficiency of these heuristics? How do we aggregate[1] data at intermediate nodes in a cost-efficient manner, and yet provide guaranteed inference performance at the fusion center? We address these issues in this paper.

A. Summary of Results

This paper considers selection of sensors to achieve optimal cost-performance tradeoff for inference. The costs are incurred in routing and aggregating the selected subset of sensor measurements, and the performance is in terms of the probability of error in inferring the correct hypothesis at the fusion center, given the aggregated data. The contributions are three fold. First, we propose a formulation for optimal sensor selection and in-network fusion known as the *prize-collecting data fusion* (PCDF). Second, we prove its reduction to a known optimization problem for certain correlation structures. Third, for general correlation, we propose two heuristics, and study their performance through simulations.

When the sensor measurements are i.i.d. and the number of sensors goes to infinity, PCDF reduces to an optimization problem known as the *prize-collecting Steiner tree* (PCST) [3]. It is defined as the sub-tree rooted at a specified vertex (fusion center in our case) that minimizes the sum of edge costs in the tree plus the penalties of the nodes not spanned by it. For PCDF with i.i.d. data, the node penalties are uniform, and given by the single-letter Kullback-Leibler divergence (KLd).

We then consider correlated sensor measurements via a *Markov random field* (MRF) model with a given (undirected) *dependency graph* [4]. For a special class of dependency graphs, a constrained form of PCDF asymptotically reduces to PCST on an augmented graph, where the augmentation involves adding new nodes and edges to account for increase in aggregation costs due to the presence of correlation. In general, finding the constrained PCDF is NP-hard and we resort to approximations via the PCST reduction. The *approximation ratio* ρ of any polynomial-time algorithm guarantees that its

[1] The terms aggregation and fusion are used interchangeably.

output is no worse than ρ times the optimal value. We give an approximation algorithm where the approximation ratio depends only on the number of "profitable" cliques in the dependency graph.

We then develop group selection heuristics for general correlation structures based on the above approximation, viz., component selection and clique selection, and study their performance through simulations. It is observed that the heuristics perform substantially better than the optimal selection scheme which routes the selected measurements to the fusion center without any aggregation at the intermediate nodes. Hence, our approach of incorporating aggregation into the sensor selection formulation substantially reduces routing costs leading to efficient selection policies. We then study the influence of node topology and observe that at sparse spatial dependencies, a clustered node placement achieves better cost-performance tradeoff compared to a uniform placement. These results have direct implications on designing good node placement strategies for cost-performance tradeoff.

B. Related Work

Energy-efficient inference in sensor networks has been considered before for some special correlation models (e.g., [5]). More relevant here is the notion of in-network aggregation, considered for specific function computation in [6]. However, the mechanisms to aggregate a subset of measurements and selection of such a subset are not considered.

In [7], [8], we consider minimum cost aggregation of *all* the sensor measurements under the Markov random field model under the constraint of achieving optimal inference at the fusion center, but we do not deal with the issue of sensor selection. In [9], we consider optimal node density for inference leading to probabilistic sleeping strategies to meet the energy constraints. In contrast, this work uses the approach of deterministic sensor selection to achieve energy efficiency.

To the best of our knowledge, the problem of optimal node selection (e.g., see survey [10]) has not been considered in conjunction with in-network fusion before. Indeed in single-hop networks, there is no need for data fusion. But most large networks are multi-hop, and routing costs are substantially reduced through fusion at intermediate nodes, as seen in simulations in Section VI.

Many works on node selection assume perfect sensing of a region (e.g., [11]). In contrast, our work explicitly models correlated imprecise measurements via a Markov random field, and is the basis for selecting "informative" sensors for inference. Indeed, there is also the issue of accuracy in learning the statistical model. Conceding this limitation, we aim to gain insights through our model-based framework.

II. SYSTEM MODEL & PROBLEM FORMULATION

In this paper, we will consider various graphs: the *dependency graphs* specifying the correlation structure of sensor measurements, the *network graphs* denoting feasible links for communication, and the *fusion digraphs* denoting links used by a policy to route and aggregate data.

A. Measurements: Correlation & Inference Model

We assume that the measurements are drawn from a Markov random field (MRF). Let $\mathbf{Y}_V = [Y_i, i \in V]^T$ denote the measurements in any set V. If $\mathbf{Y}_V$ is a MRF with dependency graph $\mathrm{DG}(V)$, then under the positivity condition, its joint pdf f_V is given by the Hammersley-Clifford theorem [4],

$$-\log f_V(\mathbf{Y}_V; \Upsilon) = \sum_{c \in \mathcal{C}} \psi_c(\mathbf{Y}_c), \tag{1}$$

where $\mathcal{C}$ is the collection of (maximal) cliques[2] in $\mathrm{DG}(V)$ and the function ψ_c is known as the normalized[3] *potential* for clique c. Hence, $\{\mathrm{DG}(V), \mathcal{C}, \psi\}$ represents a MRF. For a discussion on the use of MRF for spatial correlation, see [7].

We consider the *binary hypothesis-testing* problem with null hypothesis $\mathcal{H}_0$ and alternative $\mathcal{H}_1$. Under either hypothesis, we assume that the measurements are drawn from distinct MRFs,

$$\mathcal{H}_0 : \{\mathrm{DG}_0(V), \mathcal{C}_0, \psi_0\} \; ; \; \mathcal{H}_1 : \{\mathrm{DG}_1(V), \mathcal{C}_1, \psi_1\}. \tag{2}$$

In order to quantify inference performance, we consider the *Neyman-Pearson* criterion [2], where for a fixed false-alarm probability (type-I error), the detector at the fusion center is optimal in terms of the type-II error probability P_M.

B. Network and Cost Model

The network is connected via a *network graph* of feasible links with given routing costs. For optimization of costs, we only need to work with the *metric closure*[4] of the network graph, denoted by $G_n(V)$, and the metric cost for each node pair (i, j), denoted by $C(i, j)$. For any graph G, let $\mathrm{C}(G)$ denote the total metric cost of using all its links. Communication between the nodes is perfect and scheduled so as to avoid interference.

Nodes communicate in the form of packets. Each packet contains bits for at most one (quantized) real variable and other overhead bits. The quantization error is assumed to be small and ignored here. A node can function as an aggregator (combines incoming packets with its own measurement) or a router (forwarding packets without combination). An aggregation scheme consists of the transmitter-receiver pairs with the respective links used which form the *fusion digraph* G_f, the transmission schedule, and the aggregation algorithm.

C. Problem Formulation

The goal of this paper is to select an optimal sensor subset[5] $V_s \subset V$, given the entire set V, and to incorporate in-network aggregation of the measurements $\mathbf{Y}_{V_s}$ before delivery to the fusion center $v_0 \in V$. It is not possible to quantify

---

[2]A clique refers to a maximal clique unless otherwise mentioned.

[3]In general, finding the normalization constant is NP-hard, but can be carried out at the fusion center without sensor data.

[4]The metric closure on graph G, is defined as the complete graph where the cost of each edge (i, j) in the metric closure is the cost of the shortest path between i and j in G.

[5]The unselected nodes can still function as routers and forward data.

inference performance under arbitrary aggregation. Hence, we limit ourselves to aggregation schemes which guarantee the same inference performance as the centralized scheme, i.e., *as if* the fusion center had direct access to the selected measurements $\mathbf{Y}_{V_s}$. In this case, there is no performance loss due to aggregation at the intermediate nodes. In statistical theory, a *sufficient statistic* is a well-behaved function of the data, which is as informative as the raw data for inference [12]. Hence, a scheme which computes and delivers a sufficient statistic results in no loss of inference performance due to aggregation.

We assume that the optimal Neyman-Pearson (NP) detector is used at the fusion center, and that the inference performance is measured by the NP type-II error probability P_M. We are thus interested in subset selection $V_s \subset V$ and design of aggregation scheme $\Gamma(V_s)$ delivering a sufficient statistic of its measurements $\mathbf{Y}_{V_s}$ such that optimal linear tradeoff is achieved between the total routing costs $C(\Gamma(V_s))$ and a *penalty* function π, based on the NP type-II error $P_M(V_s)$,

$$\text{opt}(V, C, \gamma\pi) := \min_{V_s \subset V, \Gamma(V_s)} \Big[C(\Gamma(V_s)) + \gamma\pi(V \backslash V_s) \Big], \ \gamma > 0 \quad (3)$$

where $V \backslash V_s := \{ i : i \in V, i \notin V_s \}$ and π is given by

$$\pi(V \backslash V_s) := \log \frac{P_M(V_s)}{P_M(V)} > 0, \quad \forall V_s \subset V. \quad (4)$$

When we select all the sensors ($V_s = V$), (4) evaluates to zero, and there is no loss in performance since no measurement is dropped. On the other hand, for a proper subset ($V_s \subsetneq V$), we incur a loss in performance and hence, pay a positive penalty in terms of the fraction of increase in error probability due to non-selection of nodes in $V \backslash V_s$. Since we collect prizes or penalties for nodes not selected, and incorporate fusion over the selected data, we will henceforth refer to the optimal solution in (3) as the prize-collecting data fusion (PCDF) scheme.

The parameter γ is known as the *tradeoff factor*, and is used to adjust the relative importance of cost and performance. Note that the optimization in (3) is the Lagrangian dual for the problem of finding the optimal fusion scheme under a constraint on the inference performance or vice versa. Hence, once we have an algorithm to find the (approximate) solution to (3), we can use it in the constrained optimization problems. This aspect is however not studied in this paper, and we will limit to finding solutions to (3). Denote the objective in (3) as

$$\text{obj}(V_s, \Gamma(V_s); V, C, \gamma\pi) := \Big[C(\Gamma(V_s)) + \gamma\pi(V \backslash V_s) \Big], \quad (5)$$

and the optimal node subset and fusion scheme by

$$[V_*, \Gamma_*(V_*)] := \arg \min_{V_s \subset V, \Gamma(V_s)} \text{obj}(V, C, \gamma\pi). \quad (6)$$

When the tradeoff factor is sufficiently large ($\gamma \to \infty$), the optimal tradeoff problem in (3) reduces to minimum cost
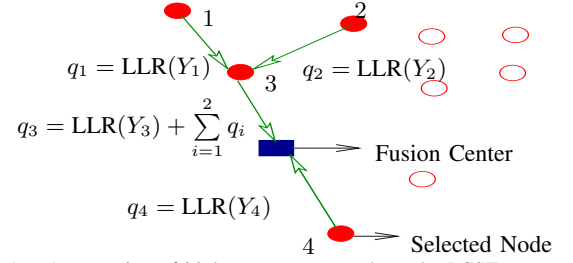

Fig. 1.   Aggregation of i.i.d. measurements along the PCST.

fusion, considered in [8], where optimal inference is required and hence, all the nodes are selected, and the goal is to find the fusion scheme which minimizes the total routing costs while ensuring delivery of a sufficient statistic to the fusion center. When the tradeoff factor is sufficiently small ($\gamma \to 0$), none of the nodes are selected.

$$\lim_{\gamma \to 0} V_*(V, C, \gamma\pi) \to \emptyset, \quad \lim_{\gamma \to \infty} V_*(V, C, \gamma\pi) \to V.$$

D. Preliminary Observations & Results

For binary hypothesis testing, the log-likelihood ratio (LLR) is *minimally* sufficient and represents maximum reduction in dimensionality of raw data. It is given by

$$\text{LLR}(\mathbf{Y}_{V_s}) := \log \frac{f_{V_s}(\mathbf{Y}_{V_s}; \mathcal{H}_0)}{f_{V_s}(\mathbf{Y}_{V_s}; \mathcal{H}_1)}, \quad (7)$$

where $f_{V_s}(\mathbf{Y}_{V_s}; \mathcal{H}_j)$ is the pdf of the measurements $\mathbf{Y}_{V_s}$ under hypothesis $\mathcal{H}_j$. Hence, the optimal aggregation scheme in (3), for a given node subset V_s, is a scheme $\Gamma(V_s)$ computing and delivering $\text{LLR}(V_s)$ to the fusion center with minimum total cost $C(\Gamma(V_s))$.

For the penalty function in (4), in general, the error probability P_M does not have a closed form, and hence, an analytical solution to (3) is not tractable. We focus on the large-network scenario, where the error probability P_M can be approximated by the *error exponent* [12]. When the type-II error $P_M(V)$ decays exponentially with the sample size $|V|$, for a fixed type-I error, the NP error exponent is given by

$$\mathcal{D} := - \lim_{|V| \to \infty} \frac{1}{|V|} \log P_M(V). \quad (8)$$

We will see that we can replace the error probability P_M in (4) by an expression based on the error exponent in (8), and yet achieve optimality with respect to (3), as the number of nodes goes to infinity.

III. IID MEASUREMENTS

We now consider the case when all the sensor measurements are i.i.d. under each hypothesis, $Y_i \overset{i.i.d.}{\sim} f(Y; \mathcal{H}_j)$, for $j = 0, 1$. We first solve a different optimization problem based on (8) and then prove its asymptotic convergence to (3).

For i.i.d. data, from Stein's Lemma [12, Thm. 12.8.1], the exponent $\mathcal{D}$ in (8) is the Kullback-Leibler divergence (KLd)

$$\mathcal{D} = D(f(Y_1;\mathcal{H}_0)||f(Y_1;\mathcal{H}_1)):=\int_y \log\frac{f(y;\mathcal{H}_0)}{f(y;\mathcal{H}_1)}f(y;\mathcal{H}_0)dy$$

We now consider a new penalty function which assigns uniform penalty to each unselected node equal to the KLd D. Hence, if V_s is the selected subset, the penalty is given by

$$\pi^{iid}(V\backslash V_s):=[|V|-|V_s|]D, \qquad (9)$$

First, we establish that the optimal solution under the penalty function π in (4) is the same as the optimal solution with penalty π^{iid}, as the number of nodes goes to infinity.

Theorem 1 (Asymptotic optimality of PCST for i.i.d. data): Under bounded link costs, we have

$$\lim_{|V|\to\infty}\frac{\mathrm{opt}(V,C,\gamma\pi)}{\mathrm{opt}(V,C,\gamma\pi^{iid})}\to 1, \quad \forall\gamma>0. \qquad (10)$$

Proof: See [13]. □

Hence, it suffices to solve the optimization with π^{iid} instead of π for asymptotic networks, given by

$$\mathrm{opt}(V,\mathrm{C},\gamma\pi^{iid}):=\min_{V_s\subset V,\Gamma(V_s)}\Big[\mathrm{C}(\Gamma(V_s))+\gamma[|V|-|V_s|]D\Big]. \quad (11)$$

In order to incorporate in-network aggregation in (11), we need an explicit form for $\mathrm{LLR}(\mathbf{Y}_{V_s})$ since it needs to be computed by the fusion scheme. For i.i.d. data, it is

$$\mathrm{LLR}(\mathbf{Y}_{V_s}) = \sum_{i\in V_s}\log\frac{f(Y_i;\mathcal{H}_0)}{f(Y_i;\mathcal{H}_1)}, \quad \forall V_s\subset V, \qquad (12)$$

which is a simple sum function in the selected nodes. In the theorem below, we prove that the optimal solution to (11) is the prize-collecting Steiner tree (PCST).

Theorem 2 (Selection & aggregation of i.i.d. data): The optimal solution to (11) is aggregation along the prize-collecting Steiner tree rooted at the fusion center v_0, and edges directed towards v_0: each node i in the PCST computes and transmits q_i to its immediate successor, given by

$$q_i = \mathrm{LLR}(Y_i) + \sum_{j\in\mathcal{N}_p(i)}q_j, \qquad (13)$$

where $\mathcal{N}_p(i)$ is the set of immediate predecessors of i in the directed PCST.

Proof: The LLR sum function in (12) over a selected subset V_s can be computed along the edges of a tree spanning V_s, rooted at and directed towards the fusion center, and V_s should be selected so as to achieve optimality in (11). By definition, it is given by the PCST. □

Hence, the optimal aggregation for i.i.d. data is along the directed PCST. A schematic of the scheme is shown in Fig.1. In general, finding the PCST is NP-hard. In [3], an approximation algorithm for the PCST with approximation ratio $2 - (|V| - 1)^{-1}$ for any node set V is proposed, and is referred to as the *Goemans-Williamson* (GW) algorithm.

Theorem 2 establishes the optimality of PCST for the penalty function π^{iid} in (9). From Theorem 1, the PCST is also optimal for the penalty function π in (4), when the network size goes to infinity. Hence, the PCDF in (3) reduces to aggregation along the PCST for i.i.d data, as the network size goes to infinity, and the GW-algorithm approximates the PCST with a proven guarantee of $2 - (|V| - 1)^{-1}$.

IV. CORRELATED MEASUREMENTS: MRF MODEL

We now generalize the results to the case when the measurements are correlated according to a Markov random field model, described in Section II-A. Several new challenges arise here. First, the LLR is no longer a simple sum function as in the i.i.d. case in (12). Hence, the structure of fusion schemes computing the LLR is not clear. Second, the error exponent $\mathcal{D}$ is no longer the single-letter KLd as for i.i.d data, and hence, the exponent-based penalty may not be separable in the nodes. Third, nodes cannot be assigned uniform penalties as in the i.i.d. case, since they affect inference performance differently in the presence of correlation.

With the above challenges, it is not tractable to solve the PCDF problem, defined in (3). Instead, we solve (3) under an additional constraint that the subsets V_s considered are only those that span a sub-collection of cliques of the dependency graph $\mathcal{C}_s\subset\mathcal{C}$, and is referred to as the *constrained* PCDF,

$$\mathrm{opt\_clique}(V,\mathrm{C},\gamma\pi):=\min_{\substack{V_s\subset\mathcal{C}_s\subset\mathcal{C}\\\Gamma(V_s)}}\Big[\mathrm{C}(\Gamma(V_s))+\gamma\pi(V\backslash V_s)\Big]. \quad (14)$$

In other words, the selection policy is coarser since it selects or rejects cliques of nodes instead of individual ones. Since we are ruling out certain subsets for selection, we cannot guarantee optimality with respect to (3).

A. In-network Aggregation of LLR

In order to design a fusion scheme for computing the LLR, we need its explicit characterization. For testing of MRFs in (2), define the joint dependency graph, $\mathrm{DG}(V):=\mathrm{DG}_0(V)\cup\mathrm{DG}_1(V)$. Henceforth, we only work with $\mathrm{DG}(V)$. Using the MRF form in (1), the LLR of the measurements $\mathbf{Y}_V$ in (7) is based on the cliques in $\mathrm{DG}(V)$

$$\begin{aligned}\mathrm{LLR}(\mathbf{Y}_V) &:= \log\frac{f_V(\mathbf{Y}_V;\mathcal{H}_0)}{f_V(\mathbf{Y}_V;\mathcal{H}_1)},\\ &= \sum_{a\in\mathcal{C}_0}\psi_{1,a}(\mathbf{Y}_a) - \sum_{b\in\mathcal{C}_1}\psi_{0,b}(\mathbf{Y}_b) \quad (15)\\ &:= \sum_{c\in\mathcal{C}}\phi_c(\mathbf{Y}_c), \quad \mathcal{C}:=\mathcal{C}_0\cup\mathcal{C}_1. \qquad (16)\end{aligned}$$

Comparing the above form with that for i.i.d data in (12), we see that correlation increases the complexity of the LLR.
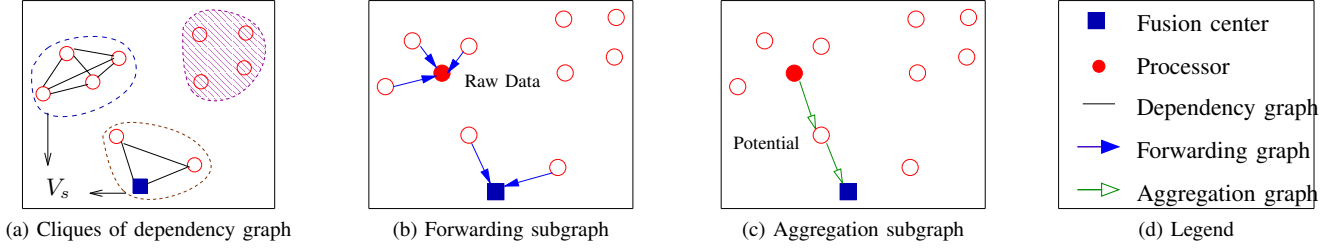
Fig. 2. In-network Aggregation for inference: computation of the log-likelihood ratio $\mathrm{LLR}(\mathbf{Y}_{V_s})$ of a given node subset V_s.

For any subset $V_s \subset V$, its marginal LLR can also be expressed based on the clique set $\mathcal{C}'$ of its dependency graph $\mathrm{DG}'(V_s)$

$$\mathrm{LLR}(\mathbf{Y}_{V_s}) = \sum_{c \in \mathcal{C}'} \phi_c'(\mathbf{Y}_c), \qquad (17)$$

where $\mathrm{DG}'(V_s):=\mathrm{DG}_0'(V_s) \cup \mathrm{DG}_1'(V_s)$, and $\mathrm{DG}_j'(V_s)$ is the dependency graph of the marginal pdf $f_{V_s}(\mathbf{Y}_{V_s}; \mathcal{H}_j)$, for $j = 0, 1$. In general, $\mathrm{DG}'(V_s)$ is not a subgraph of $\mathrm{DG}(V)$ and $\mathcal{C}'$ is not contained in $\mathcal{C}$. Hence, the structure of the marginal LLR and its fusion scheme change with the selected set V_s.

We now describe the structure of fusion schemes computing the LLR of a given subset V_s. See Fig.2. The issue of optimal selection of V_s will be considered later. Given the dependency graph $\mathrm{DG}'(V_s)$, the computation is in two stages. First, the data $\mathbf{Y}_c$ are forwarded from all the members of clique $c \in \mathcal{C}'$ to compute its potential $\phi_c'(\mathbf{Y}_c)$ at an assigned *processor*, denoted by $\mathrm{Proc}(c)$. The set of links used for such data forwarding in all the cliques form the *forwarding* graph (FG).

In the second stage of LLR computation, all the clique potentials are summed up and delivered to the fusion center, using a set of links referred to as the *aggregation* subgraph (AG). The tuple with the forwarding and aggregation subgraphs of a fusion scheme is the *fusion digraph*, $G_f:=\{\mathrm{FG}, \mathrm{AG}\}$, since it is the complete set of links used by the fusion scheme. The total routing costs of the fusion scheme is

$$\mathrm{C}(G_f) = \mathrm{C}(\mathrm{FG}) + \mathrm{C}(\mathrm{AG}). \qquad (18)$$

For finding the constrained PCDF in (14), we thus need to find a fusion scheme which minimizes the sum of routing costs in the two stages of LLR computation.

B. Error Exponent & Penalty Function

Along the lines of our approach for i.i.d. data, in the constrained PCDF problem in (14), we replace the error-probability based penalty π with the error exponent $\mathcal{D}$ for MRF hypothesis testing. Under some conditions [13], it can be shown that the error exponent $\mathcal{D}$ is given by

$$\mathcal{D} = \mathrm{p} \lim_{n \to \infty} \frac{1}{n} \sum_{c \in \mathcal{C}} \mathbb{E}(\phi_c(\mathbf{Y}_c)|V; \mathcal{H}_0), \qquad (19)$$

Hence, the exponent is given by the limit of the normalized sum of functions over the dependency cliques.

We define a new penalty function π^{clq} based on the error exponent to be used in the optimization in (14), where the unselected cliques are assigned penalty

$$\pi^{clq}(\mathcal{C} \backslash \mathcal{C}_s):= \sum_{c \in \mathcal{C} \backslash \mathcal{C}_s} \left(\mathbb{E}(\phi_c(\mathbf{Y}_c)|V; \mathcal{H}_0) \right)^+, \qquad (20)$$

and use it instead of the original penalty function π in (4) based on the error probability.

C. Special Case of MRF: Disjoint Cliques

We now provide approximation guarantees and convergence results for (14) under a special class of dependency graphs. This in turn inspires the development of a general class of heuristics for any dependency graph in Section V. We consider the special case when all the cliques in the joint dependency graph $\mathrm{DG}(V)$ are disjoint. This can occur for instance, when nodes are placed according to a cluster process and the dependency graph is given by a disk graph. See Section VI. Here, the form of the LLR in (17) and the exponent in (19) are simplified further.

For disjoint cliques, the dependency graph $\mathrm{DG}'(V_s)$ is a subgraph of $\mathrm{DG}(V)$, for any node subset V_s spanning a sub-collection of cliques $\mathcal{C}_s \subset \mathcal{C}$, and hence,

$$\mathrm{LLR}(\mathbf{Y}_{V_s}) = \sum_{c \in \mathcal{C}_s} \phi_c(\mathbf{Y}_c). \qquad (21)$$

Hence, it is simpler to design fusion schemes in this case since the dependency structure does not change for different nodes subsets, as long as the nodes span a sub-collection of cliques.

For disjoint cliques, the penalty function for each clique in (20) simplifies to the KLd of measurements in clique $c \in \mathcal{C}$

$$\pi^{clq}(c) = D(f_c(\mathbf{Y}_c; \mathcal{H}_0) || f_c(\mathbf{Y}_c; \mathcal{H}_1)):=D_c. \qquad (22)$$

Hence, if nodes in a clique c is not selected, then a penalty equal to its KLd D_c is paid.

We now prove the asymptotic optimality of using the exponent-based penalty function π^{clq} in (22), instead of the original penalty function π in (4) in (14).

(a) Augmented graph using Map   (b) PCST on augmented graph   (c) Scheme to compute LLR via RevMap
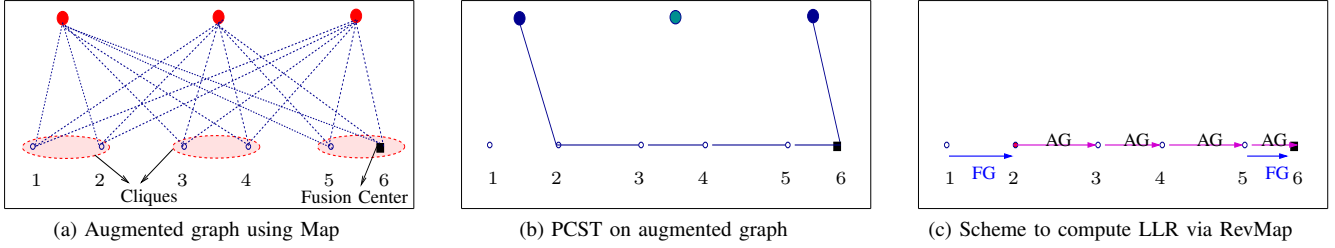
Fig. 3. Illustration of Clique Selection and Data Fusion via PCST Reduction for Binary Cliques.

Theorem 3 (Asymptotic Optimality): When the number of cliques grows with network size ($|\mathcal{C}| \to \infty$, as $|V| \to \infty$), and the link costs are bounded, we have

$$\lim_{|V| \to \infty} \frac{\text{opt\_clique}(V, C, \gamma\pi)}{\text{opt\_clique}(V, C, \gamma\pi^{clq})} = 1, \quad \forall \gamma > 0. \qquad (23)$$

Proof: Along the lines of Theorem 1. See [13]. □

Hence, using the penalty function π^{clq} in (22) instead of π is suitable for networks with large number of cliques. An example where this does *not* occur is when the dependency graph is complete, and has a single clique. We therefor need a sparse dependency graph to guarantee the asymptotic convergence of the constrained PCDF in (14) to the optimal solution under penalty π^{clq}. Along the lines of our approach for the i.i.d. case, we now prove that under π^{clq}, the optimal solution reduces to a PCST.

Theorem 4 (PCST Reduction): opt_clique($V, C, \gamma\pi^{clq}$) has an approximation-ratio preserving PCST reduction.

Proof: By simplifying an integer program [13]. □

The above result implies that any approximation algorithm for the PCST can be transformed to an approximation for opt_clique($V, C, \gamma\pi^{clq}$), with its approximation ratio preserved. One such instance, called the approximate prize-collecting data fusion (Approx_PCDF), is given in Fig.4. It builds an approximate PCST on an augmented graph using the GW-algorithm [3].

The augmented graph is given by the function Map in Fig.5, where for each non-trivial clique c (size greater than one) of the dependency graph, it adds a virtual node v_c and connects it to the nodes $v \in V$. The costs of new edges reflect the cost of forwarding raw data to candidate processors to compute the clique potentials in the first stage of LLR computation, which is not needed for i.i.d. data. Hence, the routing costs are increased in the presence of correlation due to additional complexity of the LLR.

The penalty of each virtual node v_c is $\pi^{clq}(c)$ in (22) and the penalties of all nodes $v \in V$ are set to zero. After building the approximate PCST on the augmented graph, the function RevMap in Fig.6 maps it to a valid output, viz., the set of selected cliques and the fusion scheme to compute its LLR. An example of the PCST reduction is shown in Fig.3.

As in the i.i.d. case, an approximate PCST is built on the augmented graph using the GW-algorithm [3]. Since the

augmented graph has $|V| + |\mathcal{C}^{nt}|$ number of nodes, where $\mathcal{C}^{nt}$ is the set of non-trivial cliques, the approximation ratio of Approx_PCDF(Map) with respect to opt_clique($V, C, \gamma\pi^{clq}$) is $2 - (|V| + |\mathcal{C}^{nt}| - 1)^{-1}$.

We now improve its approximation ratio based on some simple observations regarding the GW-algorithm. Define the collection of *profitable cliques* $\mathcal{C}_p \subset \mathcal{C}$ as those generating a net "profit" after reducing their scaled KLd by the costs of raw-data routing to any processor

$$\mathcal{C}_p := \{c : c \in \mathcal{C}, |c| = 1 \text{ or } |c| > 1 \text{ and}$$
$$\gamma D_c \geq \min_{i \in V} \sum_{v_k \subset c_j, k \neq i} C(v_i, v_k)\}, \qquad (24)$$

and let Map$'$ be the modified version of Map which only adds virtual nodes for non-trivial profitable cliques, i.e., $c \in \mathcal{C}_p, |c| > 1$, instead of adding for all non-trivial cliques, $c \in \mathcal{C}^{nt}$, as done by Map. Below, we give the improved approximation ratio.

Theorem 5 (Improved Approx. Ratio): On using the Map$'$ function, the approximation ratio for Approx_PCDF with respect to opt_clique($V, C, \gamma\pi^{clq}$) is

$$\rho(\text{Approx\_PCDF(Map}')) = 2 - \frac{1}{\max(|\mathcal{C}_p| - I(v_0 \in \mathcal{C}_p), 1)}.$$

Proof: Only profitable cliques can be selected in the optimal solution. See [13]. □

Hence, the approximation ratio for Approx_PCDF(Map$'$) depends only on the number of profitable cliques $|\mathcal{C}_p|$, which may be substantially smaller than the size of the augmented graph $|V| + |\mathcal{C}^{nt}|$ leading to improved approximation guarantees. In fact, when there are no profitable cliques ($\mathcal{C}_p = \emptyset$), the algorithm outputs the optimal solution ($\rho = 1$) of not selecting any of the nodes.

V. NODE SELECTION HEURISTICS

The results in the previous section inspire the development of two heuristics for a general dependency graph, viz., clique selection and component selection. The Approx_PCDF algorithm in the previous section, based on the PCST reduction, can be generalized as follows: form groups of nodes according to some criterion as candidates for selection, and define a penalty function for not selecting each group. Apply the PCST reduction as before by augmenting the graph with virtual nodes

Require: $V = \{v_0, \ldots, v_{|V|-1}\}$ nodes, v_0= Fusion center,
$\quad \mathcal{M} = \{c_0, \ldots, c_{|\mathcal{M}|-1}\}$= Candidate node groups
$\quad G_n$= Metric closure of network, C = Link costs
$\quad \Pi_m$ = Penalty of group m, γ = tradeoff factor
$\quad \{G', V_m, \pi\} \leftarrow \text{Map}(G_n; \mathcal{M}, C, \Pi, \gamma)$
$\quad \text{PCST}(G; C, \pi)$ = (Approx.) Prize-collecting Steiner tree on
$\quad G$ using GW algorithm with cost C, node penalty fn. π
$\quad \text{DPCST} = \text{PCST}(G')$ directed towards v_0
$\quad \{\mathcal{M}_s, \Gamma\} \leftarrow \text{RevMap}(\text{DPCST}; V_m, V, \mathcal{M}, \text{Algo})$
$\quad$**return** $\{\mathcal{M}_s, \Gamma\}$

Fig. 4. Approx_PCDF(Map,Algo): outputs selected groups $\mathcal{M}_s$ and fusion scheme Γ. For Algo=Clique Selection, $\mathcal{M} = \mathcal{C}$ is the clique set of DG(V) and $\Pi = \pi^{clq}$ in (20). For Algo=Component Selection, $\mathcal{M}$ is the set of components of DG(V) and $\Pi = \pi\_cmp$ in (25).

```
1:  function MAP(G_n(V); M, C, Π, γ)
2:      N_u(v; G) = Neighborhood of v in undirected G
3:      G' ← G_n, V_m ← ∅, n ← |V|, π(v) ← 0, ∀v ∈ V
4:      for j ← 0 to |M| − 1 do ▷ Let V and M be ordered
5:          if |m_j| > 1 then
6:              V_m ← v_{n−1+j}
7:              Add new node v_{n−1+j} to G'
8:              Assign penalty γπ(v_{n−1+j}) ← γΠ_{m_j},
9:              for all v_i ∈ V do
10:                 Add node v_i to N_u(v_{n−1+j}; G')
11:                 C(v_{n−1+j}, v_i; G') ← Σ_{v_k ⊂ c_j, k≠i} C(v_i, v_k; G_n)
12:             end for
13:         else
14:             V_m ← v_i, π(v_i) ← γΠ_{m_j}, v_i ⊂ m_j ▷ 1-groups
15:         end if
16:     end for
17:     return {G', V_m, π}
18: end function
```

Fig. 5. $\text{Map}(G_n; \mathcal{M}, C, \Pi, \gamma)$ adds nodes for each non-trivial group, and returns augmented graph G' with penalty π and group-representative set V_c.

```
1:  function REVMAP(G'; V_c, V, M, Algo)
2:      N_s(v; G), N_p(v; G) = Imm. successor, predecessor
3:      < i, j >= Directed edge from i to j
4:      Initialize G ← G', n ← |V|, M_s ← ∅
5:      for all v_j ∈ V_c with N_s(v_j; G') ≠ ∅ do
6:          if j > n − 1 then
7:              k ← j − n + 1, M_s ← M_s ∪ m_k
8:              Proc(m_k) ← N_s(v_j; G'), for m_k ∈ M,
9:              V_j ← c_k \ Proc(m_k), Delete < v_j, Proc(m_k) >
    in G, add < V_j, Proc(m_k) >, mark them
10:             if  N_p(v_j; G)  ≠  ∅  then Replace
    < N_p(v_j), v_j > in G with edges < N_p(v_j), Proc(m_k) >
11:             end if
12:         else
13:             Proc(m_l) ← v_j, for v_j ⊂ m_l, M_s ← M_s ∪ m_l
14:         end if
15:     end for
16:     FG ← Marked edges of G, AG ← G \ FG
17:     Retain only one edge in FG if there are parallel links
18:     Let V(Proc) be set of all processors
19:     Let V_s ← nodes in V spanning the groups M_s
20:     if Algo=Clique Selection then
21:         Let C' be clique set of DG'(V_s)
22:         for all c ∈ C' \ M_s do
23:             Proc(c) ← arg min    Σ      C(i, j)
                          i∈V(Proc) j:j⊂c
                                    <j,i>∉FG
24:             Add < j, Proc(c) >, j ⊂ c \ Proc(c) to FG if
    not already present
25:         end for
26:         M_s ← C'
27:     end if
28:     Γ ← {Proc, FG, AG}
29:     return {M', Γ}
30: end function
```

Fig. 6. $\text{RevMap}(G; V_c, V, \mathcal{M})$ returns the selected groups $\mathcal{M}_s$ and maps tree G' to fusion scheme Γ with processor assignment Proc, forwarding subgraph FG and aggregation subgraph AG.

for each group. Using the RevMap, the output is a selected sub-collection of groups and a fusion scheme which computes a sum function over the selected groups.

The desired output for cost-performance tradeoff is however *not* a fusion scheme for computing a sum function, but for computing the marginal LLR of the selected nodes. As we discussed in Section IV-B, the LLR structure (dependency graph) changes with the selected node set in general. We now overcome this hurdle by grouping nodes in such a manner that the LLR of any selected sub-collection of groups is indeed a sum function over those groups.

For general dependency graphs, such groups are given by the components of the dependency graph, i.e., if all or none of the nodes belonging to each component of the graph are selected, then the LLR of the selected subset is a simple sum function over the selected components

$$\text{LLR}(\mathbf{Y}_{V_s}) = \sum_{v \subset m, m \in \mathcal{M}} \text{LLR}(\mathbf{Y}_m),$$

where $m \in \mathcal{M}$ is a component in the dependency graph. Moreover, we can define penalty for each component by collecting the terms of the error exponent in (19) consisting of all the cliques contained in it, given by

$$\pi\_cmp(m) := \sum_{c \subset m, c \in \mathcal{C}} \mathbb{E}[\phi_c(\mathbf{Y}_c); \mathcal{H}_0] = D_m, \quad (25)$$

where D_m is the KLd of the component m, and the penalties of different components are additive. We term such a policy considering different components of the dependency graph as candidates for selection as the *component selection* heuristic.

Optimal cost-performance tradeoff is however not guaranteed for the component selection heuristic since we may be severely limiting our choices of node subsets for selection. For instance, if the graph has a single component, then the heuristic reduces to a binary decision of selecting all or none

of the nodes. We now propose another heuristic which may perform better in such instances.

As in the previous section, we consider the cliques of the dependency graph as the groups, i.e., candidates for selection, and the penalty function for each clique in (20). This is referred to as the clique selection heuristic. However, as noted, the output fusion scheme is not guaranteed to compute the marginal LLR of the selected node set which is a requirement for inference. In Fig.6, we add additional lines from (17) to (26) to ensure that the marginal LLR is indeed computed. For each new clique in the marginal dependency graph, not present in the dependency graph over all the nodes $DG(V)$, we ensure that its clique potential is computed by adding edges from its members to a processor to the forwarding subgraph (FG) of the fusion scheme. However, since new edges are added, routing costs increase, and we can no longer provide optimality results for the clique selection heuristic for a general MRF, as we did in the previous section.

The component and clique selection policies represent group selection of nodes with aggregation for efficient cost-performance tradeoff. The component selection heuristic can be viewed as coarse selection or rejection of nodes as a full component, while the clique selection heuristic is more fine-grained, depending on the graph. For graphs having very few components, and yet, a large number of cliques, we expect the clique selection policy to have better cost-performance tradeoff than component selection, since there are more candidates for selection. On the other hand, for sparse graphs with large number of components, we expect the component selection policy to do better, and this is validated by our simulations.

VI. Numerical Analysis

A. Simulation Environment

We assume that the sensor measurements are Gaussian under either hypothesis with the same covariance matrix $\mathbf{Y}_V \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_V)$, under $\mathcal{H}_i$, $i = 0, 1$. This scenario arises when the sensors measure a deterministic signal with additive (correlated) Gaussian noise under each hypothesis. The KLd D and the type-II error probability P_M have closed forms for Gaussian variables [2], [12].

In our setup, n (expected) number of nodes are distributed in a square. We consider two node placement distributions: uniform and Matern cluster process[6] [14]. The routing cost between any two nodes i and j for direct transmission is given by the power-weighted distance $|i, j|^\nu$. We present the result when the set of feasible direct connections is the complete graph and the path-loss $\nu = 2$: similar trends were observed for any connected graph and $\nu \in [2, 4]$.

B. Results: IID Measurements

We first consider the case when all the measurements are i.i.d. conditioned on each hypothesis with unit variance ($\boldsymbol{\Sigma}_V = \mathbf{I}$). We compare the performance of our fusion scheme

[6]Here, a parent Poisson process first generates points. A child Poisson process then generates nodes in a disc around each point of the parent process.

Approx_PCDF in Fig. 4 with the following simple schemes: choosing all the nodes and conducting fusion along the MST, choosing none of the nodes (paying penalty for all the nodes), and additionally, optimal selection with no aggregation, i.e., routing all the selected data to the fusion center via the shortest path routes (SPR). It is given by the set of "profitable" nodes

$$V_*^{\text{SPR}}(V, C, \gamma\pi^{iid}) = \{i : i \in V, \gamma D > C(i, v_0)\}, \qquad (26)$$

where C is the cost of shortest path. In Fig.7a, we find that the tradeoff function obj in (5) for Approx_PCDF is significantly better than those for the other schemes. Hence, incorporating fusion into cost-performance tradeoff significantly reduces the costs and achieves better tradeoff.

Fig.7b shows that more nodes are selected by Approx_PCDF as the tradeoff factor γ increases, since the penalty is given by $\gamma\pi$. In Fig.7c, we plot the average (per-node) routing cost for aggregation of selected measurements versus the resulting error probability for Approx_PCDF under different γ. We see that the exponent-based approximation e^{-nD} is close to the actual error probability P_M.

C. Results: Correlated Measurements

We employ the GMRF model in [15], where the dependency graph $DG(V)$ is a disk graph with radius δ and the coefficients of the potential matrix $\mathbf{A}_V := \boldsymbol{\Sigma}_V^{-1}$ are given in [13]. For Gaussian measurements, the maximum clique size is two and higher order clique potentials are zero [7]. Hence, the clique selection heuristic in Fig.4 reduces to selection of the dependency edges, and is called the *edge selection policy*.

In Fig.8a, we compare the component and edge selection heuristics under uniform placement. We fix the disk radius $\delta = 1.2$ and here, the disk graph is connected (single component). We expect the edge selection heuristic to perform better since it has more choices here when compared to component selection, which has to make a binary choice whether to select all or none of the nodes. We find that for γ shown in the figure, this indeed is the case; the edge selection heuristic performs better and selects some nodes, while the component selection heuristic selects none of the nodes thereby incurring high penalty in terms of error probability.

In Fig.8b and Fig.8c, we study the influence of node placement on our heuristics, and consider uniform and Matern cluster process with component selection heuristic. We observe that at low values of δ, the clustered process is more efficient; here, more nodes are chosen, and the tradeoff function obj is lower. However, as δ increases, the two processes have nearly the same performance. As in the i.i.d. case, the exponent-based penalty π_cmp is close to π, based on the error probability in all the instances.

We can provide an intuitive explanation for the above behavior. At low dependency (small values of the disk radius δ), clustering the nodes is more efficient than uniform placement since it leads to significantly smaller number of components, thereby providing more choices to the component selection

(a) Objective Value obj vs. Tradeoff Factor (b) % of Nodes Not Selected vs. Tradeoff Factor (c) Cost vs. Performance for Selected Set V_s
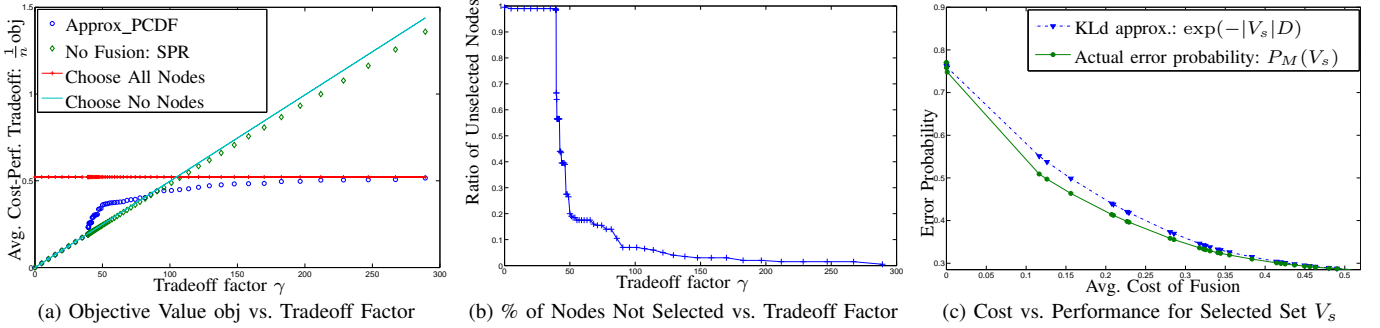
Fig. 7.   Cost-performance tradeoff for i.i.d. measurements under uniform placement for $n = 200$ nodes. See Theorem 2. For objective function obj, see (5).

(a) Component/Edge Selection $\delta = 1.2, n = 50$ (b) Component Selection, $\gamma = 140, n = 200$ (c) Component Selection, $\gamma = 140, n = 200$
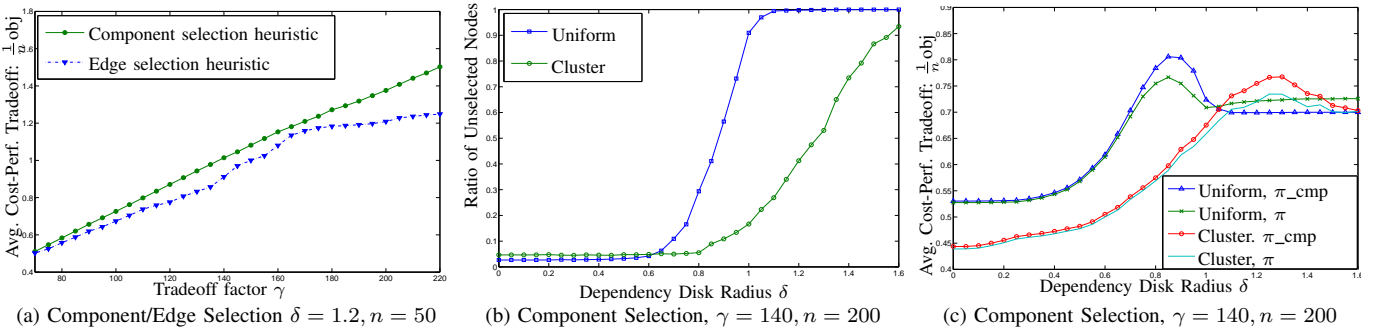
Fig. 8.   Cost-performance tradeoff for correlated Gaussian measurements under 60 simulation runs. See (4), (5), and (25).

heuristic. Moreover, the routing costs within the components are also significantly reduced upon clustering since nodes are nearer, and hence, more nodes are selected leading to improved tradeoff. However, as δ increases, there are fewer and larger components, leading to increased routing costs and fewer choices for selection. Hence, the cluster process is a good node-placement strategy for achieving efficient cost-performance tradeoff at sparse spatial dependencies, and our heuristic has good performance in this regime.

VII. CONCLUSION

In this paper, we considered optimal node selection for tradeoff between routing costs and inference performance. We explicitly incorporated the effect of correlation between the sensor measurements via the dependency graph of a Markov random field model and considered in-network aggregation of measurements to reduce routing costs. We provided theoretical and numerical results to show the efficiency of our schemes for node selection and data aggregation.

There are many future directions to pursue such as the development of better algorithms. We have only considered offline and centralized sensor selection and extension to local selection and coordination is of interest. The effect of quantization and scheduling warrants investigation.

REFERENCES

[1] A. Anandkumar, L. Tong, A. Swami, and A. Ephremides, "Cost-Performance Tradeoff in Multi-hop Aggregation for Statistical Inference," in *Proc. of IEEE ISIT*, Toronto, Canada, July 2008.

[2] H. V. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, 1994.

[3] M. Goemans and D. Williamson, "A General Approximation Technique for Constrained Forest Problems," *SIAM J. on Computing*, vol. 24, p. 296, 1995.

[4] P. Brémaud, *Markov Chains: Gibbs fields, Monte Carlo simulation, and queues*. Springer, 1999.

[5] Y. Sung, S. Misra, L. Tong, and A. Ephremides, "Cooperative Routing for Signal Detection in Large Sensor Networks," *IEEE J. Select. Area Comm.*, vol. 25, no. 2, pp. 471–483, 2007.

[6] A. Giridhar and P. Kumar, "Computing and communicating functions over sensor networks," *IEEE J. Selec. Area Comm.*, vol. 23, no. 4, pp. 755–764, 2005.

[7] A. Anandkumar, A. Ephremides, A. Swami, and L. Tong, "Routing for statistical inference in sensor networks," in *Handbook on Array Processing and Sensor Networks*, S. Haykin and R. Liu, Eds.   John Wiley & Sons, 2009, ch. 25.

[8] A. Anandkumar, L. Tong, A. Swami, and A. Ephremides, "Minimum Cost Data Aggregation with Localized Processing for Statistical Inference," in *Proc. of INFOCOM*, Phoenix, USA, April 2008, pp. 780–788.

[9] A. Anandkumar, L. Tong, and A. Swami, "Optimal Node Density for Detection in Energy Constrained Random Networks," *IEEE Trans. Signal Proc.*, vol. 56, no. 10, pp. 5232–5245, Oct. 2008.

[10] H. Rowaihy, S. Eswaran, M. Johnson, D. Verma, A. Bar-Noy, T. Brown, and T. La Porta, "A survey of sensor selection schemes in wireless sensor networks," in *Proc. of SPIE*, 2007.

[11] Y. Nakamura, K. Tei, Y. Fukazawa, and S. Honiden, "Region-Based Sensor Selection for Wireless Sensor Networks," 2008, pp. 326–331.

[12] T. Cover and J. Thomas, *Elements of Information Theory*.   John Wiley & Sons, Inc., 1991.

[13] A. Anandkumar, M. Wang, L. Tong, and A. Swami, "Prize-Collecting Data Fusion for Cost-Performance Tradeoff in Distributed Inference," Cornell University, Tech. Rep., Jan. 2009. [Online]. Available: http://acsp.ece.cornell.edu/papers/Anandkumar09TR.pdf

[14] A. Lawson and D. Denison, *Spatial Cluster Modelling*.   CRC Press, 2002.

[15] A. Pettitt, I. Weir, and A. Hart, "A Conditional Autoregressive Gaussian Process for Irregularly Spaced Multivariate Data with Application to Modelling Large Sets of Binary Data," *Statistics and Computing*, vol. 12, no. 4, pp. 353–367, 2002.