Distributed Threshold Selection for Aggregate Threshold Monitoring in Sensor Networks

Mohammad S. Talebi
School of Computer Science, IPM, Iran
mstalebi@ipm.ir

Ahmad Khonsari
School of Computer Science, IPM, Iran
ECE Department, University of Tehran
ak@ipm.ir

Ali Abbasi
ECE Department, University of Tehran
a.abbasi@ece.ut.ac.ir

***Abstract*—Motivated by applications like sensor, peer to peer, ad hoc networks there has been growing interest in monitoring large scale distributed systems. In these applications typically we wish to monitor a global system condition which is defined as a function of local network elements parameters. In this paper, we study *Aggregate Threshold Queries* in sensor networks, which are used to detect when an aggregate value of all sensor measurements crosses a predetermined threshold. The major constraint in designing monitoring applications is reducing the amount of communication burden which is the dominant factor of energy drain in wireless sensor networks. In this study, we address the aggregate threshold monitoring problem by proposing a distributed algorithm to set local thresholds on each sensor node so that only those sensors whose measurements crosses their local thresholds commence communication. We adopt the FPTAS optimization formulation of the problem [1] and propose a distributed algorithm as the solution to the problem. Simulation results demonstrate the validity of the proposed distributed algorithm in attaining very close performance as the centralized scheme.**

***Index Terms*—Aggregate Threshold Monitoring, Sensor Networks, Optimization, Iterative Methods.**

I. Introduction

In recent years, with the abundance of emerging large scale distributed systems such as sensor networks, and peer to peer networks, monitoring scenarios is progressively more considered vital to track these systems. In monitoring either we are interested in supervising the network itself (e.g. traffic engineering, routing optimization, anomaly detection in data networks), or environment that network deployed in it (e.g. wildlife behavior, moving objects in sensor networks). Network monitoring includes measuring system parameters to react to different network conditions. There are two ways of getting knowledge from the network; the first is to send requests into the network to poll all relevant information. Another option is that all network elements push all possibly important readings to the management station. The nature of network monitoring applications imposes specific requirements on design of monitoring algorithm: 1- The algorithm should provide real-time detection of noteworthy events. 2- The algorithm must be scalable to large number of nodes. 3- The detection process should incur minimum communication in the network.

Thereby, a challenge in monitoring applications is to devise plans to reduce communication while fulfilling application requirements. A common method is to install local constraints or filters at remote sources to filter out unnecessary updates. Local constraints should have the property such that preserving all of them ensures that there was not any anomalous event. Each new measurement is compared to the filter and in case it violates (exits) the corresponding range of the filter an update is sent to the base station. Apparently efficient decomposition of global system constraint into local constraint has a great impact on reducing communication. Filter setting is dependent on a number of parameters including the associated cost with sending an update from each node to the management station and the changing patterns of measurements.

Often the monitoring task is the need to detect when a function of individual network element readings crosses a given threshold. Typically, monitored functions are aggregate functions like SUM, or AVG which gives global insights concerning the state of the network. Consider the following queries:

- Report when the number of enemy troops detected in a region of the network crosses 20.
- Report when variance of temperature in a building crosses 3.

Aggregate functions like SUM, AVG are linear functions of individual sensor readings, thus the global property of $\sum_i A_i T_i \leq T$ can be decomposed to a set of $x_i \leq T_i$ constraints which checked locally at each node. In this work, we concentrate on the problem of determining optimal values for T_i's. Previous work [1] assumed that the monitoring station computes the optimal thresholds based on individual nodes probability distribution functions and assigns thresholds for nodes. Thus, each node should update its histogram constructed over recent measurements either periodically or based on a change detection algorithm to the base station. But there is limitation on the applicability of such an approach for sensor networks. The structure of such networks has necessitated the design of asynchronous, distributed and fault-tolerant computation and information exchange algorithms [4]. In this work, we propose a distributed algorithm for optimal threshold assignment so as to minimize the probability of global system polling.

The remainder of the paper is organized as follows: Section II reviews related works. Section III defines the system model

and problem definition. Section IV provides the optimal solution to the threshold assignment problem. Section V presents a distributed threshold assignment algorithm based on the problem's optimal solution. Section VI validates experimental evaluation of the proposed method. Finally, Section VII concludes the study.

II. Related Works

In recent years, continuous query processing for monitoring distributed data streams has attracted much research. Monitoring aggregate threshold queries in networks initially mentioned by the pioneering work of Raz *et al* [8]. They introduced installing local mathematical constraints at remote sites and present a simple approach for threshold assignment assuming uniform data distributions of system variables.

Keralpera *et al* [12] presented several algorithms for static and adaptive threshold setting for monitoring thresholded counts queries and analyzed the communication complexity of each algorithm. Sharfman *et al* [16] introduced a geometric approach for monitoring arbitrary threshold functions. Recent work of Kashyap *et al* [10] considered the problem of non-zero slack threshold assignment which adaptively dedicates the fraction of total threshold to monitoring node to absorb small local threshold crossing that eliminates the need for global system polling.

III. System Model and Problem Formulation

We model the topology of the sensor network by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The vertex set $\mathcal{V} = \{1, 2, \ldots, n\}$ and edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ denotes the set of nodes and links, respectively. For the edge set, we have $(i, j) \in \mathcal{E}$ if and only if there is a connection between node i and j. We also refer to the neighbors of node i as $\mathcal{N}_i = \{j \in \mathcal{V} | (i, j) \in \mathcal{E}\}$.

We assume that there is a base station being responsible for monitoring the network. The base station disseminates queries to sensors and responds to user queries. Each sensor node i continuously measures (reads) the requested local phenomenon x_i (for example, humidity, light) which is in the range $[0, M_i]$ at a fixed sampling rate. The goal is to detect when the aggregate value of all the measurements $\sum_{i=1}^{n} A_i x_i$ crosses a predetermined threshold T.

To compute the aggregate we can perform value monitoring via global polling or algorithm like [14]. According to [5] value monitoring directly solves threshold monitoring, and running $O(\frac{1}{\epsilon} \log T)$ instances of a threshold monitoring algorithm for thresholds $1, 1+\epsilon, (1+\epsilon)^2, \ldots, T$ solves value monitoring with relative error ϵ. So, the two variants differ by at most a factor of $O(\frac{1}{\epsilon} \log T)$. Thus determining whether an aggregate has crossed a threshold does not require knowing the exact value of the aggregate.

A common method to reduce communication is to install local constraints or filters (in the form of an interval $[0, T_i]$) at remote sources. Local constraints setting should maintain the property that preserving all of them, ensures that the aggregate value has not exceeded the threshold (covering property). Each sensor i after measuring the new value of x_i, checks condition $L_i \equiv (x_i \in [0, T_i])$, and the condition upon being violated, an update is sent to the base station to initiate global aggregate computation. Using this method, vast amount of updates are filtered out at the source and are not transmitted to the base station. The efficiency of this approach is largely dependent on filter setting method. But what values should we choose for T_i's? Selection of values must satisfy the global property $\sum_{i=1}^{n} A_i T_i \leq T$ to conform covering property. However, this equation leads to great flexibility in choosing T_i's. In this respect, we face to another question: what is the best selection?

Our goal is to minimize communication cost, towards which we should minimize the probability of global polling. Every local filter violation leads to global computation, thus we should maximize the probability of preserving all local constraints $\max F(x_1 \leq T_1, \ldots, x_n \leq T_n)$, where F is the joint cumulative frequency distribution over all the sensors. Computing the multi-dimensional histograms is somewhat cumbersome in terms of communication burden, and hence we consider more communication convenient assumption of independence between sensor's data distributions. With this simplifying assumption, we have

$$\Pr(\mathcal{L} = \text{true}) = \prod_{i=1}^{n} \frac{F_i(T_i)}{F_i(M_i)} \tag{1}$$

Since $F_i(M_i)$s are constant, the optimization problem can be formulated as

$$\max_{\mathbf{T}} \prod_{i=1}^{n} F_i(T_i) \tag{2}$$

subject to:

$$\sum_{i=1}^{n} A_i T_i \leq T \tag{3}$$

where $\mathbf{T} = (T_i, i = 1..n)$ is the vector representation of local thresholds. According to [1] local threshold selection problem is *NP-hard*. They introduced a centralized scheme, referred to as FPTAS, to solve the problem within ϵ relative error for arbitrarily small ϵ. The solution was based on the assumption that all the values of $F_i(T_i)$ are integral powers of a constant α, which is slightly greater than 1. Thus, each $F_i(T_i)$ will correspond to some α^{r_i} and maximizing $\prod_{i=1}^{n} F_i(T_i)$ will be equivalent to maximizing $\alpha^{\sum r_i}$. Let $T_i(r_i)$ be the local threshold of node i such that

$$\alpha^{r_i} \leq F_i(T_i(r_i)) < \alpha^{r_i+1} \tag{4}$$

Then, the optimization problem becomes

$$\max_{\mathbf{r}} \sum_{i=1}^{n} r_i \tag{5}$$

subject to:

$$\sum_{i=1}^{n} A_i T_i(r_i) \leq T \tag{6}$$

where $\mathbf{r} = (r_i, i = 1..n)$ is the vector representation of powers. In order for (5) to admit a unique maximizer, we argue that F_i must be an increasing and strictly concave function. Indeed,

we assume that optimization variable, $r_i, i = 1..n$, belongs to a domain in which F_i satisfies these assumptions.

IV. Optimal Solution

In this section, we solve problem (5). Problem (5) is a constrained problem, whose constraint (6) is coupled across the network. Such a constrained optimization problem can be efficiently solved using Interior Point Method [4], which necessitates the coordination among possibly all nodes of the networks, which is undesirable or infeasible. However, in the context of wireless ad-hoc and sensor networks, we are interested in distributive algorithms to solve (5).

Towards this end, we aim at solving problem (5) through its dual. In this respect, problem (5) is called primal problem. In the sequel, we proceed to derive the dual problem of (5) and then present a distributively iterative algorithm as the solution to the dual problem.

A. Deriving Dual Problem

We start by writing the Lagrangian of problem (5), as follows

$$L(\mathbf{r}, \mu) = \sum_{i=1}^{n} r_i - \mu \left(\sum_{i=1}^{n} A_i T_i(r_i) - T \right) \tag{7}$$

where $\mu > 0$ is the Lagrange multiplier associated with constraint (6). Using Karush-Kuhn-Tucker (KKT) conditions for convex optimization, to find optimal primal variables, i.e. optimal powers $\mathbf{r}^*$, we should find the stationary points of the Lagrangian and satisfy complementary slackness conditions. The complementary slackness conditions for optimal primal variable $\mathbf{r}^*$ and dual variable μ^*, are

$$\mu^* \geq 0; \tag{8}$$

$$\sum_{i=1}^{n} A_i T_i(r_i^*) \leq T; \tag{9}$$

$$\mu^* \left(\sum_{i=1}^{n} A_i T_i(r_i^*) - T \right) = 0 \tag{10}$$

In order to find the stationary points of the Lagrangian, we solve

$$\nabla L_{\mathbf{r}}(\mathbf{r}^*, \mu^*) = \mathbf{0} \tag{11}$$

where $\mathbf{0}$ is a vector with all zero. For the ith element of (11) we have

$$\frac{\partial L}{\partial r_i} = 1 - \mu A_i \frac{dT_i(r_i)}{dr_i} \tag{12}$$

From (4), recall that $T_i(r_i)$ is selected so that (4) is satisfied. Also, recall that α is a constant slightly greater than 1. Therefore, the lower and upper bounds in (4) are sufficiently close to each other, leading us to approximate $F_i(T_i(r_i))$, using the concept of geometric mean as following:

$$\begin{aligned} F_i(T_i(r_i)) &\approx \sqrt{\alpha^{r_i} \alpha^{r_i+1}} \\ &\approx \sqrt{\alpha} \alpha^{r_i} \approx \alpha^{r_i} \end{aligned} \tag{13}$$

Recalling the monotonicity assumption of CDFs, each CDF admits a unique inverse, which yields the explicit expression for T_i as

$$T_i(r_i) = F_i^{-1}(\alpha^{r_i}) \tag{14}$$

Substituting (14) in (12), yields

$$\begin{aligned} \frac{\partial L}{\partial r_i} &= 1 - \mu A_i \frac{dF_i^{-1}(\alpha^{r_i})}{dr_i} && (15) \\ &= 1 - \mu A_i \frac{d\alpha^{r_i}}{dr_i} \frac{dF_i^{-1}}{dr_i}\bigg|_{\alpha^{r_i}} \\ &= 1 - \mu A_i \alpha^{r_i} \ln \alpha \frac{dF_i^{-1}}{dr_i}\bigg|_{\alpha^{r_i}} && (16) \end{aligned}$$

Setting (16) to zero and doing some algebraic manipulations, gives an explicit expression for the optimal power $\mathbf{r}^*$ in terms of optimal Lagrange multiplier μ^*. For the sake of presentation, we define

$$G_i(z) = \frac{dF_i^{-1}}{dr_i}\bigg|_z \tag{17}$$

Substituting $G_i(.)$ in (16), we come up to the following implicit equation to obtain $\mathbf{r}^*$

$$\alpha^{r_i^*} G_i(\alpha^{r_i^*}) = \frac{1}{\mu^* A_i \ln \alpha} \tag{18}$$

In order to solve problem (5) through its dual, we need to obtain the Lagrange dual function, or simply dual function. The Lagrange dual function $D(\mu)$ is defined as the maximum of the Lagrangian $L(\mathbf{r}, \mu)$ over the primal variable $\mathbf{r}$, for a given μ. Thus, $D(\mu)$ can be expressed as

$$D(\mu) = \max_{\mathbf{r}} L(\mathbf{r}, \mu) \tag{19}$$

Based on the KKT condition and using (18), for $D(\mu)$ we get

$$D(\mu) = L(\mathbf{r}^*, \mu) \tag{20}$$

The dual problem is formulated as

$$\min_{\mu \geq 0} D(\mu) \equiv \min_{\mu \geq 0} L(\mathbf{r}^*, \mu) \tag{21}$$

Dual problem defined above can be solved using iterative methods. In order to obtain a distributed algorithm, we solve the dual problem (21) using Gradient Projection algorithm. We postpone solving (21) to the next subsection.

B. Solving The Dual

In this subsection, we solve the dual problem using Gradient Projection Algorithm. To solve the dual problem, Gradient Projection Algorithm adjusts μ in opposite direction to the Gradient of dual function, i.e. $\nabla D(\mu)$. Precisely speaking, in the kth iteration step, $\mu^{(k)}$ is updated as follows

$$\mu^{(k+1)} = \left[\mu^{(k)} - \gamma \frac{dD(\mu^{(k)})}{d\mu} \right]^+ \tag{22}$$

where $[z]^+ = \max\{z, 0\}$ and γ is a sufficiently small constant step size. Using the Danskin's Theorem [3], the derivative of $D(\mu)$ is given by

$$\frac{dD(\mu)}{d\mu} = T - \sum_{i=1}^{n} A_i T_i(r_i) \tag{23}$$

Substituting (23) in (22), yields

$$\mu^{(k+1)} = \left[\mu^{(k)} + \gamma \left(\sum_{i=1}^{n} A_i T_i(r_i^{(k)}) - T\right)\right]^+ \tag{24}$$

where $r_i^{(k)}$ is the solution to (18) for a given $\mu^{(k)}$. In this equation, γ is chosen sufficiently small so as to guarantee the convergence.

In the economics literature, Lagrange multiplier or dual variable, μ is called *shadow price* [11] and accordingly, (24) is called *shadow price update*. This stems from the interpretation of its role in solving the primal problem via its dual. From (18) it's apparent that $\mathbf{r}^*$ is a decreasing function of μ; therefore μ can be construed as the price which must be paid by node i to achieve the threshold $T_i(r_i)$. As the nature of such a price is hidden to nodes from the primal problem perspective, it is called *shadow price*.

(18) and (24) form an iterative solution to problem (21) and thereby problem (5). At each iteration step k, dual variable μ will be updated based on the history of itself and the primal variables $\mathbf{r}$. Then, it would be utilized by each node to update primal variable $\mathbf{r}$, accordingly. Therefore, after spending enough iteration steps, primal and dual variables tends to primal-optimal $\mathbf{r}^*$ and dual-optimal μ^*, respectively.

Based on the above iterative solution, we propose a distributed algorithm as a distributed solution to the threshold selection problem. We deffer the algorithm until Section V.

V. Algorithm

In this section, we propose a distributed algorithm based on the iterative solution obtained in Section IV.

Considering (24) and (18), it is clear that the iterative solution to problem (5) can be exploited to design a distributed algorithm.

We elaborate to employ a heuristic mechanism, proposed by Jadbabaei *et al* [9] to come up with a fully distributed algorithm which only requires local information of the network.

Since each sensor i is supposed to update μ locally, the realized update evolution will differ between sensor nodes. To avoid confusion between the updated value at each node, we introduce $\hat{\mu}_i^{(k)}$ notation, by which we mean the updated value of the shadow price at node i in iteration step k.

The algorithm evolves as following: At each iteration step k, each sensor i exchanges its data, $\hat{\mu}^{(k)}$ and $r_i^{(k)}$, with its neighboring nodes, i.e. those belonging to $\mathcal{N}_i$. Then, it proceeds to update its estimation of the shadow price $\hat{\mu}^{(k)}$, according to the *nearest neighbor rule*, proposed by Jadbabaei *et al* [9], which is modified as following:

NDTA Neighbor-based Distributed Threshold Assignment Algorithm

Initialization

Initialize the following items:

1. A_is and T.
2. $S = \sum_{i=1}^{n} A_i$.
3. $\forall i = 1..n, S_i = \sum_{i \in \mathcal{N}_i} A_i$.

Main Loop

Do until $\max_i |r_i^{(k+1)} - r_i^{(k)}| < \epsilon$

At each sensor node i,

1. Update the shadow price as following:

$$\hat{\mu}_i^{(k+1)} = \left[\hat{\mu}_i^{(k)} + \gamma \left(\sum_{i \in N_i} A_i T_i(r_i^{(k)}) - T\frac{S_i}{S}\right)\right]^+$$

2. Update $r_i^{(k)}$ according to the following equation:

$$\alpha^{r_i^{(k+1)}} G_i(\alpha^{r_i^{(k+1)}}) = \frac{1}{\hat{\mu}_i^{(k)} A_i \ln \alpha}$$

NDTA. Neighbor-based Distributed Threshold Assignment Algorithm

$$\hat{\mu}_i^{(k+1)} = \left[\hat{\mu}_i^{(k)} + \gamma \left(\sum_{i=1}^{n} A_i T_i(r_i^{(k)}) - \frac{\sum_{i \in \mathcal{N}_i} A_i}{\sum_{i=1}^{n} A_i} T\right)\right]^+ \tag{25}$$

where $\sum_{i \in \mathcal{N}_i} A_i / \sum_{i=1}^{n} A_i$ denotes the fraction of total threshold T assigned to the virtual subnetwork established by nodes $\{i\} \cup \mathcal{N}_i$. We will refer to this algorithm as *Neighbor-based Distributed Threshold Assignment* Algorithm or for short *NDTA* Algorithm. The NDTA Algorithm is stated below.

VI. Experimental Evaluation

We have conducted simulation experiments to evaluate the performance of our proposed algorithm. We verify that NDTA Algorithm, locally executed on each node, may indeed achieve the desired global optimal threshold assignment. In our simulation scenario, we consider a sensor network consisting of 100 sensor nodes. We assume that each node i incessantly takes measurements of a physical phenomenon, whose CDF obeys an exponential distribution with the exponent parameter λ_i. Although such a distribution may sound to be of limited interest, it is worth mentioning that many significant real world applications might fall within such a framework. Amongst such applications are monitoring the dwell time of a traffic flow which pursue a Poisson distribution. The corresponding coefficient A_i is assumed to be randomly drawn from a uniform distribution over $[0, 5]$. Step size is chosen to be $\gamma = 1.2$ and the total threshold T is set to 10. The most significant issues of interest are the evolutions of primal and dual (shadow price) variables. Evolution of assigned thresholds T_is and shadow price μ for NDTA Algorithm are depicted in Fig. 1 and 2, respectively. It is apparent from these figures that by spending less than 300 iteration steps, convergence was achieved and thereafter, μ and T_is had intangible variations.

NDTA is indeed a suboptimal algorithm. In order to compare the performance of NDTA with the optimal scheme, threshold assignment for both NDTA and optimal scheme

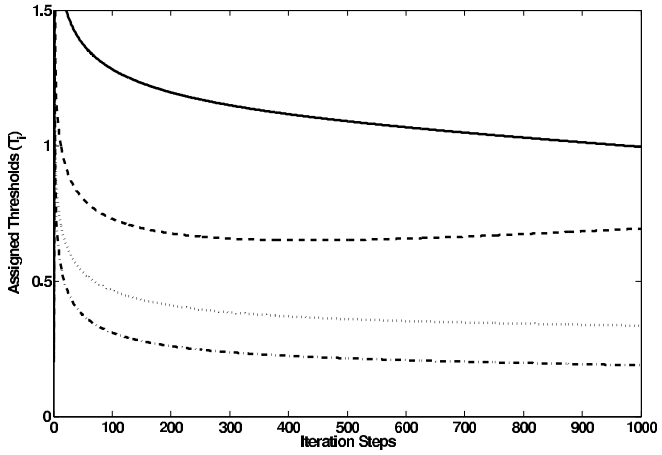

Fig. 1. Evolution of The Assigned Thresholds $T_i(r_i)$ for Some Nodes Using NDTA Algorithm

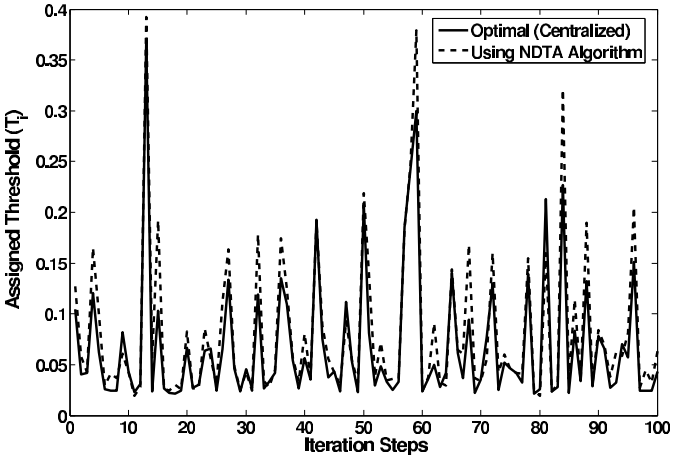

Fig. 3. Comparison Between NDTA and Centralized Scheme

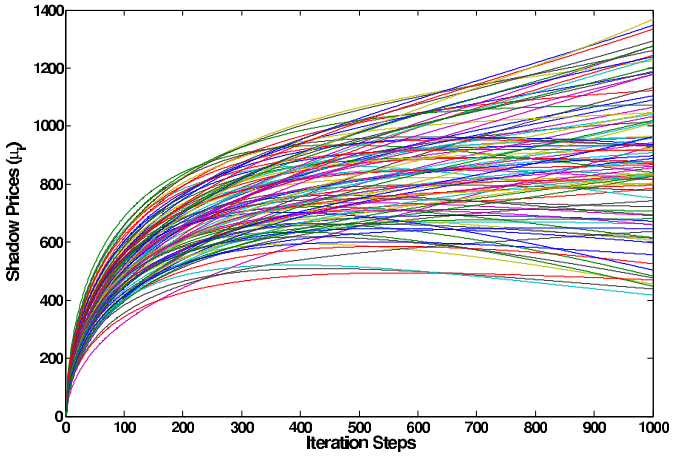

Fig. 2. Evolution of The Shadow Price μ for Some Nodes Using NDTA Algorithm

which is obtained using a centralized algorithm is depicted in Fig. 3, which shows that the result of NDTA is quite close to the optimal case.

VII. Conclusion

There have been many studies exploring various applications of WSNs such as monitoring. In this paper, we addressed the problem of distributed threshold selection for aggregate threshold monitoring in WSNs. The nature of such networks has induced the design of distributed algorithms for exchanging the information among the sensor nodes. We formulated threshold selection problem as an optimization problem that considers the data distribution of distinct monitoring variables, and whose objective is to minimize the probability of global polling. The original problem was non-convex, thus we adopted the so called FPTAS reformulation which was convex and has been solved using a centralized approach. We elaborated this method and solved the problem via its dual so as to achieve a distributed solution. Our distributed solution leads to the fully distributed algorithm called NDTA, which acts using only local information of the network using the data which has been provided by the neighboring nodes. The results extracted from the experimental evaluation was promising and demonstrated the achieved performance of the NDTA algorithm is quite comparable to the results of the centralized approach.

References

[1] S. Aggrawal, S. Deb, K. V. M. Naidu and R. Rastogi, "Efficient Detection of Distributed Constraint Violations," In *ICDE*, pp. 1320-1324, 2007.
[2] B. Babcock and C. Olston, "Distributed Top-k Monitoring," In *SIGMOD*, pp. 28-39, 2003.
[3] D. Bertsekas, *Nonlinear Programming*, Athena Scientific, 1999.
[4] S. Boyd, A. Ghosh, B. Prabhakar and D. Shah, "Gossip Algorithms: Design, Analysis, and Applications," In *INFOCOM*, pp. 1653-1664, 2005.
[5] G. Cormode, S. Muthurikshnan, K. Yi. "Algorithms for Distributed Functional Monitoring," In *SODA*, 2008.
[6] A. Deshpande, C. Guestrin, S. Madden, J. M. Hellerstein and W. Hong, "Model-Driven Data Acquisition in Sensor Networks," In *VLDB*, pp. 588-599, 2004.
[7] A. Deshpande, C. Guestrin and S. Madden, "Using Probabilistic Models for Data Management in Acquisitional Environments," In *CIDR*, pp. 317-328, 2005.
[8] M. Dilman and D. Raz, "Efficient Reactive Monitoring," In *INFOCOM*, pp. 1012-1019, 2001.
[9] A. Jadbabaie, J. Lin and A. Stephen Morse,"Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules," In *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988-1001, June 2003.
[10] S. Kashyap, J. Ramamirtham, R. Rastogi and P. Shukla, "Efficient Constraint Monitoring Using Adaptive Thresholds," In *ICDE*, pp. 526-535, 2006.
[11] F. P. Kelly, A. Maulloo and D. K. H. Tan, "Rate Control for Communication Networks: Shadow Prices, Proportional Fairness, and Stability," *Operational Research Society*, vol. 49, no. 3, pp. 237-252, 1998.
[12] R. Keralapura, G. Cormode and J. Ramamirtham, "Communication-Efficient Distributed Monitoring of Thresholded Counts," In *SIGMOD*, pp. 289-300, 2006.
[13] D. Kifer, B. Shai and J. Gehrke, "Detecting Change in Data Streams," In *VLDB*, pp. 180-191, 2004.
[14] C. Olston, J. Jiang, and J. Widom. "Adaptive filters for continuous queries over distributed data streams." In *SIGMOD*, 2003.
[15] V. Poosala and Y. Ioannidis, "Selectivity Estimation Without The Attribute Value Independence Assumption," In *VLDB*, pp. 486-495, 1997.
[16] I. Sharfman, A. Schuster and D. Keren, "A Geometric Approach to Monitoring Threshold Functions over Distributed Data Streams," In *SIGMOD*, pp. 301-312, 2006.