Decentralized Average Environmental Temperature Estimation

Mario Coutino
MSc. Electrical Engineering
Delft University of Technology
Std. Num: **4410475**
M.A.Coutinomingue@student.tudelft.nl

Kris Shrishak S
MSc. Electrical Engineering
Delft University of Technology
Std. Num: **4411811**
K.S.Sridaran@student.tudelft.nl

I. Introduction

In this project, a simulation of a sensor network for a plant with dimensions $30m \times 60m$ is implemented. The sensors are placed on the ceiling and are used to monitor the environmental temperature. The sensors have a transmission radius of $3m$ and communicate through wireless data transmission. The computation of the average temperature and transmission in the network is performed in a decentralized manner. Two different methods, based on dual decomposition and constraint enforcement, are presented and compared. Problems as noisy measurements and package losses are also simulated and the performance of the methods under different update schemes (synchronous and asynchronous) is reported.

II. Design Overview

As the problem statement does not put any constraints on the energy consumption or actual sensing range (sensitivity of the sensor), we did not delve into it. The most important criterion considered in designing the network was maximum coverage (with a considerable amount of sensors). We have considered the communication range to be 3m and assumed the sensing range as the half of the communication range. Maximum coverage is expressed in terms of the number of sensor nodes[1]. It is expressed by the following equation:

$$NCovered_p = \begin{cases} 1 & d(s_i, m_p) \leq Rs \\ 0 & \text{otherwise} \end{cases}$$

where
R_s is the sensing range.
$d(s_i, m_p)$ is the Euclidean distance between the sensing node i and the monitoring point p.

In addition, we make sure that every monitoring location has only one sensor node. We designed the sensor network in a rectangular grid with 231 sensors. Each sensor is at a distance of $3m$ from another. We have 11 rows and 21 columns of sensors. In this way every part of the plant is covered. Each sensor in the network is identified by its ID and its position in the grid. To keep it simple, a sensor at position (i, j) communicates with its neighbors along row i and column j. They do not communicate with sensors along the diagonals. Each of the four corner sensors have two neighbors, the edge sensors have three neighbors while other sensors have four neighbors. The sensors store the ID of its neighbors in order to communicate.

III. ADMM for Decentralized Consensus Optimization

As no fusion center is available in our sensor network $\mathcal{G}(V, E)$, the decentralized ADMM is implemented as first solution to the average estimation problem.

Lets define the optimization problem to be solved as

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \sum_{i=1}^{N} \frac{1}{2}(x_i - a_i)^2 \\ \text{s.t} \quad & x_i - z_{ij} = 0,\ x_j - z_{ij} = 0,\ \forall (i,j) \in E \end{aligned}$$

After the Augmented Lagrangian of the previous problem is formulated, it is clearly observed that by introducing the decouple variables z_{ij} the minimizing process can now be solved in a distributed way.

$$\begin{aligned} L_\rho(x, z, v) = \sum_{i \in V} \frac{1}{2}(x_i - a_i)^2 + \sum_{(i,j) \in E} (v_{ij}^T (x_i - z_{ij}) \\ + \frac{\rho}{2}\|x_i - z_{ij}\|_2^2 + v_{ji}^T (x_j - z_{ij}) + \frac{\rho}{2}\|x_j - z_{ij}\|_2^2) \end{aligned}$$

Now the decentralized average estimation problem is reduced to find the x_i^*, z_{ij}^*, v_{ij} and v_{ji} that maximize the Augmented Lagrangian, which can be solve through an iterative method that only requires computations inside each node and transmission of messages between neighbors.

IV. Gradient Based Average Approximation (GBA)

Approaching the consensus problem as an optimization problem, it is possible to use a similar idea as the Augmented Lagrangian in ADMM case to devise a method based in a *relaxed* way to enforce the desired constraints.

Assuming a probability distribution $p(\mathbf{t})$ over the graph $\mathcal{G}(V, E)$ defined by our mesh of sensors, we want to find $\mathbf{t}^*$ that maximizes the given probability distribution and at the same time be a proper estimate of the average temperature given the measurements in each node. In addition, the decentralized property must hold, otherwise the idea of a system without central unit will not be possible.

Let the argument of our exponential function for the probability distribution be

$$f(\mathbf{t}) = -\frac{1}{2}\sum_{i\in V}(a_i - t_i)^2 - \frac{1}{2}\sum_{i\in V}\sum_{j\in\mathcal{N}(i)}(t_i - t_j)^2$$

where
a_i is the measurement at node i
t_i is the average temperature estimate at node i
$\mathcal{N}(i)$ neighboring nodes of node i

From the previous equation can be seen that the first term is exactly the same as the proposed cost for ADMM case. The difference comes when the constraints are enforced. Instead of using the Augmented Lagrangian and work under the dual problem, a straight forward function which tries to account for differences in the estimate temperature at the edges is used.

After casting the problem as a new unconstrained one, some advantages can be observed. The separability of the cost is conserved. Less number of variables have to be computed in each iterations and hence smaller packages.

Unfortunately, the algorithm now makes a trade off between convergence speed and convergence tolerance (not all t_i reach the same value). This trade off can be accounted by adding a penalizing variable γ to the first term, which is responsible for differences between nodes estimates. As $\gamma \to 0$ the method will find an optimal $\mathbf{t}^*$ with equal entries, which probably will not be exactly the true mean.

The final implementation of GBA uses a gradient descent approach with fix step size ρ (typically in the range $[0.2 - 0.4]$ and a tunable cost γ (of the order of 10^{-6}). During experimental trials, it was observed that the selection of this parameters is fairly independent of the mesh size.

V. Results

A comparison under different operation conditions and update schemes is made between ADMM and GBA. In the experimental setup, under the proposed sensor network $\mathcal{G}(V, E)$, conditions of transmission failures and noise in the measurements in each iteration is simulated.

The general stop criteria for both methods, in both updating schemes, was selected as

$$\epsilon_k = (T - \mathbf{t}_k)^2/|V| < 10^{-2}$$

where
T is the true mean of the temperature measurements
$\mathbf{t}$ is the k estimate of the network
$|V|$ is the number of nodes in the graph

If this criteria is not achieved in any given realization, the maximum number of iterations allowed for the synchronous case was set at 500 and for the asynchronous case at 10^5.

The simulations consist of 100 realizations with each node (no restriction for their values) initialized to random temperatures. The average and worst case results obtained are shown in the tables below.

A. *ADMM Performance*

Synchronous Scheme

Channel Conditions	Num. Iterations/ Worst Case	Num. Transmissions/ Worst Case
Ideal	59/145	50,740/124,700
Noise 5%	61/159	52,460/136,740
Packet Loss 30%	63/166	54,180/142,760
Packet Loss 10% + Noise 10%	58/161	49,880/138,460

TABLE I: GBA Simulation Results. Mean and worst case values

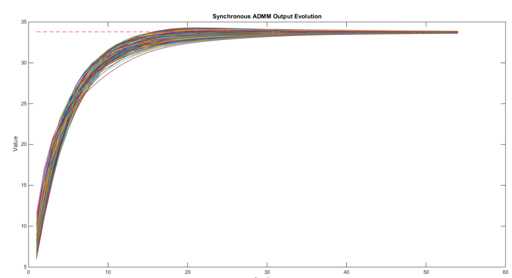

Fig. 1: Synchronous ADMM Simulation Output

Asynchronous Scheme

Channel Conditions	Num. Iterations/ Worst Case	Num. Transmissions/ Worst Case
Ideal	86,436/10e4	643,670/745,026
Noise 5%	-/-	744,570/744,842
Packet Loss 30%	89,907/10e4	669,530/744,803
Packet Loss 10% + Noise 10%	-/-	744,630/745,094

TABLE II: GBA Simulation Results. Mean and worst case values

Under asynchronous update for the noisy cases, it was not possible to achieve convergence under the established terms. However, if the number of iteration is increased or the margin of error accepted per node is relaxed, ADMM will deliver a good estimation of the average temperature.

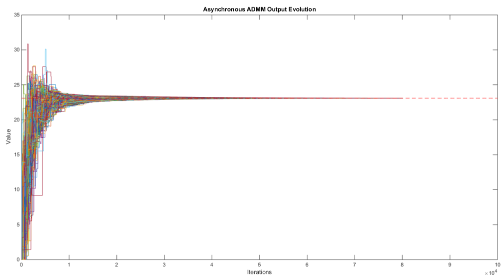

Fig. 2: Asynchronous ADMM Simulation Output

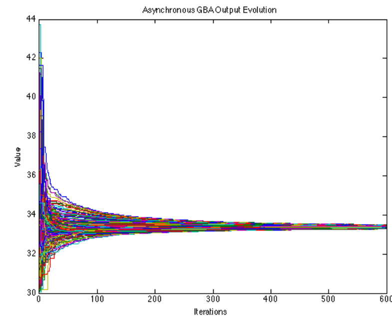

Fig. 4: Asynchronous GBA Simulation Output

B. GBA Performance

Both updating schemes are simulated using the same method parameters $\rho = 0.4$ and $\gamma = 0$ (these value for γ and ρ were found the best after several tests).

Synchronous Scheme

Channel Conditions	Num. Iterations/ Worst Case	Num. Transmissions/ Worst Case
Ideal	70/110	60,130/94,490
Noise 5%	72/138	61,848/118542
Packet Loss 30%	140/240	120,260/206,160
Packet Loss 10% + Noise 10%	70/141	60,130/121,119

TABLE III: GBA Simulation Results. Mean and worst case values

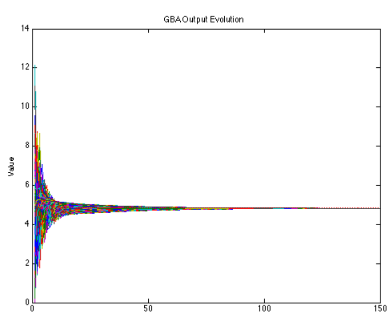

Fig. 3: Synchronous GBA Simulation Output

Asynchronous Scheme

Channel Conditions	Num. Iterations/ Worst Case	Num. Transmissions/ Worst Case
Ideal	6.3e4/10e4	80,259/372,076
Noise 5%	6.8e4/10e4	102,392/372,119
Packet Loss 30%	8.2e4/10e4	145,300/372,262
Packet Loss 10% + Noise 10%	6.8e4/10e4	112,421/372,158

TABLE IV: GBA Simulation Results. Mean and worst case values

VI. CONCLUSION

In this project two approaches to solve the distributed computation of the average temperature of a plant was compared. Their individual performance was discussed in terms of the number iterations and number of transmission for the network. Even though both approaches come from the almost the same principle, a separable convex optimization problem with enforced constraints, they show slightly different performance. Particularly, GBA also shows that even when the design of the algorithm is meant as a suboptimal solution, its experimental performance is good and it performs better in the asynchronous update under noise, which is the closest experimental set up to a real scenario.

REFERENCES

[1] H. Zainol Abidin, N.M. Din and N.A.M. Radzi, *Deterministic Static Sensor Node Placement in Wireless Sensor Network based on Territorial Predator Scent Marking Behavior*, IJCNIS,2013

[2] O. Hua et al *Stochastic Alternating Direction Method of Multipliers*, Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013.

[3] M. Hong et al *A Distributed, Asynchronous and Incremental Algorithm for Nonconvex Optimization: An ADMM Based Approach* arXiv:1412.6058