Distributed Computation of Quantiles via ADMM

Franck Iutzeler

***Abstract*—In this paper, we derive distributed synchronous and asynchronous algorithms for computing quantiles of the agents local values. These algorithms are based on the formulation of a suitable problem, explicitly solvable by the Alternating Direction Method of Multipliers (ADMM), and recent randomized optimization methods.**

***Index Terms*—Quantile, Median, Gossiping, ADMM, Distributed Algorithms.**

I. Introduction

Consider a connected network of N agents, each with a scalar value $(a_i)_{i=1,..,N}$. The problem of distributively computing a function of these values by exchanging information locally over the network has received a lot of interest [1], [2], [3]. Particularly, solving this problem in an asynchronous way attracted the attention of the community, notably with applications to Wireless Sensor Networks [4].

The most famous gossiping problems include average computation [4], [5], [6], [7], [8], greatest value spreading [9], [10], [11], or some more involved means (see [12] and references therein). However, to the best of our knowledge, no algorithm for quantiles or median computation are available when they can have great potential applications notably in the field of distributed statistical signal processing. Indeed, the median for instance is a more robust estimator of the central tendency than the average, and quantiles are generally more robust than variance when looking at dispersion.

Recently, distributed optimization algorithms were developed [13], [14], along with randomized, asynchronous versions [15], [16]. These methods are based on formulating proper distributed optimization problems, and solving them using splitting-based optimization methods such as the popular Alternating Direction Method of Multipliers (ADMM) [17], [18], [19], or Primal-Dual algorithms [20], [21]; then, randomized versions are derived using suited coordinate descent [15], [22].

In this note, we first design a convex optimization problem whose solution meets the sought quantile to compute. Then, in Section III, we use a distributed formulation of ADMM to derive a distributed algorithm for quantile computation. In Section IV, an asynchronous version which communication scheme mimics Random Gossip [4] is proposed. Finally, we illustrate the relevance of our algorithms in Section V.

II. Quantile finding as an Optimization Problem

Let $a = (a_i)_{i=1,..,N}$ be a size-N real vector and $\mathfrak{s}$ be a permutation that sorts a in increasing order. For a given $q \in [0,1]$, our goal is to find a $q\%$ *quantile estimator* of vector a in the sense that we wish to find a real value $a^\star$ such that

$$\begin{array}{rll} a^\star \in & \left[a_{\mathfrak{s}(\lfloor qN \rfloor)}; a_{\mathfrak{s}(\lceil qN \rceil)}\right] & \text{if } q \in [1/N, 1[\\ \text{and } & a^\star \leq \min_i a_i & \text{if } q < 1/N \\ & a^\star \geq \max_i a_i & \text{if } q = 1 \end{array} . \tag{1}$$

Note that the degenerate cases where $qN < 1$ and $qN = N$ can also boil down to finding the minimum and the maximum of the agents values respectively (see e.g. [11]).

In this section, we formulate an optimization problem, based on proposed quantile objective functions, which solutions verify the sought conditions of Eq. (1).

A. Objective functions

Let us define *quantile objective functions* $(f_a^\beta)_{a,\beta}$ as

$$\begin{array}{rccl} f_a^\beta : & \mathbb{R} & \to & \mathbb{R} \\ & x & \mapsto & f_a^\beta(x) = \begin{cases} \beta(a-x) & \text{if } x < a \\ x - a & \text{if } x \geq a \end{cases} \end{array} \tag{2}$$

which rely on two parameters:

- $a \in \mathbb{R}$; a *point* of the set;
- $\beta > 0$; a *scale* parameter.

It is easy to see that for any $a \in \mathbb{R}, \beta > 0$, f_a^β is convex and continuous. In addition, its proximity operator can be explicitly computed for any $\gamma > 0$ and any $z \in \mathbb{R}$ as (See Appendix A)

$$\begin{aligned} \mathbf{prox}_{\gamma f_a^\beta}(z) &\triangleq \operatorname*{arg\,min}_{w \in \mathbb{R}} \left\{ f_a^\beta(w) + \frac{1}{2\gamma}\|w - z\|_2^2 \right\} \\ &= \begin{cases} z + \gamma\beta & \text{if } z < a - \gamma\beta \\ z - \gamma & \text{if } z > a + \gamma \\ a & \text{if } z \in [a - \gamma\beta; a + \gamma] \end{cases} . \end{aligned} \tag{3}$$

The fact that f_a^β is not differentiable but has an explicit proximal operator naturally leads us to consider proximal minimization algorithms like ADMM or primal-dual algorithms for problems involving such functions (see [23, Chap. 27]).

B. Equivalent Problem

Consider the optimization problem:

$$\min_{x \in \mathbb{R}} f(x) \triangleq \sum_{i=1}^N f_{a_i}^\beta(x). \tag{4}$$

Lemma 1: Let $\beta = \frac{q}{1-q}$. Then, the minimizers of Pb. (4) verify Eq. (1).

The proof is reported in Appendix B. Notice that, although other choices for β are possible, we emphasize the fact that the present one do not necessitates any knowledge of the total number of agents N which is a very attractive property in practical networked systems.

F. Iutzeler is with Laboratoire Jean Kuntzmann, Université Grenoble Alpes, Grenoble, France

III. Distributed Quantile computation

Consider a network of N agents linked through an undirected connected graph $\mathcal{G} = (\{1,..,N\}, E)$ where E is the set of edges: $E = \{e = \{i,j\} : i$ and j are connected$\}$. $\mathcal{N}_i$ is the neighborhood of i, that is the agents j such that $\{i,j\} \in E$ ($i \notin \mathcal{N}_i$), and $d_i = |\mathcal{N}_i|$ is the degree of i.

Let each agent i have a (local) point $a_i \in \mathbb{R}$; the objective of this section is to design an algorithm that distributively computes a $q\%$ *quantile estimator* of the vector $(a_i)_{i=1,..,N}$.

To do so, we proceed as such: i) we reformulate Problem (4) so that it is distributed-ready; and ii) apply ADMM on it.

Distributed formulation. Each agent i has a local point a_i and thus can maintain local function $f^\beta_{a_i}$. To lead to distributed algorithms, the new problem must feature a local variable $x(i) \in \mathbb{R}$ at each agent/function i; thus, in order to recover the solutions of Prob. 4, one must add a *consensus* constraint $x(1) = x(2) = .. = x(N)$.

There are several ways to formulate consensus constraints over graphs [13], [24], [16]; for brevity and in line with our target algorithms, we proceed as in [24]: for all edges $e = \{i,j\} \in E$, we create a size-2 additional variable $y(e) = [y_1(e), y_2(e)]^\mathrm{T} \in \mathbb{R}^2$ such that $y_1(e) = x(i)$ and $y_2(e) = x(j)$. Then, imposing $y_1(e) = y_2(e)$ for all $e \in E$, by adding in the cost function a convex indicator function $\iota(y(e)) = 0$ if $y_1(e) = y_2(e)$ and $+\infty$ elsewhere, is the same as imposing the consensus as soon as the graph is connected. For clarity, at edge $e = \{i,j\}$ we note $y_1(e) = y(i,e)$ if $i < j$ and $y_2(e) = y(i,e)$ elsewhere.

We note $x = [x(1),..,x(N)]^\mathrm{T} \in \mathbb{R}^N$ and $y = [y(1),..,y(|E|)]^\mathrm{T} \in \mathbb{R}^{2|E|}$; similarly, we will adopt the same notation for any variable with the same size. Then, our distributed-ready problem reads:

$$\min_{x\in\mathbb{R}^N, y\in\mathbb{R}^{2|E|}} \underbrace{\sum_{i=1}^N f^\beta_{a_i}(x(i))}_{F(x)} + \underbrace{\sum_{e\in E} \iota(y(e))}_{G(y)} \tag{5}$$

$$\text{s.t. } Mx = y \qquad \text{where } M = \begin{bmatrix} M_1 \\ \vdots \\ M_{|E|} \end{bmatrix}$$

with M_e the $2\times N$ matrix such that $M(1,i) = M(2,j) = 1$ if $e = \{i,j\}$ and zeros elsewhere, so that $M_e x = [x(i), x(j)]^\mathrm{T} = y(e)$. This makes M size $2|E| \times N$ and full column-rank.

Algorithm. As F and G have explicit proximal operators, it is natural to use ADMM on Pb. (5), which leads to the following algorithm after simplification of some variables.

Distributed quantile computation

Init.: $x_0, \bar{x}_0, z_0, \lambda_0 \in \mathbb{R}^N$, $\rho > 0$, $\beta = \frac{q}{1-q}$.
At each iteration k:

- Each agent i compute

$$z_k(i) = \frac{x_k(i) + \bar{x}_k(i)}{2} - \lambda_k(i)$$
$$x_{k+1}(i) = \mathbf{prox}_{f^\beta_{a_i}/(\rho d_i)}(z_k(i))$$
$$= \begin{cases} z_k(i) + \frac{\beta}{\rho d_i} & \text{if } z_k(i) < a_i - \frac{\beta}{\rho d_i} \\ z_k(i) - \frac{1}{\rho d_i} & \text{if } z_k(i) > a_i + \frac{1}{\rho d_i} \\ a_i & \text{elsewhere} \end{cases}$$

- All agents send their version of x_{k+1} to their neighbors

$$\bar{x}_{k+1}(i) = \frac{1}{d_i} \sum_{j\in\mathcal{N}_i} x_{k+1}(j)$$

- Each agent i update

$$\lambda_{k+1}(i) = \lambda_k(i) + \frac{x_{k+1}(i) - \bar{x}_{k+1}(i)}{2}$$

Theorem 1: Let $q \in]0,1[$, $\beta = \frac{q}{1-q}$, and $\rho > 0$. The *Distributed quantile computation* algorithm converges to a consensus over a value verifying Eq. (1).

$$x_k \to (x^\star, x^\star, .., x^\star) \text{ with } x^\star \text{ verifying Eq. (1).}$$

Proof: This theorem comes from the succession of three results: i) the *Distributed quantile computation* algorithm is an instantiation of the ADMM on Prob. (5) of the form $\min_{x,y} F(x) + G(y)$ s.t. $Mx = y$ with F and G convex; so it converges the algorithm converges to a solution of (5) [17], [18]; ii) by construction of Prob. (5), its solutions have the form $[x^\star, x^\star, .., x^\star]^\mathrm{T}$ where $x^\star$ is a solution of Prob. (4); and iii) by Lemma 1, the solutions of (4) verify Eq. (1). ■

Remark 1: An efficient initialization of the algorithm is to set $x_0(i) = \bar{x}_0(i) = a_i$. λ_0 can be taken as the null vector, and $\rho \approx 1$. This kind of initialization is performed in the numerical experiments, along with discussions over the choice of ρ.

IV. Asynchronous Gossiping for Quantile estimation

Randomized versions of the ADMM have been introduced in [15] in order to produce asynchronous gossip-like distributed algorithms; this kind of result was later extended to more general algorithms in [22], [25]. In this section, we build upon the distributed problem Eq. (5) and a careful application of the Asynchronous Distributed ADMM of [15] to produce a gossip-based algorithm for quantile estimation.

As our formulation is edge-based, at each iteration k, we select an edge $e_k = (i,j) \in E$ following an i.i.d. random process; then, only agents i and j update and exchange during this iteration (see [15] for derivation details). This kind of gossiping (drawing two nodes who average their values) is similar to the *Random Gossip* averaging algorithm [4].

Asynchronous Gossip for quantile computation

Init.: $x_0, z_0 \in \mathbb{R}^N$, $\bar{x}_0 \in \mathbb{R}^{|E|}, \lambda_0 \in \mathbb{R}^{2|E|}$, $\rho > 0$, $\beta = \frac{q}{1-q}$.
At each iteration k, draw an edge $e_k = \{i,j\} \in E$:

- Agents $v \in \{i,j\}$ compute

$$z_k(v) = \frac{1}{d_v} \sum_{n\in\mathcal{N}_v} (\bar{x}_k(\{v,n\}) - \lambda_k(v, \{v,n\})$$
$$x_{k+1}(v) = \mathbf{prox}_{f^\beta_{a_v}/(\rho d_v)}(z_k(v))$$
$$= \begin{cases} z_k(v) + \frac{\beta}{\rho d_v} & \text{if } z_k(v) < a_v - \frac{\beta}{\rho d_v} \\ z_k(v) - \frac{1}{\rho d_v} & \text{if } z_k(v) > a_v + \frac{1}{\rho d_v} \\ a_v & \text{elsewhere} \end{cases}$$

- Agents i and j exchange their copy of x_{k+1}

$$\bar{x}_{k+1}(e_k) = \frac{x_{k+1}(i) + x_{k+1}(j)}{2}$$

- Agents $v \in \{i, j\}$ update

$$\lambda_{k+1}(v, e_k) = \lambda_k(v, e_k) + \rho(x_{k+1}(v) - \bar{x}_{k+1}(e_k))$$

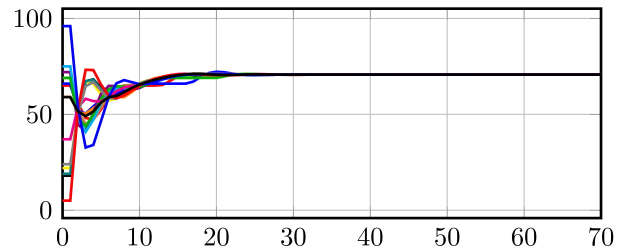
Theorem 2: Let $q \in]0, 1[$, $\beta = \frac{q}{1-q}$, and $\rho > 0$. If (e_k) is an i.i.d. sequence valued in E such that for all $e \in E$, $\mathbb{P}[e_1 = e] > 0$, then the *Asynchronous Gossip for quantile computation* algorithm converges almost surely to a consensus over a value verifying Eq. (1).

$$x_k \to (x^\star, x^\star, .., x^\star)\text{a.s. with } x^\star \text{ verifying Eq. (1).}$$

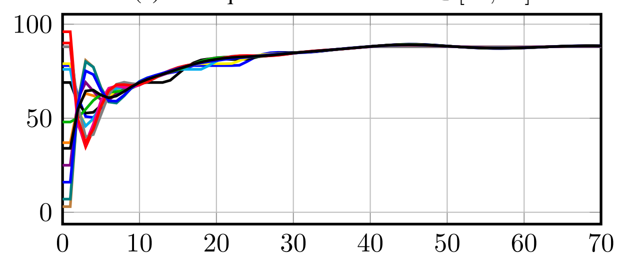
Proof: As the *Asynchronous Gossip for quantile computation* algorithm is obtained by randomized coordinate descent on distributed ADMM, [15, Th. 3] gives the almost sure convergence to a solution of Prob. (4); and, as for Th. 1, we get that the solutions of (4) verify Eq. (1) by Lemma 1. ■

Remark 2: An efficient initialization of the algorithm is to set $\bar{x}_0(i, j) = a_i$. λ_0 can be taken null and $\rho \approx 1$.

Remark 3: For different gossiping schemes such as asynchronous one-way communication (e.g. broadcast) over undirected graphs, we refer the reader to [25].



(a) 80% quantile: $x^\star = 70.71 \in [69, 72]$



(b) 80% quantile: $x^\star = 88$

Fig. 1: *Distributed quantile computation* on different data

V. Numerical Experiments

In this section, we illustrate the features of both our distributed synchronous and asynchronous algorithms. In all experiments, we consider $N = 15$ agents linked by a connected undirected graph with 72 edges (out of 105 possible). The values (a_i) of the sensors are taken randomly in the integers between 0 and 100.

In Fig. 1, we represent all the sensors values $(x_k(i))$ versus the number of iterations for the *Distributed quantile computation* with $q = 0.8$ *i.e.* a quantile at 80% is sought. We set $\beta = q/(1 - q)$ as prescribed and $\rho = 0.1$. In Fig. 1a, the objective in the sense of Eq. 1 is to reach a value in $[69, 72]$ while in Fig. 1b, we manually set $a_{\mathfrak{s}(\lceil qN \rceil)} = a_{\mathfrak{s}(\lfloor qN \rfloor)}$ in the (a_i) so that the objective is to reach exactly 88. We observe that the agents have the sought behavior however, the convergence is slightly slower in the second case, but not seriously so. In both cases, one can notice that two goals collide in order to reach consensus over a quantile: i) the consensus itself, being close to its neighbors; and ii) minimizing the objective; the relative importance of these two sub-problems is actually tuned by free parameter ρ.

In Fig. 2, we keep everything as in Fig. 1a but, instead of taking $\rho = 0.1$, we use $\rho = 3$ in Fig. 2a and $\rho = 0.001$ in Fig. 2b. In Figure 2c, we plot for both $\rho = 3$ and $\rho = 0.001$ at each iteration i) the consensus error, equal to the distance between the agents value and the mean of the agents values $\|x_k - \frac{1}{N}\sum_i x_k(i)\|^2$; ii) the objective error, equal to the difference between the mean value and the sought optimum $\|\frac{1}{N}\sum_i x_k(i) - x^\star\|^2$; and iii) the total error, equal to the distance between the agents values and the sought optimum $\|x_k - x^\star \mathbf{1}\|^2$. One can notice that the lower ρ the slower the consensus; however, when ρ get too big, the consensus is very fast but the agents move towards the sought objective more slowly. In conclusion, while any value for ρ brings convergence theoretically and practically, the convergence times can be very different. Heuristics to chose a correct parameter have appeared in the literature [26] but no distributed version exists to the best of our knowledge.

In Fig. 3, we illustrate the performance of our *Asynchronous Gossip for quantile computation* under the same configuration as Fig. 1a. In Fig. 3a, we display the agents values over time, we notice that the convergence has a quite different shape than the synchronous algorithm. In Fig. 3b, we plot the norm of the error over the mean of the reached value; as this value is an acceptable quantile, this evaluates both the error towards a consensus and an acceptable value. We compare our distributed synchronous and asynchronous algorithms over the decrease of this errors versus the number of full uses of the communication graph, that is 1 per iteration for the synchronous algorithm, and $1/|E| = 1/72 \approx 0.014$ for the asynchronous one. Although the synchronous algorithm still over-performs its asynchronous counterpart in this setup, both reach machine precision within a few hundreds full graph uses.

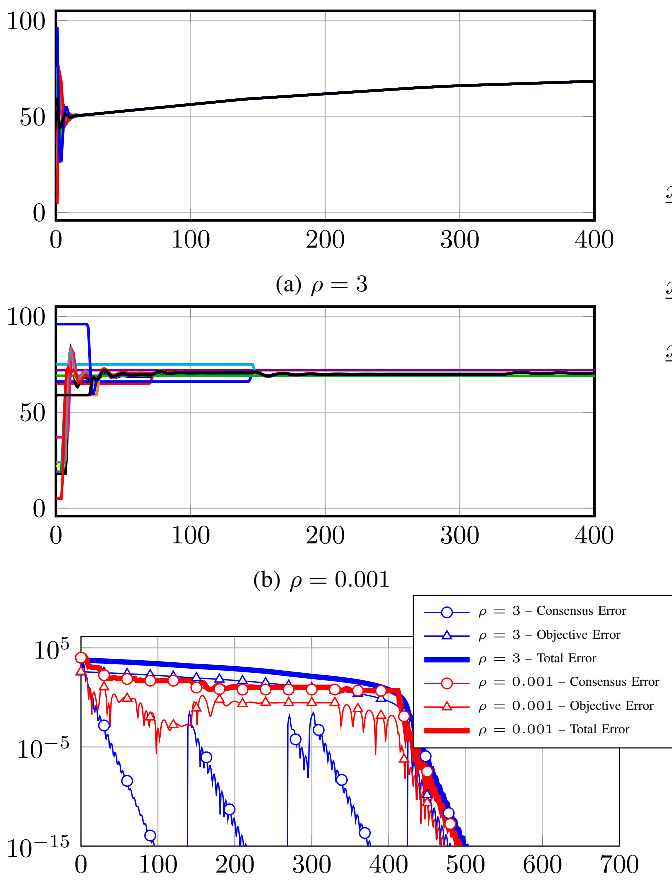
VI. Conclusion

In this note, we proposed a distributed algorithm for quantile computation along with an asynchronous gossip-based one. The derivation and convergence proofs of these algorithm rely on the application of (randomized) ADMM on a well chosen distributed problem.

Appendix A
Derivation of the proximal operator of f_a^β

For any $\gamma > 0$ and any $z \in \mathbb{R}$, the proximity operator of f_a^β is defined as

$$x = \mathbf{prox}_{\gamma f_a^\beta}(z) = \operatorname*{arg\,min}_{w \in \mathbb{R}} \left\{ \underbrace{f_a^\beta(w) + \frac{1}{2\gamma}\|w - z\|_2^2}_{g(w)} \right\}.$$

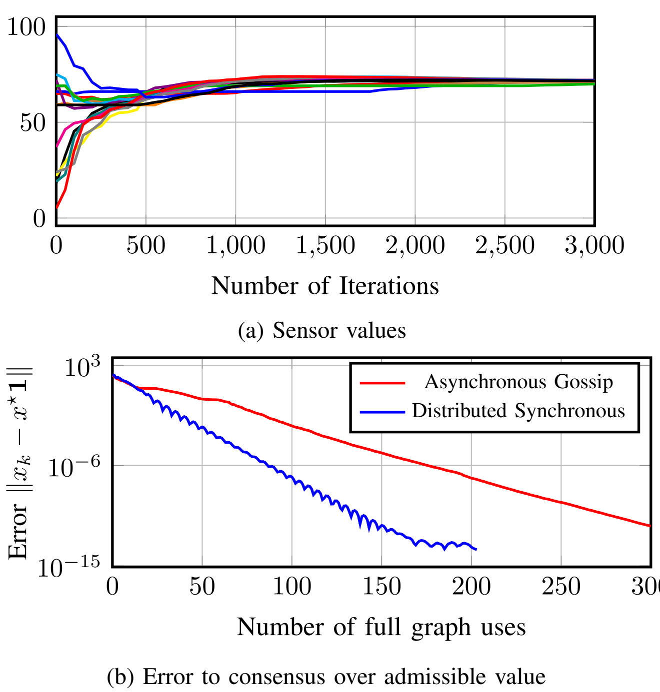


(a) $\rho = 3$

(b) $\rho = 0.001$

(c) Comparison

Fig. 2: *Distributed quantile computation* with different ρ



(a) Sensor values

(b) Error to consensus over admissible value

Fig. 3: *Asynchronous Gossip for quantile computation*

The derivation of this operator is similar to the one of the soft-thresholding operator as proximity operator of the ℓ_1-norm. Using the fact that x is the (unique) point such that 0 belongs to the subdifferential of g (see [23, Chap. 16, 26, 27]), we obtain $0 \in \partial f_a^\beta(x) + 1/\gamma(x-z)$. Now let us look at this equation for three cases: $x < a$, $x > a$, and $x = a$.

$\underline{x < a}$ $\partial f_a^\beta(x) = -\beta$ thus $0 = -\beta + 1/\gamma(x-z)$ so $x = z + \gamma\beta$. This corresponds to the case where $x < a$, that is $z + \gamma\beta < a$, otherwise said $z < a - \gamma\beta$.

$\underline{x > a}$ $\partial f_a^\beta(x) = 1$ thus $0 = 1 + 1/\gamma(x-z)$ so $x = z - \gamma$. Similarly, this corresponds to the case $z > a + \gamma$.

$\underline{x = a}$ This final case, $x = a$ corresponds to the values of z not covered in the previous cases.

Appendix B
Minimizers of Problem (4)

For any real number x, define the following quantities:

$$\left| \begin{array}{ll} B(x) & := \textbf{Card}\left\{a_i : a_i < x; i = 1, .., N\right\} \\ E(x) & := \textbf{Card}\left\{a_i : a_i = x; i = 1, .., N\right\} \\ A(x) & := \textbf{Card}\left\{a_i : a_i > x; i = 1, .., N\right\} \end{array} \right. . \quad (6)$$

Thanks to the convexity of problem, Fermat's rule tells us that the minimizers of Problem (4) are the zeros of the sub-differential of f (see [23, Chap. 16, 26]):

$$\begin{aligned} \partial f(x) &= \sum_{i: x < a_i} -\beta + \sum_{i: x > a_i} 1 + \sum_{i: x = a_i} [-\beta, 1] \\ &= -\beta A(x) + B(x) + E(x)[-\beta, 1] \\ &= -\beta(N - B(x) - E(x)) + B(x) + E(x)[-\beta, 1] \\ &= -\beta N + (\beta + 1)B(x) + [0, (1+\beta)E(x)]. \end{aligned}$$

Now, take $\beta = \frac{q}{1-q}$ and let us look at the zeros. First, due to the convexity and polyhedral form of f, we get that the zeros of ∂f are necessarily either one of the (a_i) or a segment of the form $[a_{\mathfrak{s}(i)}; a_{\mathfrak{s}(i+1)}]$.

First, if $E(x) = 0$, $0 = \partial f(x)$ implies that $B(x) = \frac{\beta}{\beta+1} N = qN$. As $B(x)$ is an integer, it is necessarily equal to $\lfloor qN \rfloor$; as there is exactly $\lfloor qN \rfloor$ entries of a below x, we get that $x \in]a_{\mathfrak{s}(\lfloor qN \rfloor)}; a_{\mathfrak{s}(\lceil qN \rceil)}[$ if $q \geq 1/N$, and $x < a_{\mathfrak{s}(1)} = \min_i a_i$ which verifies the sought condition (4).

Then, if $E(x) \neq 0$, define x^- (resp. x^+) as a real number strictly between x and the next entry of a strictly smaller (resp. greater) than x. Then, $E(x^-) = 0$ and $\partial f(x^-) = -\beta N + (\beta + 1)B(x^-)$, the same thing holds with x^+. A sufficient condition so that x is the only point such that $E(x) \neq 0$ and $0 \in \partial f(x)$ is that $\partial f(x^-) < 0$ and $\partial f(x^+) \geq 0$. With the prescribed value for β, we get that $\partial f(x^-) < 0$ if and only if $B(x^-) < qN$ and $\partial f(x^+) \geq 0$ if and only if $B(x^+) \geq qN$. Thus, with our choice of β, x is a zero of ∂f if there are at least $\lfloor qN \rfloor$ values below or equal to x (from the x^+ part) and strictly less than $\lfloor qN \rfloor$ strictly below (from the x^- part); thus, the condition is verified.

References

[1] M. H. DeGroot, "Reaching a consensus," *Journal of the American Statistical Association*, vol. 69, no. 345, pp. 118–121, 1974.

[2] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.

[3] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. Sturgis, D. Swinehart, and D. Terry, "Epidemic algorithms for replicated database maintenance," in *ACM Symposium on Principles of Distributed Computing (PODC)*, 1987, pp. 1–12.

[4] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508–2530, 2006.

[5] A. D. G. Dimakis, A. D. Sarwate, and M. J. Wainwright, "Geographic Gossip: Efficient Averaging for Sensor Networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 3, pp. 1205–1216, 2008.

[6] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, "Broadcast Gossip Algorithms for Consensus," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2748–2761, 2009.

[7] F. Iutzeler, P. Ciblat, and W. Hachem, "Analysis of sum-weight-like algorithms for averaging in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 61, no. 11, pp. 2802–2814, 2013.

[8] S. Wu and M. G. Rabbat, "Broadcast Gossip Algorithms for Consensus on Strongly Connected Digraphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 16, pp. 3959–3971, 2013. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6517485

[9] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking, "Randomized rumor spreading," in *IEEE Symposium on Foundations of Computer Science (FOCS)*, 2000, pp. 565–574.

[10] G. Giakkoupis and T. Sauerwald, "Rumor spreading and vertex expansion," in *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2012, pp. 1623–1641.

[11] F. Iutzeler, P. Ciblat, and J. Jakubowicz, "Analysis of max-consensus algorithms in wireless channels," *IEEE Transactions on Signal Processing*, vol. 60, no. 11, pp. 6103–6107, 2012.

[12] J. Cortés, "Distributed algorithms for reaching consensus on general functions," *Automatica*, vol. 44, no. 3, pp. 726–737, 2008.

[13] I. Schizas, A. Ribeiro, and G. Giannakis, "Consensus in Ad Hoc WSNs With Noisy Links – Part I: Distributed Estimation of Deterministic Signals," *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 350 –364, jan. 2008.

[14] E. Wei and A. Ozdaglar, "Distributed Alternating Direction Method of Multipliers," in *IEEE Conference on Decision and Control (CDC)*, 2012, pp. 5445–5450.

[15] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem, "Asynchronous distributed optimization using a randomized alternating direction method of multipliers," in *52nd IEEE Conference on Decision and Control*. IEEE, 2013, pp. 3671–3676.

[16] E. Wei and A. Ozdaglar, "On the o (1= k) convergence of asynchronous distributed alternating direction method of multipliers," in *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*. IEEE, 2013, pp. 551–554.

[17] P. Lions and B. Mercier, "Splitting algorithms for the sum of two nonlinear operators," *SIAM Journal on Numerical Analysis*, vol. 16, no. 6, pp. 964–979, 1979.

[18] J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, pp. 293–318, 1992.

[19] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[20] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2011.

[21] L. Condat, "A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms," *Journal of Optimization Theory and Applications*, vol. 158, no. 2, pp. 460–479, 2013.

[22] P. L. Combettes and J.-C. Pesquet, "Stochastic quasi-fejér block-coordinate fixed point iterations with random sweeping," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 1221–1248, 2015.

[23] H. H. Bauschke and P. L. Combettes, "Convex analysis and monotone operator theory in Hilbert spaces," 2011.

[24] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem, "Explicit convergence rate of a distributed alternating direction method of multipliers," *IEEE Transactions on Automatic Control*, vol. 61, no. 4, pp. 892–904, 2013.

[25] P. Bianchi, W. Hachem, and F. Iutzeler, "A coordinate descent primal-dual algorithm and application to distributed asynchronous optimization," *IEEE Transactions on Automatic Control*, vol. 61, no. 10, pp. 2947–2957, 2016.

[26] Z. Xu, M. A. Figueiredo, and T. Goldstein, "Adaptive ADMM with Spectral Penalty Parameter Selection," *arXiv preprint arXiv:1605.07246*, 2016.