Practical Interactive Scheme for Extremum Computation in Distributed Networks

Solmaz Torabi, Jie Ren, and John MacLaren Walsh
Dept. of Electrical and Computer Engineering
Drexel University
Philadelphia, PA 19104
Email: solmaz.t@drexel.edu, jie.ren@drexel.edu, jwalsh@ece.drexel.edu

***Abstract*—Several users observing independent random variables exchange error-free messages with one another and a central receiver, the central estimation officer (CEO), with the aim of enabling the CEO to compute either the maximum across users (the $\max$), or a a user attaining this maximum (the $\arg\max$) for each element of their local observation sequences. The fundamental lower bound on the information exchange rate required over all quantization schemes, both scalar and vector, is computed for this interactive problem with a known iterative convex geometric method. Next, an optimal dynamic program achieving the minimum expected rate and expected rate delay tradeoff over all scalar quantization schemes is presented, and the benefits of enabling users to overhear each others messages is assessed. Finally, a series of substantially reduced complexity dynamic programs are shown, both theoretically and empirically, to obtain performance close to the fundamental limits, and to scale favorably as the number of users grow.**

I. Introduction

A key theme in big-data processing is the design of distributed algorithms and computations to scale as best as possible as data becomes massive, and a key component to consider in this scaling is the sorts of cloud storage and processing architectures these algorithms will run on [1]. While recent work [2, 3] has empirically explored the power of coding to improve modern distributed computing frameworks by replicating redundant calculations over different nodes in order to mitigate stragglers, the impact of results regarding coding for distributed function computation in the design of modern distributed computing architectures is substantially less explored. As the information transfer required to complete a calculation over massive data, and its unpredictability, is typically a key bottleneck in the practical design of massive data processing architectures, a proper theory of optimal design for these platforms would determine the best network computation architectures and codes, in the sense of minimizing information transfer at a reasonable localized complexity, for given functions.

Over the years, substantial theoretical attention has been given by the information theory community to obtaining the minimum amount of information that must be exchanged where the goal is compute a function rather than batch data transfer. After an initial result [4] demonstrated the potential for savings relative to batch information transmission [5] for non-interactive lossless function computation, connections with graph entropy and conditional graph entropy for this non-interactive lossless case were explored [6–8]. For continuous sources, the performance of high-resolution quantizers for decentralized function computation has been studied in [9, 10]. Allowing interaction enables substantial rate savings [11], which can be theoretically well quantified in the context of collocated networks [12]. Polar codes have been adapted to interactive distributed function computation over collocated networks in [13]. However, even these elegant results require a distribution derived from solving the complex iterative convex geometric program in [12], which is unlikely to be computationally feasible in many contexts, and requires a somewhat high complexity blocking construction. For the case of computing extrema, a relatively simple interactive scheme built from several rounds of scalar quantization followed by Huffman coding in [14] was considered, with the twist that at each round all of the participating users must utilize a common (homogeneous) scalar quantizer, and must transmit their messages at each round in parallel.

In this paper, building upon this prior work, we study the design of interactive scalar quantizers for distributed lossless function computation in a collocated network for particular problem of computing extrema. A precise formulation of the problem under study follows.

A. Problem Formulation

Consider a network with m source terminals and a single sink terminal depicted in Fig. 1. Each source terminal $j \in [m]$ observes a sequence of discrete random variables $\boldsymbol{X}_j^n = (X_j(1), ..., X_j(n)) \in \mathcal{X}_j^n$ that is independent and identically distributed. The sources are also independent across users, so the probability mass function for $\boldsymbol{X}_{1:m}^n = (\boldsymbol{X}_1^n, \ldots, \boldsymbol{X}_m^n)$ factors as $P_{\boldsymbol{X}_{1:m}^n}(.) = \prod_{i=1}^n \prod_{j=1}^m P_{X_j}(.)$. The numerical examples and results in the paper will primarily focus on the case that each $X_j(i)$ is independent and identically distributed (in both $j \in [m]$ and $i \in [n]$) according to a uniform discrete distribution whose support set is taken to be $\mathcal{X} = \{1, ..., L\}$ without loss of generality. The sink terminal wishes to compute an extremum function $f : \mathcal{X}_1 \times ... \times \mathcal{X}_m \to \mathcal{Z}$, elementwise, obtaining the the sequence $\boldsymbol{Z}^n = (Z(1), ..., Z(n))$ with $Z(i) = f(X_1(i), X_2(i), ..., X_m(i))$. Here, by an extremum function, we mean that the function f under study will either specify the maximum $Z(i) = f_M(X_{1:m}(i)) = \max_{j \in [m]} X_j(i)$, indicate at least one user attaining it, $Z(i) = f_A(X_{1:m}(i))$ such that

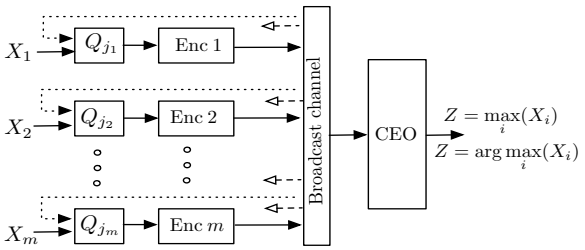

Fig. 1: System diagram

$X_{f_A(X_{1:m}(i))} = f_M(X_{1:m}(i))$, or indicate both of these, as studied in [14, 15]. In order to enable the sink to determine one of these functions, the sources interact with the sink in an interactive communication model known as *interactive function computation in a collocated network* [16, 17]. In this interaction model, the source terminals take turns broadcasting individual messages one-by-one over t rounds. Each message is received perfectly both at the sink node and all other terminals. The communication is initiated by Terminal 1, and the users send messages one by one following the order of their indices. By contrast, the interactive communications model in [14, 15] requires that the users send their messages in parallel, and that those users sending messages in a given parallel communication round utilize the same encoder. Once a user in [14, 15] does not send any rate in any round of this parallel communication, they do not send any rate thereafter. This paper aims to study the difference between these homogenous parallel interactive function computation strategies in [14, 15] and the collocated network model, with a specific focus on the sum rate required, and the tradeoff between the expected sum rate and the expected delay at which the function can be computed. In both cases, we will require the sink to compute the function in a lossless manner.

A substantial theoretical benefit of investigating the collocated network model is that the fundamental lower bound, over all codes, for the required sum rate to compute the function at t rounds is known [12, 16, 18] and can be calculated, at least in principle, and from a practical standpoint, for small problems, using a large series of convex hulls [12]. Bearing this in mind, in §II we review how to apply the results of [12] to extrema functions for uniform distributions, computing these fundamental limits for several small supports $\mathcal{X}$. The next issue we consider is the design of low-complexity encoding schemes to approach these fundamental limits. We will consider simple encoding schemes composed of scalar quantization followed by Huffman codes, deriving optimal such strategies as solutions to a certain dynamic program (DP) in §III, then show that even lower complexity schemes created by drastically limiting the search space of quantizers to optimize over in the DP to a specific family of quantizers can yield comparable performance in §IV.

II. Minimum Sum-rate for Computing Extrema

Formally, a t-message distributed source code consists of block encoding functions $\psi_i : \mathcal{X}_j^n \times \bigotimes_{l=1}^{i-1} \mathcal{M}_l \to \mathcal{M}_i$ for $i \in [t]$ with $j = ((i-1) \bmod m) + 1$, and a decoding function $\phi : \bigotimes_{l=1}^{t} \mathcal{M}_l \to \mathcal{Z}$ enabling lossless computation of the function. For each t, $R_t = (1/n)\log|\mathcal{M}_t|$ is called the t-th bock coding rate and the associated sum rate is $R_{sum,t} = \sum_{i=1}^{t} R_i$. A convex geometric approach in [12] enables the computation of the minimum sum-rate over all encoding and decoding strategies for any finite and infinite number of rounds t.

This minimum sum rate for all finite t is the solution to the optimization $R_{sum,t} = \min_{p_{U^t|X^m} \in \mathcal{C}} I(X^m; U^t)$, where $\mathcal{C}$ requires the message random variables $U_i, i \le t$ to obey the Markov conditions associated with the information available until round i, as well as the entire collection of them after t rounds, $\boldsymbol{U}^t$ to determine the function $f(X_1, \ldots, X_m)$. To compute this for any t, another quantity, the rate reduction functional is defined as $\rho_t = H(X^m) - R_{sum,t} = \max_{p_{u^t|X^m} \in \mathcal{C}} H(X^m|U^t)$, the reduction in the minimum rate for function computation in comparison to the rate required for source reproduction. The rate reduction function is of interest because the auxiliary random variables U^t appears as conditioned random variables, enabling additional rounds of communication to be linked with convex hulls. To compute the sum rate for losslessly t-round $\arg\max, \max$ computation following [12], we start with an important quantity, that is the rate reduction for zero messages defined as follows

$$\rho_0 = \begin{cases} H(X^m) & \text{if } p_{X^m} \in \mathcal{P} \\ -\infty & \text{o.w.} \end{cases} \tag{1}$$

where $\mathcal{P}$ is a set of distributions such that for any $p_{X^m} \in \mathcal{P}$ the extremization function of interest is deterministically determined without sending any message. i.e. $H(f(X_1, .., X_m)) = 0$. After defining these zero-message distributions, we initialize the zero-message rate reduction function ρ_0. Then, for each $t \in \mathbb{Z}$, for every set of marginal distributions $\{q_{X_j}\}_{j=1, j\neq k}$ for $k = ((t-1)\bmod m) + 1$, we construct the upper boundary of the convex hull of the hypograph of the function $\rho_{t-1}(q_{X^m})$ on the set of all PMFs on alphabet $\mathcal{X}_k$ [12].

Following this process, in order to apply this method to the case of extrema functions of interest in this paper, one simply needs to determine this set of zero-message distributions $\mathcal{P}$. In the following lemma this set $\mathcal{P}$ is determined for the case of computing the $\max$.

Lemma 1. *If for all $i \in [m]$, $\mathcal{X}_i = \mathcal{X}$ is a bounded and discrete support, the set of distributions that result in the zero message for computing the max can be found by*

$$\mathcal{P}_M = \Big\{ p_{X^m} \Big| \forall a \in \mathcal{X}, \forall i \in [m], p_i(a) = 1,\ p_j(b) = 0, \\ \forall b > a \in \mathcal{X},\ \textit{s.t.} \sum_{x \in \mathcal{X}} p_i(x) = 1 \Big\}.$$

Shifting attention to the $\arg\max$ extremization function, to find the set of zero-message distributions $\mathcal{P}_A$ for the computing $\arg\max$ we first define the following.

Definition 1. If $S_1, S_2 \subset \mathcal{X}$, we say S_2 dominates S_1, and write it as $S_1 < S_2$ if and only if $|S_1 \cap S_2| = 0$ or 1, and $\max(S_1) \le \min(S_2)$.

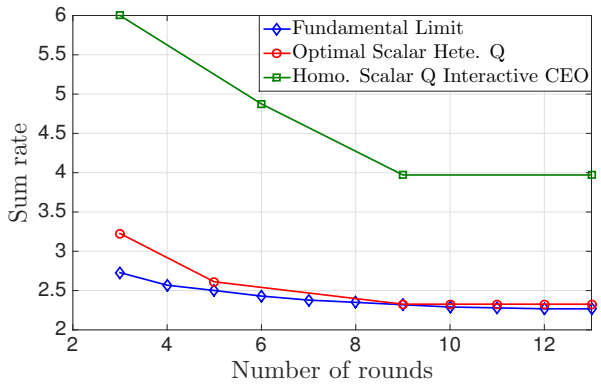

Fig. 2: Number of users= 3, and size of the support set $L = 4$

In computing the $\arg\max$, a tie happens when two or more users attain the maximum value, and in this case, the CEO can choose any user that achieves the maximum. Bearing this in mind to handle the case of a singleton overlap between S_1 and S_2, we see that a set of distributions will be in $\mathcal{P}_A$.

Lemma 2. *If for all $i \in [m]$, $\mathcal{X}_i = \mathcal{X}$ is a bounded and discrete support, the set of distributions that result in the zero message for computing the* $\arg\max$ *is*

$$\mathcal{P}_A = \Big\{ p_{X^m} \Big| \forall i \in [m], \forall S_i \subset \mathcal{X}, \forall j \in [m] \setminus \{i\}, \\ S_j < S_i \subset \mathcal{X}, \sum_{x \in S_i} p_i(x) = 1, \sum_{x \in S_j} p_j(x) = 1 \Big\}.$$

Proof. The set of distributions $\mathcal{P}_A$ for the $\arg\max$ function, will be those which are supported on sets for which there is a user that dominates the other users. The user with a dominating support set can be returned as the $\arg\max$ because the sink does not need to know the exact value it attains. □

Utilizing these results, in Figure 2 the fundamental lower bound on the sum-rate for computing the $\arg\max$ is plotted, via this convex geometric approach for the case of 3 users and a support size of 4. The sum-rates required by two practical encoding schemes are also displayed in this plot. Both of these practical coding schemes are built by cascading scalar quantization with Huffman coding, with the scalar quantizer being utilized being selected through the solution to a dynamic program. The line labeled "Optimal Scalar Hete. Q" utilizes the dynamic program based on the collocated network communication model (§I-A) to be described in §III to minimize the sum rate subject to a deterministic delay bound t, while the line labelled "Homo. Scalar Q Interactive CEO" utilizes a DP to find the best strategy over all parallel homogenous communications strategies from [14] achieving the same bounded delay. The collocated network model of series message transmission in the interactive strategy is shown to enable substantial rate savings relative to parallel message transmission, and additionally optimized encoding based on scalar quantization is shown to achieve sum-rate very close to the fundamental limit. Motivated by this observation, in the next section we describe the dynamic programming oriented approach utilized to create optimal such collocated network scalar quantization strategies, with an additional focus on not just minimizing sum-rate, but also expected delay.

III. Optimal Interactive Scalar Quantization

An especially simple collection of achievable schemes for interactive function computation in collocated networks utilizes scalar quantizers followed by Huffman coding. Fixing notation, in these schemes, the quantization function $Q_t : \mathcal{X}_i \times \bigotimes_{l=1}^{t-1} \mathcal{Q}_l \to \mathcal{Q}_t$ for user $i = ((t-1) \bmod m + 1)$, partitions the support set L^t into k_t intervals, where k_t is the number of bins, and the bin sizes are $q_t = (n_1^t, ..., n_{k_t}^t)$, such that $\sum_{i=1}^{k_t} n_i^t = L^t$. In another words, the first n_1^t elements of L^t are mapped to index 1, and the next n_2^t elements of L^t are mapped to index 2, and so on. We define $\mathcal{L}_i^t, i = 1, .., k_t$ to be the interval associated with each bin.

The quantization function Q_t then maps the value drawn from the support to the index of the interval associated with it. After a sufficient number of rounds, these quantization functions give the CEO enough information to compute the function, via the map $\psi : \bigotimes_{l=1}^{T} \mathcal{Q}_l \to \mathcal{Z}$. The interval associated with the quantized message at round t is $\mathcal{L}^t = \mathcal{L}^t_{Q_t(X_i)}$, the interval in which X_t lies in. A key realization is that receiving this quantized message enables the support of the maximum to shrink because the maximum can be no lower than the interval that the most recent message has indicated that the previous source is in. Based on this information, only those users whose values are greater than the lower end of this interval should continue participating in the interaction.

The optimal quantizers to utilize in this interactive scheme, in terms of minimizing expected sum-rate, delay, or a weighted combination of the pair, can be selected through a dynamic programming framework. Fixing notation again, at each round t, the users keep track of the public knowledge about the state of the system via $s^t = (N^t, L^t, \mathcal{N}^t, \mathcal{A}^t)$, wherein L^t is remaining possible support for the maximum at time t. Here, $\mathcal{A}_i^t$ is the most recent public interval that user i was announced to be in at time t. N^t is the number of active users at time t – those users which could still yield the $\max$ based on the collected knowledge from the previous $t-1$ messages, i.e. those users j with $A_j \cap L^t \neq \emptyset$. At round $t+1$, let the next user sending its message be $X_j \in \mathcal{N}^t$. If $x_j \notin L^t$, then this user simply sends no message and incurs no rate penalty in doing so, decreasing N^t accordingly. Otherwise, if $x_j \in L^t$, X_j utilizes a k_{t+1}-bin quantizer $(n_1^{t+1}, ..., n_{k_{t+1}}^{t+1})$ to quantize his message, then Huffman encodes the result $Q_{t+1}(X_j)$. After overhearing these messages, the state of the system (i.e. the support and number of active users, etc) are updated at each user according to

$$\mathcal{N}^{t+1} = \begin{cases} \mathcal{N}^t \setminus \{j\} & x_j \notin L^t \\ \{i \in \mathcal{N}^t \big| |\mathcal{A}_i^t \cup \mathcal{L}^{t+1}| > 1\} & x_j \in L^t \end{cases} \quad (2)$$

$$\mathcal{A}_i^{t+1} = \begin{cases} \mathcal{L}^{t+1} & i = j, X_j \in L^t \\ \mathcal{A}_i^t & \text{otherwise} \end{cases}, \quad N^{t+1} = |\mathcal{N}^{t+1}| \quad (3)$$

$$L^{t+1} = \begin{cases} L^t \setminus \{\mathcal{L}_1^{t+1} \cup ... \cup \mathcal{L}_{Q_{t+1}-1}^{t+1}\} & x_j \in L^t \\ L^t & x_j \notin L^t \end{cases} \quad (4)$$

As it is shown in (3), (4), the number of active users N^{t+1} and the cardinality of the support set is non increasing. Thus after X_j transmits, the updated support set will be all the elements which are greater or equal to the elements in $\mathcal{L}^{t+1}_{Q_{t+1}(X_j)}$. If it becomes clear from the public interval $\mathcal{A}^t_i$ and L^{t+1} that user i can not be the maximum, user i ceases to communicate, and no longer participates in any upcoming rounds.

If at round t the state of the system is $s^t = (N^t, L^t, \mathcal{N}^t, \mathcal{A}^t)$, then the next online user X_j selects its quantization function Q_t such that the combination $C(s^t) =$

$$\min_{Q_t} \left\{(1-\lambda)R(q_t, s^t)+\lambda D(q_t, s^t)+\mathbb{E}(C(s^{t+1})|s^t, Q_t)\right\}, \quad (5)$$

of its rate and the delay and the expected cost to go is minimized. Let $\mathcal{S}^*_A$ be the set of states of the system for which the $\arg\max$ function can be deterministically computed, then rate $R(q_t, s^t) = H(Q_t(X_j))$, and the delay $D(q_t, s^t) = 1$ for $s^t \notin \mathcal{S}^*_A$. The dynamic program terminates, $s^t \in \mathcal{S}^*_A$, when there is only one active user remaining, or the support set shrinks to only one element. Hence, If $s^t \in \mathcal{S}^*_A$, the CEO can compute f and the interaction is over. Performing some basic calculations [19], the expected cost to go can be written as shown in the following proposition.

Proposition 1. *Suppose at the time t, a k_t-bin quantizer Q_t with the bin sizes $q_t = (n^t_1, ..., n^t_{k_t})$ is chosen by X_j, then the expected cost to go, which consists of the expected rate, and expected delay can be obtained by the following DP.*

$$\mathbb{E}(C(s^{t+1})|s^t, Q_t) = (1 - \frac{L^t}{L})C(s')+ \quad (6)$$
$$\sum_{j=1}^{k_t} \frac{n^t_j}{L^t} \sum_{k=0}^{N^t-1} p_{k,j} C\left(N^t - k, L^t - \sum_{i=1}^{j-1} n_i, \mathcal{N}^t \backslash \mathcal{B}_k, \mathcal{A}^t \backslash \mathcal{C}_k\right)$$

where $s' = (N^t - 1, L^t, \mathcal{N}^t \backslash \{j\}, \mathcal{A}^t \backslash \{\mathcal{A}^t_j\})$ and $\mathcal{B}_k = \{i \in \mathcal{N}^t \big| |\mathcal{A}^t_i \cup \mathcal{L}^{t+1}| \leq 1\}$ such that $|\mathcal{B}_k| = k$, and $\mathcal{C}_k = \{\mathcal{A}^t_i | i \in \mathcal{B}_k\}$ and $p_{k,j} = \binom{N^t-1}{k}(1 - \frac{(L^t - n^t_j + 1)}{L})^k (\frac{L^t - n^t_j + 1}{L})^{N^t - 1 - k}$.

The set of terminating states is smaller when the CEO wishes to determine the $\max$ because for computing the $\arg\max$, communication can stop when a single user is left, regardless of the set of remaining possible values.

$$\mathcal{S}^*_A = \{s^t | N^t = 1 \text{ or } L^t = 1 \text{ or } |\mathcal{N}^t| = 1 \text{ or } |\mathcal{A}^t| = 1\} \quad (7)$$
$$\mathcal{S}^*_M = \{s^t |\, L^t = 1\}. \quad (8)$$

To trace out the rate-delay trade-offs of interactive scalar quantization, the dynamic program of (5) is solved for multiple values of λ and the corresponding rate-delay values are plotted parametrically. Fig. 3 in the next section plots this optimal rate-delay tradeoff for $m = 3$ users each with a support size of 4. Additionally, Fig. 4 displays the minimum sum-rate (setting $\lambda = 0$ in the DP) obtained by this optimal dynamic program as a function of the support size for two users. Motivated by the somewhat rapid growth of the search space required to solve the optimal dynamic program (5), these plots also include reduced complexity programs restricted to smaller quantizer search spaces as we shall describe in the next section.

IV. Reduced Complexity Quantization Strategies

In this section, we consider several simple quantization strategies with substantially lower complexity than solving for the optimal scalar quantizer under the optimal DP introduced in §III. Via both analytical and numerical arguments we show that these reduced complexity quantization schemes yield sum-rates and rate delay tradeoffs close to those set out by the optimal DP. Additionally, the scaling law of the expected sum rate is explored for computing $\max$ using interactive scalar quantization.

We begin by showing that an especially simple interactive scalar quantization scheme can yield a expected sum-rate which, for a fixed support size $|\mathcal{X}| = L$, is constant in the number of users m.

Theorem 1. *For a fixed $|\mathcal{X}| = L$ the expected sum rate required in an interactive scheme to compute the max is $\Theta(1)$ in the number of users m.*

Proof. See [19]. □

The simple scheme utilized to prove Thm. 1 is already sufficient to not require a number of bits which grows in the number of users for a fixed support size, however, a more carefully selected family of quantizers can save more rate. In particular consider the family of quantizers

$$\mathcal{U}(L) = \{\mathbf{1}_{k-1} \oplus (L - k + 1) : k = 2, ..., L\}, \quad (9)$$

where $\mathbf{1}_{k-1}$ is $(k-1)$-tuple of all ones and $\oplus$ denotes the tuple concatenation. This family of quantizers is beneficial for the sum-rate because it can rapidly shrinks the support set, enabling subsequent iterations to require less rate. Beyond this obvious intuition, this family of quantizers can be proven meritorious in the present, uniformly distributed observations, context in the sense that selecting the quantizers from this family yields a lower expected rate than any of their permutations for the case $m = 2$.

Lemma 3. *If there are $m = 2$ users, and the aim is to compute the $\arg\max$ or $\max$ of the users' values, the k-level quantizer $\mathcal{U}(L) = \{\mathbf{1}_{k-1} \oplus (L - k + 1) : k = 2, ..., L\}$, has lower cost than any other quantizer obtained by permutation $\mathcal{U}_s(L) = \{\mathbf{1}_s \oplus (L - k + 1) \oplus \mathbf{1}_{k-s+1}\}$.*

Proof. See [19]. □

While Lemma 3 demonstrates that the family of quantizers (9) is efficacious for the case $m = 2$, the extension of its optimality over quantizers from $\mathcal{U}_s$ to $m > 2$ does not hold. This can be observed by comparing the expected-rate vs. expected delay tradeoff performance of the quantizers labeled $\mathcal{L}$ and $\mathcal{L}_{k-1}$ in Fig. 3, which plots the case $m = 3$ and $|\mathcal{X}| = 4$. Note that the lower bound over all scalar quantization schemes provided by solving the dynamic program (5) is labelled "optimal" in Fig. 3. Extending the family $\mathcal{U}$ to $\mathcal{L} = \mathcal{U} \cup \mathcal{U}_{k-1}$ results in reasonable rate-delay trade-offs as compared to the fundamental limit (9) for these parameters. Also note in Fig. 3 that as we get to the larger delays, two-bin quantizers become

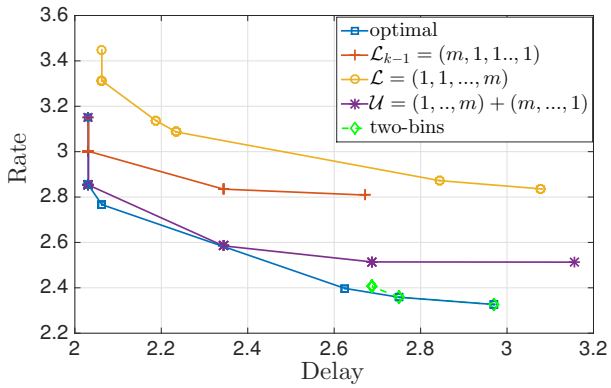

Fig. 3: Rate-delay tradeoff for different families of quantizers for $m = 3$ users and $|\mathcal{X}| = 4$.

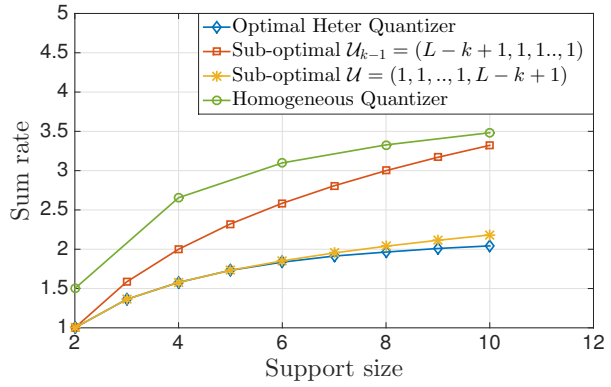

Fig. 4: Sum rate vs. the support size $|\mathcal{X}|$ for $m = 2$ users.

optimal. This result is consistent with the result in [14], that single threshold binary quantizers appears to be optimal in terms of rate. All in all, the proposed sets of quantizers described in the this section, are shown to be pretty close the the optimal rate-delay trade-off. There is about 3 to 3.5 bits ($1 - 1.16$ bit/user) rate savings versus the case that users send their observations.

In Fig. 4 the minimum sum rate obtained by the optimized interactive scalar quantization is plotted for a fixed number of users, and for a varying support size, by setting $\lambda = 0$ in (5). Searching over the quantizer family $\mathcal{U}$ that is described in (9), not only substantially reduces the size of the search space, but also it approaches the optimal expected sum-rate. It is clear from this plot that the minor expense of increasing the complexity to that associated with choosing heterogenous quantizers from $\mathcal{U}$ provides most of the rate improvement over the homogenous scalar quantizers in [14] that any heterogenous one-at-a-time optimized communications strategies can attain. This result together with the near sum-rate of heterogenous scalar quantizers to heterogenous vector quantizers demonstrated in Fig. 2 argues that the simple low complexity scalar quantization selection strategies introduced in this section provide performance very close to the best attainable, at least for the small numbers of users m and support size $|\mathcal{X}| = L$ that we have investigated.

V. Conclusion

This paper calculated the fundamental sum-rate and rate-delay tradeoff limits for calculating extrema over a collocated network. A family of practical designs based on multiple rounds of Huffman coded scalar quantization optimized through a dynamic program were shown to yield performance comparable to fundamental limits over all designs. The search space of the dynamic program was then reduced to several quantization families which were then shown to still yield performance close to the fundamental limit.

References

[1] L. R. Varshney and K. C. Ratakonda, "An information-theoretic view of cloud workloads," in *Cloud Engineering (IC2E), 2014 IEEE International Conference on*. IEEE, 2014, pp. 466–471.

[2] S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, "Coded mapreduce," *arXiv preprint arXiv:1512.01625*, 2015.

[3] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding up distributed machine learning using codes," *arXiv preprint arXiv:1512.02673*, 2015.

[4] J. Korner and K. Marton, "How to encode the modulo-two sum of binary sources (corresp.)," *IEEE Trans. Inform. Theory*, vol. 25, no. 2, pp. 219–221, 1979.

[5] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, vol. 19, no. 4, pp. 471–480, Jul. 1973.

[6] A. Orlitsky and J. R. Roche, "Coding for computing," *IEEE Trans. Inform. Theory*, vol. 47, no. 3, pp. 903–917, March 2001.

[7] M. Sefidgaran and A. Tchamkerten, "Distributed function computation over a rooted directed tree," submitted to *IEEE Trans. Inf. Theory*. [Online]. Available: {http://arxiv.org/pdf/1312.3631v1.pdf}

[8] V. Doshi, D. Shah, M. Medard, and S. Jaggi, "Functional compression through graph coloring," *IEEE Trans. Inform. Theory*, vol. 56, no. 8, pp. 3901–3917, August 2010.

[9] V. Misra, V. K. Goyal, and L. R. Varshney, "Distributed scalar quantization for computing: high-resolution analysis and extensions," *IEEE Trans. Inform. Theory*, vol. 57, no. 8, pp. 5298–5325, August 2011.

[10] J. Z. Sun and V. K. Goyal, "Intersensor collaboration in distributed quantization networks," *IEEE Trans. Comm.*, vol. 61, no. 9, pp. 3931–3942, 2013.

[11] N. Ma and P. Ishwar, "Interaction strictly improves the wyner-ziv rate-distortion function," in *2010 IEEE Int. Symp. Inf. Theory (ISIT)*, 2010, pp. 61–65.

[12] N. Ma, P. Ishwar, and P. Gupta, "Interactive source coding for function computation in collocated networks," *IEEE Trans. Inf. Theory*, vol. 58, no. 7, pp. 4289–4305, 2012.

[13] T. C. Gulcu and A. Barg, "Interactive function computation via polar coding," in *52nd Allerton Conf. Comm., Control, Comp.*, 2014, pp. 820–827.

[14] Bradford D. Boyle, Jie Ren, John MacLaren Walsh, and Steven Weber, "Interactive Scalar Quantization for Distributed Resource Allocation," *IEEE Trans. Signal Process.*, To Appear. Accepted for publication September 16, 2015. [Online]. Available: http://dx.doi.org/10.1109/TSP.2015.2483479

[15] Jie Ren, Bradford Boyle, Gwanmo Ku, Steven Weber, John MacLaren Walsh, "Overhead Performance Tradeoffs – A Resource Allocation Perspective," *IEEE Trans. Inf. Theory*, To Appear. Accepted for publication January 16, 2016. [Online]. Available: http://arxiv.org/abs/1408.3661

[16] N. Ma and P. Ishwar, "Some results on distributed source coding for interactive function computation," *IEEE Trans. Inform. Theory*, vol. 57, no. 9, pp. 6180–6195, September 2011.

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

[18] A. H. Kaspi, "Two-way source coding with a fidelity criterion," *IEEE Trans. Inform. Theory*, vol. 31, no. 6, p. 735, November 1985.

[19] S. Torabi, J. Ren, and J. M. Walsh, "Practical interactive scheme for extremum computation in distributed networks." [Online]. Available: http://www.ece.drexel.edu/walsh/SQISIT16.pdf