On the Multi-Round Group-Broadcast Scheme in Collocated Gaussian Networks

Nezihe Merve Gürel
LINX, École Polytechnique Fédérale de Lausanne

June 14, 2014

Abstract: Multi-round group-broadcast scheme to efficiently deduce the desired function and reduce redundancy in collocated Gaussian networks is studied. We focus on a particular type-threshold function, which is a subclass of symmetric functions. We numerically provide the optimum multi-round group-broadcast scheme for given parameters. An existing solution to non-linear discrete optimization problems is analyzed. An algorithm to evaluate the broadcast scheme based on numerical observations is presented.

I. INTRODUCTION

Wireless sensor networks is a distributed collection of small sensor nodes that perform a collaborative measurement process for particular applications. The application area is numerous such as forest fire detection, traffic congestion, electromagnetic level monitoring and many others. The sensor nodes are specifically designed to measure small amounts of data, and equipped with several devices, mainly, microcontrollers, sensors, battery, communication modules and Secure Digital Memory card. The sensor network is autonomous since it integrates different functional capabilities such as sensing and monitoring the event due to processor and memory on the sensor nodes.

The network structure is composed of sensor nodes and a fusion center. These elements in the network occupy the following roles: the sensor nodes take measurement from the environment and convey these measurement to fusion center, then fusion center detects the pre-defined function of the measurements. A focus on increasing the lifetimes of the sensor nodes through power generation and management also becomes important in recent years. In [1], the authors showed that the multi-round group-broadcast scheme efficiently computes the desired function in fully connected Gaussian networks with equal channel gains so-called collocated Gaussian networks.

The main objective of this work is to evaluate multi-round group-broadcast scheme to increase the lifetimes of the sensor nodes in the network. We argue in this paper that how the broadcasting scheme changes to reduce redundancy with the network parameters. Specifically, we consider collocated Gaussian networks. The paper is organized as follows. In next section, we formulate our problem for our network model and find the mathematical expression of the entropy to be reduced. In Section III, we propose a new generating algorithm for broadcast-scheme and survey another existing algorithm. In Section IV, we give numerical and theoretical results concerning the group-broadcast scheme. An observation-based algorithm is also presented. In next Section, discrete Lagrange multipliers as a solution to local minimization of our problem is analyzed. Finally, Section VI concludes the report.

II. PROBLEM FORMULATION

Throughout the report, we denote

$$H_{bino}(c,p) = -\sum_{i=0}^{c} \binom{c}{i} p^i (1-p)^i \log_2 \binom{c}{i} p^i (1-p)^i$$

that is, entropy of *binomial* random variable with parameters c and p. Let $\mathbf{1}_{\{.\}}$ denote the indicator function of an event.

We consider a network consisting of n sensor nodes and a fusion center to detect the so-called desired function of the measurements taken by the sensor nodes. We assume that each sensor node has a joint source-channel coder and sources are discrete, independent and *Bernoulli* distributed random variables with parameter p, and labelled as S_i, where $i \in \{1, 2, ..., n\}$. The fusion center estimates the desired function of the sources. In this work, we focus on the class of type-threshold functions. To model the desired function, let us provide the definitions of type-threshold functions and type,frequency histogram function[1].

Definition 1. *For any integer $q \geq 1$, Type, Frequency histogram function of the sources is defined as the vector $[b_1 b_2 ... b_{q-1}]$, where each term b_l is given by the equation*

$$b_l = \sum_{i=1}^{n} \mathbf{1}_{\{S_i = l\}}$$

Definition 2. *Let us define the non-negative integer vector $[\theta_1 \theta_2 ... \theta_{q-1}]$ as threshold vector, $\bar{b}_l = \min\{\theta_l, b_l\}$*

[1]We refer to Definition 5 and 6 in [1] for formal definitions

as clipped frequency of l and f_n as a symmetric function, where $f_n : [0 : q-1]^n \rightarrow \Lambda$ for a finite alphabet Λ, then f_n is called type-threshold function if it satisfies the followings

$$f_n(s_1, s_2, ..., s_n) = g(\bar{b}_0, \bar{b}_1, ..., \bar{b}_{q-1})$$

where g is a function such that $g : [0 : \theta_1] \times [0 : \theta_2] \times ... \times [0 : \theta_{q-1}] \rightarrow \Lambda$.

Noticing that the main goal of this work is to evaluate multi-round group-broadcast scheme by reducing the information received at the fusion center to compute the desired function, we utilize both superposition and broadcast properties of the network as in [1]. It is clear from the definition of type-threshold function that clipped frequencies have enough information to evaluate type-threshold function. Thus, throughout this work, we deal with the clipped frequencies rather than type-threshold function.

Notation: Let s_i be the instance of sensor node i, namely realization of the source information S_i, then we call the group of source instances the description and denote it by $U_1^l, U_2^l, ..., U_k^l$ where k is the number of groups.

The broadcast scheme of the network is as follows:

- Activate each group U_i^l in a round once, i.e., set of sensor nodes in that group, and each sensor in that group broadcast its instance S_j,
- Update the frequency accumulated after each group and clip if it exceeds threshold and then terminate the broadcasting

Clearly, each node hears other nodes since we assume that the network is collocated. Clearly, cardinality of each group can be written as composition of n.[2] Thus we denote cardinality of U_i by c_i. Consequently, U_m^l is given by [1]

$$U_m^l = U_{m-1}^l + \sum_{i \in c_m} \mathbf{1}_{\{U_{m-1}^l < \theta_l\} \cap \{S_i = l\}}$$

Recall that the sources are independently drawn from *Bernoulli* distributions with parameter p. Hence, the frequency histogram function has only two components denoted by b_0 and b_1. In this work, we work on the multi-group broadcast scheme for the simplest case of type-threshold function, that is, we assume that $[\theta_0\ \theta_1]$ is given by [0 1]. This assumption implies that the fusion center is only interested in the frequency at b_1. Based on the network model, let us state the mathematical expression for the redundancy in the network. The entropy of the descriptions $U_1^1, U_2^1, ..., U_k^1$ can be evaluated as follows

$$\begin{aligned} H(U_1^1, U_2^1, ..., U_k^1) &= \sum_{i=1}^{k} H(U_i^1 | U_{i-1}^1) \\ &= \sum_{i=1}^{k} P(U_{i-1}^1 = 0) H(U_i^1 | U_{i-1}^1 = 0) \\ &= \sum_{i=1}^{n} (1-p)^{\sum_{j=1}^{i-1} c_j} H_{bino}(c_i, p) \end{aligned} \tag{1}$$

first equality follows from independence of sources, second equality follows from the fact that we assume $\theta_1 = 1$. Thus, any U_i^1 is deterministic given that $U_{i-1}^1 \geq 1$. Last equality follows since U_i^1 is Binomial random variable given that $U_{i-1}^1 = 0$ due to the fact that $p(\{U_{i-1}^1 = 0\}) = p(\{U_1^1 = 0, U_2^1 = 0, ..., U_{i-1}^1 = 0\})$, which in turn implies that $p(\{U_{i-1}^1 = 0\}) = (1-p)^{\sum_{j=1}^{i-1} c_j}$.

To simplify notation, let $H(n, \mathbf{c}, p)$ be equal to the entropy given by descriptions, where $\mathbf{c}$ is denoted by the sequence $c_1, c_2, ..., c_k$[3] for any $1 \leq k \leq n$. Therefore, the main objective of this work is to find the sequence $c_1, c_2, ..., c_k$, where the entropy of the descriptions $H(n, \mathbf{c}, p)$ is minimized. To state the problem, we have the following discrete optimization problem

$$\begin{aligned} &\textit{minimize} \quad \sum_{i=1}^{n} (1-p)^{\sum_{j=1}^{i-1} c_j} H_{bino}(c_i, p) \quad \textit{over} \ \ c_1, ..., c_k \\ &\textit{subject to} \quad 0 \leq c_i \leq n \ \ \textit{for} \ \ i, k \in \{1, 2, ..., n\} \\ &\qquad\qquad\quad c_1 + c_2 + ... + c_k = n \end{aligned} \tag{2}$$

In the next section, we propose an algorithm to generate compositions of n and provide an existing algorithm that generates k composition of n[4].

III. COMPOSITION GENERATING ALGORITHMS

A. First Algorithm

Here we propose an algorithm to generate all compositions of n.

Theorem 1. *The distinct compositions of n+1 can be generated by adding 1 to the first term of all distinct compositions of n and inserting a 1 as a new term to left to the first term of the distinct compositions of n.*

Proof. Consider 2 disjoint subsets A_1 and A_2 of integer compositions of n+1 such that their union is a set of all compositions of n+1 and the first term of the compositions satisfies $c_1 \geq 2$ and $c_1 = 1$ in the sets A_1 and A_2, respectively. The set A_1 includes all possible combinations of the non-negative integer sequences of the form $c_1, c_2, ..., c_k$, where $c_1 \geq 2$ and $c_1 + c_2 + ... + c_k = n+1$. It

[2] In Combinatorics, a composition of an integer n is defined as ordered non-negative sequences that sum to n.

[3] One can also consider the sequence as a vector $[c_1\ c_2\ ...\ c_k]$.

[4] k compositions of n is referred to the compositions of n that have exactly k strictly positive parts.

is easy to see that replacing c_1 with c_1-1 gives a set of all possible non-negative integer sequences that sum to n, that is, integer compositions of n. Similarly, the set A_2 is a collection of all sequences $c_1, c_2, ..., c_k$ that satisfy $c_1 = 1$ and c_1+c_2+...+c_k=n+1. Hence, removing c_1 from those sequences in A_2 gives a set of all possible non-negative integer sequences that sum to n, namely integer compositions of n. Hence, either subtracting 1 from the first terms of the compositions in the set A_1 or removing the first term from the compositions in the set A_2 gives a set of all distinct compositions of n. Similar discussion holds for any fixed index of the compositions in these sets.

□

Corollary 1. *Number of partitions of a positive integer n, where n can be written as a sum of ordered positive integers, is given by* 2^{n-1}.

Proof. The above argument shows that 2 identical sets of distinct compositions of n can be obtained from one set of distinct compositions of n+1. Hence, the number of compositions is given by 2^{n-1} since number of distinct compositions is 1 for $n = 1$. □

Here is a sketch of the algorithm for generating compositions of n:

- initialize composition by 1;
- at each step, update the set of compositions by combining two sets that are generated by adding

1 to the first terms and inserting 1 to the beginning of the compositions in the set of compositions

generated at the previous step. That is, denoting the all compositions of i generated at i^{th} step by

$c_1^{(i)}, c_2^{(i)}, ..., c_k^{(i)}$, the set of compositions of $i+1$ is given by the union of 2 sets of all $c_1^{(i)}+1, c_2^{(i)}, ..., c_k^{(i)}$ and $1, c_1^{(i)}, c_2^{(i)}, ..., c_k^{(i)}$ sequences. Thus, the compositions of $i+1$ is generated at $(i+1)^{th}$ step.

The above algorithm has the time and space complexity $\Theta(n)$ and $\Theta(2^n)$, respectively. The problem is *intractable*.[5] We refer to Fig.1 for an illustration.

B. Second Algorithm, Colex

In [2], the authors develop an algorithm named *Colex* to generate the set of *k non-negative composition* of n denoted by *C(n,k)*, where *C(n,k)* includes all distinct sequences of the form $c_1, c_2, ..., c_k$ for all c_i, $1 \le i \le k \le n$, $0 \le c_i \le n$ and $\sum_{i=1}^{k} c_i = n$.

Definition 3. *For a given k-sequence* $\mathbf{b} = b_1, b_2, ..., b_k$, *the set of compositions of n,* $C^{\mathbf{b}}(n,k)$ *is called* **b**-*bounded if* $0 \le c_i \le b_i$ *for all i,* $1 \le i \le k$.

[5]In Computer Science, the term *intractable* is used for the problems that have non-polynomial complexity, but rather have exponential complexities.

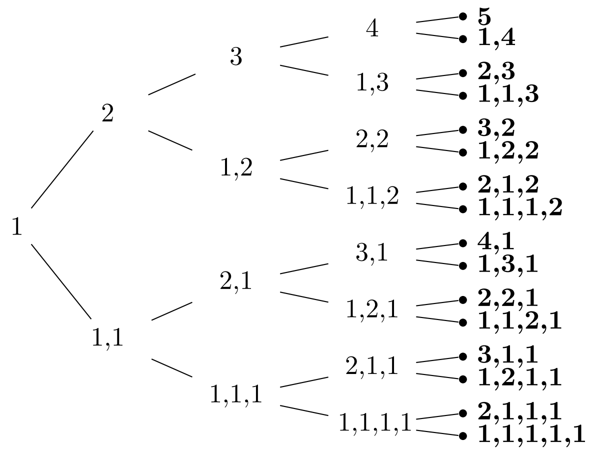

Figure 1: The tree induced by the call of the algorithm for n=5. Each column corresponds to compositions of the integers smaller than or equal to n in the increasing order. Terminal nodes are in bold-face.

The algorithm is based on generating the $C^{\mathbf{b}}(n,k)$ in colexicographic order, that is, at each step, the sequence $c_1, c_2, ..., c_i$ is updated where $c_{i+1}, c_{i+2}, ..., c_k$ remains unchanged.

Definition 4. *An increasable position is the corresponding index i in the sequence* $c_1, c_2, ..., c_k$ *where* $c_1 = c_2 = ... = c_{i-1} = 0$ *and there exists a sequence* **d** $= d_1, d_2, ..., d_k \in C^{\mathbf{b}}(n,k)$ *such that* $c_i < d_i$ *and* $c_j = d_j$ *for all j,* $i+1 \le j \le n$.

First, the sequence $c_1, c_2, ..., c_k$ is initialized to zero vector. Then, all increasable positions of the current sequences are found and corresponding entries to increasable positions are increased by one. The algorithm calls this procedure recursively until the resulting composition is in $C^{\mathbf{b}}(n,k)$.

The Matlab implementation to find restricted integer compositions with fixed number of parts based on the algorithm n,k, minimum and maximum values in the composition as inputs, and outputs $\binom{n-1}{k-1}$ distinct integer compositions of n of length-k.[3] Obviously, minimum number in an positive integer composition is 1. In the implementation, the non-negative compositions in $C^{\mathbf{b}}(n-k,k)$ are generated based on the algorithm in [2] by choosing all entries of **b** as $(n-k)$. In order to avoid the generation of redundant sequences, c_i is replaced with **b** - $\sum_{j=i+1}^{n} c_j$ if index i is the only increasable position at that node.

In the Matlab implementation, all distinct sequences in $C^{\mathbf{b}}(n-k,k)$ are generated, and unity vector of length-k is added to all sequences by leaving integer compositions of n at the output.

In Fig.2, the tree is induced by the call of the algorithm for $C^{\mathbf{3,3,3}}(3,3)$, where $n = 6$ and $k = 3$. Hence the Matlab implementation adds unity vector

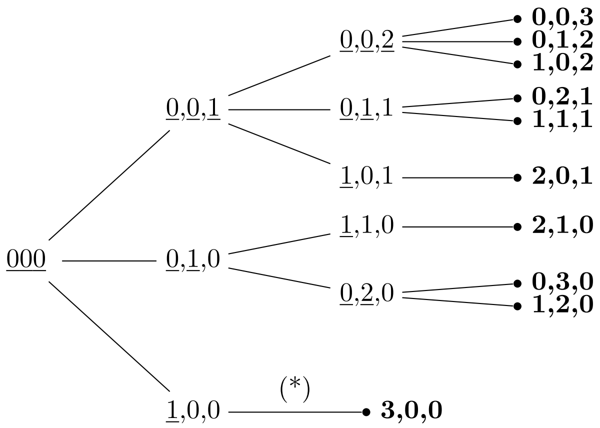

Figure 2: Tree for generation $C^{\mathbf{3,3,3}}(3,3)$. At each branch, the corresponding value of the increasable positions(underlined) is increased by 1 until the sum becomes 3. At (*) labelled branch, c_1 is fixed to $3-\sum_{j=2}^{3} c_j = 3$ where index i is the only increasable position at that node. Terminal nodes are in bold-face.

to those sequences and outputs 3-compositions of 6 given by (1,1,4), (1,2,3), (2,1,3), (1,3,2), (2,2,2), (3,1,2), (3,2,1), (1,4,1), (2,3,1), and (4,1,1).

IV. MULTI-ROUND GROUP-BROADCAST SCHEME

A. *Fixed number of groups*

Consider the case where we put constraints on the parts on the compositions. In this section, we investigate the solution of optimization problem for the simplest case, namely, compositions of n into 2 distinct terms. The motivation is to have insights on the general structure of the *best* compositions where the *best* composition of n is referred to the composition that minimizes the entropy in the network.

Analytically speaking, the entropy $H(n,p)$ can be minimized over $c_1 c_2$ where $c_2 = n - c_1$. Let we denote c_1 and c_2 by c and $n-c$, respectively. Then the entropy is given by:

$$H(n,c,p) = H_{bino}(c,p) + (1-p)^c H_{bino}(n-c,p) \quad (3)$$

where there is a slight change in notation of $H(n,p)$. Thus, it is sufficient to find the unique value of c where difference equation $H(n,c+1,p) - H(n,c,p)$ changes its sign from *negative* to *positive*, namely, the value of c where entropy starts to increase. We have implemented the above algorithm numerically by double-checking it with *Colex* algorithm. In Fig.3, we provide a plot of the value of normalized c that minimizes the entropy for corresponding p and different values of n. The figure has significant importance since it shows that c takes the values $n-1$ and 1 over a significant interval of p. Moreover, the value of c is significantly decreasing as p gets large.

Fig.4 illustrates that the value of p, where the *best* composition of changes from $(n-1,1)$ to $(n-2,2)$ sequence, gets smaller for large n as naturally expected due to the fact that more sensor nodes broadcast for small values of p. Thus the broadcast scheme $(n_{large}-2,2)$ appears to be more efficient than $(n_{small}-1,1)$.

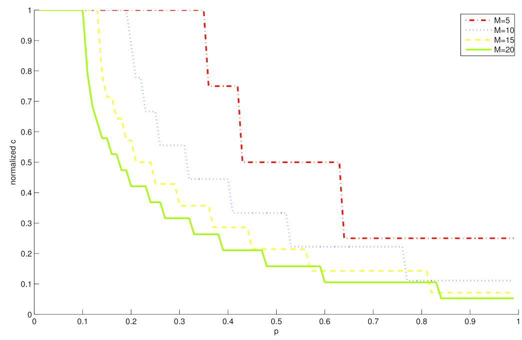

Figure 3: First part of the *best* composition versus p for $k=2$ and $n=5$,(dash-dot curve) $n=10$,(dot curve) $n=15$(dash curve) and $n=20$(solid curve)

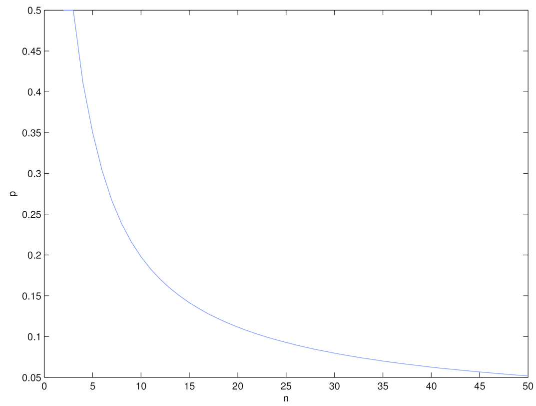

Figure 4: The value of p where the *best* composition changes from $(n-1,1)$ to $(n-2,2)$

B. *Numerical Results on the broadcast scheme*

Theorem 2. *The best composition is given by the sequence* $\underbrace{1,1,...,1}_{n}$ *for any* $p \geq 0.5$ *and* $n \geq 6$.

Lemma 1. *We have*

$$c - \sum_{i=0}^{c} \binom{c}{i} p^i (1-p)^{c-i} \log_2 \binom{c}{i} \geq 2 \quad (4)$$

Proof. Eqn.4 can be re-written as

$$c - 2 \geq \sum_{i=0}^{c} \binom{c}{i} p^i (1-p)^{c-i} \log_2 \binom{c}{i} \quad (5)$$

A moment of thought reveals that $\binom{c}{i}$ is upper bounded by $\binom{c}{\lfloor\frac{c}{2}\rfloor+1}$. Noticing $\binom{c}{i}p^i(1-p)^{c-i}$ is the probability distribution of *Binomial* random variable, the right hand side of Eqn.5 can be written as

$$\log_2\binom{c}{\lfloor\frac{c}{2}\rfloor} \geq \sum_{i=0}^{c}\binom{c}{i}p^i(1-p)^{c-i}\log_2\binom{c}{i} \tag{6}$$

Looking more closely at Eqn.6, we have

$$\sum_{x=\lfloor\frac{c}{2}\rfloor+1}^{c}\log_2(x) \geq \log_2\binom{c}{\lfloor\frac{c}{2}\rfloor+1} \tag{7}$$

Using the concavity of logarithm, we can write

$$\log_2\Bigg(\sum_{x=\lfloor\frac{c}{2}\rfloor+1}^{c} x\Bigg) \geq \sum_{i=0}^{c}\binom{c}{i}p^i(1-p)^{c-i}\log_2\binom{c}{i} \tag{8}$$

Consequently, we have

$$\log_2\Big(\frac{3c^2+2c+1}{8}\Big) \geq \log_2\Big(\sum_{x=\lfloor\frac{c}{2}\rfloor+1}^{c} x\Big) \tag{9}$$

Thus, noticing that $c-2 \geq \log_2\left(\frac{3c^2+2c+1}{8}\right)$ is satisfied for $c=6$, clearly, Eqn.4 holds true for any $c \geq 6$. □

Proof. Proof of theorem

By a straightforward calculation one can show that

$$H_{bino}(c,p) = cH_{bino}(1,p) - \sum_{i=0}^{c}\binom{c}{i}p^i(1-p)^{c-i}\log_2\binom{c}{i} \tag{10}$$

Dividing the both sides of the above equation implies that

$$\frac{H_{bino}(c,p)}{H_{bino}(1,p)} \geq c - \sum_{i=0}^{c}\binom{c}{i}p^i(1-p)^{c-i}\log_2\binom{c}{i} \tag{11}$$

where we use the fact that $H_{bino}(1,p) \leq 1$. By the Lemma, it is easy to see

$$\begin{aligned} H_{bino}(c,p) &\geq 2H_{bino}(1,p) \\ &\geq \frac{1-p^c}{1-p}H_{bino}(1,p) \\ &= \sum_{i=1}^{c}(1-p)^{i-1}H_{bino}(1,p) \end{aligned} \tag{12}$$

for any $p \geq \frac{1}{2}$. The main result of the above equations that entropy of the Binomial random variable is lower bounded by $\sum_{i=1}^{c}(1-p)^{i-1}H_{bino}(1,p)$. Consequently, all remains is to see that any $H_{bino}(c,p)$ term in $H(n,c_1,c_2,...,c_k,p)$ can be lower bounded by $\sum_{i=1}^{c}(1-p)^{i-1}H_{bino}(1,p)$ and yet leaving with the expression of the entropy for the composition of n given by 11...1 sequence for $p \geq 0.5$. We emphasize that above theorem holds for $n \geq 6$, but holds for all $n \geq 1$, which in turn can be shown for all n in $\{1,2,3,4,5\}$. □

Notation: $C(1^i, 2^j)$ represents all the combinations of the partitions of the form $\underbrace{1,1,...,1}_{\text{i}}\underbrace{2,2,...,2}_{\text{j}}$., i.e. $C(1^2,2^2)$ is the set of combinations of the partition (1,1,2,2), that is $\{(1,1,2,2)$, $(1,2,1,2)$, $(1,2,2,1)$, $(2,2,1,1)$, $(2,1,2,1)$, $(2,1,1,2)\}$.

Theorem 3. *Let $c_1,c_2,...,c_k$ be a solution to the minimization problem $H(n,\mathbf{c},p)=\sum_{k=1}^{n}(1-p)^{\sum_{j=1}^{k-1}c_j}H_{bino}(c_k,p)$. Then, at $p=0.5$, the problem has many solutions, that is, for any $0 \leq i \leq \lfloor\frac{n}{2}\rfloor$ and $p=0.5$, any $c_1,c_2,...,c_k$ in $C(1^{n-2i},2^i)$ minimizes the entropy $H(n,\mathbf{c},p)$.*

Proof. Recall from Theorem 2 that $\underbrace{1,1,...,1}_{\text{n}}$ sequence is the solution of the optimization problem for $p \geq 0.5$. Thus, it remains to prove that the entropy for the compositions in $C(1^{n-2i},2^i)$ is equal to the entropy $\sum_{i=1}^{n}\frac{1}{2}^{i-1}H_{bino}(1)$ given by $\underbrace{1,1,...,1}_{\text{n}}$ sequence, where $1 \leq i \leq \lfloor\frac{n}{2}\rfloor$. It is easy to see that $H_{bino}(2) = H_{bino}(1)+\frac{1}{2}H_{bino}(1)$. and for any index $1 \leq j \leq n$, the entropy given by $\underbrace{1,1,...,1}_{\text{j-1}}\underbrace{2}_{\text{j}}\underbrace{11...1}_{\text{n-j}}$ sequence is as follows

$$\begin{aligned} &H_{bino}(1) + \frac{1}{2}H_{bino}(1) + ... + \frac{1}{2^{j-2}} + \frac{1}{2^{j-1}}H_{bino}(2) \\ &+ \frac{1}{2^{j+1}}H_{bino}(1) + ... + \frac{1}{2^{n-1}}H_{bino}(1) \end{aligned} \tag{13}$$

Using the equality of $H_{bino}(2)$ and $H_{bino}(1) + \frac{1}{2}H_{bino}(1)$, clearly, replacing $H_{bino}(2)$ term with $H_{bino}(1)+\frac{1}{2}H_{bino}(1)$ does not change the value of the entropy given by $\underbrace{1,1,...,1}_{\text{j-1}}\underbrace{2}_{\text{j}}\underbrace{1,1,...,1}_{\text{n-j}}$ sequence. Consequently, we have

$$\begin{aligned} H(n,\frac{1}{2}) = &H_{bino}(1,\frac{1}{2}) + \frac{1}{2}H_{bino}(1,\frac{1}{2}) + ... \\ &+ \frac{1}{2^{j-2}}H_{bino}(1,\frac{1}{2}) + \frac{1}{2^{j-1}}H_{bino}(1,\frac{1}{2}) \\ &+ \frac{1}{2^{j}}H_{bino}(1,\frac{1}{2}) + \frac{1}{2^{j+1}}H_{bino}(1,H_{bino}(1,\frac{1}{2})) \\ &+ ... + \frac{1}{2^{n-1}}H_{bino}(1,H_{bino}(1,\frac{1}{2})) \end{aligned} \tag{14}$$

Hence, it is clear from the Eqn.13 that Eqn.14 is equal to the entropy given by $\underbrace{1,1,...,1}_{\text{n}}$ sequence. The same discussion holds true for any number of index $0 \leq i \leq \lfloor \frac{n}{2} \rfloor$.

□

Using the composition generating algorithm we proposed, we investigate the structure of the *best* composition for different values of p. Remarkably, there is a repetitive pattern between the parts of the *best* composition. For example, the *best* composition is given by the sequence $c, c, .., c, n-(k-1)c$ in most of the cases. In addition, the value of c decreases as p gets larger, and becomes 1 for $p \geq 0.5$, which is proved in Thm.4. Motivated by the repetitive pattern of the group sizes and relation between group sizes and p, we expect that $\lceil \frac{1}{p} \rceil$ might be preferable for the choice of c. This is because of the fact that to reach the threshold value is much easier task for larger values of p. Thus smaller number of sensors in a group can be enough for the fusion center to compute the desired function. Fig.5 illustrates the Hamming distance between the *best* composition and $\lceil \frac{1}{p} \rceil, \lceil \frac{1}{p} \rceil, ..., \lceil \frac{1}{p} \rceil, n-(J-1)\lceil \frac{1}{p} \rceil$, where J is the number of parts of the composition, and it is easy to see that $J = \lfloor \frac{n}{\lceil \frac{1}{p} \rceil} \rfloor + 1$. It is worth to remark that we fix the sequence to $\underbrace{1, 1, ..., 1}_{\text{n}}$ for $p \geq 0.5$. The Hamming distance is the largest around $0.2 \leq p \leq 0.4$ for $n \leq 20$. As remarked, there is a quick change in c around those values of p. However, the values of the parts are limited to $\{3, 4, 5\}$. Consequently, we expect the Hamming distance will increase for larger values of n around those probabilities. In this regard, we compare the entropies given by two different approaches. Fig.6 shows the difference between the entropies. The difference is large around small values of p since $q = 1-p$ term gets larger and $H(n, \mathbf{c}, p)$ increases more significantly for small values of p.

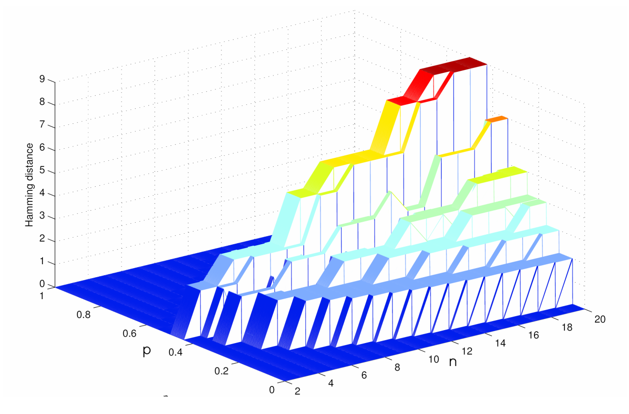

Figure 5: The *Hamming* distance between the best composition and the estimated composition

C. An efficient algorithm as an approximate solution to the problem

In the present section, we propose a composition

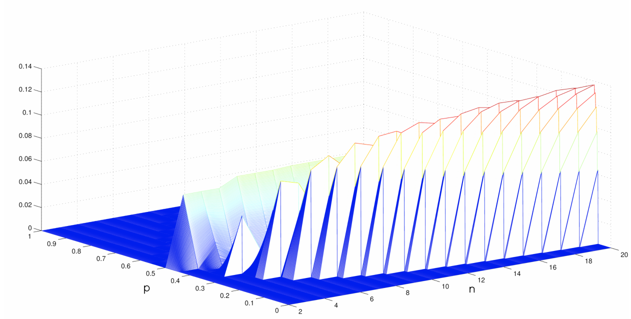

Figure 6: The error in the entropy given by $(np, np...)$ sequence for $n \leq 20$ and all values of p

generating algorithm that finds approximate solutions for the minimization problem. The algorithm has the remarkable property that search space size is decreased from 2^n to n+1, that is, complexity of the algorithm is reduced from $\Theta(2^n)$ to $\Theta(n)$. The relation between the compositions n and n+1 that minimize the entropy function gives interesting insights, and in particular tells us how the *best* composition of n+1 can be generated by knowing the *best* composition of n.

The sketch of the generating procedure is as follows:

- initialize the sequence by 1,
- at each step, find the composition that minimizes the entropy over all possible sequences whose Hamming distance to the sequence $0, c_1, c_2, ..., c_k 0$ is 1, where $c_1, c_2, ..., c_k$ is the composition of n stored in an array, in the previous step.

For example, let the *best* composition of n be given by $c, c, .., c, (n-(k-1)c)$ sequence, then the *best* composition of n+1 is given by one of the sequences that has Hamming distance 1 to the *best* composition of n depending on the value of p. Using the proposed algorithm, we have plotted in Fig.7 the difference between the entropies generated by the algorithm and the actual solution to optimization problem for $n \leq 20$.

The numerical evidence in Fig.7 suggests that the optimization problem can be approximately solved by decreasing the search space size by a maximum penalty of 0.15 in the entropy. Figure illustrates that the difference is decreasing for large number of sensor nodes. This is because the estimated composition by the call of the proposed algorithm is closer to the *best* composition for large n, since number of iterations increases.Thus the algorithm estimates the composition that minimizes $H(n, \mathbf{c}, p)$ more accurately.

It is worth emphasizing that after adding *Colex* algorithm to the approximate solution, we numerically find the exact solution to the optimization problem for $n \leq 20$. The idea here is based on that the approximate solution given by the above algorithm is a solution around the global minima. In other words, the number of parts of the approximate composition is numerically found to be the same with the number of parts of the *best* composition. Thus, *Colex* algorithm re-evaluates the entropy of k compositions of n, where k is the number of parts evaluated by the above algorithm. The collection of two algorithm, yet not proved, finds the global minima of the entropy over compositions of n, where $n \leq 20$ and the complexity of the overall algorithm is given by $\Theta(\binom{n}{k})$.

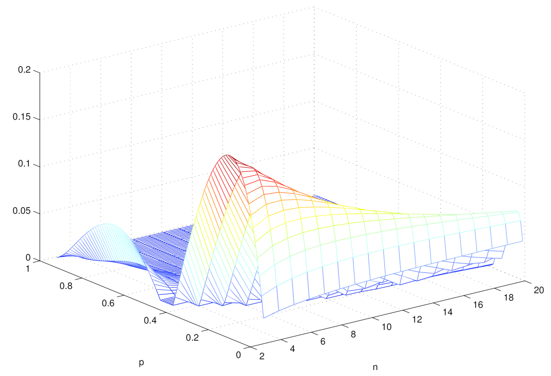

Figure 7: The error in the estimated entropy by the call of the algorithm for $n \leq 20$ and all values of p

V. AS A LOCAL MINIMIZATION METHOD IN DISCRETE SPACE: DISCRETE LAGRANGE MULTIPLIERS FOR NON-LINEAR DISCRETE OPTIMIZATION

Discrete Lagrange Multipliers for Non-linear Discrete Optimization

The theory of discrete Lagrange multipliers for discrete constrained optimization problems have been studied in [4]. The focus of the present section is on providing an algorithm on the local search methods of *nonlinear, nonconvex* optimization problem obtained in [4].

A general *nonlinear, nonconvex, discrete constrained minimization problem* is formulated as follows[4]

minimize *f(x)*

subject to *g(x)* $\leq$ *0 h(x)=0* where $x = (x_1, x_2, ..., x_n)$ is a vector of discrete variables.

Applying the above formulation to the present entropy minimization problem, we can reformulate the problem as follows

$f(x) = \sum_{i=1}^{n}(1-p)^{\sum_{j=1}^{i-1} x_j} H_{bino}(x_i)$ for n ≥ 2

$g(x) = -x \leq 0$

$h(x) = x_1 + x_2 + ... + x_n - n = 0$

To apply the search algorithm in [4], we introduce the concepts of neighbourhood and discrete constraint local minima.

Definition 5. *Neighbourhood of a point x in space X, which is denoted by N(x), is a set of points x' $\in$ X such that x $\notin$ N(x) and that x' $\in$ N(x) $\iff$ x $\in$ N(x').*

As remarked in [1], the choice of neighbourhoods is provided by user and the local search algorithm is valid for any kind of neighbourhood choice.

Definition 6. *A feasible point x is a discrete constraint local minimum if it satisfies f(x') $\geq$ f(x) for all feasible points x' in the neighbourhood of x.*

Similar to the **method of Lagrange multipliers** in continuous space, a new strategy for finding local extremum points for discrete space is proposed in [4].

To understand how the theory of Lagrange multipliers is adapted to discrete optimization problems in [4], let us introduce discrete Lagrangian function and maximum potential drop.

Definition 7. *Discrete Lagrangian function is defined to be:*

$$L_d(x, \lambda, \mu) = f(x) + \lambda H(h(x)) + \sum_{i=1}^{n} \mu_i H(max(0, g_i(x)))$$

where H(x) can be any continuous function such that H(x) = 0 for x = 0, and it should be either strictly positive or strictly negative for non-zero values of x.

It is worth to remark that *maximum* operator transforms inequality constraints into equality constraints and μ is a vector of Lagrange multipliers that is applied to transformed equality constraints.

Definition 8. *In discrete space, direction of maximum potential drop(DMPD) for $L_d(x, \lambda, \mu)$ at a point x and fixed λ and μ is defined to be*

$$\Delta_x L_d(x, \lambda, \mu) = (y_1 - x_1, y_2 - x_2, ..., y_n - x_n)$$

where $y \in N(x) \cup \{x\}$ and $L_d(y, \lambda, \mu) = \min_{x' \in N(x) \cup \{x\}} L_d(x', \lambda, \mu)$.

The main theorem in [4] is that the set of discrete saddle points is equal to the set of all solutions that satisfies $\Delta_x L_d(x, \lambda, \mu) = 0$, $h(x) = 0$ and $g(x) \leq 0$. Moreover, sufficient condition for saddle points to be discrete local minimum is that H(x) should be either strictly positive or strictly negative continuous function that satisfies H(0) = 0. Thus, any search strategy for saddle points is equivalent to searching for local minima as long as above constraints on $H(x)$ are satisfied. Therefore, the iterative discrete first order method that searches for discrete saddle points is as follows[4]

$$x^{k+1} = x^k \oplus \Delta_x L_d(x^k, \lambda^k, \mu^k)$$
$$\lambda^{k+1} = \lambda^k + cH(h(x^k))$$
$$\mu^{k+1} = \mu^k \oplus \mathbf{d}H(max(0, g(x^k)))$$

where c and $\mathbf{d}$ are positive stepwise real number and matrix controlling how fast Lagrange multipliers change, respectively.

To change to another saddle point, one can alter the penalties of the constraints. By reducing the penalties, the cost function is changed, and therefore will allow gradient descent to move to other regions in the search space. At that point, one may increase the penalties again to find a local saddle point. However, it is a formidable task to find the number of the local minimums of the entropy over the compositions of n. Thus, the algorithm based on Lagrange-multiplier formulation does not guarantee to find the global minimum of the entropy.

VI. DISCUSSIONS

Let us briefly summarize our main results. We study the algorithms that provide multi-round group-broadcast schemes in collocated Gaussian networks for a particular type-threshold function. Numerical evidence provides the optimum scheme for the computation of type-threshold function by minimizing the information received at fusion center in the network. The intuitive discussions as well as the numerical observations help us in building a rigorous framework for the analysis of the optimum scheme. Besides numerical calculations, more rigorous approach to the optimization problem, that is, Discrete Lagrange multipliers is studied.

The group size of the sensor nodes gets smaller for large p as we naturally expect due to the fact that reaching the threshold of the frequency is more probable for small number of sensors when individual probability of that frequency of the source is large. The same idea can also be extended to different type-threshold functions. Consider that $[\Theta_0\, \Theta_1]$ is assumed to be $[0\, l]$ instead of $[01]$. In the same setting, one can expect that general structure remains the same but group sizes will be larger in general since more sensor nodes in a group allows the fusion center to compute the desired function efficiently due to large threshold.

An interesting avenue for the future research can be evaluating the multi-round group-broadcast scheme for a type, frequency histogram function where at least 2 frequency information is to be received by fusion center to evaluate desired function,i.e., $[\Theta_0 \Theta_1] = [l_1\, l_2]$, and to analyze the optimum broadcasting scheme where more correlated relation is expected.

REFERENCES

[1] C.-Y. Wang, S.-W. Jeon, and M. Gastpar, "Interactive Computation of Type-Threshold Functions in Collocated Broadcast-Superposition Networks", *CoRR* abs/1310.7112, 2013.

[2] V. Vajnovszki, "An efficient Gray code algorithm for generating all permutations with a given major index", *J. Discrete Algorithms*, vol. 26, pp. 77-78, Jan. 2014.

[3] S. Eger, " Restricted integer compositions with fixed number of parts". Retrieved from http://www.math works.com/matlabcentral/fileexchange/authors/399812, May 2013

[4] B. W. Wah, and Z. Wu, "The Theory of Discrete Lagrange Multipliers for Nonlinear Discrete Optimization", *CP*, pp. 28-42, 1999.