

Optimal ordering of transmissions for computing Boolean threshold functions

Hemant Kowshik

CSL and Department of ECE
University of Illinois Urbana-Champaign
Email: kowshik2@illinois.edu

P. R. Kumar

CSL and Department of ECE
University of Illinois Urbana-Champaign
Email: prkumar@illinois.edu

Abstract—We address a sequential decision problem that arises in the computation of symmetric Boolean functions of distributed data. We consider a collocated network, where each node’s transmissions can be heard by every other node. Each node has a Boolean measurement and we wish to compute a given Boolean function of these measurements. We suppose that the measurements are independent and Bernoulli distributed. Thus, the problem of optimal computation becomes the problem of optimally ordering nodes’ transmissions so as to minimize the total expected number of bits.

We solve the ordering problem for the class of Boolean threshold functions. The optimal ordering is dynamic, i.e., it could potentially depend on the values of previously transmitted bits. Further, it depends only on the ordering of the marginal probabilities, but not on their exact values. This provides an elegant structure for the optimal strategy. For the case where each node has a block of measurements, the problem is significantly harder, and we conjecture the optimal strategy.

I. INTRODUCTION

Most sensor network applications are typically interested only in computing some relevant *function* of the correlated data at distributed sensors. For example, one might want to compute the mean temperature for environmental monitoring, or the maximum temperature in fire alarm systems. On the other hand, sensor nodes are severely limited in terms of power and bandwidth, and are generating enormous quantities of data. Thus, we seek efficient in-network computation and communication strategies for the function of interest.

Computing and communicating functions of distributed data presents several challenges. On the one hand, the wireless medium being a broadcast medium, nodes have to deal with interference from other transmissions. On the other hand, nodes can exploit these overheard transmissions, and the structure of the function to be computed, to achieve a more efficient description of their own data. Moreover, the strategy for computation may benefit from interactive information exchange between nodes.

We consider a collocated network where each node’s transmissions can be heard by every other node. At most one node

is allowed to transmit successfully at any time. Each node has a Boolean variable and we focus on the specific problem of symmetric Boolean function computation. We adopt a deterministic formulation of the problem of function computation, allowing zero error. We suppose that node measurements are independent and distributed according to given marginal Bernoulli distributions. In this paper, we focus on optimal strategies for Boolean threshold functions, which are equal to 1 if and only if the number of nodes with measurement 1 is greater than a certain threshold. The set of admissible strategies includes all interactive strategies, where a node may exchange several messages with other nodes.

In the case where each node has a single bit, the communication problem is rendered trivial, since it is optimal for the transmitting node to simply indicate its bit value. Thus, it only remains to determine the optimal ordering of nodes’ transmissions so as to minimize the expected number of bits exchanged. For the class of Boolean threshold functions, we present a simple indexing policy for ordering the transmissions and prove its optimality. The optimal policy is dynamic, possibly depending on the previously transmitted bits. Further, the optimal policy depends only on the ordering of the marginal probabilities, but surprisingly not on their values.

The problem of optimally ordering transmissions of nodes is a sequential decision problem and can indeed be solved by dynamic programming. However, this would require solving the dynamic program for all thresholds and all probability distributions, which is computationally hard. We avoid this, and establish a more insightful solution, in the form of a simple rule defining the optimal policy.

In Section III, we formulate the problem of single instance computation, and derive the resulting dynamic programming equation. We then propose the indexing policy and present a detailed proof of optimality, by induction on the number of nodes in the network. In Section IV, we consider the extension to the case of block computation, where each node has a block of measurements and we are allowed block coding. This problem is significantly harder, and we conjecture the structure of an optimal multi-round policy, building on the optimal policy for single instance computation.

This material is based upon work partially supported by AFOSR under Contract FA9550-09-0121, NSF under Contract No. CNS-07-21992, and USARO under Contract Nos. W911NF-08-1-0238 and W-911-NF-0710287. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the above agencies.

II. RELATED WORK

The problem of worst-case block function computation with zero error was formulated in [1]. The authors identify two classes of symmetric functions namely *type-sensitive* functions exemplified by Mean, Median and Mode, and *type-threshold* functions, exemplified by Maximum and Minimum. The maximum rates for computation of type-sensitive and type-threshold functions in random planar networks are shown to be $\Theta(\frac{1}{\log n})$ and $\Theta(\frac{1}{\log \log n})$ respectively, where n is the number of nodes. If we impose a probability distribution on the node measurements, one can show that the average case complexity of computing type-threshold functions is $\Theta(1)$ [2].

In this paper, we require that every node must compute the function. This approach naturally allows the use of tools from communication complexity [3]. In communication complexity [3], we seek to find the minimum number of bits that must be exchanged between two nodes to achieve worst-case zero-error computation of a function of the node variables. The problem of worst-case Boolean function computation was first considered in [4], where the complexity of the Boolean AND function was shown to be $\log_2 3$ bits. In [5], this was considerably generalized to derive the exact complexity of computing Boolean threshold functions.

If the measurements are drawn from some joint probability distribution and one is allowed block computation, we arrive at a distributed source coding problem with a fidelity criterion that is function-dependent, for which little is known. One special case, a source coding problem for function computation with side information, has been studied in [6]. The problem of interactive function computation in collocated networks has been studied in [7].

The problem of minimizing the depth of decision trees for Boolean threshold queries is considered in [8]. In [9], an interesting problem in sequential decision making is studied, where, n nodes have i.i.d. measurements, and a central agent wishes to know the identities of the nodes with the k largest values. One is allowed questions of the type “Is $X \geq t$ ”, to which the central agent receives the list of all nodes which satisfy the condition. Under this framework, the optimal recursive strategy of querying the nodes is found. A key difference in our formulation is that we are only allowed to query particular nodes, and not all nodes at once.

III. OPTIMAL ORDERING FOR SINGLE INSTANCE COMPUTATION

Consider a collocated network with nodes 1 through n , where each node i has a Boolean measurement $X_i \in \{0, 1\}$. The X_i s are independent of each other and drawn from a Bernoulli distribution with $P(X_i = 1) =: p_i$. Without loss of generality, we assume that $p_1 \leq p_2 \leq \dots \leq p_n$.

We address the following problem. *Every* node wants to compute the same function $f(X_1, X_2, \dots, X_n)$ of the measurements. We seek to find communication schemes which achieve correct function computation at each node, with minimum expected total number of bits exchanged. Throughout this paper, we consider the broadcast scenario where each node’s

transmission can be heard by every other node. We also suppose that collisions do not convey information thus restricting ourselves to *collision-free strategies* as in [1]. This means that for the k^{th} bit b_k , the identity of the transmitting node T_k depends only on previously broadcast bits b_1, b_2, \dots, b_{k-1} , while the value of the bit it sends can depend arbitrarily on all previous broadcast bits as well as its own measurements X_{T_k} .

First, we note that since each node has exactly one bit of information, it is optimal to set $b_k = X_{T_k}$. Indeed, for any other choice $b'_k = g(b_1, \dots, b_{k-1}, X_{T_k})$, the remaining nodes can reconstruct b'_k since they already know b_1, \dots, b_{k-1} . Thus the only freedom available is in choosing the transmitting node T_k as a function of b_1, b_2, \dots, b_{k-1} , for otherwise the transmission itself could be avoided. We call this the *ordering problem*. Thus, by definition, the order can dynamically depend on the previous broadcast bits. In this paper, we address the ordering problem for a class of Boolean functions, namely threshold functions.

Notation: The set of measurements of nodes 1 through n is denoted by (X_1, X_2, \dots, X_n) which is abbreviated as \mathbf{X}^n . In the sequel, we will use \mathbf{X}_{-i}^n to denote the set of measurements $(X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n)$. As a natural extension, we use $\mathbf{X}_{-(i,j)}^n$ to denote the set of measurements $(X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_{j-1}, X_{j+1}, \dots, X_n)$, where $i < j$.

A. Optimal ordering for computing Boolean threshold functions

Definition 1 (Boolean threshold functions): A Boolean threshold function $\Pi_\theta(X_1, X_2, \dots, X_n)$ is defined as

$$\Pi_\theta(X_1, X_2, \dots, X_n) = \begin{cases} 1 & \text{if } \sum_i X_i \geq \theta, \\ 0 & \text{otherwise.} \end{cases}$$

Given a function $\Pi_{n-k}(\mathbf{X}^n)$, the ordering problem can indeed be solved using dynamic programming. Let $C(\Pi_{n-k}(\mathbf{X}^n))$ denote the minimum expected number of bits required to compute $\Pi_{n-k}(\mathbf{X}^n)$. The dynamic programming equation is

$$C(\Pi_{n-k}(\mathbf{X}^n)) = \min_i \{1 + p_i C(\Pi_{n-k-1}(\mathbf{X}_{-i}^n)) + (1 - p_i) C(\Pi_{n-k}(\mathbf{X}_{-i}^n))\}.$$

However solving this equation is computationally complex. Further, it is unclear at the outset if the optimal strategy will depend only on the ordering of the p_i s, or their particular values. This makes the explicit solution of (III-A) for all n, k and (p_1, p_2, \dots, p_n) notoriously hard. We present a very simple characterization of the optimal strategy for each n and $0 \leq k \leq n-1$ and show that this is independent of the particular values of the p_i s, but only depends on the ordering.

To begin with, we argue that solving the ordering problem for Boolean threshold functions, is equivalent to solving the following problem for each n and k : In the optimal strategy for computing $\Pi_{n-k}(X_1, X_2, \dots, X_n)$ determine which node must transmit first. Indeed, if $T(1)$ is the first node to transmit under the optimal strategy, then, depending on whether $X_{T(1)} = 0$ or $X_{T(1)} = 1$, the rest of the nodes would need to compute $\Pi_{n-k}(\mathbf{X}_{-T(1)}^n)$ or $\Pi_{n-k-1}(\mathbf{X}_{-T(1)}^n)$. Since we solved the problem for all n and k , we can determine which node should transmit next in either case.

Theorem 1: In order to compute the Boolean threshold function $\Pi_{n-k}(\mathbf{X}^n)$, it is optimal for node $k+1$ to transmit first. This result is true for all n and all $0 \leq k \leq n-1$ and all probability distributions with $p_1 \leq p_2 \leq \dots \leq p_n$.

Proof: Define $\tilde{C}(\Pi_{n-k}(\mathbf{X}^n)) := C(\Pi_{n-k}(\mathbf{X}^n)) - 1$ for notational convenience. We also define the following expressions.

$$T_{m,k,i}(\mathbf{X}^m) := p_{k+1}\tilde{C}(\Pi_{m-k-1}(\mathbf{X}_{-(k+1)}^m)) + (1-p_{k+1})\tilde{C}(\Pi_{m-k}(\mathbf{X}_{-(k+1)}^m)) - p_i\tilde{C}(\Pi_{m-k-1}(\mathbf{X}_{-i}^m)) - (1-p_i)\tilde{C}(\Pi_{m-k}(\mathbf{X}_{-i}^m)). \quad (1)$$

$T_{m,k,i}$ is the difference between the expected number of bits when node $k+1$ transmits first, and the expected number of bits when node i transmits first.

$$S_{m,k,i}^{(1)}(\mathbf{X}^m) := (p_{k+1}-p_i)C(\Pi_{m-k-1}(\mathbf{X}_{-(k+1,i)}^m)) + (1-p_{k+1})\tilde{C}(\Pi_{m-k}(\mathbf{X}_{-(k+1)}^m)) - (1-p_i)\tilde{C}(\Pi_{m-k}(\mathbf{X}_{-i}^m)) \quad (2)$$

$$S_{m,k,i}^{(2)}(\mathbf{X}^m) := (p_i-p_{k+1})C(\Pi_{m-k-1}(\mathbf{X}_{-(i,k+1)}^m)) + p_{k+1}\tilde{C}(\Pi_{m-k-1}(\mathbf{X}_{-(k+1)}^m)) - p_i\tilde{C}(\Pi_{m-k-1}(\mathbf{X}_{-i}^m)) \quad (3)$$

We do not yet have an interpretation for $S_{m,k,i}^{(1)}$ and $S_{m,k,i}^{(2)}$. However, we will use these expressions in the sequel.

We establish the above theorem by induction on the number of nodes n . However, we need to load the induction hypothesis. Consider the following induction hypothesis.

- (a) $T_{m,k,i}(\mathbf{X}^m) \leq 0$ for all $0 \leq k \leq (m-1), 1 \leq i \leq m$
- (b) $S_{m,k,i}^{(1)}(\mathbf{X}^m) \leq 0$ for all $0 \leq k \leq (m-1), k+2 \leq i \leq m$
- (c) $S_{m,k,i}^{(2)}(\mathbf{X}^m) \leq 0$ for all $0 \leq k \leq (m-1), 1 \leq i \leq k$

Observe that part (a) immediately establishes that $k+1$ should transmit first in the optimal strategy for computing the function $\Pi_{m-k}(\mathbf{X}^m)$.

The basis step for $m=1, k=1$ is trivially true. Let us suppose the induction hypothesis is true for all $m \leq n$. We now proceed to prove the hypothesis for $m=n+1$.

Lemma 1: For fixed k and $i \geq k+2$, we have $S_{n+1,k,i}^{(1)}(\mathbf{X}^{n+1}) \leq 0$.

Proof: See Appendix A.

Lemma 2: For fixed k and $i \leq k$, we have $S_{n+1,k,i}^{(2)}(\mathbf{X}^{n+1}) \leq 0$.

Proof: See Appendix B.

Lemmas 1 and 2 establish the induction step for parts (b) and (c) of the induction hypothesis. We now proceed to show the induction step for part (a).

Lemma 3: For fixed k and $i \geq k+2$, we have $T_{n+1,k,i}(\mathbf{X}^{n+1}) \leq S_{n+1,k,i}^{(1)}(\mathbf{X}^{n+1})$.

Proof: See Appendix C

Lemma 4: For fixed k and $i \leq k$, we have $T_{n+1,k,i}(\mathbf{X}^{n+1}) \leq S_{n+1,k,i}^{(2)}(\mathbf{X}^{n+1})$.

Proof: See Appendix D

Using Lemmas 3 and 4 together with Lemmas 1 and 2, we see that $T_{n+1,k,i}(\mathbf{X}^{n+1}) \leq 0$ for all $0 \leq k \leq n$ and $i \neq k+1$. For the case $i=k+1$, we have $T(n+1, k, k+1) = 0$ trivially. This completes the induction step for part (a), and the proof of the Theorem. \square

IV. OPTIMAL ORDERING FOR BLOCK COMPUTATION

We now shift attention to the case where we allow for nodes to accumulate a block of N measurements, and thus achieve improved efficiency by using block codes. We consider the class of all interactive strategies for computation, where the k th bit can depend arbitrarily on all previously broadcast bits. We require that all nodes compute the function with zero error for the block. We present a conjecture for the optimal strategy based on the insight gained from the single instance solution.

Conjecture 1: In order to compute the Boolean threshold function $\Pi_{n-k}(\mathbf{X}^n)$, it is optimal for node $k+1$ to transmit first, using the Huffman code. This result is true for all n and all $0 \leq k \leq n-1$ and all probability distributions with $p_1 \leq p_2 \leq \dots \leq p_n$.

Observe that after node $k+1$ transmits, we are left with two block computation problems. For the instances where $X_{k+1} = 0$, we need to compute $\Pi_{n-k}(\mathbf{X}_{-(k+1)}^n)$ and for the instances where $X_{k+1} = 1$, we need to compute $\Pi_{n-k-1}(\mathbf{X}_{-(k+1)}^n)$. Thus the conjectured strategy can be recursively applied, yielding an interactive multi-round strategy. However, proving the optimality of this strategy is significantly harder. For worst case block computation, the lower bound is established using fooling sets [5]. Adapting this idea to the probabilistic scenario remains an interesting challenge for the future.

V. CONCLUDING REMARKS

We have considered a sequential decision problem, that arises in the context of optimal computation of Boolean threshold functions in collocated networks. For single instance computation, we show that the optimal strategy has an elegant structure, which depends only on the ordering of the marginal probabilities, and not on their exact values. The extension to the case of block computation is harder and remains a challenge for the future. It is also interesting to extend this result to the case of correlated measurements

REFERENCES

- [1] A. Giridhar and P. R. Kumar. Computing and communicating functions over sensor networks. *IEEE Journal on Selected Areas in Communication*, 23(4):755–764, April 2005.
- [2] H. Kowshik and P. R. Kumar. Zero-error function computation in sensor networks. In *Proceedings of the 48th IEEE Conference on Decision and Control(CDC)*, December 2009.
- [3] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1997.
- [4] R. Ahlswede and Ning Cai. On communication complexity of vector-valued functions. *IEEE Transactions on Information Theory*, 40:2062–2067, 1994.
- [5] H. Kowshik and P. R. Kumar. Optimal strategies for computing boolean functions in collocated networks. In *Proceedings of the Information Theory Workshop, Cairo*, January 2010.
- [6] A. Orlitsky and J. R. Roche. Coding for computing. *IEEE Transactions on Information Theory*, 47:903–917, 2001.
- [7] N. Ma, P. Ishwar, and P. Gupta. Information-theoretic bounds for multi-round function computation in collocated networks. In *IEEE International Symposium on Information Theory (ISIT)*, 2009.
- [8] Yosi Ben-Asher and Ilan Newman. Decision trees with boolean threshold queries. *Journal of Computer and System Sciences*, 51, 1995.
- [9] K. J. Arrow, L. Pesotchinsky, and M. Sobel. On partitioning of a sample with binary-type questions in lieu of collecting observations. *Journal of the American Statistical Association*, 76(374):402–409, June 1981.

APPENDIX

A. Proof of Lemma 1

$$\begin{aligned}
& (p_{k+1} - p_i)C(\Pi_{n-k}(\mathbf{X}_{-(k+1,i)}^{n+1})) + (1 - p_{k+1})\tilde{C}(\Pi_{n-k+1}(\mathbf{X}_{-(k+1)}^{n+1})) - (1 - p_i)\tilde{C}(\Pi_{n-k+1}(\mathbf{X}_{-i}^{n+1})) \\
= & (p_{k+1} - p_i) \left[1 + p_k C(\Pi_{n-k-1}(\mathbf{X}_{-(k,k+1,i)}^{n+1})) + (1 - p_k)C(\Pi_{n-k}(\mathbf{X}_{-(k,k+1,i)}^{n+1})) \right] \\
& + (1 - p_{k+1}) \left[p_k C(\Pi_{n-k}(\mathbf{X}_{-(k,k+1)}^{n+1})) + (1 - p_k)C(\Pi_{n-k+1}(\mathbf{X}_{-(k,k+1)}^{n+1})) \right] \\
& - (1 - p_i) \left[p_k C(\Pi_{n-k}(\mathbf{X}_{-(k,i)}^{n+1})) + (1 - p_k)C(\Pi_{n-k+1}(\mathbf{X}_{-(k,i)}^{n+1})) \right] \tag{4} \\
= & p_k \left[(p_{k+1} - p_i)C(\Pi_{n-k-1}(\mathbf{X}_{-(k,k+1,i)}^{n+1})) + (1 - p_{k+1})\tilde{C}(\Pi_{n-k}(\mathbf{X}_{-(k,k+1)}^{n+1})) - (1 - p_i)\tilde{C}(\Pi_{n-k}(\mathbf{X}_{-(k,i)}^{n+1})) \right] \\
& + (1 - p_k) \left[(p_{k+1} - p_i)C(\Pi_{n-k}(\mathbf{X}_{-(k,k+1,i)}^{n+1})) + (1 - p_{k+1})\tilde{C}(\Pi_{n-k+1}(\mathbf{X}_{-(k,k+1)}^{n+1})) - (1 - p_i)\tilde{C}(\Pi_{n-k+1}(\mathbf{X}_{-(k,i)}^{n+1})) \right] \\
\leq & p_k \left[(p_{k+1} - p_i)C(\Pi_{n-k-1}(\mathbf{X}_{-(k,k+1,i)}^{n+1})) + (1 - p_{k+1})\tilde{C}(\Pi_{n-k}(\mathbf{X}_{-(k,k+1)}^{n+1})) - (1 - p_i)\tilde{C}(\Pi_{n-k}(\mathbf{X}_{-(k,i)}^{n+1})) \right] + (1 - p_k)S_{n,k-1,i-1}^{(1)}(\mathbf{X}_{-k}^{n+1}) \\
\leq & p_k \left[(p_{k+1} - p_i)C(\Pi_{n-k-1}(\mathbf{X}_{-(k,k+1,i)}^{n+1})) + (1 - p_{k+1})\tilde{C}(\Pi_{n-k}(\mathbf{X}_{-(k,k+1)}^{n+1})) - (1 - p_i)\tilde{C}(\Pi_{n-k}(\mathbf{X}_{-(k,i)}^{n+1})) \right] \tag{5} \\
= & p_k \left[(p_{k+1} - p_i)C(\Pi_{n-k-1}(\mathbf{X}_{-(k,k+1,i)}^{n+1})) + (1 - p_{k+1})\tilde{C}(\Pi_{n-k}(\mathbf{X}_{-(k,k+1)}^{n+1})) \right. \\
& \left. - (1 - p_i)[p_{k+1}C(\Pi_{n-k-1}(\mathbf{X}_{-(k,k+1,i)}^{n+1})) + (1 - p_{k+1})C(\Pi_{n-k}(\mathbf{X}_{-(k,k+1,i)}^{n+1}))] \right] \tag{6} \\
= & p_k(1 - p_{k+1}) \left[\tilde{C}(\Pi_{n-k}(\mathbf{X}_{-(k,k+1)}^{n+1})) - p_i C(\Pi_{n-k-1}(\mathbf{X}_{-(k,k+1,i)}^{n+1})) - (1 - p_i)C(\Pi_{n-k}(\mathbf{X}_{-(k,k+1,i)}^{n+1})) \right] \leq 0. \tag{7}
\end{aligned}$$

Equation (4) follows from the optimal ordering for computing $\Pi_{n-k}(\mathbf{X}_{-(k+1,i)}^{n+1})$, $\Pi_{n-k+1}(\mathbf{X}_{-(k+1)}^{n+1})$ and $\Pi_{n-k+1}(\mathbf{X}_{-i}^{n+1})$, which is true by the induction hypothesis for $m = n$. The inequality (5) follows from the induction hypothesis that $S_{n,k-1,i}^{(1)}(\mathbf{X}_{-k}^{n+1}) \leq 0$. Equality in (6) and (7) follows from the optimal ordering for computing $\Pi_{n-k}(\mathbf{X}_{-(k,i)}^{n+1})$ and $\Pi_{n-k}(\mathbf{X}_{-(k,k+1)}^{n+1})$ respectively. \square

B. Proof of Lemma 2

$$\begin{aligned}
& (p_i - p_{k+1})C(\Pi_{n-k}(\mathbf{X}_{-(i,k+1)}^{n+1})) + p_{k+1}\tilde{C}(\Pi_{n-k}(\mathbf{X}_{-(k+1)}^{n+1})) - p_i\tilde{C}(\Pi_{n-k}(\mathbf{X}_{-i}^{n+1})) \\
= & (p_i - p_{k+1}) \left[1 + p_{k+2}C(\Pi_{n-k-1}(\mathbf{X}_{-(i,k+1,k+2)}^{n+1})) + (1 - p_{k+2})C(\Pi_{n-k}(\mathbf{X}_{-(i,k+1,k+2)}^{n+1})) \right] \\
& + p_{k+1} \left[p_{k+2}C(\Pi_{n-k-1}(\mathbf{X}_{-(k+1,k+2)}^{n+1})) + (1 - p_{k+2})C(\Pi_{n-k}(\mathbf{X}_{-(k+1,k+2)}^{n+1})) \right] \\
& - p_i \left[p_{k+2}C(\Pi_{n-k-1}(\mathbf{X}_{-(i,k+2)}^{n+1})) + (1 - p_{k+2})C(\Pi_{n-k}(\mathbf{X}_{-(i,k+2)}^{n+1})) \right] \tag{8} \\
= & p_{k+2} \left[(p_i - p_{k+1})C(\Pi_{n-k-1}(\mathbf{X}_{-(i,k+1,k+2)}^{n+1})) + p_{k+1}\tilde{C}(\Pi_{n-k-1}(\mathbf{X}_{-(k+1,k+2)}^{n+1})) - p_i\tilde{C}(\Pi_{n-k-1}(\mathbf{X}_{-(i,k+2)}^{n+1})) \right] \\
& + (1 - p_{k+2}) \left[(p_i - p_{k+1})C(\Pi_{n-k}(\mathbf{X}_{-(i,k+1,k+2)}^{n+1})) + p_{k+1}\tilde{C}(\Pi_{n-k}(\mathbf{X}_{-(k+1,k+2)}^{n+1})) - p_i\tilde{C}(\Pi_{n-k}(\mathbf{X}_{-(i,k+2)}^{n+1})) \right] \\
\leq & p_{k+2} \left[S_{n,k,i}^{(2)}(\mathbf{X}_{-(k+2)}^{n+1}) \right] + (1 - p_{k+2}) \left[(p_i - p_{k+1})C(\Pi_{n-k}(\mathbf{X}_{-(i,k+1,k+2)}^{n+1})) + p_{k+1}\tilde{C}(\Pi_{n-k}(\mathbf{X}_{-(k+1,k+2)}^{n+1})) - p_i\tilde{C}(\Pi_{n-k}(\mathbf{X}_{-(i,k+2)}^{n+1})) \right] \\
\leq & (1 - p_{k+2}) \left[(p_i - p_{k+1})C(\Pi_{n-k}(\mathbf{X}_{-(i,k+1,k+2)}^{n+1})) + p_{k+1}\tilde{C}(\Pi_{n-k}(\mathbf{X}_{-(k+1,k+2)}^{n+1})) - p_i\tilde{C}(\Pi_{n-k}(\mathbf{X}_{-(i,k+2)}^{n+1})) \right] \tag{9} \\
= & (1 - p_{k+2}) \left[(p_i - p_{k+1})C(\Pi_{n-k}(\mathbf{X}_{-(i,k+1,k+2)}^{n+1})) + p_{k+1}\tilde{C}(\Pi_{n-k}(\mathbf{X}_{-(k+1,k+2)}^{n+1})) \right. \\
& \left. - p_i[p_{k+1}C(\Pi_{n-k-1}(\mathbf{X}_{-(i,k+1,k+2)}^{n+1})) + (1 - p_{k+1})C(\Pi_{n-k}(\mathbf{X}_{-(i,k+1,k+2)}^{n+1}))] \right] \tag{10} \\
= & (1 - p_{k+2})p_{k+1} \left[\tilde{C}(\Pi_{n-k}(\mathbf{X}_{-(k+1,k+2)}^{n+1})) - p_i C(\Pi_{n-k-1}(\mathbf{X}_{-(i,k+1,k+2)}^{n+1})) - (1 - p_i)C(\Pi_{n-k}(\mathbf{X}_{-(i,k+1,k+2)}^{n+1})) \right] \leq 0. \tag{11}
\end{aligned}$$

Equation (8) follows from the optimal ordering for computing $\Pi_{n-k}(\mathbf{X}_{-(i,k+1)}^{n+1})$, $\Pi_{n-k}(\mathbf{X}_{-(k+1)}^{n+1})$ and $\Pi_{n-k}(\mathbf{X}_{-i}^{n+1})$, which follows from the induction hypothesis for $m = n$. The inequality (9) follows from the induction hypothesis that $S_{n,k,i}^{(2)}(\mathbf{X}_{-(k+2)}^{n+1}) \leq 0$. Equations (10) and (11) follow from the optimal ordering for computing $\Pi_{n-k}(\mathbf{X}_{-(i,k+2)}^{n+1})$ and $\Pi_{n-k}(\mathbf{X}_{-(k+1,k+2)}^{n+1})$ respectively. \square

C. Proof of Lemma 3

First, we observe that

$$T_{n+1,k,i}(\mathbf{X}^{n+1}) - S_{n+1,k,i}^{(1)}(\mathbf{X}^{n+1}) = p_{k+1}\tilde{C}(\Pi_{n-k}(\mathbf{X}_{-(k+1)}^{n+1})) - p_i\tilde{C}(\Pi_{n-k}(\mathbf{X}_{-i}^{n+1})) - (p_{k+1} - p_i)C(\Pi_{n-k}(\mathbf{X}_{-(k+1,i)}^{n+1}))$$

Thus it is enough to show that

$$p_{k+1}\tilde{C}(\Pi_{n-k}(\mathbf{X}_{-(k+1)}^{n+1})) - p_i\tilde{C}(\Pi_{n-k}(\mathbf{X}_{-i}^{n+1})) \leq (p_{k+1} - p_i)C(\Pi_{n-k}(\mathbf{X}_{-(k+1,i)}^{n+1})) \quad \text{for } i \geq k+2$$

$$\begin{aligned} & p_{k+1}\tilde{C}(\Pi_{n-k}(\mathbf{X}_{-(k+1)}^{n+1})) - p_i\tilde{C}(\Pi_{n-k}(\mathbf{X}_{-i}^{n+1})) \\ = & p_{k+1} \left[p_{k+2}C(\Pi_{n-k-1}(\mathbf{X}_{-(k+1,k+2)}^{n+1})) + (1 - p_{k+2})C(\Pi_{n-k}(\mathbf{X}_{-(k+1,k+2)}^{n+1})) \right] \\ & - p_i \left[p_{k+1}C(\Pi_{n-k-1}(\mathbf{X}_{-(k+1,i)}^{n+1})) + (1 - p_{k+1})C(\Pi_{n-k}(\mathbf{X}_{-(k+1,i)}^{n+1})) \right] \end{aligned} \quad (12)$$

$$\begin{aligned} = & p_{k+1} \left[p_{k+2}C(\Pi_{n-k-1}(\mathbf{X}_{-(k+1,k+2)}^{n+1})) - p_iC(\Pi_{n-k-1}(\mathbf{X}_{-(k+1,i)}^{n+1})) \right] \\ & + p_{k+1}(1 - p_{k+2})C(\Pi_{n-k}(\mathbf{X}_{-(k+1,k+2)}^{n+1})) - p_i(1 - p_{k+1})C(\Pi_{n-k}(\mathbf{X}_{-(k+1,i)}^{n+1})) \\ \leq & p_{k+1} \left[(1 - p_i)C(\Pi_{n-k}(\mathbf{X}_{-(k+1,i)}^{n+1})) - (1 - p_{k+2})C(\Pi_{n-k}(\mathbf{X}_{-(k+1,k+2)}^{n+1})) \right] \\ & + p_{k+1}(1 - p_{k+2})C(\Pi_{n-k}(\mathbf{X}_{-(k+1,k+2)}^{n+1})) - p_i(1 - p_{k+1})C(\Pi_{n-k}(\mathbf{X}_{-(k+1,i)}^{n+1})) \\ = & (p_{k+1} - p_i)C(\Pi_{n-k}(\mathbf{X}_{-(k+1,i)}^{n+1})) \end{aligned} \quad (13)$$

Equation 12 follows from the optimal order for computing $\Pi_{n-k}(\mathbf{X}_{-(k+1)}^{n+1})$ and $\Pi_{n-k}(\mathbf{X}_{-i}^{n+1})$. The inequality in 13 follows from the induction hypothesis $T_{n,k,i}(\mathbf{X}_{-(k+1)}^{n+1}) \leq 0$. \square

D. Proof of Lemma 4

First, we observe that

$$T_{n+1,k,i}(\mathbf{X}^{n+1}) - S_{n+1,k,i}^{(2)}(\mathbf{X}^{n+1}) = (1 - p_{k+1})\tilde{C}(\Pi_{n-k+1}(\mathbf{X}_{-(k+1)}^{n+1})) - (1 - p_i)\tilde{C}(\Pi_{n-k+1}(\mathbf{X}_{-i}^{n+1})) - (p_i - p_{k+1})C(\Pi_{n-k}(\mathbf{X}_{-(i,k+1)}^{n+1}))$$

Thus it is enough to show that

$$(1 - p_{k+1})\tilde{C}(\Pi_{n-k+1}(\mathbf{X}_{-(k+1)}^{n+1})) - (1 - p_i)\tilde{C}(\Pi_{n-k+1}(\mathbf{X}_{-i}^{n+1})) \leq (p_i - p_{k+1})C(\Pi_{n-k}(\mathbf{X}_{-(i,k+1)}^{n+1})) \quad \text{for } i \leq k$$

$$\begin{aligned} & (1 - p_{k+1})\tilde{C}(\Pi_{n-k+1}(\mathbf{X}_{-(k+1)}^{n+1})) - (1 - p_i)\tilde{C}(\Pi_{n-k+1}(\mathbf{X}_{-i}^{n+1})) \\ = & (1 - p_{k+1}) \left[p_kC(\Pi_{n-k}(\mathbf{X}_{-(k,k+1)}^{n+1})) + (1 - p_k)C(\Pi_{n-k+1}(\mathbf{X}_{-(k,k+1)}^{n+1})) \right] \\ & - (1 - p_i) \left[p_{k+1}C(\Pi_{n-k}(\mathbf{X}_{-(i,k+1)}^{n+1})) + (1 - p_{k+1})C(\Pi_{n-k+1}(\mathbf{X}_{-(i,k+1)}^{n+1})) \right] \end{aligned} \quad (14)$$

$$\begin{aligned} = & (1 - p_{k+1}) \left[(1 - p_k)C(\Pi_{n-k+1}(\mathbf{X}_{-(k,k+1)}^{n+1})) - (1 - p_i)C(\Pi_{n-k+1}(\mathbf{X}_{-(i,k+1)}^{n+1})) \right] \\ & + p_k(1 - p_{k+1})C(\Pi_{n-k}(\mathbf{X}_{-(k,k+1)}^{n+1})) - p_{k+1}(1 - p_i)C(\Pi_{n-k}(\mathbf{X}_{-(i,k+1)}^{n+1})) \\ \leq & (1 - p_{k+1}) \left[p_iC(\Pi_{n-k}(\mathbf{X}_{-(i,k+1)}^{n+1})) - p_kC(\Pi_{n-k}(\mathbf{X}_{-(k,k+1)}^{n+1})) \right] \\ & + p_k(1 - p_{k+1})C(\Pi_{n-k}(\mathbf{X}_{-(k,k+1)}^{n+1})) - p_{k+1}(1 - p_i)C(\Pi_{n-k}(\mathbf{X}_{-(i,k+1)}^{n+1})) \\ = & (p_i - p_{k+1})C(\Pi_{n-k}(\mathbf{X}_{-(i,k+1)}^{n+1})) \end{aligned} \quad (15)$$

Equation 14 follows from the optimal order for computing $\Pi_{n-k+1}(\mathbf{X}_{-(k+1)}^{n+1})$ and $\Pi_{n-k+1}(\mathbf{X}_{-i}^{n+1})$. The inequality in 15 follows from the induction hypothesis $T_{n,k-1,i}(\mathbf{X}_{-(k+1)}^{n+1}) \leq 0$. \square