

Adaptivity Gaps for the Stochastic Boolean Function Evaluation Problem

Lisa Hellerstein¹, Devorah Kletenik², Naifeng Liu³, and R. Teal Witter¹

¹ NYU Tandon, Brooklyn NY 11201, USA

² Brooklyn College, Brooklyn NY 11210, USA

³ CUNY Graduate Center, New York NY 10016, USA

Abstract. We consider the Stochastic Boolean Function Evaluation (SBFE) problem where the task is to efficiently evaluate a known Boolean function f on an unknown bit string x of length n . We determine $f(x)$ by sequentially testing the variables of x , each of which is associated with a cost of testing and an independent probability of being true. If a strategy for solving the problem is adaptive in the sense that its next test can depend on the outcomes of previous tests, it has lower expected cost but may take up to exponential space to store. In contrast, a non-adaptive strategy may have higher expected cost but can be stored in linear space and benefit from parallel resources. The adaptivity gap, the ratio between the expected cost of the optimal non-adaptive and adaptive strategies, is a measure of the benefit of adaptivity. We present lower bounds on the adaptivity gap for the SBFE problem for popular classes of Boolean functions, including read-once DNF formulas, read-once formulas, and general DNFs. Our bounds range from $\Omega(\log n)$ to $\Omega(n/\log n)$, contrasting with recent $O(1)$ gaps shown for symmetric functions and linear threshold functions.

1 Introduction

We consider the question of determining adaptivity gaps for the Stochastic Boolean Function Evaluation (SBFE) problem, for different classes of Boolean formulas. In an SBFE problem, we are given a (representation of a) Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$, a positive cost vector $c = [c_1, \dots, c_n]$, and a probability vector $p = [p_1, \dots, p_n]$. The problem is to determine the value $f(x)$ on an initially unknown random input $x \in \{0, 1\}^n$. The value of each x_i can only be determined by performing a test, which incurs a cost of c_i . Each x_i is equal to 1 (is true) with independent probability p_i . Tests are performed sequentially and continue until $f(x)$ can be determined. We say $f(x)$ is determined by a set of tests if $f(x) = f(x')$ for all $x' \in \{0, 1\}^n$ such that $x'_i = x_i$ for every i in the set of tests.

For example, if $f(x) = x_1 \vee \dots \vee x_n$ then testing continues until a test is performed on some x_i such that $x_i = 1$ at which point we know $f(x) = 1$, or until all n tests have been performed with outcome $x_i = 0$ for each x_i so we know $f(x) = 0$. The problem is to determine the order to perform tests that minimizes the total expected cost of the tests.

We will call a testing order a strategy which we can think of as a decision tree for evaluating f . A strategy can be *adaptive*, meaning that the choice of the next test x_i can depend on the outcome of previous tests. In some practical settings, however, it is desirable to consider only *non-adaptive* strategies. Non-adaptive strategies often take up less space than adaptive strategies, and they may be able to be evaluated more quickly if tests can be performed in parallel [20], such as in the problem of detecting network faults [23] or in group testing for viruses, such as the coronavirus [28]. A non-adaptive testing strategy is a permutation of the tests where testing continues in the order specified by the permutation until the value of $f(x)$ can be determined from the outcomes of the tests performed so far. A non-adaptive strategy also corresponds to a decision tree where all non-leaf nodes on the same level contain the same test x_i .

The *adaptivity gap* measures how much benefit can be obtained by using an adaptive strategy. Consider a class F of n -variable functions $f : \{0, 1\}^n \rightarrow \{0, 1\}$. Let $\text{OPT}_{\mathcal{N}}(f, c, p)$ be the expected evaluation cost of the optimal non-adaptive strategy on function f under costs c and probabilities p . Similarly, $\text{OPT}_{\mathcal{A}}(f, c, p)$ is the expected evaluation cost of the optimal adaptive strategy on f under c and p . The adaptivity gap of the function class F is

$$\max_{f \in F} \sup_{c, p} \frac{\text{OPT}_{\mathcal{N}}(f, c, p)}{\text{OPT}_{\mathcal{A}}(f, c, p)}.$$

The SBFE problem for a class of Boolean formulas F restricts the evaluated f to be a member of F . In this paper we prove bounds on the adaptivity gaps for the SBFE problem on read-once DNF formulas, DNF formulas, and read-once formulas. (See Section 1.3 for definitions.) A summary of our results can be found in Table 1.

All the bounds in the table have a dependence on n , meaning that none of the listed SBFE problems has a constant adaptivity gap. This contrasts with recent work of Ghuge et al. [14], which shows that the adaptivity gaps for the SBFE problem for symmetric Boolean functions and linear threshold functions are $O(1)$.

For any SBFE problem, the non-adaptive strategy of testing the x_i in increasing order of c_i has an expected cost that is within a factor of n of the optimal adaptive strategy [25]. Thus n is an upper bound on the adaptivity gap for all SBFE problems.

Table 1. A summary of our results. We also prove an $O(\sqrt{n})$ upper bound for tribes formulas i.e., read-once DNFs with unit costs where every term has the same number of variables. We say all probabilities are equal if $p_1 = p_2 = \dots = p_n$.

Formula Class	Adaptivity Gap
Read-once DNF	$\Theta(\log n)$ for unit costs, uniform distribution
	$\Omega(\sqrt{n})$ for unit costs
	$\Omega(n^{1-\epsilon}/\log n)$ for uniform distribution
Read-once	$\Omega(\epsilon^3 n^{1-2\epsilon/\ln(2)})$ for unit costs, equal probabilities
DNF	$\Omega(n/\log n)$ for unit costs, uniform distribution
	$\Theta(n)$ for uniform distribution

Outline: We present our results on formula classes in increasing order of generality. In Section 2, we warm up with a variety of results on read-once DNF formulas in different settings. In Section 3, we prove our main technical result for read-once formulas, drawing on branching process identities and concentration inequalities. In Section 4, we prove our most general results on DNF formulas (the bounds also apply to the restricted class of DNF formulas with a linear number of terms). Note that we state our lower bound results in the most restricted context because they of course apply to more general settings. Due to space constraints, we defer some proofs to Appendix A.

1.1 Connection to *st*-connectivity in uncertain networks

Our result for read-once formulas has implications for a problem of determining *st*-connectivity in an uncertain network, studied by Fu et al. [13]. The input is a multi-graph with a source node s and a destination node t . Each edge corresponds to a variable x_i indicating whether it is usable, which

is true with probability p_i . Testing the usability of edge i costs c_i . The st -connectivity function for the multi-graph is true if and only if there is a path of usable edges from s to t . The problem is to find a strategy to evaluate the st -connectivity function that has minimum expected cost.

The st -connectivity function associated with a multi-graph can be represented by a read-once formula if and only if the multi-graph is a two-terminal series-parallel graph. This type of graph has two distinguished nodes, s and t , and is formed by recursively combining disjoint series-parallel graphs either in series, or in parallel (see [12] for the precise definitions).⁴ Fu et al. performed experiments with both adaptive and non-adaptive strategies for this problem, comparing their performance, but did not prove theoretical adaptivity gap bounds. Since the st -connectivity function on a series-parallel graph is a read-once formula, our lower bound on the adaptivity gap for read-once formulas applies to the problem of st -connectivity.

1.2 Related work

It is well-known that the SBFE problem for the Boolean OR function given by $f(x) = x_1 \vee \dots \vee x_n$ has a simple solution: test the variables x_i in increasing order of the ratio c_i/p_i until a test reveals a variable set to true, or until all variables are tested and found to be false (cf. [30]). This strategy is non-adaptive, meaning that the SBFE problem for the Boolean OR function has an adaptivity gap of 1. That is, there is no benefit to adaptivity.

Gkenosis et al. [16] introduced the Stochastic Score Classification problem, which generalizes the SBFE problem for both symmetric Boolean functions and for linear threshold functions. Ghuge et al. [14] showed that the Stochastic Score Classification problem has an adaptivity gap of $O(1)$. In the unit-cost case, Gkenosis et al. showed the gap is at most 4 for symmetric Boolean functions, and at most ϕ (the golden ratio) for the not-all-equal function [16].

Adaptivity gaps were introduced by Dean et al. [7] in the study of the stochastic knapsack problem which, in contrast to the SBFE problem for Boolean functions, is a maximization problem. It has an adaptivity gap of 4 [7,9]. Adaptivity gaps have also been shown for other stochastic maximization problems (e.g., [8,19,5,24,2]). Notably, the problem of maximizing a monotone submodular function with stochastic inputs, subject to any class of prefix-closed constraints, was shown to have an $O(1)$ adaptivity gap [20,5].

Adaptivity gaps have also been shown for stochastic covering problems, which, like SBFE, are minimization problems. Goemans et al. [17] showed that the adaptivity gap for the Stochastic Set Cover problem, in which each item can only be chosen once, is $\Omega(d)$ and $O(d^2)$, where d is the size of the target set to be covered. If the items can be used repeatedly, the adaptivity gap is $\Theta(\log d)$.

Agarwal et al. [1] and Ghuge et al. [15] proved bounds of $\Omega(Q)$ and $O(Q \log Q)$ respectively, on the adaptivity gap for the more abstract Stochastic Submodular Cover Problem in which each item can only be used once. Applied to the special case of Stochastic Set Cover, the upper bound is $O(d \log d)$, which improves the above $O(d^2)$ bound. They also gave bounds parameterized by the number of rounds of adaptivity allowed. We note that, as shown by Deshpande et al. [10], one approach to solving SBFE problems is to reduce them to special cases of Stochastic Submodular Cover. However, this approach does not seem to have interesting implications for SBFE problem adaptivity gaps.

⁴ The term *series-parallel circuits* (systems) refers to a set of parallel circuits that are connected in series (see, e.g., [11,31]). Viewed as graphs, they correspond to the subset of two-terminal series-parallel graphs whose st -connectivity functions correspond to read-once CNF formulas. We note that Kowshik used the term “series-parallel graph” in a non-standard way to refer only to this subset; Fu et al. in citing Kowshik, used the term the same way [27,13].

1.3 Preliminaries

Consider a Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$, a positive cost vector $c = [c_1, \dots, c_n]$, and probability vector $p = [p_1, \dots, p_n]$. We assume $c_i > 0$ and $0 < p_i < 1$ for $i \in [n]$ where $[n]$ denotes the set $\{1, \dots, n\}$. Let strategy S be a decision tree for evaluating $f(x)$ on an unknown input $x \in \{0,1\}^n$. We define $\text{cost}_c(f, x, S)$ as the total cost of the variables tested by S on input x until $f(x)$ is determined. We say $x \sim p$ if $\Pr(x) = \prod_{i:x_i=1} p_i \prod_{i:x_i=0} (1 - p_i)$. Then $\text{cost}_{c,p}(f, S) := \mathbb{E}_{x \sim p}[\text{cost}_c(f, x, S)]$ is the expected evaluation cost of strategy S when x is drawn according to the product distribution induced by p .

For fixed n , let $\mathcal{A}$ be the set of adaptive strategies on n variables and $\mathcal{N}$ be the set of non-adaptive strategies on n variables. We are interested in the quantities

$$\text{OPT}_{\mathcal{A}}(f, c, p) := \min_{S \in \mathcal{A}} \text{cost}_{c,p}(f, S) \text{ and } \text{OPT}_{\mathcal{N}}(f, c, p) := \min_{S \in \mathcal{N}} \text{cost}_{c,p}(f, S).$$

We will omit c and p from the notation when the costs and probabilities are clear from context. A (Boolean) *read-once formula* is a tree, each of whose internal nodes are labeled either $\vee$ or $\wedge$. The internal nodes have two or more children. Each leaf is labeled with a Boolean variable $x_i \in \{x_1, \dots, x_n\}$. The formula computes a Boolean function in the usual way.⁵ A (Boolean) *DNF formula* is a formula of the form $T_1 \vee T_2 \vee \dots \vee T_m$ for some $m \geq 1$, such that each *term* T_i is the conjunction ($\wedge$) of literals. A literal is a variable x_i or a negated variable $\neg x_i$. The DNF formula is read-once if distinct terms contain disjoint sets of variables, without negations. Read-once DNF formulas whose terms all contain the same number w of literals are sometimes known as tribes formulas of width w (cf. [29]).

2 Warm Up: Adaptivity Gaps for Read-Once DNFs

Let $f : \{0,1\}^n \rightarrow \{0,1\}$ be a read-once DNF formula. Boros and Ünyürlüt [3] showed that the following approach gives an optimal adaptive strategy for evaluating f (this has been rediscovered in later papers [18,27,26]). Let $f = T_1 \vee T_2 \vee \dots \vee T_k$ be a DNF formula with k terms. For each term T_j , let $\ell(j)$ be the number of variables in term T_j . Order the variables of T_j as $x_{j_1}, x_{j_2}, \dots, x_{j_{\ell(j)}}$ in non-decreasing order of the ratio $c_i/(1 - p_i)$, i.e., so that $c_{j_1}/(1 - p_{j_1}) \leq c_{j_2}/(1 - p_{j_2}) \leq \dots \leq c_{j_{\ell(j)}}/(1 - p_{j_{\ell(j)}})$. For evaluating the single term T_j , an optimal strategy tests the variables in T_j sequentially, in the order $x_{j_1}, x_{j_2}, \dots, x_{j_{\ell(j)}}$, until a variable is found to be false, or until all variables are tested and found to be true.

Denote the probability of the term evaluating to true as $P(T_j) = \prod_{i=1}^{\ell(j)} p_{j_i}$ and the expected cost of this evaluation of the term as

$$C(T_j) = \sum_{i=1}^{\ell(j)} \left(\sum_{k=1}^i c_{j_k} \prod_{r=1}^{i-1} p_{j_r} \right).$$

An optimal algorithm for evaluating f applies the above strategy sequentially to the terms T of f , in non-decreasing order of the ratio $C(T)/P(T)$, until either some term is found to be satisfied by x , so $f(x) = 1$, or all terms have been evaluated and found to be falsified by x , so $f(x) = 0$. We will use this optimal adaptive strategy in the remainder of the section.

In what follows, we will frequently describe non-adaptive strategies as performing the n possible tests in a particular order. We mean by this that the permutation representing the strategy lists the tests in this order. The testing stops when the value of f can be determined.

⁵ Some definitions of a read-once formula allow negations in the internal nodes of the formula. By DeMorgan's laws, these negations can be "pushed" into the leaves of the formula, resulting in a formula whose internal nodes are $\vee$ and $\wedge$, such that each variable x_i appears in at most one leaf.

2.1 Unit Costs and the Uniform Distribution

Algorithm 1: Evaluating a read-once DNF where each variable has unit cost and uniform distribution.

Input : $n > 0$, read-once DNF $f : \{0, 1\}^n \rightarrow \{0, 1\}$ with m terms
Output: π // $O(\log n)$ -approximation non-adaptive strategy for f
 $\pi \leftarrow []$ // empty list
for $i = 1$ **to** m **do**
 if $|T_i| \leq 2 \log n$ **then** // T_i is i th shortest term in f
 $\pi \leftarrow \pi + \text{all variables in } T_i$
 else
 $\pi \leftarrow \pi + \text{first } 2 \log n \text{ variables in } T_i$
 end
end
 $\pi \leftarrow \pi + \text{remaining variables not in } \pi$

We begin by showing that the adaptivity gap for read-once DNFs, in the case of unit costs and the uniform distribution, is at most $O(\log n)$.

Theorem 1. *Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$ be a read-once DNF formula. For unit costs and the uniform distribution, there is a non-adaptive strategy S such that $\text{cost}(f, S) \leq O(\log n) \cdot \text{OPT}_{\mathcal{A}}(f)$.*

Proof (Proof Sketch). Using the characterization of the optimal adaptive strategy due to Boros and Ünyülür [3], we show that Algorithm 1 gives a non-adaptive strategy that has expected cost at most $O(\log n)$ times the optimal adaptive strategy. The algorithm crucially relies on the observation that the optimal adaptive algorithm tests terms in non-decreasing order of length for unit costs and the uniform distribution. To see this, observe $C(T)/P(T)$ is non-decreasing when terms are ordered by length in this setting. For terms with length at most $2 \log n$, we can test every variable without paying more than $O(\log n)$ times the optimal adaptive strategy. For terms with length greater than $2 \log n$, we can test $2 \log n$ variables and only need to continue testing with probability $1/n^2$.

We complement Theorem 1 with a matching lower bound. We prove the theorem by exhibiting a read-once DNF with $\sqrt{n}$ identical terms. We upper bound the optimal adaptive strategy and argue any non-adaptive strategy has to make $\log n$ tests per term to verify $f(x) = 0$ which occurs with constant probability.

Theorem 2. *Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$ be a read-once DNF formula. For unit costs and the uniform distribution, $\text{OPT}_{\mathcal{N}}(f) \geq \Omega(\log n) \cdot \text{OPT}_{\mathcal{A}}(f)$.*

2.2 Unit Costs and Arbitrary Probabilities

We give an upper bound of the adaptivity gap for read-once DNF formulas with unit costs and arbitrary probabilities in the special case where all terms have the same number of variables. This is known as a tribes formula [29]. Let the number of terms be m . We now describe two non-adaptive strategies which yield a n/m -approximation and a m -approximation, respectively. Then, by choosing the non-adaptive strategy based on the the number of terms m , we are guaranteed a $\min\{n/m, m\} \leq O(\sqrt{n})$ -approximation.

Lemma 1. *Consider a read-once DNF $f : \{0,1\}^n \rightarrow \{0,1\}$ where each term has the same number of variables. For unit costs and arbitrary probabilities, there is a non-adaptive strategy $S \in \mathcal{N}$ such that $\text{cost}(f, S) \leq n/m \cdot \text{OPT}_{\mathcal{A}}(f)$.*

Proof (Proof of Lemma 1). Consider a random input x and the optimal adaptive strategy described at the start of this section. If $f(x) = 0$, the optimal adaptive strategy must certify that each term is 0 which requires at least m tests. Since any non-adaptive strategy will make at most n tests, the ratio between the cost incurred on x by a non-adaptive strategy, and by the optimal adaptive strategy, is at most n/m . Otherwise, if $f(x) = 1$, the optimal adaptive strategy will certify that a term is true after testing some number of false terms. Now consider the non-adaptive version of this optimal adaptive strategy which tests terms in the same fixed order but must test all variables in a term before proceeding to the next term. For each false term that the optimal adaptive strategy tests, the non-adaptive strategy will test every variable for a total of n/m tests. Since the optimal adaptive strategy must make at least one test per false term, the ratio between the cost incurred on x by the non-adaptive strategy, and the cost incurred by the optimal strategy, is at most n/m . Since the ratio n/m holds for all x , the lemma follows.

Lemma 2. *Consider a read-once DNF $f : \{0,1\}^n \rightarrow \{0,1\}$ where each term has the same number of variables. For unit costs and arbitrary probabilities, there is a non-adaptive strategy $S \in \mathcal{N}$ with expected cost $\text{cost}(f, S) \leq m \cdot \text{OPT}_{\mathcal{A}}(f)$.*

Proof (Proof of Lemma 2). Fix a random input x . If $f(x) = 0$, the optimal adaptive strategy certifies that every term is false. Let C_i be the number of tests it makes until finding a false variable on the i th term. Consider the non-adaptive “round-robin” strategy which progresses in rounds, making one test in each term per round. Within a term, the non-adaptive strategy tests variables in the same fixed order as the optimal adaptive strategy. Then the cost of the non-adaptive strategy is $m \cdot \max_i C_i$ whereas the cost of the optimal adaptive strategy is $\sum_{i=1}^m C_i$. It follows that the adaptivity gap is at most m . Otherwise, if $f(x) = 1$, the optimal adaptive strategy must certify that a term is true by making at least n/m tests. Any non-adaptive strategy will make at most n tests so the adaptivity gap is at most m .

Together, the $O(n/m)$ - and $O(m)$ -approximations imply the following result.

Theorem 3. *Let $f : \{0,1\}^n \rightarrow \{0,1\}$ be a read-once DNF formula where each term has the same number of variables. For unit costs and arbitrary probabilities, there is a non-adaptive strategy $S \in \mathcal{N}$ with $\text{cost}(f, S) \leq O(\sqrt{n}) \cdot \text{OPT}_{\mathcal{A}}(f)$.*

We complement Theorem 3 with a matching lower bound. We prove the theorem by exhibiting a read-once DNF with $2\sqrt{n}$ identical terms. By making one special variable in each term have a low probability of being true and arguing it must always be tested first, the non-adaptive strategy has to search at random for which special variable is true when every other special variable is false which happens with constant probability.

Theorem 4. *Let $f : \{0,1\}^n \rightarrow \{0,1\}$ be a read-once DNF formula. For unit costs and arbitrary probabilities, $\text{OPT}_{\mathcal{N}}(f) \geq \Omega(\sqrt{n}) \cdot \text{OPT}_{\mathcal{A}}(f)$.*

2.3 Arbitrary Costs and the Uniform Distribution

We prove Theorem 5 by exhibiting a read-once DNF with 2^ℓ terms each of length ℓ . Within each term, the cost of each variable increases geometrically with a ratio of 2. The challenge is choosing ℓ so that $2^\ell \ell = n$. We accomplish this by using a modified Lambert W function [6] which is how we calculate n_ϵ .

Theorem 5. *For all $\epsilon > 0$, there exists $n_\epsilon > 0$ such that the following holds for all read-once DNF formulas $f : \{0, 1\}^n \rightarrow \{0, 1\}$ where $n > n_\epsilon$: There exists a cost assignment such that for the uniform distribution, $\text{OPT}_{\mathcal{N}}(f) \geq \Omega(n^{1-\epsilon}/\log n) \cdot \text{OPT}_{\mathcal{A}}(f)$.*

3 Main Result: Read-Once Formulas

Theorem 6. *Fix $\epsilon > 0$. There is a read-once formula $f : \{0, 1\}^n \rightarrow \{0, 1\}$, such that for unit costs and $p_i = \frac{1+\epsilon}{2}$ for all $i \in [n]$, $\text{OPT}_{\mathcal{N}}(f) \geq \Omega(\epsilon^3 n^{1-2\epsilon/\log 2}) \cdot \text{OPT}_{\mathcal{A}}(f)$.*

Before we prove Theorem 6, we describe the read-once formula f and present the technical lemmas we use in the proof. Without loss of generality, assume $n = 2 \cdot 2^d - 2$ for some positive integer d . We define the function $f(x)$ on inputs $x \in \{0, 1\}^n$ in terms of a binary tree with depth d . The edges of the tree are numbered 1 through n , and variable x_i corresponds to edge i . Each variable x_i has a $\frac{1+\epsilon}{2}$ probability of being true. Say that a leaf of the tree is “alive” if $x_i = 1$ for all edges i on the path from the root to the leaf. We define $f(x) = 1$ if and only if at least one leaf of the tree is alive. A strategy for evaluating f will continue testing until it can certify that there is at least one alive leaf, or that no alive leaf exists.

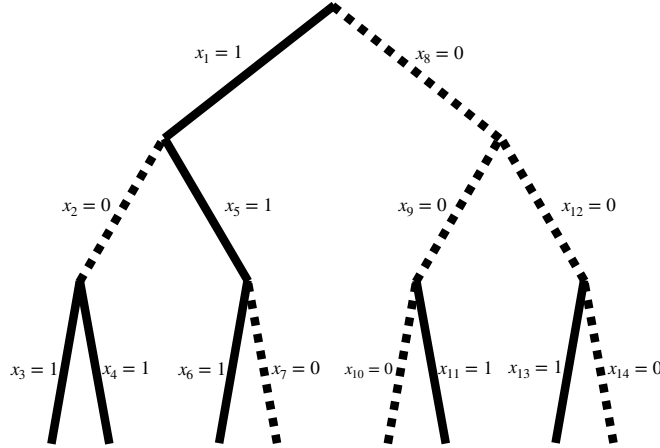


Fig. 1. The binary tree corresponding to the read-once formula we construct when $n = 14$. In particular, $f(x) = (x_1 \wedge ((x_2 \wedge (x_3 \vee x_4)) \vee (x_5 \wedge (x_6 \vee x_7)))) \vee (x_8 \wedge ((x_9 \wedge (x_{10} \vee x_{11})) \vee (x_{12} \wedge (x_{13} \vee x_{14}))))$. Notice that $f(x) = 1$ for this x because the third leaf from the left is alive (all its ancestors are true).

Now consider the multi-graph that is produced from the tree by merging all leaves into a single node. The function f is the st -connectivity function of this multi-graph. It is easy to show, by induction on the depth of the tree, that the multi-graph is two-terminal series-parallel, with s the root, and t the node produced by merging the leaves of the tree. Thus f is computed by a read-once formula.

We refer to the edges of the tree that join a leaf to its parent as leaf edges, and the other edges as internal edges. We say that a non-adaptive strategy S is leaf-last if it first tests all non-leaf edges of the tree, and then tests the leaf edges.

In the proof of Theorem 6, we consider an alternative cost assignment where we pay unit costs for the tests on leaf edges, as usual, but tests on internal edges are free. The expected cost of a strategy under the usual unit cost assignment is clearly lower bounded by its expected cost when internal edges are free. Note that when internal edges are free, there is no disadvantage in performing all the tests on internal edges first, so there is an optimal non-adaptive strategy which is leaf-last in the sense that all the leaf edges appear last. Our first technical lemma describes a property of a leaf-last strategy. We defer the proof of this lemma, and of the ones that follow, to the end of this section. In all of the lemma statements, we assume f is as just described, and expected costs are with respect to unit costs and test probabilities $p_i = \frac{1+\epsilon}{2}$. We use L to denote the number of leaves in the tree.

Lemma 3. *There exists a leaf-last non-adaptive strategy S for evaluating f which, conditioned on the event that there is at least one alive leaf, has minimum expected cost when internal edges are free relative to all non-adaptive strategies. Further, for any such S and any $\ell \in [L - 1]$, conditioned on the existence of at least one alive leaf, the probability that S first finds an alive leaf on the ℓ th leaf test is at least the probability S first finds an alive leaf on the $(\ell + 1)$ st leaf test.*

The next lemma gives us an inequality that we will use to lower bound the cost of the optimal non-adaptive strategy.

Lemma 4. *Let L be a positive integer and $p_1 \geq p_2 \geq \dots \geq p_L$ be non-negative real numbers. Now let $p \geq p_1$ and define $L' = \lfloor \sum_{\ell=1}^L p_\ell / p \rfloor$. Then $\sum_{\ell=1}^L \ell p_\ell \geq \sum_{\ell=1}^{L'} \ell p$.*

Our analysis depends on there being at least constant probability that $f(x) = 1$, or equivalently, that there is at least one alive leaf. The next lemma assures us that this is indeed the case. The proof of the lemma depends on our choice of having each p_i be slightly larger than $1/2$; it would not hold otherwise.

Lemma 5. *With probability at least ϵ , there is at least one alive leaf in the binary tree representing f .*

With these key lemmas in hand, we prove Theorem 6.

Proof (Proof of Theorem 6). We will show that the adaptivity gap is large. Intuitively, we rely on the fact that if there is at least one alive leaf, then an adaptive strategy can find an alive leaf cheaply, by beginning at the root of the tree and moving downward only along edges that are alive. In contrast, a non-adaptive strategy cannot stop searching along “dead” branches. However, it is not immediately clear that the cost of the non-adaptive strategy is high because there are conditional dependencies between the probabilities that two leaves with the same ancestor(s) are alive. To prove the desired result, we need to show that, despite these dependencies, the optimal non-adaptive strategy must have high expected cost.

We begin by showing that the expected cost of any non-adaptive strategy is at least $\frac{\epsilon^2}{16} n^{1 - \frac{\epsilon}{\log 2}}$. We want to lower bound the expected cost of the optimal strategy $\text{OPT}_{\mathcal{N}}(f)$:

$$\min_{S \in \mathcal{N}} \mathbb{E}_x[\text{cost}(f, x, S)] \geq \min_S \mathbb{E}[\text{cost}^L(f, x, S)] = \min_{S'} \mathbb{E}[\text{cost}^L(f, x, S')] \quad (1)$$

where $\text{cost}^L(f, x, S)$ is the number of leaf tests S makes on x until $f(x)$ is determined, and S' is a leaf-last strategy. Then

$$\begin{aligned}
(1) &= \min_{S'} \left(\sum_{x:f(x)=1} \Pr(x) \cdot \text{cost}^L(f, x, S') + \sum_{x:f(x)=0} \Pr(x) \cdot \text{cost}^L(f, x, S') \right) \\
&\geq \min_{S'} \sum_{x:f(x)=1} \Pr(x) \cdot \text{cost}^L(f, x, S') \\
&= \min_{S'} \sum_{\ell=1}^L \ell \Pr(S' \text{ first finds alive leaf on } \ell\text{th leaf test})
\end{aligned}$$

where $L = 2^d$ is the number of leaves in the binary tree. Initially, all leaves have a $(\frac{1+\epsilon}{2})^d$ probability of being alive where $d = \log_2((n+2)/2)$. By Lemma 3, the probability that the next leaf is alive cannot increase as the optimal non-adaptive strategy S^* performs its test. Set

$$p = \left(\frac{1+\epsilon}{2} \right)^d$$

and $p_\ell = \Pr(S^* \text{ first finds alive leaf on } \ell\text{th test})$. Therefore, Lemma 4 with p and p_ℓ tells us that

$$\sum_{\ell=1}^L \ell p_\ell \geq \sum_{\ell=1}^{L'} \ell \left(\frac{1+\epsilon}{2} \right)^d \geq \left(\frac{1+\epsilon}{2} \right)^d \frac{\epsilon^2}{8 \left(\frac{1+\epsilon}{2} \right)^{2d}} \geq \frac{\epsilon^2}{8 \left(\frac{n+2}{2} \right)^{\log_2 \left(\frac{1+\epsilon}{2} \right)}} \geq \frac{\epsilon^2}{16} n^{1 - \frac{\epsilon}{\log 2}} \quad (2)$$

where we use the inequality that $L' \geq \epsilon / (2 \left(\frac{1+\epsilon}{2} \right)^d)$. To see this, recall that $\sum_{\ell=1}^L p_\ell / p \geq L'$ and, since the right-hand side is greater than 1, $2L' \geq \sum_{\ell=1}^L p_\ell / p$. By Lemma 5, $\sum_{\ell=1}^L p_\ell \geq \epsilon$ so $L' \geq \epsilon / (2 \left(\frac{1+\epsilon}{2} \right)^d)$. The last inequality in Equation (2) follows from $\log_2 \left(\frac{1+\epsilon}{2} \right) \leq \frac{\epsilon}{\log 2} - 1$ which can be shown by comparing the y -intercepts and derivatives for $\epsilon > 0$.

Next, we show that the expected cost of the adaptive strategy is at most $(n+1)^{\frac{\epsilon}{\log 2}} / \epsilon$. Consider an adaptive strategy which starts by querying the two edges of the root and recurses as follows: if an edge is alive it queries its two child edges and otherwise stops. Observe that this simple depth-first search adaptive strategy will make at most two tests for every alive edge in the binary tree. Therefore the expected number of tests an adaptive strategy must make is at most twice the expected number of alive edges. By the branching process analysis in the proof of Lemma 5, twice the expected number of alive edges is

$$2 \sum_{i=0}^d (1+\epsilon)^i \leq 2 \sum_{i=0}^{\log_2 n} (1+\epsilon)^i = 2 \frac{(1+\epsilon)^{\log_2(n)+1} - 1}{(1+\epsilon) - 1} \leq 4 \frac{n^{\log_2(1+\epsilon)}}{\epsilon} \leq 4 \frac{n^{\frac{\epsilon}{\log 2}}}{\epsilon} \quad (3)$$

where the last inequality follows from $\log_2(1+\epsilon) \leq \frac{\epsilon}{\log 2}$ which we can see by comparing the y -intercepts and slopes for $\epsilon > 0$. Then Theorem 6 follows from Equations (2) and (3).

Proof (Proof of Lemma 3). A leaf-last non-adaptive strategy S satisfying the given conditional optimality property clearly exists, because any non-adaptive strategy can be made leaf-last by moving the leaf tests to the end without affecting its cost when internal edges are free. Define $p_\ell(S)$ as the probability that S finds an alive leaf for the first time on leaf test ℓ . We may write the expected number of leaf tests of S as $\sum_{\ell=1}^L \ell p_\ell(S)$.

Now suppose for contradiction that there is some ℓ' such that $p_{\ell'}(S) < p_{\ell'+1}(S)$. Let S' be S but with the ℓ' th and $(\ell' + 1)$ th tests swapped. We will show that the expected number of leaf tests made by S' is strictly lower than the expected number of leaf tests made by S . Observe that

$$\sum_{\ell=1}^L \ell p_{\ell}(S) - \sum_{\ell=1}^L \ell p_{\ell}(S') = \ell'(p_{\ell'}(S) - p_{\ell'}(S')) + (\ell' + 1)(p_{\ell'+1}(S) - p_{\ell'+1}(S')).$$

Notice that $p_{\ell'}(S) < p_{\ell'+1}(S) \leq p_{\ell'}(S')$ where the first inequality follows by assumption and the second inequality follows because moving a test on a particular leaf edge to appear earlier in the permutation can only increase the probability that its leaf is the first alive leaf found. In addition, since the combined probability we first find an alive leaf in either the ℓ' th or $(\ell' + 1)$ th test is the same in either order of tests, $p_{\ell'}(S) + p_{\ell'+1}(S) = p_{\ell'}(S') + p_{\ell'+1}(S')$. Together, we have that $-(p_{\ell'}(S) - p_{\ell'}(S')) = p_{\ell'+1}(S) - p_{\ell'+1}(S') > 0$. Therefore $\sum_{\ell=1}^L \ell p_{\ell}(S) - \sum_{\ell=1}^L \ell p_{\ell}(S') = p_{\ell'+1}(S) - p_{\ell'+1}(S') > 0$ and S' makes fewer leaf tests in expectation even though S was optimal by assumption. A contradiction!

Proof (Proof of Lemma 4). Define $\delta_{\ell} = p - p_{\ell} \geq 0$ for $\ell \in [L']$. Observe that $L' \leq L$ since $p \geq p_{\ell}$ for all $\ell \in [L']$. Then

$$\sum_{\ell=1}^L \ell p_{\ell} = \sum_{\ell=1}^{L'} \ell(p - \delta_{\ell}) + \sum_{\ell=L'+1}^L \ell p_{\ell} = \sum_{\ell=1}^{L'} \ell p - \sum_{\ell=1}^{L'} \ell \delta_{\ell} + \sum_{\ell=L'+1}^L \ell p_{\ell}. \quad (4)$$

Now all that remains to be shown is that the sum of the last two terms in Equation (4) is non-negative. Notice that

$$\sum_{\ell=1}^{L'} p_{\ell} + \sum_{\ell=L'+1}^L p_{\ell} \geq \sum_{\ell=1}^{L'} p \implies \sum_{\ell=L'+1}^L p_{\ell} \geq \sum_{\ell=1}^{L'} (p - p_{\ell}) = \sum_{\ell=1}^{L'} \delta_{\ell}.$$

Then

$$\sum_{\ell=1}^{L'} \ell \delta_{\ell} \leq L' \sum_{\ell=1}^{L'} \delta_{\ell} \leq L' \sum_{\ell=L'+1}^L p_{\ell} \leq \sum_{\ell=L'+1}^L \ell p_{\ell}.$$

Proof (Proof of Lemma 5). Let Z_i denote the number of alive edges at level i . Then the statement of Lemma 5 becomes $\Pr(0 < Z_d) \geq \epsilon$. Using standard results from the study of branching processes, we know that

$$\mathbb{E}[Z_d] = \mu^d \quad \text{and} \quad \text{Var}(Z_d) = (\mu^{2d} - \mu^d) \frac{\sigma^2}{\mu(\mu - 1)}$$

where μ is the expectation and σ is the variance of the number of alive “children” from a single alive edge (see e.g., p. 6 in Harris [22]). In our construction,

$$\mu = 2 \left(\frac{1 + \epsilon}{2} \right) = (1 + \epsilon) \quad \text{and} \quad \sigma^2 = 2 \left(\frac{1 + \epsilon}{2} \right) \left(1 - \frac{1 + \epsilon}{2} \right) = \frac{1 - \epsilon^2}{2}$$

since the children of one edge follow the binomial distribution. Then $\mathbb{E}[Z_d] = (1 + \epsilon)^d$ and

$$\text{Var}(Z_d) = ((1 + \epsilon)^{2d} - (1 + \epsilon)^d) \frac{1 - \epsilon^2}{2(1 + \epsilon)\epsilon} \leq \frac{1}{2\epsilon} ((1 + \epsilon)^d)^2 = \frac{1}{2\epsilon} \mathbb{E}[Z_d]^2.$$

We will now use Cantelli's inequality (see page 46 in Boucheron et al. [4]) to show that $\Pr(Z_d > 0) \geq \epsilon$. Cantelli's tells us that $\Pr(X - \mathbb{E}[X] \geq \lambda) \leq \frac{\text{Var}(X)}{\text{Var}(X) + \lambda^2}$ for any real-valued random variable X and $\lambda > 0$. Choose $X = -Z_d$ and $\lambda = \mathbb{E}[Z_d]$. Then

$$\Pr(Z_d \leq \mathbb{E}[Z_d] - \mathbb{E}[Z_d]) \leq \frac{\text{Var}(Z_d)}{\text{Var}(Z_d) + \mathbb{E}[Z_d]^2}$$

and, by taking the complement,

$$\Pr(Z_d > 0) \geq \frac{\mathbb{E}[Z_d]^2}{\text{Var}(Z_d) + \mathbb{E}[Z_d]^2} \geq \frac{\mathbb{E}[Z_d]^2}{\frac{1}{2\epsilon} \mathbb{E}[Z_d]^2 + \mathbb{E}[Z_d]^2} = \frac{2\epsilon}{1 + 2\epsilon} \geq \epsilon \quad (5)$$

for $0 < \epsilon \leq 1/2$. Then Lemma 5 follows from Equation (5).

4 DNF Formulas

We will show near-linear and linear in n lower bounds for DNF formulas under the uniform distribution with unit and arbitrary costs, respectively. Since the function we exhibit has linear terms, the lower bounds also apply to the class of linear-size DNF formulas.

Theorem 7. *Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$ be a DNF formula. For unit costs and the uniform distribution, $\text{OPT}_{\mathcal{N}}(f) \geq \Omega(n/\log n) \cdot \text{OPT}_{\mathcal{A}}(f)$.*

Proof. Without loss of generality, assume $n = 2^d + d$ for some positive integer d . Consider the address function f with 2^d terms which each consist of d shared variables appearing in all terms, and a single dedicated variable appearing only in that term. We may write $f = T_0 \vee T_1 \vee \dots \vee T_{2^d-1}$ where T_i consists of the shared variables negated according to the binary representation of i and the single dedicated variable.

By testing the d shared variables, the optimal adaptive strategy can learn which single term is unresolved and test the corresponding dedicated variable in a total of $d + 1$ tests. In contrast, any non-adaptive strategy has to search for the unresolved dedicated test at random which gives expected $2^d/2$ cost. (We can ensure the non-adaptive strategy tests the shared variables first by making them free which can only decrease the expected cost.) It follows that the adaptivity gap is $\Omega(2^d/d) = \Omega(n/\log n)$.

We can easily modify the address function in the proof of Theorem 7 to prove an $\Omega(n)$ lower bound for DNF formulas under the uniform distribution and arbitrary costs. In particular, make the cost of each shared variable $1/d$. Then the adaptive strategy pays $d \cdot (1/d) + 1 = 2$ while the non-adaptive strategy still pays $\Omega(n)$. The $O(n)$ upper bound comes from the increasing cost strategy and analysis in [25].

Theorem 8. *Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$ be a DNF formula. For the uniform distribution, $\text{OPT}_{\mathcal{N}}(f) \geq \Theta(n) \cdot \text{OPT}_{\mathcal{A}}(f)$.*

5 Conclusion and Open Problems

We have shown bounds on the adaptivity gaps for the SBFE problem for well-studied classes of Boolean formulas. Our proof of the lower bound for read-once formulas depended on having p_i 's that are slightly larger than $1/2$ but we conjecture that a similar or better lower bound holds for the

uniform distribution. We note that our lower bound for read-once formulas also applies to (linear-size) monotone DNF formulas, since the given read-once formula based on the binary tree has a DNF formula with one term per leaf. Another open question is to prove a lower bound for monotone DNF formulas that matches our lower bound for general DNF formulas.

A long-standing open problem is whether the SBFE problem for read-once formulas has a polynomial-time algorithm (cf. [18,30]). The original problem only considered adaptive strategies, and it is also open whether there is a polynomial-time (or pseudo polynomial-time) constant or $\log n$ approximation algorithm for such strategies. Happach et al. [21] gave a pseudo polynomial-time approximation algorithm for the non-adaptive version of the problem, which outputs a non-adaptive strategy with expected cost within a constant factor of the optimal non-adaptive strategy. Because of the large adaptivity gap for read-once formulas, as shown in this paper, the result of Happach et al. does not have any implications for the open question of approximating the adaptive version of the SBFE problem for read-once formulas.

References

1. Agarwal, A., Assadi, S., Khanna, S.: Stochastic submodular cover with limited adaptivity. In: Chan, T.M. (ed.) *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*. pp. 323–342. SIAM (2019)
2. Asadpour, A., Nazerzadeh, H., Saberi, A.: Stochastic submodular maximization. In: Papadimitriou, C.H., Zhang, S. (eds.) *Proceedings of the Internet and Network Economics, 4th International Workshop, WINE*. Lecture Notes in Computer Science, vol. 5385, pp. 477–489. Springer (2008)
3. Boros, E., Unyulurt, T.: Sequential testing of series-parallel systems of small depth. In: Laguna, M., Velarde, J.L.G. (eds.) *Computing Tools for Modeling, Optimization and Simulation: Interfaces in Computer Science and Operations Research*, pp. 39–73. Springer US, Boston, MA (2000)
4. Boucheron, S., Lugosi, G., Massart, P.: *Concentration inequalities: A nonasymptotic theory of independence*. Oxford University Press (2013)
5. Bradac, D., Singla, S., Zuzic, G.: (Near) optimal adaptivity gaps for stochastic multi-value probing. In: Achlioptas, D., Végh, L.A. (eds.) *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2019*, September 20–22, 2019, Massachusetts Institute of Technology, Cambridge, MA, USA. LIPIcs, vol. 145, pp. 49:1–49:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2019)
6. Bronstein, M., Corless, R.M., Davenport, J.H., Jeffrey, D.J.: Algebraic properties of the Lambert W function from a result of Rosenlicht and of Liouville. *Integral Transforms and Special Functions* **19**(10), 709–712 (2008)
7. Dean, B.C., Goemans, M.X., Vondrák, J.: Approximating the stochastic knapsack problem: The benefit of adaptivity. In: *45th Symposium on Foundations of Computer Science (FOCS 2004)*, 17–19 October 2004, Rome, Italy, Proceedings. pp. 208–217. IEEE Computer Society (2004)
8. Dean, B.C., Goemans, M.X., Vondrák, J.: Adaptivity and approximation for stochastic packing problems. In: *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2005*, Vancouver, British Columbia, Canada, January 23–25, 2005. pp. 395–404. SIAM (2005)
9. Dean, B.C., Goemans, M.X., Vondrák, J.: Approximating the stochastic knapsack problem: The benefit of adaptivity. *Math. Oper. Res.* **33**(4), 945–964 (2008)
10. Deshpande, A., Hellerstein, L., Kletenik, D.: Approximation algorithms for stochastic submodular set cover with applications to boolean function evaluation and min-knapsack. *ACM Trans. Algorithms* **12**(3), 42:1–42:28 (Apr 2016)
11. El-Newehi, E., Proschan, F., Sethuraman, J.: Optimal allocation of components in parallel-series and series-parallel systems. *Journal of Applied Probability* **23**(3), 770–777 (1986)
12. Eppstein, D.: Parallel recognition of series-parallel graphs. *Information and Computation* **98**(1), 41–55 (1992)

13. Fu, L., Fu, X., Xu, Z., Peng, Q., Wang, X., Lu, S.: Determining source-destination connectivity in uncertain networks: Modeling and solutions. *IEEE/ACM Trans. Netw.* **25**(6), 3237–3252 (2017)
14. Ghuge, R., Gupta, A., Nagarajan, V.: Non-adaptive stochastic score classification and explainable half-space evaluation. *CoRR* **abs/2111.05687** (2021)
15. Ghuge, R., Gupta, A., Nagarajan, V.: The power of adaptivity for stochastic submodular cover. In: *Proceedings of the 38th International Conference on Machine Learning, ICML (2021)*
16. Gkenosis, D., Grammel, N., Hellerstein, L., Kletenik, D.: The stochastic score classification problem. In: *26th Annual European Symposium on Algorithms, ESA 2018, August 20–22, 2018, Helsinki, Finland*. pp. 36:1–36:14 (2018)
17. Goemans, M.X., Vondrák, J.: Stochastic covering and adaptivity. In: Correa, J.R., Hevia, A., Kiwi, M.A. (eds.) *LATIN 2006: Theoretical Informatics, 7th Latin American Symposium, Valdivia, Chile, March 20–24, 2006, Proceedings. Lecture Notes in Computer Science*, vol. 3887, pp. 532–543. Springer (2006)
18. Greiner, R., Hayward, R., Jankowska, M., Molloy, M.: Finding optimal satisficing strategies for and-or trees. *Artif. Intell.* **170**(1), 19–58 (2006)
19. Gupta, A., Nagarajan, V., Singla, S.: Algorithms and adaptivity gaps for stochastic probing. In: Krauthgamer, R. (ed.) *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10–12, 2016*. pp. 1731–1747. SIAM (2016)
20. Gupta, A., Nagarajan, V., Singla, S.: Adaptivity gaps for stochastic probing: Submodular and XOS functions. In: Klein, P.N. (ed.) *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16–19*. pp. 1688–1702. SIAM (2017)
21. Happach, F., Hellerstein, L., Lidbetter, T.: A general framework for approximating min sum ordering problems. *INFORMS Journal on Computing* (Advance online publication). <https://doi.org/https://doi.org/10.1287/ijoc.2021.1124>
22. Harris, T.E., et al.: *The Theory of Branching Processes*, vol. 6. Springer Berlin (1963)
23. Harvey, N.J., Patrascu, M., Wen, Y., Yekhanin, S., Chan, V.W.: Non-adaptive fault diagnosis for all-optical networks via combinatorial group testing on graphs. In: *IEEE INFOCOM 2007–26th IEEE International Conference on Computer Communications*. pp. 697–705. IEEE (2007)
24. Hellerstein, L., Kletenik, D., Lin, P.: Discrete stochastic submodular maximization: Adaptive vs. non-adaptive vs. offline. In: *Proceedings of the 9th International Conference on Algorithms and Complexity (CIAC) (2015)*
25. Kaplan, H., Kushilevitz, E., Mansour, Y.: Learning with attribute costs. In: Gabow, H.N., Fagin, R. (eds.) *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22–24, 2005*. pp. 356–365. ACM (2005)
26. Kaplan, H., Kushilevitz, E., Mansour, Y.: Learning with attribute costs. In: *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, (STOC)*. pp. 356–365 (2005)
27. Kowshik, H.J.: Information aggregation in sensor networks. University of Illinois at Urbana-Champaign (2011)
28. Liva, G., Paolini, E., Chiani, M.: Optimum detection of defective elements in non-adaptive group testing. In: *2021 55th Annual Conference on Information Sciences and Systems (CISS)*. pp. 1–6. IEEE (2021)
29. O’Donnell, R.: *Analysis of Boolean Functions*. Cambridge University Press, USA (2014)
30. Ünliyurt, T.: Sequential testing of complex systems: a review. *Discret. Appl. Math.* **142**(1–3), 189–205 (2004)
31. Wikipedia: Series and parallel circuits — Wikipedia, the free encyclopedia (2022), [Online; accessed 08-February-2022]

A Additional Proofs

Proof (Proof of Theorem 1). Suppose f is a read-once DNF formula. We will prove that for unit costs and the uniform distribution, there is a non-adaptive strategy S such that $\text{cost}(f, S) \leq O(\log n) \cdot \text{OPT}_A(f)$.

Let m be the number of terms in f . Because each variable x_i appears in at most one term, we have that $m \leq n$. As a warm-up, we begin by proving adaptivity gaps for two special cases of f .

Case 1: All terms have at most $2 \log n$ variables Under the uniform distribution and with unit costs, the p_i are all equal, and the c_i are all equal. Thus in this case, the optimal adaptive strategy described previously tests terms in increasing order of length. The adaptive strategy *skips* in the sense that if it finds a variable in a term that is false, it moves to the next term without testing the remaining variables in the term. Suppose we eliminate skipping from the optimal adaptive strategy, making the strategy non-adaptive. Since all terms have at most $2 \log n$ variables, this increases the testing cost for any given x by a factor of at most $2 \log n$. Thus the cost of evaluating $f(x)$ for a fixed x increases by a factor of at most $2 \log n$ from an optimal adaptive strategy to a non-adaptive strategy, leading to an adaptivity gap of at most $2 \log n$.

Case 2: All terms have more than $2 \log n$ variables Consider the following non-adaptive strategy that operates in two phases. In Phase 1, the strategy tests a fixed subset of $2 \log n$ variables from each term, where the terms are taken in increasing length order. In Phase 2, it tests the remaining untested variables in fixed arbitrary order. Since each term has more than $2 \log n$ variables, the value f can only be determined in Phase 1 if a false variable is found in each term during that phase.

Say that an assignment x is *bad* if the value of f cannot be determined in Phase 1, meaning that a false variable is not found in every term during the phase. The probability that a random x satisfies all the tested $2 \log n$ variables of a particular term is $1/n^{2 \log n}$. Then, by the union bound, the probability that x is bad is at most $m/n^2 \leq n/n^2 = 1/n$.

Now let us focus on the *good* (not bad) assignments x . For each good x , our strategy must find a false variable in each term of f , which requires at least one test per term for any adaptive or non-adaptive strategy. The cost incurred by our non-adaptive strategy on a good x is at most $2m \log n$, since the strategy certifies that $f(x) = 0$ by the end of Phase 1. Therefore, the expected cost incurred by our non-adaptive strategy S is

$$\begin{aligned} \text{cost}(f, S) &\leq \Pr(x \text{ good}) \cdot \mathbb{E}[\text{cost}(f, x, S) | x \text{ good}] \\ &\quad + \Pr(x \text{ bad}) \cdot \mathbb{E}[\text{cost}(f, x, S) | x \text{ bad}] \\ &\leq 1 \cdot 2m \log n + \frac{1}{n} \cdot n \leq 3m \log n \end{aligned}$$

using the fact that $\mathbb{E}[\text{cost}(f, x, S) | x \text{ bad}] \leq n$, since there are only n tests, with unit costs.

The expected cost of any strategy, including the optimal adaptive strategy, is at least

$$\begin{aligned} \text{OPT}_{\mathcal{A}}(f) &\geq \min_{S \in \mathcal{A}} \Pr(f(x) = 0) \cdot \mathbb{E}[\text{cost}(f, x, S) | f(x) = 0] \geq P(f(x) = 0) \cdot m \\ &= (1 - \Pr(f(x) = 1)) \cdot m \geq (1 - \Pr(x \text{ bad})) \cdot m \geq \left(1 - \frac{1}{n}\right) \cdot m \geq \frac{m}{2} \end{aligned}$$

for $n \geq 2$. It follows that the adaptivity gap is at most $6 \log n$.

Case 3: Everything else We now generalize the ideas in the above two cases. Let f be a read-once DNF that does not fall into Case 1 or Case 2. We can break this DNF into two smaller DNFs, $f = f_1 \vee f_2$ where f_1 contains the terms of f of length at most $2 \log n$ and f_2 contains the terms of f of length greater than $2 \log n$.

Let S be the non-adaptive strategy that first applies the strategy in Case 1 to f_1 and then, if $f_1(x) = 0$, the strategy in Case 2 to f_2 . Since S cannot stop testing until it determines the value of f , in the case that $f_1(x) = 0$, it will test all variables in f_1 and then proceed to test variables f_2 .

Let S^* be the optimal adaptive strategy for evaluating read-once DNFs, described above. We know S^* will test terms in non-decreasing order of length since all tests are equivalent. So, like S ,

S^* tests f_1 first and then, if $f_1(x) = 0$, it continues to f_2 . It follows that we can write the expected cost of S on f as

$$\mathbb{E}[\text{cost}(f, x, S)] = \mathbb{E}[\text{cost}(f_1, x, S_1)] + \Pr(f_1(x) = 0) \cdot \mathbb{E}[\text{cost}(f_2, x, S_2) | f_1(x) = 0]$$

where S_1 is the first stage of S , where f_1 is evaluated, and S_2 is the second stage of S , where f_2 is evaluated. Notice that, by the independence of variables, $\mathbb{E}[\text{cost}(f_2, x, S_2) | f_1(x) = 0] = \mathbb{E}[\text{cost}(f_2, x, S_2)]$. We can similarly write the expected cost of S^* on f . Then the adaptivity gap is

$$\frac{\text{OPT}_{\mathcal{N}}(f)}{\text{OPT}_{\mathcal{A}}(f)} \leq \frac{\mathbb{E}[\text{cost}(f_1, x, S_1)] + \Pr(f_1(x) = 0) \cdot \mathbb{E}[\text{cost}(f_2, x, S_2)]}{\mathbb{E}[\text{cost}(f_1, x, S_1^*)] + \Pr(f_1(x) = 0) \cdot \mathbb{E}[\text{cost}(f_2, x, S_2^*)]} \quad (6)$$

where S_1^* is S^* applied to f_1 and S_2^* is S^* beginning from the point when it starts evaluating f_2 .

Using the observation that $(a+b)/(c+d) \leq \max\{a/c, b/d\}$ for positive real numbers a, b, c, d , we know that

$$(6) \leq \max \left\{ \frac{\mathbb{E}[\text{cost}(f_1, x, S_1)]}{\mathbb{E}[\text{cost}(f_1, x, S_1^*)]}, \frac{\mathbb{E}[\text{cost}(f_2, x, S_2)]}{\mathbb{E}[\text{cost}(f_2, x, S_2^*)]} \right\} = O(\log n)$$

where the upper bound follows from the analysis of Cases 1 and 2.

Proof (Proof of Theorem 2). Suppose f is a read-once DNF formula. For unit costs and the uniform distribution, we will show that $\text{OPT}_{\mathcal{N}}(f) \geq \Omega(\log n) \cdot \text{OPT}_{\mathcal{A}}(f)$.

For ease of notation, assume $\sqrt{n}$ is an integer. Consider a read-once DNF f with $\sqrt{n}$ terms where each term has $\sqrt{n}$ variables. By examining the number of tests in each term, we can write the optimal adaptive cost as

$$\text{OPT}_{\mathcal{A}}(f) \leq \sqrt{n} \sum_{i=1}^{\sqrt{n}} \frac{i}{2^i} \leq \sqrt{n} \sum_{i=1}^{\infty} \frac{i}{2^i} = 2\sqrt{n}.$$

The key observation is that, within a term, the adaptive strategy queries variables in any order since each variable is equivalent to any other. Then the probability that the strategy queries exactly $i \leq \sqrt{n}$ variables is $1/2^i$.

Next, we will lower bound the expected cost of the optimal non-adaptive strategy

$$\begin{aligned} \text{OPT}_{\mathcal{N}}(f) &= \min_{S \in \mathcal{N}} \mathbb{E}_{x \sim \{0,1\}^n} [\text{cost}(f, x, S)] \\ &\geq \min_{S \in \mathcal{N}} \Pr(f(x) = 0) \mathbb{E}[\text{cost}(f, x, S) | f(x) = 0] \end{aligned}$$

where $x \sim \{0,1\}^n$ indicates x is drawn from the uniform distribution. First, we know $\Pr(f(x) = 0) \geq .5$. To see this, consider a random input $x \sim \{0,1\}^n$. The probability that a particular term is true is $1/2^{\sqrt{n}}$ so the probability that all terms are false (i.e., $f(x) = 0$) is

$$\left(1 - \frac{1}{2^{\sqrt{n}}}\right)^{\sqrt{n}} = \left(\left(1 - \frac{1}{2^{\sqrt{n}}}\right)^{2^{\sqrt{n}}}\right)^{\sqrt{n}/2^{\sqrt{n}}} \geq \left(\frac{1}{2e}\right)^{\sqrt{n}/2^{\sqrt{n}}} \geq .5$$

where the first inequality follows from the loose lower bound that $(1 - 1/x)^x \geq 1/(2e)$ when $x \geq 2$ and the second inequality follows when $n \geq 8$. Second, we know

$$\begin{aligned} &\mathbb{E}[\text{cost}(f, x, S) | f(x) = 0] \\ &\geq \Pr(\text{one term needs } \Omega(\log n) \text{ tests} | f(x) = 0) \cdot \frac{\log_4 n}{2} \cdot \frac{\sqrt{n}}{2} \end{aligned}$$

where we used the symmetry of the terms to conclude that if any term needs $\Omega(\log n)$ tests to evaluate it then any non-adaptive strategy will have to spend $\Omega(\log n)$ on half the terms in expectation.

All that remains is to lower bound the probability one term requires $\Omega(\log n)$ tests given $f(x) = 0$. Observe that this probability is

$$\begin{aligned} & 1 - (1 - \Pr(\text{a particular term needs } \Omega(\log n) \text{ tests} | f(x) = 0))^{\sqrt{n}} \\ & \geq 1 - \left(1 - \frac{1}{\sqrt{n}}\right)^{\sqrt{n}} \geq 1 - \frac{1}{e} \geq .63 \end{aligned}$$

where we will now show the first inequality. We can write the probability that a particular term needs $\log_4(n)/2$ tests given $f(x) = 0$ as

$$\begin{aligned} & \Pr(x_1 = 1 | f(x) = 0) \cdots \Pr(x_{\log_4(n)/2} = 1 | f(x) = 0, x_1 = \cdots = x_{\log_4(n)/2-1} = 1) \\ & = \frac{2^{\sqrt{n}-1} - 1}{2^{\sqrt{n}} - 1} \cdots \frac{2^{\sqrt{n}-1-\log_4(n)/2} - 1}{2^{\sqrt{n}-\log_4(n)/2} - 1} \geq \left(\frac{2^{\sqrt{n}-1-\log_4(n)/2} - 1}{2^{\sqrt{n}-\log_4(n)/2} - 1}\right)^{\log_4(n)/2} \\ & \geq \left(\frac{1}{4}\right)^{\log_4(n)/2} = \frac{1}{\sqrt{n}}. \end{aligned}$$

For the first equality, we use the observation that conditioning on $f(x) = 0$ eliminates the possibility every variable is true so the probability of observing a true variable is slightly smaller. For the first inequality, notice that $(2^{i-1} - 1)/(2^i - 1)$ is monotone increasing in i . For the second, observe that $i \geq \sqrt{n} - \log_4(n)/2$ for our purposes and so $(2^{i-1} - 1)/(2^i - 1) \geq 1/4$ when $n \geq 16$.

Proof (Proof of Theorem 4). Suppose f is a read-once DNF. For unit costs and arbitrary probabilities, we prove $\text{OPT}_{\mathcal{N}}(f) \geq \Omega(\sqrt{n}) \cdot \text{OPT}_{\mathcal{A}}(f)$.

Consider the read-once DNF with $m = 2\sqrt{n}$ identical terms where each term has $\ell = \sqrt{n}/2$ variables. In each term, let one variable have $1/\ell$ probability of being true and the remaining variables have a $(\ell/m)^{1/(\ell-1)}$ probability of being true. Within a term, the optimal adaptive strategy will test the variable with the lowest probability of being true first. Using this observation, we can write

$$\begin{aligned} \text{OPT}_{\mathcal{A}}(f) & \leq [\Pr(x_1 = 0) \cdot 1 + \Pr(x_1 = 1) \cdot \ell] \cdot m \\ & \leq [(1 - 1/\ell) \cdot 1 + (1/\ell) \cdot \ell] \cdot m \leq 4\sqrt{n} \end{aligned}$$

where x_1 is the first variable tested in each term. The first inequality follows by charging the optimal adaptive strategy for all ℓ tests in the term if the first one is true. The second inequality follows since the variable with probability $1/\ell$ of being true is tested first for $n \geq 18$ (i.e., $1/\ell < (\ell/m)^{1/(\ell-1)}$ for such n).

In order to lower bound the cost of the optimal non-adaptive strategy, we will argue that there is a constant probability of an event where the non-adaptive strategy has to test $\Omega(n)$ variables. In particular,

$$\begin{aligned} \text{OPT}_{\mathcal{N}}(f) & \geq \min_{S \in \mathcal{N}} \Pr(\text{exactly one term is true}) \\ & \quad \cdot \mathbb{E}[\text{cost}(f, x, S) | \text{exactly one term is true}]. \end{aligned}$$

By the symmetry of the terms, observe that

$$\mathbb{E}[\text{cost}(f, x, S) | \text{exactly one term is true}] \geq \sqrt{n}/2 \cdot \sqrt{n} = n/2.$$

That is, the optimal non-adaptive strategy has to search blindly for the single true term among all $2\sqrt{n}$ terms, making $\sqrt{n}/2$ tests each for half the terms in expectation.

All that remains is to show there is a constant probability exactly one term is true. The probability a particular term is true is $(1/\ell)((\ell/m)^{1/(\ell-1)})^{(\ell-1)} = 1/m$. Since all variables are independent, the probability that exactly one of the m terms is true is

$$\begin{aligned} & m \cdot \Pr(\text{a term is true}) \cdot \Pr(\text{a term is false})^{m-1} \\ &= m \cdot \frac{1}{m} \cdot \left(1 - \frac{1}{m}\right)^{m-1} \geq \frac{1}{2e} \stackrel{(m-1)/m}{\geq} \frac{1}{2e}. \end{aligned}$$

It follows that $\text{OPT}_{\mathcal{N}}(f) \geq \frac{1}{2e} \cdot \frac{n}{2} = \Omega(n)$ so the adaptivity gap is $\Omega(\sqrt{n})$.

Proof (Proof of Theorem 5). Suppose f is a read-once formula. For arbitrary costs and the uniform distribution, $\text{OPT}_{\mathcal{N}}(f) \geq \Omega(n^{1-\epsilon}/\log n) \cdot \text{OPT}_{\mathcal{A}}(f)$.

Define $W(w) := w^{1-\epsilon} \log_2(w^{1-\epsilon})$ for positive real numbers w .⁶ We will choose n_ϵ in terms of the function W so that $W(n) < n$ for $n \geq n_\epsilon$. First, consider the first and second derivatives of W :

$$\begin{aligned} W'(w) &= \frac{1-\epsilon}{w^\epsilon} \left(\log_2(w^{1-\epsilon}) + \frac{1}{\log 2} \right) \\ W''(w) &= \frac{1-\epsilon}{w^{1+\epsilon}} \left[-\epsilon \left(\log_2(w^{1-\epsilon}) + \frac{1}{\log 2} \right) + \frac{1-\epsilon}{\log 2} \right]. \end{aligned}$$

For fixed $\epsilon > 0$, observe that as w goes to infinity, $W(w) < w$, $W'(w) < 1$, and $W''(w) < 0$. Therefore there is some point n_ϵ so that for all $n \geq n_\epsilon$, the slope of W is decreasing, the slope of W is less than the slope of n , and $W(n)$ is less than n . Equivalently, $n \geq W(n) = n^{1-\epsilon} \log_2(n^{1-\epsilon})$. We will use this inequality when lower bounding the asymptotic behavior of the adaptivity gap.

For $n \geq n_\epsilon$ we construct the n -variable read-once DNF formula f as follows. First, let r_n be a real number such that $n = n^{1-r_n} \log_2(n^{1-r_n})$. We know that r_n exists for all $n \geq 4$ by continuity since $n^{1-0} \log_2(n^{1-0}) \geq n \geq n^{1-1} \log_2(n^{1-1})$. Let f be the read-once DNF formula with m terms of length ℓ , where $\ell = \log_2(n^{1-r_n})$ and $m = 2^\ell$. Thus the total number of variables in f is $m\ell = n^{1-r_n} \log_2(n^{1-r_n}) = n$ as desired. We assume for simplicity that ℓ is an integer. The bound holds by a similar proof without this assumption.

To obtain our lower bound on evaluating this formula, we consider expected evaluation cost with respect to the uniform distribution and the following cost assignment: in each term, choose an arbitrary ordering of the variables and set the cost of testing the i th variable in the term to be 2^{i-1} .

Consider a particular term. Recall the optimal adaptive strategy for evaluating a read-once DNF formula presented at the start of Section 2. Within a term, this optimal strategy tests the variables in non-decreasing cost order, since each variable has the same probability of being true. Since it performs tests within a term until finding a false variable or certifying the term is true, we can upper bound the expected cost of this optimal adaptive strategy in evaluating f as follows:

$$\text{OPT}_{\mathcal{A}}(f) \leq m \cdot \left[\frac{1}{2} \cdot (1) + \frac{1}{4} \cdot (1+2) + \dots + \frac{1}{2^\ell} \cdot (1 + \dots + 2^{\ell-1}) \right] \leq m \cdot \ell.$$

In contrast, the optimal non-adaptive strategy does not have the advantage of stopping tests in a term when it finds a false variable. We will lower bound the expected cost of the optimal non-adaptive strategy in the case that exactly one term is true. By symmetry, any non-adaptive strategy will have to randomly search for the term and so pay 2^ℓ for half the terms in expectation.

⁶ Notice that W is similar to a Lambert W function $e^y y$, after changing the base of the logarithm and substituting $y = \log(w^{1-\epsilon})$ [6].

All that remains is to show there is a constant probability exactly one term is true. The probability that a particular term is true is $1/2^\ell$ and so the probability that exactly one term is true is

$$m \cdot \frac{1}{2^\ell} \cdot \left(1 - \frac{1}{2^\ell}\right)^{m-1} \geq \frac{m}{2^\ell} \cdot \left(\frac{1}{2e}\right)^{(m-1)/2^\ell} \geq \frac{1}{2e}$$

where the last inequality follows since $m = 2^\ell$. Then the expected cost $\text{OPT}_{\mathcal{N}}(f)$ of the optimal non-adaptive strategy is at least

$$\Pr(\text{exactly one term is true}) \cdot 2^\ell \cdot \frac{m}{2} = \Omega(m \cdot 2^\ell) = \Omega(m \cdot n^{1-r_n}) \geq \Omega(m \cdot n^{1-\epsilon})$$

where we used that $2^\ell = n^{1-r_n}$ and $n^{1-r_n} \log_2(n^{1-r_n}) = n \geq n^{1-\epsilon} \log_2(n^{1-\epsilon})$ since $n \geq n_\epsilon$. It follows that the adaptivity gap is $\Omega(n^{1-\epsilon}/\log n)$.