Abstract We characterize the best possible trade-off achievable when optimizing the construction of a decision tree with respect to both the worst and the expected cost. It is known that a decision tree achieving the minimum possible worst case cost can behave very poorly in expectation (even exponentially worse than the optimal), and the vice versa is also true. Led by applications where deciding which optimization criterion might not be easy, several authors recently have focussed on the bicriteria optimization of decision trees. Here we sharply define the limits of the best possible trade-offs between expected and worst case cost. More precisely, we show that for every $\rho > 0$ there is a decision tree $D$ with worst testing cost at most $(1 + \rho)OPT_W$ and expected testing cost at most $\frac{1}{1-e^{-\rho}}OPT_E$, where $OPT_W$ and $OPT_E$ denote the minimum worst testing cost and the minimum expected testing cost of a decision tree for the given instance. We also show that this is the best possible trade-off in the sense that there are infinitely many instances for which we cannot obtain a decision tree with both worst testing cost smaller than $(1 + \rho)OPT_W$ and expected testing cost smaller than $\frac{1}{1-e^{-\rho}}OPT_E$.

Keywords Decision trees · Approximation algorithms · Combinatorial optimization
1 Introduction

We consider a very general model of the decision tree construction problem: We have a set of objects $S = \{s_1, \ldots, s_n\}$ which is partitioned into $m$ classes $C_1, \ldots, C_m$. Objects are characterized by the value they take with respect to a set of tests $T$. Each test $t \in T$ has a finite number of possible values, which we assume are bounded above by some fixed value $\ell$. Each test $t$ has also an associated rational positive cost $c(t)$ which has to be paid in order to use the test. The goal is to design a procedure for discovering the classification of an unknown object in the most efficient way, where efficiency is measured with respect to the (expected) cost of the tests used. Tests are performed to acquire information on the object to classify. Each new test performed (adaptively chosen from $T$ on the basis of the result of the previous tests) reveals the value taken for the object to classify. Hence performing a test restricts the set of possible classifications to those of the objects matching the result of the test. The procedure stops when the objects agreeing with the results of the tests performed belong to the same class, which must also be the class of the object that had to be classified. We assume that the set of tests is complete. This means that for any of objects from distinct classes, there exists a test $t$ that separates them, that is, it has different values for the two objects. We also assume a probability distribution $p$ over the set of objects. This distribution reflects the belief to which objects are chosen.

More formally, by an instance $I$ we mean a quintuple $I = (S, C, T, p, c)$ where $S$ is the set of objects; $C$ is the family of classes defining the partition of $S$; $T$ is the set of tests: each test $t \in T$ defines a partition $\{S^1_t, \ldots, S^\ell_t\}$ for the objects of set $S$, where $S^i_t$ is the set of objects in $S$ for which $t$ outputs value $i$; $p$ is the probability distribution on $S$ and $c$ is the cost function assigning a non-negative cost to each test in $T$.

A typical application of the above model is to a diagnosis problem. For instance the objects could represent diseases, e.g., $\{\text{flu}, \text{dengue}, \text{cancer}\}$, divided into infectious and non-infectious, e.g., infectious $= \{\text{flu}\}$ and non-infectious $= \{\text{dengue, cancer}\}$. Tests might be medical tests with different costs and the probability might refer to the prior knowledge of the incidence of the three possible diseases among the patients who are going to ask to be diagnosed. The goal is to have a strategy that can be used to quickly/cheaply determine whether a patient who is ill, and whose disease is not known, is infectious or not.

Any testing procedure can be represented by a decision tree, which is a tree where every internal node is associated with a test. The branches stemming out from a node are associated with the possible outcomes of the test associated with the node. Every leaf is associated with a set of objects that belong to the same class. More formally, a decision tree $D$ over the set of objects $S$ can be inductively defined as follows: (1) if every object of $S$ belongs to the same class $i$, then $D$ is a single leaf associated with class $i$; (2) otherwise, the root $r$ of $D$ is associated with some test $t \in T$ and the children of $r$ are decision trees for the non empty sets in the partition $\{S^1_t, \ldots, S^\ell_t\}$ of $S$. We also assume that there exist at least two non-empty sets in the partition. This latter condition is meant to avoid considering decision trees where some nodes do not provide any information to the classification procedure.

Given a decision tree $D$, rooted at $r$, we can identify the class of an object $s$ by following a path from $r$ to a leaf as follows: first, we ask for the result of the test
associated with $r$ when performed on $s$; then, we follow the branch of $r$ associated with the result of the test to reach a child $r'$ of $r$; next, we apply the same steps recursively starting from $r'$. The procedure ends when a leaf is reached, which determines the class of $s$. We also say that this is the leaf associated to $s$.

We define $\text{cost}(D, s)$ as the sum of the tests’ cost on the path from the root of $D$ to the leaf associated with object $s$. Then, the worst testing cost and the expected testing cost of $D$ are, respectively, defined as

$$
\text{cost}_W(D) = \max_{s \in S} \{\text{cost}(D, s)\} \quad \text{and} \quad \text{cost}_E(D) = \sum_{s \in S} \text{cost}(D, s) p(s)
$$

Figure 1 shows a decision tree $D$ for 3 objects: The instance is given by the set of objects $S = \{s_1, s_2, s_3\}$, with probabilities $p(s_1) = 0.4$, $p(s_2) = 0.5$ and $p(s_3) = 0.1$; each object $s_i$ belongs to a different class $C_i = \{s_i\}$; there are two tests, $t_1, t_2$ with costs $c(t_1) = 1$, $c(t_2) = 2$, respectively. Then, for the decision tree $D$ depicted in the figure, we have $\text{cost}_W(D) = \max\{1, 1+2\} = 3$ and $\text{cost}_E(D) = 0.4 \times 1 + 0.5 \times 3 + 0.1 \times 3 = 2.2$.

Most of the research on decision tree optimization focuses on building a decision tree that minimizes only one of the above measures [1,5,7,12–15,18,23]. From an application point of view, the choice of the optimization criterion reflects different assumptions on the data model: a more optimistic perspective on the knowledge of the underlying distribution might elicit the minimization of the expected testing cost; a more pessimistic perspective might prefer the more conservative minimization of the worst case testing cost.

However, the two different optimization criteria can lead to very different trees: a decision tree minimizing the expected cost for a very skewed distribution can have a skewed shape with a very high worst case cost, even exponentially larger than the worst cost of a decision tree optimized with respect to the worst testing cost. Conversely, optimizing with respect to the worst testing cost can lead to a tree with poor performance in expectation.

As an example let us consider the problem of constructing a prefix code for an alphabet with $n$ symbols [26]. This is a particular case of the decision tree optimization problem given above, where each symbol corresponds to an object, each one of the $n$ symbols (objects) belongs to a different class, the testing costs are uniform and the set of tests is in one to one correspondence with the set of all binary strings of length $n$. In particular, a test outputs 0 (resp. 1) for an object $s_i$ if the $i$th bit of the binary string associated to the test is 0 (resp. 1).
Let us consider the case where the probability distribution on the objects is given by $p(s_i) = 2^{-i}$ for each $i = 1, \ldots, n - 1$ and $p(s_n) = 2^{-(n-1)}$. Let $D^*_E$ and $D^*_W$ be decision trees with, respectively, minimum expected cost and minimum worst testing cost for the instance. $D^*_E$ can be constructed by the Huffman’s algorithm, and it is not difficult to verify that we have $\text{cost}_E(D^*_E) \leq 3$ and $\text{cost}_W(D^*_E) = n - 1$. In addition, one possibility for $D^*_W$ is the decision tree that implements a binary search and, in this case, we have that $\text{cost}_E(D^*_W) = \text{cost}_W(D^*_W) = \Theta(\log n)$. Therefore, we have here an example where the minimization of the expected testing cost produces a decision tree whose worst testing cost is exponentially worse than the cost of the worst-cost-optimal tree and vice versa the minimization of the worst testing cost produces a decision tree whose expected testing cost is $\Theta(\log n)$ larger than the expected testing cost of the decision tree that minimizes this measure.

The choice of the “wrong” optimization criterion might have serious consequences in practical applications (see, e.g., [17] for such a study in the economics literature). Therefore, it makes sense to look for a trade-off between minimizing these two measures.

These arguments have motivated recent work on decision tree constructions optimizing both worst and the expected testing cost [3,9,16,19]. In [9], which was our starting point in this line of research, we provided an algorithm that builds a decision tree guaranteeing simultaneously $O(\log n)$-approximation for both worst and expected testing costs, which is the best possible approximation achievable for either criterion under standard complexity assumptions.

Here, we address and exactly answer a more fundamental question regarding the existence of arbitrarily good trade-offs between expected and worst cost: does there exist in general (asymptotically for any instance) a decision tree with worst testing cost and expected testing cost arbitrarily close, respectively, to the optimal worst testing cost and the optimal expected testing cost? Or, otherwise, what is the threshold for the best trade-off we can hope for?

### 1.1 Our Results

In Section 2, we show that for every $\rho > 0$ and every instance $I$ there exists a decision tree $D$ with worst testing cost at most $(1 + \rho)\text{OPT}_W(I)$ and expected testing cost at most $(1 - e^{-\rho})\text{OPT}_E(I)$, where $\text{OPT}_W(I)$ (resp. $\text{OPT}_E(I)$) denotes the cost of the decision tree with minimum worst testing cost (resp. minimum expected testing cost) for the instance $I$.

We then show, in Sect. 3, that this is a sharp characterization of the best possible trade-off attainable, in the sense that there are infinitely many instances for which we cannot obtain a decision tree with both worst testing cost smaller than $(1 + \rho)\text{OPT}_W(I)$ and expected testing cost smaller than $(1 - e^{-\rho})\text{OPT}_E(I)$.

To obtain the upper bound, we present a general procedure that merges decision trees built according to different optimization criteria: given a parameter $\rho > 0$, a decision tree $D_W$ with worst testing cost $W$ and a decision tree $D_E$ with expected testing cost $E$, our merging procedure produces a decision tree $D$ with worst testing...
cost at most \((1 + \rho)W\) and expected testing cost at most \(\frac{1}{1-e^{-\rho}}E\). For the analysis of our procedure we employ techniques from non-linear programming (NLP) [4].

For the lower bound, we make use of the the probability distribution used in the analysis of the upper bound—obtained by the optimal solution of the NLP—as a starting point for constructing non-trivial instances that guarantee that the upper bound is tight.

The upper bound achieved and the techniques we employed are similar to those in [2,25] to obtain tight trade-offs between the minimization of the expected completion time and the makespan for scheduling problems. On the other hand, these results give no information on how to obtain a tight lower bound for our problem.

### 1.2 Related Work

There are some studies related to the simultaneous minimization of the expected testing cost and the worst testing cost for the prefix code problem [6,11,20–22]. As explained above, the problem of constructing a prefix code is a particular case of decision tree optimization.

In contrast to our present findings, the results of Milidiu and Laber [22] imply that in the case of the prefix code problem, asymptotically, there exists a decision tree that is arbitrarily close to the optimum with respect to both expected and worst cost. More precisely, for every instance \(I\) with \(n\) objects and any \(\rho > 0\), there is a decision tree \(D\) such that \(\text{cost}_W(D)/\text{OPT}_W(I) \leq (1+\rho)\) and \(\text{cost}_E(D)/\text{OPT}_E(I) \leq 1+1/\psi^\rho \log n^{-1}\), where \(\psi\) is the golden ratio \((1 + \sqrt{5})/2\).

### 2 Trade-Off: Upper Bound

In this section, we show our upper bound on the achievable trade-off between worst and expected testing cost for the decision tree optimization problem. Our proof is constructive, that is, we show a procedure for constructing a decision tree guaranteeing the desired trade off.

Given a positive number \(j\), and two decision trees \(D_E\) and \(D_W\) for instance \(I\), the procedure CombineTrees(\(D_E, D_W, j\)) (see Algorithm 1) constructs a new decision tree \(D^j\) for \(I\) whose worst testing cost is increased by at most \(j\) w.r.t the worst testing cost of \(D_W\), i.e., \(\text{cost}_W(D^j) \leq j + \text{cost}_W(D_W)\). Our algorithm uses the definition of a \(j\)-replaceable node, by which we mean a node \(v\) in \(D\) such that the total cost of the tests on the path from the root of \(D\) to \(v\) (including \(v\)) is larger than \(j\) and the cost of the path from the root of \(D\) to the parent of \(v\) is smaller than or equal to \(j\). The procedure Trade-Off repeatedly uses CombineTrees to create several decision trees (one of these trees being \(D_W\)) with increasingly worst testing cost and chooses the one with the best expected testing cost. We will show that this way it can guarantee the best possible trade off.
Algorithm 1 Computes trade off tree between $D_W$ and $D_E$

Procedure CombineTrees($D_E, D_W, j$)
1: $D^j \leftarrow D_E$
2: Traverse $D^j$ and construct $R = \{v \mid v \text{ is a } j\text{-replaceable node of } D^j\}$
3: for each $v \in R$ do
4: Replace in $D^j$ the subtree rooted at $v$ with $D_W$
5: return $D^j$

Procedure Trade-Off($D_E, D_W, C$)
1: for $j = 0, \ldots, C$ do
2: $D^j \leftarrow \text{CombineTrees}(D_E, D_W, j)$
3: $j^* \leftarrow \arg \min_{0 \leq j \leq C} \text{cost}_E(D^j)$
4: return $D^{j^*}$

Proposition 1 The decision tree $D^j$ returned by CombineTrees has worst testing cost at most $j + \text{cost}_W(D_W)$.

Proof Let $s$ be an object in $S$. In the following, we identify $s$ with the leaf associated to it. To establish the proof we show that for all $s \in S$ it holds that $\text{cost}(D^j, s) \leq j + \text{cost}_W(D_W)$.

If $s$ is not a descendant of a replaceable node in $D_E$ then the cost of the path from the root of $D_E$ to $s$ is at most $j$. Since this path remains the same in $D^j$, we have that $\text{cost}_W(D^j, s) \leq j$. On the other hand, if $s$ is a descendant of a replaceable node $v$ in $D_E$, then $\text{cost}(D^j, s) \leq j + \text{cost}_W(D_W)$ because $\text{cost}(D^j, s)$ is the sum of (1) the cost of the path from the root of $D^j$ to the parent of $v$, which is at most $j$, and (2) the cost to reach $s$ in the decision tree $D_W$, which is at most $\text{cost}_W(D_W)$. □

Now we analyze the decision tree $D = D^{j^*}$ output by Trade-Off($D_E, D_W, C$), where $C$ is an integer parameter. Notice that $D$ is the decision tree with minimum expected testing cost among the decision trees $D^0, D^1, D^2, \ldots, D^C$, where $D^j$ is the decision tree returned by CombineTrees($D_E, D_W, j$). It follows from the previous proposition that $\text{cost}_W(D) \leq C + \text{cost}_W(D_W)$.

The analysis of the expected testing cost of $D$ is more involved. In order to simplify the notation we will let $W = \text{cost}_W(D_W)$. We also assume for simplicity in the following that test costs are integers. Given a decision tree $D'$ and an object/leaf $s \in S$ with $\text{cost}(D', s) = \kappa$ we will say that $s$ has cost $\kappa$ in $D'$.

Let $p_i$, with $i = 1, \ldots, C$, be the sum of the probabilities of objects with cost $i$ in $D_E$ and $p_{C + 1}$ be the sum of the probabilities of the objects with cost larger than $C$ in $D_E$. Clearly:

$$\text{cost}_E(D_E) \geq \sum_{i=1}^{C+1} p_i \cdot i$$

Furthermore, for $j = 0, \ldots, C$, we have that:

$$\text{cost}_E(D^j) \leq \sum_{i=1}^{j} p_i \cdot i + \left( j + W \sum_{i=j+1}^{C+1} p_i \right)$$
because the objects whose cost in $D_E$ is larger than $j$ have cost at most $j + W$ in $D^j$. Thus,

$$\frac{\text{cost}_E(D)}{\text{cost}_E(D_E)} = \min_{j=0, \ldots, C} \frac{\text{cost}_E(D^j)}{\text{cost}_E(D_E)}$$  \quad (2)$$

$$\leq \min_{j=0, \ldots, C} \left\{ \frac{\sum_{i=1}^{j} p_i \cdot i + (j + W) \sum_{i=j+1}^{C+1} p_i}{\sum_{i=1}^{C+1} p_i \cdot i} \right\}$$  \quad (3)$$

$$\leq \max_{p \in \mathcal{P}} \min_{j=0, \ldots, C} \left\{ \frac{\sum_{i=1}^{j} p_i \cdot i + (j + W) \sum_{i=j+1}^{C+1} p_i}{\sum_{i=1}^{C+1} p_i \cdot i} \right\},$$  \quad (4)$$

where $\mathcal{P} = \{(p_1, p_2, \ldots, p_{C+1})| \sum_{i=1}^{C+1} p_i = 1$ and $p_1, p_2, \ldots, p_{C+1} \geq 0\}$. Thus, we can conclude that $\text{cost}_E(D) / \text{cost}_E(D_E) \leq z^*$, where $z^*$ is the maximum achieved by the following non linear program(NLP):

$$z^* = \max z \quad \text{s.t.}$$

$$z \left( \sum_{i=1}^{C+1} i \cdot p_i \right) - \sum_{i=1}^{j} i \cdot p_i - (j + W) \left( \sum_{i=j+1}^{C+1} p_i \right) \leq 0, \quad j = 0, \ldots, C$$  \quad (6)$$

$$\sum_{i=1}^{C+1} p_i = 1,$$  \quad (7)$$

$$p_i \geq 0, \quad i = 1, \ldots, C + 1$$  \quad (8)$$

Perhaps surprisingly we can prove that the optimal solution $p^* = (p^*_1, p^*_2, \ldots, p^*_{C+1}, z^*)$ of the NLP is given by

$$p^*_i = \begin{cases} 
\frac{(W-1)^{i-1}}{W^i}, & i = 1, \ldots, C \\
\frac{(W-1)^C}{W^C}, & i = C + 1 
\end{cases}$$  \quad (9)$$

$$z^* = \frac{1}{1 - \left( \frac{W-1}{W} \right)^{C+1}}$$  \quad (10)$$

The proof, presented in the appendix, consists of expanding the feasible region of the NLP in order to get an open set $X$; showing that the functions that define the $C + 1$ non linear constraints are convex in $X$; showing the point $p^*$ is feasible and, finally, showing that it satisfies the Karush-Kuhn-Tucker conditions [4]. Thus, by setting $C = \lfloor \rho W \rfloor$ we get the following theorem.

**Theorem 1** Fix an instance $I$ of the decision tree optimization problem and let $D_E$ be a decision tree such that $\text{cost}_E(D_E) = \text{OPT}_E(I)$. For every $\rho > 0$ there exists a decision tree $D$ such that
cost_W(D) \leq (1 + \rho)OPT_W(I) \quad \text{and} \quad cost_E(D) \leq \left( \frac{1}{1 - \rho} \right) OPT_E(I).

**Proof** Let \( W = OPT_W(I), \) and \( C = \lfloor \rho W \rfloor. \) Let \( D_W \) be a decision tree such that \( cost_W(D_W) = W. \) It follows from the analysis above that the decision tree \( D \) output by \( \text{Trade-Off}(D_E, D_W, C) \) has worst testing cost at most \( C + W < (1 + \rho)W \) and expected testing cost smaller than

\[
\frac{1}{1 - \left( \frac{W-1}{W} \right)^{C+1}} \text{OPT}_E(I) \leq \left( \frac{1}{1 - \left( \frac{W-1}{W} \right)^{\rho W}} \right) \text{OPT}_E(I) \leq \left( \frac{1}{1 - \rho} \right) \text{OPT}_E(I)
\]

\( \square \)

### 3 Trade-Off: Lower Bound

In this section we show that the result of Theorem 1 is tight in the sense that no better trade off is possible in general. This is proved in Theorem 3 below. The main ingredient of this result is the construction of a family of instances for which the following theorem holds.

**Theorem 2** Fix integers \( W > 1 \) and \( C > 0, \) and let \( \eta = \eta(W, C) \) be a number such that \( \eta < \frac{1}{W^{2(C+1)}}. \) There exists an instance \( I \) such that the following hold:

1. \( OPT_W(I) \leq W. \)
2. \( OPT_E(I) \leq (1 - \eta) \left( W \left( 1 - \left( \frac{W-1}{W} \right)^{C} \right) + \lfloor \log W \rfloor \left( \frac{W-1}{W} \right)^{C} \right) + (W + C + \lfloor \log W \rfloor) \eta. \)
3. If a decision tree \( D \) for \( I \) is such that \( cost_W(D) \leq W + C \) then it holds that \( cost_E(D) \geq W(1 - \eta) - \eta C. \)

**Theorem 3** For any fixed \( \rho > 0 \) and \( \epsilon > 0, \) there are infinitely many instances \( I \) of the decision tree problem such that no decision tree can simultaneously guarantee worst testing cost smaller than \( OPT_W(I)(1 + \rho) \) and expected testing cost smaller than \( OPT_E(I) \left( \frac{1}{1 - \rho} \right) - \epsilon. \)

**Proof** Fix integers \( W > 1/\rho \) and \( C = \lfloor \rho W \rfloor. \) Then, let a value \( \eta \) and an instance \( I \) be defined as by the previous theorem. From this result, it follows that every decision tree \( D, \) with \( cost_W(D) \leq (1 + \rho)W \leq W + C, \) satisfies \( cost_E(D) \geq W(1 - \eta) - \eta C \geq W(1 - \eta(1 + \rho + 1/W)). \) Thus,

\[
\begin{align*}
\frac{cost_E(D)}{OPT_E(I)} &\geq \frac{1 - \eta(1 + \rho + 1/W)}{\left( 1 - \left( \frac{W-1}{W} \right)^{C} + \frac{\lfloor \log W \rfloor}{W} \left( \frac{W-1}{W} \right)^{C} \right)(1 - \eta) + \frac{(W + C + \lfloor \log W \rfloor) \eta}{W}} \\
&\geq \frac{1 - \eta(1 + \rho + 1/W)}{\left( 1 - \left( \frac{W-1}{W} \right)^{(\rho W + 1)} + \frac{\lfloor \log W \rfloor}{W} \left( \frac{W-1}{W} \right)^{(\rho W + 1)} \right)(1 - \eta) + \frac{(W + \rho W + 1 + \lfloor \log W \rfloor) \eta}{W}}.
\end{align*}
\]
By definition \( \eta \to 0 \) for \( W \to \infty \). Accordingly, it is not hard to see that the right hand side expression goes to \( \frac{1}{1-e^{-\rho}} \) as \( W \to \infty \). Therefore, for any \( \epsilon > 0 \) there exists \( W_\epsilon \) such that for every \( W \geq W_\epsilon \) the right hand side of (12) is larger than \( \frac{1}{1-e^{-\rho}} - \epsilon \), hence the instance \( I \) has the desired property.

3.1 The Structure of the Instance \( I \) in Theorem 2

Given the integers \( W > 1 \) and \( C > 0 \) and the number \( \eta < \frac{1}{W^{2(C+1)}} \), we define the following instance \( I = (S, T, C, p, e) \).

The set of objects \( S \) For the sake of simplifying notation, let \( L_W = \lfloor \log W \rfloor \). The set of objects is divided into the objects of type \( i \) (for each \( i = 1, \ldots, C + L_W \)) and light objects. The latter will have total probability mass \( \eta \) which will be asymptotically 0, i.e., negligible with respect to the probability of the other (non-light) objects. For each \( i = 1, \ldots, C + L_W \) there are \( 2^i \) objects of type \( i \), which we denote by \( S(i) = \{o_1^{(i)}, \ldots, o_{2^i}^{(i)}\} \).

For each \( i = 1, \ldots, C \) and \( j = 1, \ldots, 2^i \), the probability of \( o_j^{(i)} \) is

\[
\frac{(W - 1)^{i-1}}{2^i W^i} (1 - \eta).
\]

Hence, the total probability of objects of type \( i \) is

\[
p(S^{(i)}) = \frac{(W - 1)^{i-1}}{W^i} (1 - \eta).
\]

Note that this is exactly the probability distribution of the optimal solution of the NLP presented in the previous section adjusted by \((1 - \eta)\).

For each \( i = C + 1, \ldots, C + L_W \) and \( j = 1, \ldots, 2^i \), the probability of \( o_j^{(i)} \) is

\[
(1 - \eta) \left( \frac{W - 1}{W} \right)^{C} \frac{1}{2^{2^i - C + i - 2}}.
\]

Hence, for the total cumulative probability of objects of type larger than \( C \) we have

\[
p \left( S^{(C+1)} \cup \ldots \cup S^{(C+L_W)} \right) = \left( \frac{W - 1}{W} \right)^{C} (1 - \eta).
\]

Finally, there exists one light object for each non-light object. Each light object has the same probability and we denote by \( S^L \) the set of the light objects, and set \( p(S^L) = \eta \).

The partition into classes \( C \) Each object belongs to a different class.

A tree representation of the non-light objects For later purposes it is convenient to visualize the set of non-light objects as a complete binary tree \( T \) of depth \( C + L_W \) as shown in Fig. 2. By the \( i \)th level of \( T \) we understand the set of nodes at distance \( i \) from the root.

For \( i = 1, \ldots, C + L_W \) the objects of type \( i \) are identified with the nodes at level \( i \) of \( T \). Therefore, for \( i = 1, \ldots, C + L_W \) and \( j = 1, \ldots, 2^i \), the \( j \)th node (counting
from left to right) in level $i$ is identified with object $o_j^{(i)}$ of $S^{(i)}$. We use $O_j^{(i)}$ to denote the set of objects of the subtree of $T$ rooted at $o_j^{(i)}$.

**The set of tests $T$** The set $T$ of available tests is easily explained with reference to the tree’s representation of the objects presented above. The values taken by a test can be interpreted as a partition of the set of objects, each value corresponding to the subset of objects for which the test has that value. Therefore, we describe a test by the way it partitions or splits the set of objects.

There is one test of type 1, which we denote with $t_1^{(1)}$. It splits the objects as follows:

- **Group 1.** The single object $\{o_1^{(1)}\}$;
- **Group 2.** The single object $\{o_2^{(1)}\}$;
- **Group 3.** The set $O_1^{(1)} - \{o_1^{(1)}\}$ and its corresponding light objects;
- **Group 4.** The set $O_2^{(1)} - \{o_2^{(1)}\}$ and its corresponding light objects.
- **Group 5.** The two light objects associated with the objects $\{o_1^{(1)}, o_2^{(1)}\}$;

For each $i = 2, \ldots, C + L_W$ and $j = 1, \ldots, 2^i - 2$ the set $T$ includes a test $t_j^{(i)}$ which splits the set of objects into 5 groups as follows:

- **Group 1.** The single object $\{o_{2j-1}^{(i)}\}$;
- **Group 2.** The single object $\{o_{2j}^{(i)}\}$;
- **Group 3.** $O_1^{(1)} - \{o_1^{(1)}, o_{2j-1}^{(i)}\} - O_2^{(i)}$ and its corresponding light objects;
The split for the test $t_j^{(i)}$ associated to object $o_j^{(i)}$

The tree representation of objects

Fig. 3 The split of a test $t_j^{(i)}$ corresponding to an object in the left subtree of the tree of objects. Groups are represented by different patterns. We denote by $\omega_j^{(i)}$ the light object associated with the object $o_j^{(i)}$.

- Group 4. $O_2^{(1)} \cup \{o_1^{(1)}\} \cup \left(O_2^{(j)} \setminus \{o_2^{(j)}\}\right)$ and its corresponding light objects;
- Group 5. The two light objects associated with the objects $\{o_2^{(j-1)}, o_2^{(j)}\}$;

For each $i = 2, \ldots, C + L_W$ and $j = 2^{i-2} + 1, \ldots, 2^{i-1}$ the set $T$ includes a test $t_j^{(i)}$ which splits the set of objects into 5 groups as follows:

- Group 1. The single object $\{o_2^{(j-1)}\}$;
- Group 2. The single object $\{o_2^{(j)}\}$;
- Group 3. $O_1^{(1)} \cup \{o_2^{(1)}\} \cup \left(O_2^{(j-1)} \setminus \{o_2^{(j-1)}\}\right)$ and its corresponding light objects;
- Group 4. $O_2^{(1)} - \{o_2^{(1)}, o_2^{(j)}\} - O_2^{(j-1)}$ and its corresponding light objects;
- Group 5. The two light objects associated with the objects $\{o_2^{(j-1)}, o_2^{(j)}\}$;

For each $i = 1, \ldots, C + L_W$, we will refer to tests $\{t_j^{(i)} \mid j = 1, \ldots, 2^{i-1}\}$ as the tests of type $i$. Figures 3 and 4 illustrate the split corresponding to the test $t_j^{(i)}$ for some $i > 1$. By the test associated with a non-light object $o$ we mean the non-costly test that separates the two children of $o$, that is, for $o = o_j^{(i)}$ the test associated to $o$ is $t_j^{(i+1)}$. This terminology will be extensively used in our proofs.

Finally, $T$ includes a test denoted by $t^*$ which separates each single object.

**The cost of the tests** The tests of type $i = 1, \ldots, C + L_W$ have cost 1 while the test $t^*$ has cost $W$. We will refer to test $t^*$ as the costly test.

Some simple properties of the tests that can be verified by inspection will be used in our analysis.
The following properties hold for the instance above

(a) Let $o_{2j-1}^{(i)}$ and $o_{2j}^{(i)}$ be two non-light objects that are siblings in $T$. Then, the only tests that separate them are the costly test and the test $t_j^{(i)}$.

(b) If two light objects are associated with objects that are siblings in $T$ then the only test that separates them is the costly test.

3.2 Proof of Theorem 2

Proof of 1 The first item of Theorem 2 follows because a tree with the costly test $t^*$ at the root has worst testing cost $W$. □

Proof of 2 To prove the second item we construct a decision tree $D^C(I)$ for instance $I$, which we call the Canonical Decision Tree, and we evaluate its expected testing cost. □

If we ignore the leaves—which can be added in the natural way—the structure of the nodes associated with tests in the canonical decision tree $D^C(I)$ can be obtained as follows: start with the tree of objects $T$ and remove every object at level $C + L_W$ (the last one); replace the root of $T$ with the test $t_1^{(1)}$ and each node $o_j^{(i)}$ with the test $t_j^{(i+1)}$, for $i = 1, \ldots, C + L_W - 1$. Identify the edges going from node $o_j^{(i)}$ to its left and right children with the outcomes of $t_j^{(i+1)}$ represented by group 3 and 4 in its definition. Finally, for each $i = 1, \ldots, C + L_W$ and $j = 1, \ldots, 2^{i-1}$ add a new child to the node associated to test $t_j^{(i)}$ and associate it to the costly test $t^*$. The branch leading to this new child is associated to the outcome of the test $t_j^{(i)}$ represented by the light
Fig. 5 The structure of the canonical decision tree $D^C(I)$. Here $o_k^{(i)}$ denotes the light-object associated to the (non-light) object $o_k^{(i)}$ objects in group 5 (according to the definition of tests given above). Figure 5 shows the resulting tree.

It is not to hard to verify that

$$
cost_E(D^C(I)) \leq (1 - \eta) \sum_{j=1}^{C} j \frac{(W - 1)^{j-1}}{W^j} + (1 - \eta)(C + L_W) \left( \frac{W - 1}{W} \right)^C \\
+ (W + C + L_W)\eta
$$

Inequality (13) follows by observing that in the canonical decision tree every non-light object of type larger than $C$ has cost at most $C + L_W$ and their total probability if
Accordingly, every light object has cost at most $W + C + LW$ where the $W$ accounts for the cost of the costly test needed to separate it from the other objects, and $\eta$ is the total probability mass of the light objects. In order to obtain (14) we use

$$\sum_{j=1}^{C} j \frac{(W - 1)^{j-1}}{W^j} = W - (C + W) \left( \frac{W - 1}{W} \right)^C.$$ 

Thus, we have proved point (2) of Theorem 2. \hfill \Box

Proof of 3 In order to establish the last statement of the theorem we will need some additional notation and some intermediate results. For a decision tree $D$, we use $Obj(v)$ to denote the set of non-light objects associated with the leaves in the subtree of $D$ rooted at $v$.

We will use the following propositions:

**Proposition 2** Let $D$ be a decision tree for instance $I$. Let $v$ be an internal node of $D$ such that $Obj(v)$ is non empty. Then there are two sibling nodes/objects of the tree of objects $T$, name them $x_1$ and $x_2$, such that $x_1, x_2 \in Obj(v)$ and each object in $Obj(v)$ is a descendant of either $x_1$ or $x_2$ in $T$.

**Proof** We say that an object $o$ in $Obj(v)$ is maximal if no object in $Obj(v)$ is a proper ancestor of $o$ in $T$. In order to establish the proposition it is enough to show that the set of maximal objects in $Obj(v)$ has exactly two objects and those are siblings in $T$.

Let $M(v)$ be the set of maximal objects of $Obj(v)$. By Fact 1 (a), clearly, if an object belongs to $Obj(v)$ then so does its sibling in $T$. Therefore, if an object belongs to $M(v)$ then so does its sibling in $T$. For the sake of contradiction, let us assume that we $|M(v)| \geq 3$. In this case $o_1^{(1)}$ and $o_2^{(1)}$ do not belong to $M(v)$, for otherwise we would have $M(v) = \{o_1^{(1)}, o_2^{(1)}\}$. Let $x_1, x_2 \in M(v)$ be two siblings in $T$ and let $y$ be another object in $M(v)$. We assume that $x_1, x_2 \in O_1^{(1)}$ (the argument for the other case is analogous so we can omit it). We have two cases:

(i) $y \in O_2^{(1)}$. Since $o_1^{(1)}$ does not belong to $Obj(v)$ then there is a test in $D$, which is a proper ancestor of $v$ and separates $o_1^{(1)}$ from $x_1$. This test has to satisfy at least one of the following three conditions: (a) it is the test $t_1^{(1)}$; (b) it is associated with the parent of $x_1$; (c) it is associated with an object $o$ in $O_1^{(1)}$ such that $x_1$ is not in the right subtree below $o$ in $T$. However, in all these cases, such a test would also separate $x_1$ from $y$ which is a contradiction because they are both assumed to be in $Obj(v)$.

(ii) $y \in O_1^{(1)}$. Now, let $z$ the least common ancestor of $x_1$ and $y$ in $T$. Let $z'$ be the child of $z$ that it is an ancestor of $x_1$. Note that $z' \neq x_1$, for otherwise $y$ would be a descendant of either $x_1$ or its sibling $x_2$ and, as a consequence, it would not be maximal.

Because $z'$ is not in $Obj(v)$ there is a test, say $v'$, that is a proper ancestor of $v$ in $D$ and it separates $z'$ from $x_1$. Therefore, one of the following possibilities holds:
\( \nu' \) is associated with \( z \);
\( \nu' \) is associated with the parent of \( x_1 \);
\( \nu' \) associated with an object \( o \) that simultaneously satisfy: (a) \( o \) is a proper ancestor of \( x_1 \) in \( T \); (b) \( o \) is a descendant of \( z' \) in \( T \) and (c) \( x_1 \) lies in the right subtree of \( o \) in \( T \).

If \( \nu' \) is either associated with \( z \) or the parent of \( x_1 \) then it separates \( x_1 \) from \( y \). If \( \nu' \) is associated with an object that simultaneously satisfies (a), (b) and (c) then it also separates \( x_1 \) from \( y \). In all cases, we have a contradiction because \( x_1 \) and \( y \) are together in \( \text{Obj}(\nu) \). \( \Box \)

**Proposition 3** The following inequality holds\(^1\): \( \Pr[O^{(i)}_k - o^{(i)}_k] \leq (W - 1)\Pr[o^{(i)}_k] \), for any \( 1 \leq i \leq C + L_W \) and \( 1 \leq k \leq 2^i \).

**Proof** We split the proof into two cases:

**Case 1** \( i \leq C \). In this case, we have that

\[
\Pr[O^{(i)}_k - o^{(i)}_k] = \left( 1 - \sum_{s=1}^{i-1} \sum_{j=1}^{2^s} \Pr[o^{(j)}_s] - \frac{(W-1)^{i-1}}{W^{2^i}} \right) (1 - \eta)
\]

\[
= \left( \frac{(W-1)^{i-1}}{W^{2^i}} - \frac{(W-1)^{i-1}}{W^{2^i}} \right) (1 - \eta)
\]

\[
= (W - 1)\Pr[o^{(i)}_k]
\]

**Case 2** \( C < i \leq C + L_W \). In this case, all the objects in \( O^{(i)}_k \) have the same probability \( \left( \frac{W-1}{W} \right)^C \frac{1}{2^C(2^{L_W+1} - 2)} (1 - \eta) \). Moreover we have \( |O^{(i)}_k| \leq 2L_W - 1 \leq W - 1 \). Thus, it follows that

\[
\Pr[O^{(i)}_k - o^{(i)}_k] < (W - 1) \left( \frac{W - 1}{W} \right)^C \frac{1}{2^C(2^{L_W+1} - 2)} (1 - \eta)
\]

\[
= (W - 1)\Pr[o^{(i)}_k].
\]

\( \Box \)

We say that a test \( t \) (an object \( o \)) occurs at cost level \( \kappa \) in a decision tree \( D \) if the total cost of tests on the path from the root of \( D \) to the parent of \( t \) (or \( o \)) is \( \kappa \). As an example, in Fig. 1, test \( t_2 \) occurs at cost level 1 and object \( s_2 \) occurs at cost level 3 = 1 + 2.

**Proposition 4** Let \( D \) be a decision tree for instance \( I \) and let \( D' \) be the tree obtained from \( D \) by removing all subtrees rooted at costly tests. Then, \( D' \) has at most \( 2^\ell \) objects occurring at cost level \( \ell \) for every \( \ell \).

**Proof** First, note that all leaves in \( D' \) are associated with non-light objects. Indeed, as a consequence of Fact 1 (b), the deletion of subtrees rooted at costly nodes also removes all leaves associated with light objects.

---

\(^1\) For the sake of readability here we use the notation \( \Pr[\] \) for the probability of objects.
Let \( v \) be an arbitrarily chosen internal node in \( D' \). Note that it is enough to prove that \( v \) has at most two children that are internal nodes and at most two children that are leaves because, in this case, a simple inductive argument can be used to establish that \( D' \) has at most \( 2^\ell \) objects occurring at cost level \( \ell \) for every \( \ell \).

First, we prove that \( v \) has at most two children that are leaves. Let \( v \) be a node in \( D' \) and let \( t_j^{(i)} \) be the test corresponding to \( v \). If \( v \) has more than two leaves as children, then there is an object, say \( o \), with \( o \notin \{ o_{2j-1}^{(i)}, o_{2j}^{(i)} \} \), corresponding to one of these leaves. Let \( t \) be the test corresponding to the parent of \( o \) in \( T \) (\( t_1^{(1)} \) if \( o \) has type 1).

Since, by Fact 1 (a) \( t \) is the only non-costly test separating \( o \) from its sibling and \( t \) is not the test corresponding to \( v \), we have that \( t \) must be an ancestor of \( v \). But then \( o \) cannot be in \( \text{Obj}(v) \) since it must be a leaf child of the node corresponding to \( t \). Hence we have a contradiction.

Finally, we prove that \( v \) has at most two children that are internal nodes. Note that the node \( v \) in \( D \) has at most three children that are internal nodes, corresponding to the groups 3–5 in the definition of the test’s splits. However, the internal node associated with group 5 must be a costly test, for otherwise there would be a node in \( D \) that does not provide information which is not possible according to our definition of the decision tree problem. Since all costly tests are removed it follows that at most two children are internal nodes.

**Lemma 1** Let \( D \) be a decision tree for the instance \( I \) such that \( \text{cost}_W(D) \leq W + C \). Then \( \text{cost}_E(D) \geq W(1 - \eta) - \eta C. \)

**Proof** Let \( D \) be a decision tree with minimum expected testing cost among all decision trees for \( I \) with worst testing cost not larger than \( W + C \).

First, we argue that every non-costly test in \( D \), that has at least one non-light object as a descendant, occurs at cost level at most \( C - 1 \). For the sake of contradiction, let us assume that some non-costly test that has at least one non-light object as a descendant occurs at cost level larger than or equal to \( C \). Let \( v \) be the node of \( D \) corresponding to such a test and let \( o \) be an object in \( \text{Obj}(v) \). Assume that \( o \in O_1^{(1)} \) (the proof for the other case is analogous so that we omit it). Both the light object associated with \( o \) and the light object associated with \( o \)’s sibling are also in subtree of \( D \) rooted at \( v \) because the only tests that separate them from \( o \) are the costly test and the test corresponding to the parent of \( o \) (\( t_1^{(1)} \) if \( o = o_1^{(1)} \)). However, none of these tests can be a proper ancestor of \( v \) in \( D \), for otherwise we would have \( o \notin \text{Obj}(v) \). Thus, since \( v \) does not correspond to a costly test, there must be a costly test in the subtree of \( D \) rooted at \( v \) to separate these light objects. This implies that \( \text{cost}_W(D) > C + W \), which is a contradiction.

Now, we argue that there exists a tree \( \tilde{D} \) with worst testing cost at most \( C + W \) and expected testing cost not much larger than that of \( D \) such that all costly tests in \( \tilde{D} \), which are ancestors of at least one non-light object, occur at cost level \( C \). For that, let \( v \) be an internal node of \( D \) associated with a costly test that occurs at cost level smaller than \( C \) and such that \( \text{Obj}(v) \) is non-empty. By Proposition 2 the set of non-light objects in \( \text{Obj}(v) \) can be partitioned into three groups \( \{ o_{2j-1}^{(s)}, o_{2j}^{(s)} \} \) and \( \text{Obj}(v) \setminus \{ o_{2j-1}^{(s)}, o_{2j}^{(s)} \} \subseteq \left( O_{2j-1}^{(s)} \cup O_{2j}^{(s)} \right) \setminus \{ o_{2j-1}^{(s)}, o_{2j}^{(s)} \} \), for some \( 1 \leq s \leq C + LW \).
and some $1 \leq j \leq 2^\ell$. From Proposition 3, we have

$$\Pr[\text{Obj}(v) \setminus \{o^{(s)}_{2j-1}, o^{(s)}_{2j}\}] \leq \Pr[\{O^{(s)}_{2j-1}\} \setminus \{o^{(s)}_{2j-1}\}] + \Pr[\{O^{(s)}_{2j}\} \setminus \{o^{(s)}_{2j}\}]$$

(17)

$$\leq \Pr[\{o^{(s)}_{2j-1}, o^{(s)}_{2j}\}](W - 1).$$

(18)

Let $l(v)$ be the set of light objects that are separated by the costly test associated with $v$; and let $D'$ be the decision tree obtained by replacing this test with the test $t^{(s)}_j$ and then using costly tests as children of $t^{(s)}_j$ to separate objects of $\text{Obj}(v)$ that are grouped together by $t^{(s)}_j$. This modification reduces the cost of the leaves associated with $o^{(s)}_{2j-1}$ and $o^{(s)}_{2j}$ by $W - 1$ and increases by 1 the cost of the leaves associated to the objects in $\text{Obj}(v) \setminus \{o^{(s)}_{2j-1}, o^{(s)}_{2j}\}$ and the cost of the objects in $l(v)$. In formulas, using (17)–(18), we have

$$\text{cost}_E(D') = \text{cost}_E(D) - (W - 1)\Pr[\{o^{(s)}_{2j-1}, o^{(s)}_{2j}\}]
+ \Pr[\text{Obj}(v) \setminus \{o^{(s)}_{2j-1}, o^{(s)}_{2j}\}] + \Pr[l(v)]$$

(19)

$$\leq \text{cost}_E(D) + \Pr[l(v)].$$

(20)

By repeated application of the above transformation we can obtain a decision tree $\tilde{D}$ such that: $\text{cost}_W(\tilde{D}) \leq C + W$; for each node $v$ of $\tilde{D}$ associated with a costly test, either $\text{Obj}(v)$ is empty or $v$ occurs at cost level $C$; $\text{cost}_E(\tilde{D}) \leq \text{cost}_E(D) + \eta C$. The term $\eta C$ in the last inequality comes from (19) to (20) telling that each repetition of the transformation might increases the cost of some light object by 1. Each light object can be involved in at most $C$ such transformations, since after such a number of transformation the light object would be a leaf child of a costly test at level $\geq C$. Since in total light objects have probability mass $\eta$ the cumulative increase given by the transformations is at most $\eta C$.

Now, we lower bound the expected testing cost of $\tilde{D}$ by considering only the contribution provided by the non-light objects. Our first observation is that there are at most $2^\ell$ non-light objects occurring at level $\ell$ for $\ell = 1, \ldots, C$. To see that, let $\overline{D}$ be tree obtained from $\tilde{D}$ by removing all subtrees rooted at costly tests. Because all costly tests that have at least one non-light object as a descendant occur at cost level $C$, it follows that all non-light objects that occur at cost level smaller than or equal to $C$ in $\overline{D}$ are not affected by the deletion. Moreover, it follows from Proposition 4 that there are at most $2^\ell$ leaves at level $\ell$ associated with a non-light object in $\overline{D}$, and as a consequence, also in $\tilde{D}$.

For each $\ell = 1, 2, \ldots, C$ let $\tilde{p}_\ell$ be the sum of the probabilities of the non-light objects that occur at cost level $\ell$ in $\tilde{D}$. For each $k = 1, \ldots, C$ we have that

$$\sum_{\ell=1}^k \tilde{p}_\ell \leq \left( \sum_{\ell=1}^k \frac{(W - 1)^{\ell-1}}{W^\ell} \right) (1 - \eta).$$

(21)
In fact, for each $k$ there are at most $2^{k+1} - 2$ leaves associated with non-light objects in the first $k$ levels. In addition the set of $2^{k+1} - 2$ objects of largest probability in $I$ is given by the set of objects of type $1, \ldots, k$, whose cumulative probability coincides with the right-hand-side expression.

Then, ignoring the contribution of the light objects, we can write

\[
\text{cost}_E(\tilde{D}) \geq \sum_{\ell=1}^C \ell \cdot \tilde{p}_\ell + (C + W) \left( (1 - \eta) - \sum_{\ell=1}^C \tilde{p}_\ell \right) (22)
\]

\[
= \sum_{j=1}^C \left( (1 - \eta) - \sum_{\ell=1}^{j-1} \tilde{p}_\ell \right) + W \left( (1 - \eta) - \sum_{\ell=1}^C \tilde{p}_\ell \right) (23)
\]

\[
\geq \left( \sum_{j=1}^C \left( 1 - \sum_{\ell=1}^{j-1} \frac{(W - 1)^{\ell-1}}{W^\ell} \right) + W \left( (1 - \sum_{\ell=1}^C \frac{(W - 1)^{\ell-1}}{W^\ell} \right) \right) (1 - \eta) (24)
\]

\[
= \left( \sum_{j=1}^C \frac{(W - 1)^{j-1}}{W^j} + W \left( \frac{(W - 1)^C}{W^C} \right) \right) (1 - \eta) = W(1 - \eta) (25)
\]

where (23) is a rewriting of $\text{cost}_E(\tilde{D})$ in terms of the contribution of the internal nodes/tests by cost level; and (24) follows from (23) because of (21). By the construction of $\tilde{D}$ we finally have the desired result $\text{cost}_E(D) \geq \text{cost}_E(\tilde{D}) - \eta C \geq W(1 - \eta) - \eta C$

\[\square\]

4 Further Observations and Open Problems

We proved that given an instance $I$ of the decision tree optimization problem and a parameter $\rho > 0$ there exists a decision tree $D$ with worst testing cost at most $(1 + \rho)\text{OPT}_W(I)$ and expected testing cost at most $(1 + 1/(1 - e^{-\rho}))\text{OPT}_E(I)$. In addition, we proved that this bound is essentially tight.

In the construction of the lower bound we used both non-uniform test costs and non-uniform probabilities. We can extend the lower bound to the case of uniform probabilities by the following transformation: Take an instance $I = (S, C, T, p, c)$ as described in the proof of Theorem 2 and produce the instance $\tilde{I} = (\tilde{S}, \tilde{C}, \tilde{T}, \tilde{p}, \tilde{c})$ as follows: for each object $o \in S$ create a class $\tilde{C}_o$ in $\tilde{C}$ and create $p(o) / \prod_{o \in S} p(o)$ new objects in $\tilde{S}$ setting them as all belonging to the class $\tilde{C}_o$. Set the probability of each created objects $\tilde{o} \in \tilde{S}$ to $\tilde{p}(\tilde{o}) = \prod_{o \in S} p(o)$. Hence, for each $o \in S$, the total probability of the objects in class $\tilde{C}_o$ is equal to $p(o)$.

Finally for each test $t \in T$ create a corresponding test $\tilde{t}$ in $\tilde{T}$. If object $o \in S$ is in group $g$ of test $t \in T$ then each object $o' \in \tilde{C}_o$ is in group $g$ of test $\tilde{t}$. It is not hard to realise that for the instance $\tilde{I}$ the class $\tilde{C}_o$ behaves exactly as the single objects $o$ in the instance $I$. This construction shows that the limit on the best trade-off achievable also holds for the case of uniform probabilities.
An interesting question which remains open regards the case of uniform testing costs. We ask whether for every $\epsilon > 0$, there is some integer $n_0$ such that every instance $I$ with uniform testing costs and with more than $n_0$ objects, admits a decision tree $D$ such that $\text{cost}_E(D) \leq (1 + \epsilon)\text{OPT}_E(I)$ and $\text{cost}_W(D) \leq (1 + \epsilon)\text{OPT}_W(I)$. We notice that this result holds for length restricted prefix codes [22], the special case of decision tree construction mentioned in the related work’s section.

Appendix: Optimality of $p^*$

Here, we show that the point $p^*$ defined in (9)–(10) is an optimal solution of the NLP (5)–(8) defined in Sect. 2. For that, we use Theorem 4.3.8 of [4] that we restate here in a simplified form.

Theorem 4 Let $X$ be an open subset of $R^n$. Consider the optimization problem $P$:

\[
\begin{align*}
\text{Minimize} & \quad f(x) \\
\text{subject to} & \quad g_i(x) \leq 0 \quad \text{for } i = 1, \ldots, m \\
& \quad h_i(x) = 0 \quad \text{for } i = 1, \ldots, l \\
& \quad x \in X
\end{align*}
\]

Let $x$ be a feasible solution and let $I = \{i | g_i(x) = 0\}$. Suppose there exists scalars $u_i \geq 0$ for $i = 1, \ldots, m$ and $v_i$ for $i = 1, \ldots, l$ such that

\[
-\nabla f(x) = \sum_{i \in I} u_i \nabla g_i(x) + \sum_{i=1}^l v_i \nabla h_i(x)
\]

If $f$ is linear, $g_i$ is convex on $X$ for $i \in I$ and $h_i$ is linear, then $x$ is an optimal solution of $P$.

The non linear problem defined in (5)–(8) can be rewritten in terms of problem $P$. In fact, for a point $p = (p_1, \ldots, p_{C+1}, z)$ we have that $f(p) = -z$,

\[
h_1(p) = 1 - \sum_{i=1}^{C+1} p_i,
\]

\[
g_j(p) = z \left( \sum_{i=1}^{C+1} i \cdot p_i \right) - \sum_{i=1}^j i \cdot p_i - (j + W) \left( \sum_{i=j+1}^{C+1} p_i \right),
\]

for $j = 0, \ldots, C$ and

\[
g_j(p) = -p_{j-C}
\]

\footnote{Theorem 4.3.8 of [4] requires weaker conditions on $f$, $h$ and $g$ but for our purposes it is enough to consider these stronger and more well known conditions on $x$.}
for \( j = C + 1, \ldots, 2C + 1 \).

We define an open set \( X \) that contains all feasible solutions of the problem defined in (5)–(8). The motivation is to meet the conditions of Theorem 3.3.7 of [4] that will be used to establish the convexity of \( g_i \).

Let

\[
X = \left\{ (p_1, p_2, \ldots, p_{C+1}, z) \left| \sum_{i=1}^{C+1} p_i > 1 - \frac{1}{2(C+1)} \text{ and } \right. \right. \\
\left. \left. p_i > -\frac{1}{2(C+1)^2} \right. \right. \text{ and } \left. \right. z > 0 \right\}.
\]

We have to prove that: (a) \( p^* \) is feasible and \( I = \{0, \ldots, C\} \); (b) \( f, h_1 \) and \( g_i \) satisfy the conditions of Theorem 4 and (c) there are multipliers satisfying condition (30).

Feasibility of \( p^* \)

We have that \( p^* \) is feasible because

\[
g_j(p^*) = z^* \left( \sum_{i=1}^{C+1} i \cdot p^*_i \right) - \sum_{i=1}^{j} i \cdot p^*_i - (j + W) \left( \sum_{i=j+1}^{C+1} p^*_i \right) \\
= z^* \left( W - (C + W) \frac{(W - 1)^C}{WC} + (C + 1) \frac{(W - 1)^C}{WC} \right) \\
- \left( W - (j + W) \frac{(W - 1)^j}{wj} \right) - (j + W) \frac{(W - 1)^j}{wj} \\
= z^* \left( W - \frac{(W - 1)^{C+1}}{WC+1} \right) - W = 0
\]

where the last expression holds because

\[
z^* = \frac{W}{W - \frac{(W - 1)^{C+1}}{WC+1}}.
\]

We have that \( I = \{0, \ldots, C\} \) because

\[
g_j = -p_{j-C} = -\frac{(W - 1)^{j-C-1}}{(W)^{j-C}} < 0,
\]

for \( j > C \).
Convexity of $g_j$

Because $f$ and $h_1$ are linear and $I = \{0, \ldots, C\}$, we only need to prove that $g_i$, for $i = 0, \ldots, C$, is convex in the open set $X$.

Since $g_j$, for $j = 0, \ldots, C$, is twice differentiable it follows from Theorem 3.3.7 of [4] that it is enough to show that the Hessian of $g_j$, for $j = 0, \ldots, C$, is semi-positive defined in $X$. In fact, the Hessian of $g_j$ is a matrix where all elements, except the first $C + 1$ in the last line and the first $C + 1$ in the last column, are zeros. The matrix has the structure presented below.

$$
H = \begin{bmatrix}
0 & 0 & \cdots & 0 & 1 \\
0 & 0 & \cdots & 0 & 2 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & 0 & C + 1 \\
1 & 2 & \cdots & C + 1 & 0
\end{bmatrix}
$$

We have that

$$
pHp = 2z(p_1 + 2p_2 + \cdots + (C + 1)p_{C+1}) \geq 0
$$

for all $p = (p_1, \ldots, p_{C+1}, z) \in X$ because $z > 0$ and

$$
\sum_i i \cdot p_i \geq \sum_i p_i - (C + 1) \frac{1}{2(C + 1)^2} > 1 - \frac{1}{2(C + 1)} - \frac{1}{2(C + 1)} = 0
$$

KKT Conditions

Let $\lambda_0, \ldots, \lambda_C$ be the dual variables associated with the constraints $g_j$, $j = 0, \ldots, C$. Let $\lambda_E$ be the dual variable associated with the constraint $\sum_{i=1}^{C+1} p_i = 1$.

The multipliers $(\lambda_0, \ldots, \lambda_C, \lambda_E)$ must satisfy

$$
-\nabla f(p^*) = \sum_{i=0}^{C} \lambda_i \nabla g_i(p^*) + \lambda_E \nabla h_1(p^*),
\lambda_i \geq 0 \quad \text{for } i = 0, \ldots, C
$$

Thus, we must have

$$
\sum_{j=0}^{i-1} \lambda_j (j + W) + \sum_{j=i}^{C} \lambda_j i + \lambda_E = z \sum_{j=0}^{C} \lambda_j i \quad (31)
$$
for \( i = 1, \ldots, C + 1 \), and also
\[
\sum_{j=0}^{C} \lambda_j \left( \sum_{i=1}^{C+1} i \cdot p_i^* \right) = 1 \tag{32}
\]

Let
\[
E = \sum_{i=1}^{C+1} i \cdot p_i^*
\]

By subtracting the Eq. (31) when \( i = C \) from the same equation, when \( i = C + 1 \), using the Eq. (32) and using the fact that \( E \cdot z^* = W \), we get that \( \lambda_C = \frac{1}{E^2} \). In addition, we can prove by induction that \( \lambda_{j-1} = \lambda_j \frac{W-1}{W} \). For that we subtract the Eq. (31) when \( i = j - 1 \) from the same equation, when \( i = j \), and use the induction hypothesis. Thus, we get that
\[
\lambda_i = \frac{(W-1)^{C-i}}{W^{C-i}E^2},
\]
for every \( i \) so that all multipliers are positives.

References

1. Adler, M., Heeringa, B.: Approximating optimal binary decision trees. Algorithmica 62(3), 1112–1121 (2012)
2. Aslam, J.A., Rasala, A., Stein, C., Young, N.E.: Improved bicriteria existence theorems for scheduling. In: SODA 1999, pp. 846–847 (1999)
3. Alkhalid, A., Chikalov, I., Moshkov, M.: A tool for study of optimal decision trees. LNCS 6401, 353–360 (2010)
4. Bazaraa, M.S., Sherali, H.D., Shetty, C.M.: Nonlinear Programming: Theory and Algorithms, 2nd edn. Wiley, New York (1993)
5. Bellala, G., Bhavnani, S.K., Scott, C.: Group-based active query selection for rapid diagnosis in time-critical situations. IEEE-IT 58(1), 459–478 (2012)
6. Buro, M.: On the maximum length of huffman codes. IPL 45(5), 219–223 (1993)
7. Chakaravarthy, V.T., Pandit, V., Roy, S., Awasthi, P., Mohania, M.: Decision trees for entity identification: approximation algorithms and hardness results. ACM Trans. Algorithms 7(2), 15:1–15:22 (2011)
8. Cicaelese, F., Jacobs, T., Laber, E., Molinaro, M.: On greedy algorithms for decision trees. In: Proceedings of ISAAC (2010)
9. Cicalese, F., Laber, E., Saettler, A.: Diagnosis determination: decision trees optimizing simultaneously worst and expected testing cost. In: ICML2014, pp. 414–422 (2014)
10. Garey, M.R.: Optimal binary identification procedures. SIAM J. Appl. Math. 23(2), 173–186 (1972)
11. Garey, M.R.: Optimal binary search trees with restricted maximal depth. SIAM J. Comput. 3(2), 101–110 (1974)
12. Golovin, D., Krause, A., Ray, D.: Near-optimal bayesian active learning with noisy observations. Adv. Neural Inf. Proc. Syst. 23, 766–774 (2010)
13. Guillory, A., Bilmes, J.: Average-case active learning with costs. In: ALT’09, pp. 141–155 (2009)
14. Guillory, A., Bilmes, J.: Interactive submodular set cover. In: Proceedings of ICML10, pp. 415–422 (2010)
15. Gupta, A., Nagarajan, V., Ravi, R.: Approximation algorithms for optimal decision trees and adaptive TSP problems. In: Proceedings of ICALP’10, pp. 690–701 (2010)
16. Hussain, S.: Relationships among various parameters for decision tree optimization. Stud. Comput. Intell. 514, 393–410 (2014)
17. Kelle, P., Schneider, H., Yi, H.: Decision alternatives between expected cost minimization and worst case scenario in emergency supply second revision. Int. J. Prod. Econ. 157, 250–260 (2014)
18. Kosaraju, S., Przytycka, T., Borgstrom, R.: On an optimal split tree problem. WADS 99, 157–168 (1999)
19. Krause, A.: Optimizing sensing: theory and applications. Ph.D. thesis, Carnegie Mellon University (2008)
20. Larmore, L.L.: Height restricted optimal binary trees. SICOMP 16(6), 1115–1123 (1987)
21. Larmore, L.L., Hirschberg, D.S.: A fast algorithm for optimal length-limited Huffman codes. J. ACM 37(3), 464–473 (1990)
22. Milidi, R.L., Laber, E.S.: Bounding the inefficiency of length-restricted prefix codes. Algorithmica 31(4), 513–529 (2001)
23. Moshkov, M.I.: Greedy algorithm with weights for decision tree construction. Fundamentae Informaticae 104(3), 285–292 (2010)
24. Papadimitriou, C.H., Yannakakis, M.: On the approximability of trade-offs and optimal access of web sources. FOCS 2000, 86–92 (2000)
25. Rasala, A., Stein, C., Torng, E., Uthaisombut, P.: Existence theorems, lower bounds and algorithms for scheduling to meet two objectives. In: SODA 2002, pp. 723–731 (2002)
26. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. MIT Press, Cambridge (2001)