Mixed integer linear optimization formulations for learning optimal binary classification trees

Brandon Altson[1], Hamidreza Validi[1], and Illya V. Hicks[1]

[1]Computational and Applied Mathematics, Rice University,
{bca3,hamidreza.validi,ivhicks}@rice.edu

May 9, 2022

Abstract

Decision trees are powerful tools for classification and regression that attract many researchers working in the burgeoning area of machine learning. A decision tree has two types of vertices: (i) branching vertices at which datapoints are assessed on a set of features; and (ii) leaf vertices at which datapoints are given a prediction. In classification trees, branching and leaf vertices use discrete features and discrete predictions, respectively. A binary classification tree is a specific type of classification tree in which each branching vertex has exactly two children. An optimal binary classification tree can be obtained by solving a biobjective optimization problem that seeks to (i) maximize the number of correctly classified datapoints and (ii) minimize the number of branching vertices. In this paper, we propose four mixed integer linear optimization (MILO) formulations for designing optimal binary classification trees: two flow-based formulations and two cut-based formulations. Furthermore, we provide theoretical and computational comparisons between our formulations and the most recent flow-based MILO formulation proposed by Aghaei et al. (2021). We also propose valid inequalities, fixing procedures, and a heuristic procedure to enhance our computational experiments. We see an improvement in solution time or optimality gap by an average factor of 2.80. Further, our models are able to outperform a tailored Bender's decomposition of the flow-based model. Our code and data are available on GitHub.

Discussions on the paper

1. We need to highlight the superiority of exact solvers and MILO models over CART and other heuristics.
2. Why optimality/quality of objectives are important?
3. Cite some recent papers from INFORMS journals

1 Introduction

Decision trees have been employed as tools in various applications including decision making in management science (Magee, 1964) and solving integer optimization problems in operations research (Land and Doig, 1960) since as early as the 1960s. With emergence of machine learning around 1980, Breiman et al. (1984) introduced the application of decision trees on classification and regression problems. Since 1980, multiple heuristic approaches have been proposed to find quality feasible solutions for the binary classification decision tree problem; e.g., CART (Breiman et al., 1984) and QUEST (Loh and Shih, 1997). Although Breiman et al. (1984) were aware their heuristic algorithm "[was] only one-step optimal and not overall optimal", they did not address the optimal decision tree problem due to the immature "computer technology" in the 1980s.

In this paper, we focus on binary classification decision trees: trees in which each parent has exactly two children. Due to their interpretability, binary decision trees are used in a wide range of applications, including but not limited to healthcare (Yoo et al., 2020; Li et al., 2021; Albaqami et al., 2021), geological surveying (Balk and Elder, 2000), cyber-security (Maturana et al., 2011; Kumar et al., 2013), financial analysis (Delen et al., 2013; Charlot and Marimoutou, 2014; Manogna and Mishra, 2021), and more recently fair decision making (Zhang and Ntoutsi, 2019; Barata and Veenman, 2021; Valdivia et al., 2021). Nearly all such applications use decision tree algorithms for feature selection.

Optimal binary classification decision trees select binary tests to perform at each branching vertex and classes to assign to each leaf vertex to maximize prediction accuracy or minimize misclassification rate. However, building an optimal decision tree is NP-hard shown by Hyafil and Rivest (1976) and confirmed empirically by Bessiere et al. (2009) who note greedy methods that build large trees perform inferior to methods that find the smallest decision tree. The NP-hard observation of Hyafil and Rivest (1976) led to the use of heuristic methods, as they aid in finding a solution when an exhaustive search is not practical. Heuristic approaches are often suboptimal and suffer from a number of shortcomings leading to solutions that are often tailored for better solutions after the original algorithm. Comparisons of heuristic-based tree models provide useful outlines on which methods to use given specific conditions for selecting attributes, but Murthy (2004) conclude no one heuristic for splitting is superior to the others.

This led researches to use Mixed Integer Linear Optimization (MILO) techniques. MILO formulations have well established solvers and algorithms for finding solutions and provide flexibility when choosing the objective function and constraints associated with building a decision tree. In turn, MILO formulations are more robust but also more challenging to formulate due to the trade-off between the number of decision variables and their linear optimization (LO) relaxations. Optimization solvers such as Gurobi (Gurobi Optimization, 2021) and CPLEX (Cplex, 2009), have become substantially more powerful (speedup factor of 450 billion over a 20 year period) through advancements in hardware and effective use of cutting plane theory, disjunctive programming for branching rules, and improved heuristic methods detailed by Bixby (2012). These advancements simultaneously eliminate the practical barrier for implementing MILO formulations and the prejudice relevant during the inception of MILO formulations. Using MILO formulations also allows for more robust methods with flexible branching constraints and objective functions.

In this paper, we propose four MILO formulations for finding optimal binary decision trees, show their stronger linear optimization (LO) relaxations compared to current MILO formulations in the literature, highlight the practical application of the proposed MILO formulations through experimental results on 13 publicly available datasets, and in general provide those interested in using MILO techniques for machine learning a set of formulations to use. In Section 2, . In Section 3, we propose four MILO formulations: two multi-commodity flow formulations (MCF1 and MCF2), and two cut-based formulations (CUT1 and CUT2) for finding optimal binary trees. The proposed formulations allow for imbalanced decision trees, as pruning is built-in. In Section 4, we show that linear relaxations of our proposed formulations are stronger than that of the recent flow-based formulation (FlowOCT) proposed by Aghaei et al. (2021). In Section 5, we provide computational enhancements to decrease the solution time for training decision trees using our MILO models through heuristic warm starts and decision variable fixing procedures as well as valid inequalities for producing more interpretable decision trees. In Section 6, we provide computational experiments.

2 Related Work on Binary Decision Trees

Multiple heuristic approaches have been proposed to find quality feasible solutions for the binary classification decision tree problem; e.g., CART (Breiman et al., 1984), QUEST (Loh and Shih, 1997), C4.5 (Quinlan, 1993), SPRINT (Shafer et al., 1996), SLIQ (Mehta et al., 1996), and CHAID (Kass, 1980). These heuristic methods use different metrics such as Shannon's entropy (Shannon, 1948) (generally referred to as entropy), the Gini index (Gini, 1912), information gain (Kullback and Leibler, 1951) (referred to as the Kullback–Leibler divergence in decision trees) and the Gain ratio (Quinlan, 1986), all of which use a probability distribution function $P(X)$ for each discrete random variable, $X = \{x_1, ..., x_n\}$ with probability of observing each $x_i \in X$ as $P(x_i)$. Many of the aforementioned methods take *top-down* approaches for building decision trees. Starting at the root vertex, splits are determined by solving what can be defined as an optimization problem. In the case of CART, ID3, and C4.5, a split is determined by the minimum Gini index. Using this approach is greedy and each split is determined independent of the others, leading to a tree that may not adequately capture intrinsic relationships of the trained dataset; the tree may also perform poorly on tested datasets due to over-fitting. A *top-down* approach also requires pruning to not eliminate powerful splits "hidden" behind weaker ones. These methods also have stopping criteria (complexity penalties) to grow the tree as deep as possible before pruning, resulting in a two stage learning approach.

In attempt to build more efficient trees, formulations with oblique splits are also considered. Murthy et al. (1994) extend the work of Breiman et al. (1984) by using hill-climbing techniques paired with randomization. Orsenigo and Vercellis (2003) use discrete SVM? operators counting misclassified points rather than measuring distance at each node of the tree; then, to find the complete tree sequential LP-based heuristics are employed. Menze et al. (2011) extend the oblique random forests first proposed by Breiman et al. (1984) with LDA? to find optimal internal splits. Wang et al. (2015) use logistic regression to find separating hyperplanes at branching vertices while maintaining sparsity through a weight vector. Wickramarachchi et al. (2016) employ Householder transformations to find oblique trees. Kontschieder et al. (2015) combine convolutional neural nets and stochastic and differentiable decision trees to find global parameters for split and leaf vertices. Balestriero (2017) uses a modified hashing neural net framework with sigmoid activation functions and independent multilayer percepetrons that are equivalent to vertices of a decision tree and allow for oblique functions on each splitting node which in turn imposes a global optimization problem that is differentiable with respect to tuning parameters. Yang et al. (2018) use a one-layer neural network with softmax as its activation function with a *soft* binning function for branching. Lee and Jaakkola (2020) note greedy oblique splitting methods generally find poor local optima and show decision trees are related to peace-wise linear neural nets with locally constant gradients.

Many of the methods mentioned above are counterintuitive when evaluating the true purpose of a decision tree, which is to minimize misclassified datapoints, not impurity measures on branching vertices. Breiman et al. (1984) note continued growth of the tree is indicative of successful splits and the growth is a one-step optimization problem; thus, using the unnatural objective function was acceptable. The more natural, and computationally expensive, method is to build the tree in one problem. The unaligned objective and the lack of computer technology made it infeasible to search all possible partitions for such a method. Bertsimas and Dunn (2017) attribute failure to produce optimal decision trees with heuristic methods because such methods do not address the underlying problem of building a decision tree. Steps of building a decision tree involve discrete decisions (which vertex to split on, which variable to split with) and discrete outcomes (is a datapoint correctly classified, which leaf does a datapoint end on). Therefore, one should consider building optimal decision trees using MILO formulations.

Although Breiman et al. (1984) were aware that their heuristic algorithm "is only one-step optimal and not overall optimal", they did not address this problem due to the "immature computer technology" of the 1980s. To the best of our knowledge, Bennett and Blue (1996) propose the first MILO formulation for designing optimal multivariate decision trees. They fix (i) the structure of the tree, (ii) the number of branching vertices and (iii) the classes of leaf vertices before solving the optimal decision tree problem. Further, they showed that optimal solutions of their problem lie on the extreme points of the feasible set, even with a nonconvex objective function.

Through considerable developments in integer optimization techniques and employing them in commercial solvers, Bertsimas and Shioda (2007) introduced CRIO (classification and regression via integer optimization) as a software package that employs CPLEX 8.0 for solving optimization models. Bertsimas and Dunn (2017) propose OCT (optimal classification trees) which outperforms CART in accuracy. Verwer and Zhang (2019) propose a Binary-Linear Programming (BLP) model aiming to reduce the dependence of the problem size on the size of the training dataset (BinOCT). Optimal randomized classification trees (ORCT) from Blanquero et al. (2021) uses a continuous optimization method for learning trees by replacing discrete binary decisions in traditional trees with probabilistic decisions. Dash et al. (2018) and Firat et al. (2020) propose column generation approaches. Günlük et al. (2021) formulate IP models for decision trees with categorical data. Narodytska et al. (2018) present a scalable SAT-based approach to build an optimal tree of depth D. They generate valid binary encodings for a specified number of vertices $N \leq D$, such that valid binary trees of size N can be generated, and then choose the best tree; they also provide upper bounds on N. Aglin et al. (2020) develop two branch-and-bound approaches that improve upon their previous DL8 method by caching itemsets used for cutting the search space and only including vertices not in the cache in the branch-and-bound cuts. Zhu et al. (2020) use a 1-norm SVM to train decision trees. Zantedeschi et al. (2020) propose using stochastic descent to generate branching attributes by defining some pruning preference parameter $\eta \in \mathbb{R}$ and auxiliary variables for linearity, then applying a unique tree-structured isotonic optimization algorithm to build a pruning-aware MILO formulation for decision trees.

Recently, Aghaei et al. (2021) propose a flow-based formulation whose LP relaxation is stronger than that of OCT. They modify the structure of a traditional decision tree by adding a single source vertex, s, adjacent only to the root, and a single sink vertex, t, adjacent to all vertices in the tree. The tree can now be thought of as an *directed acyclic network with a single source and sink*. For learning fair decision trees they incorporate two novel indices measuring disparate impact and disparate treatment as loss functions weighed in the objective function. They also consider a max-flow based MILO formulation where correctly classified datapoints successfully *flow* through the tree. The flow formulation of Aghaei et al. (2021) also uses Bender's Decomposition for large size instances.

3 MILO Formulations

In this section, we discuss a recent mixed integer linear optimization (MILO) formulation proposed by Aghaei et al. (2021). Then, we propose four MILO formulations (two flow-based and two cut-based) for designing optimal binary classification trees. We also compare the proposed models with each other and the MILO formulation of Aghaei et al. (2021). For bravity purposes and ease of comparisons, we employ similar notations employed by Aghaei et al. (2021). Let I, F, and K be the set of data, binary encoded features, and classes, respectively. For every datapoint $i \in I$, y^i denotes a class $k \in K$ that is assigned to datapoint i. For every datapoint $i \in I$ and every feature $f \in F$, binary parameter x^i_f equals one if datapoint i observes feature f. Graph $G_h = (V, E)$ denotes the input decision tree with height h, where $h \geq 1$. The number of vertices and edges of G_h are represented by $n := |V| = 2^{h+1} - 1$ and $m := |E| = 2^{h+1} - 2$, respectively. The vertex set V is the union of the branching vertex set, $B \subset V$, and the leaf vertex set, $L \subset V$, with $B \cap L = \emptyset$.

An optimal binary classification tree can be obtained by solving a biobjective optimization problem that seeks to (i) maximize the number of correctly classified datapoints and (ii) minimize the number of branching vertices. For every vertex $v \in V$, path $P_{1,v}$ and vertex set $V(P_{1,v})$ denote the unique $1, v$-path from vertex 1 to vertex v and the vertex set on the path $P_{1,v}$, including vertices 1 and v, respectively. For every datapoint $i \in I$ and every vertex $v \in V$, binary variable s^i_v equals one if datapoint i is correctly classified at vertex v. For every branching vertex $v \in B$ and every feature $f \in F$, binary variable b_{vf} equals one if vertex v is branched on feature f. For every vertex $v \in V$ and every class $k \in K$, binary variable w_{vk} equals one if vertex v is assigned to class k. Finally for every vertex $v \in V$, binary variable p_v equals one if a prediction class is assigned to vertex v.

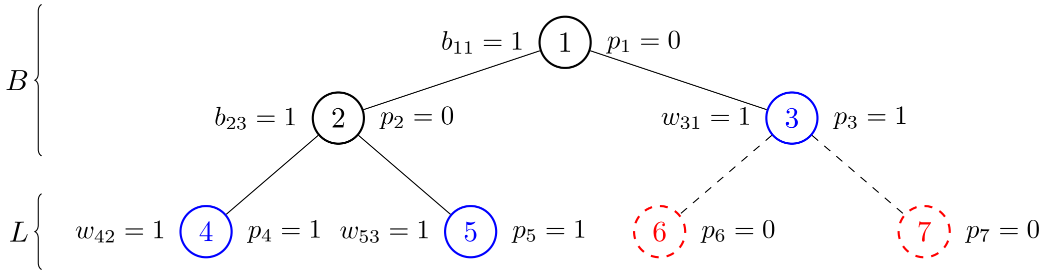


Figure 1: Input decision tree $G_2 = (B \cup L, E)$, branching vertex set $B = \{1, 2, 3\}$ and leaf vertex set $L = \{4, 5, 6, 7\}$. Here, vertices 1 and 2 are branched on features 1 and 3, respectively; vertices 3, 4, and 5 are assigned to a classes 1, 2, and 3, respectively; and vertices 6 and 7 are pruned.

A biobjective base model for designing optimal binary classification trees is provided below.

$$
\text{(BASE)}\quad
\begin{aligned}
&\max \sum_{i\in I}\sum_{v\in V} s_v^i && && \text{(1a)}\\
&\min \sum_{v\in B}\sum_{f\in F} b_{vf} && && \text{(1b)}\\
&p_v = \sum_{k\in K} w_{vk} && \forall v\in V && \text{(1c)}\\
&\sum_{f\in F} b_{vf} + \sum_{u\in V(P_{1,v})} p_u = 1 && \forall v\in V && \text{(1d)}\\
&b_{vf} = 0 && \forall v\in L,\ \forall f\in F && \text{(1e)}\\
&s_v^i \le w_{vk} && \forall i\in I : y^i = k,\ \forall k\in K,\ \forall v\in V && \text{(1f)}\\
&p\in[0,1]^n,\ s\in\{0,1\}^{|I|\times n}, && && \\
&b\in\{0,1\}^{n\times|F|},\ w\in\{0,1\}^{n\times|K|}. && && \text{(1g)}
\end{aligned}
$$

Here, objective function (1a) maximizes the number of correct classifications. Objective function (1b) minimizes the number of branching vertices. Constraints (1c) imply that a vertex is labeled with a prediction class if and only if it is assigned to a class $k \in K$. Constraints (1d) imply that a vertex is either branched on a feature, or a vertex on the $1, v$-path $P_{1,v}$ is assigned to a prediction class. Constraints (1e) imply that no leaf vertex is branched on a feature. Constraints (1f) imply that if datapoint $i \in I$ is classified at vertex $v \in V$, then vertex v is assigned to the class for which $y^i = k$. Constraints (1g) specify the domain of all decision variables of the base model. Furthermore, the polytope of the base model (1) is denoted as follows.

$$\mathscr{P}_{\text{BASE}} := \{(p, s, b, w) \in [0,1]^{n(1+|I|+|F|+|K|)} : (p, s, b, w) \text{ satisfies constraints (1c)-(1f)}\}.$$

Remark 1. *Constraints $p \in [0,1]^n$, $w \in \{0,1\}^{n\times|K|}$, and* (1c) *imply that $p \in \{0,1\}^n$.*

However, the base model does not guarantee feasible paths for datapoints to reach a correct classification vertex; that is, a datapoint can "jump up" onto the classification vertex 5 without going through branching vertex 2 in Figure 1. Hence, we need complementary MILO formulations with connectivity constraints to ensure that correctly classified datapoints can reach their corresponding classification vertex via feasible paths. In the following sections, we discuss five connectivity formulations: the flow-based connectivity formulation of Aghaei et al. (2021), two new flow-based formulations (MCF1 and MCF2), and two new cut-based formulations (CUT1 and CUT2).

3.1 Flow-based formulations

Let $D_h = (V', A)$ be a directed graph with vertex set $V' := V \cup \{t\}$ and arc set $A := \{(u, v) \mid \{u, v\} \in E,\ u < v\} \cup \{(v, t) \mid v \in V\}$. For every vertex $v \in V$, let (i) $\ell(v)$ be the left child of vertex v; (ii) $r(v)$ be the right child of vertex v; and (iii) $a(v)$ be the ancestor (parent) of vertex v in the directed graph D_h. Note that for every vertex $v \in L$, we have $\ell(v) = r(v) = \emptyset$. Furthermore, we do not employ auxiliary vertex s in tree structure of Aghaei et al. (2021) to present their formulation. For every datapoint $i \in I$ and every vertex $v \in V$, we also employ s_v^i instead of flow variable z_{vt}^i in the MILO formulation of Aghaei et al. (2021) (FlowOCT).

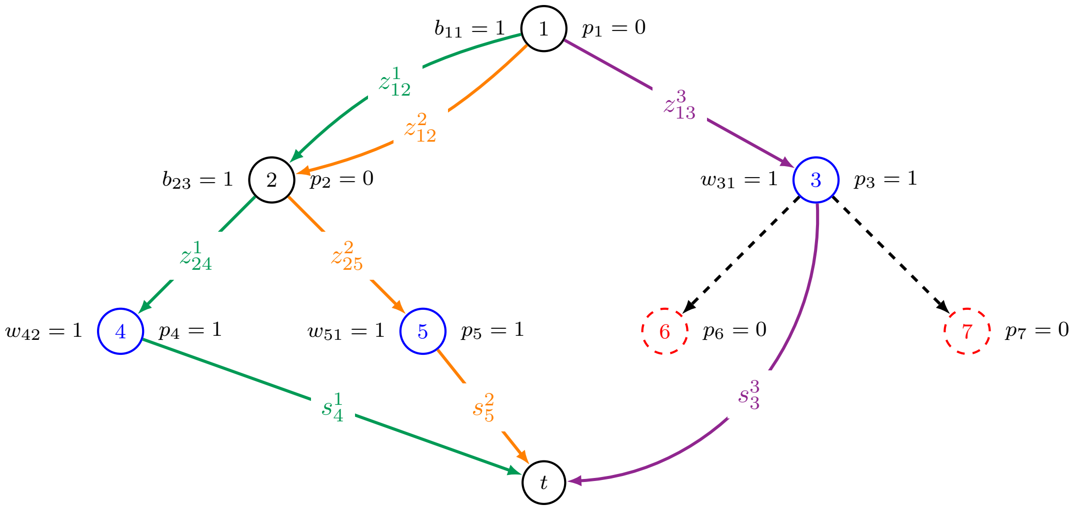


Figure 2: Directed decision tree $D_2 = (V', A)$ with height 2. Here, $I = \{1, 2, 3\}$.

3.1.1 FlowOCT (Aghaei et al., 2021)

For every datapoint $i \in I$ and every directed edge $(u, v) \in A$, variable z^i_{uv} denotes the flow of type i on edge (u, v). Because for every datapoint $i \in I$ and every vertex $v \in L$ we have $\ell(v) = r(v) = \emptyset$, we set variables $z^i_{v,\ell(v)}$ and $z^i_{v,r(v)}$ to zero.

$$
\text{(FlowOCT)}\quad
\begin{aligned}
& (p, s, b, w) \in \mathscr{P}_{\text{BASE}} && && \text{(2a)}\\
& z^i_{1,\ell(1)} + z^i_{1,r(1)} + s^i_1 \le 1 && \forall i \in I && \text{(2b)}\\
& z^i_{a(v),v} = z^i_{v,\ell(v)} + z^i_{v,r(v)} + s^i_v && \forall v \in V \setminus \{1\},\ \forall i \in I && \text{(2c)}\\
& z^i_{v,\ell(v)} \le \sum_{f \in F: x^i_f = 0} b_{vf} && && \\
& z^i_{v,r(v)} \le \sum_{f \in F: x^i_f = 1} b_{vf} && \forall v \in B,\ \forall i \in I && \text{(2d)}\\
& s \in \{0,1\}^{|I| \times n}, b \in \{0,1\}^{n \times |F|} && && \\
& w \in \{0,1\}^{n \times |K|},\ z \in \mathbb{R}^{m \times |I|}_+. && && \text{(2e)}
\end{aligned}
$$

Here, constraints (2b) imply that for every datapoint $i \in I$, at most one unit of flow of type i can emanate from vertex 1. Constraints (2c) imply flow conservation for any datapoint $i \in I$ at any vertex $v \in V \setminus \{1\}$. Constraints (2d) imply that if a flow of type $i \in I$ is directed towards the left (right) child of a vertex, then the vertex is branched on a feature $f \in F$ with $x^i_f = 0$ ($x^i_f = 1$). Figure 2 illustrates directed graph D'_2 and the corresponding flow variables of formulation FlowOCT.

Now, we define the polytope of the FlowOCT formulation as follows.

$$\mathscr{P}_{\text{FlowOCT}} \coloneqq \{(p, s, b, w) \in \mathscr{P}_{\text{BASE}},\ z \in \mathbb{R}^{m \times |I|}_+ : (p, s, b, w, z) \text{ satisfies constraints (2b)-(2d)}\}.$$

Remark 2. *The following constraints are implied in $\mathscr{P}_{\text{FlowOCT}}$.*

$$\sum_{v\in V} s_v^i \leq 1 \qquad\qquad \forall i \in I.$$

Proof. Let $(\hat{p}, \hat{s}, \hat{b}, \hat{w}, \hat{z})$ be a point in $\mathscr{P}_{\text{FlowOCT}}$. For every datapoint $i \in I$, we have

$$\begin{aligned}
\sum_{v\in V} \hat{s}_v^i &= \sum_{v\in V} \hat{s}_v^i + 0 && \text{(3a)}\\
&= \sum_{v\in V} \hat{s}_v^i + \left(\sum_{v\in V} \left(\hat{z}^i(\delta^+(v)) + \hat{s}_v^i - \hat{z}^i(\delta^-(v)) \right) - \sum_{v\in V} \hat{s}_v^i \right) && \text{(3b)}\\
&= \hat{z}^i(\delta^+(1)) + \hat{s}_1^i - \hat{z}^i(\delta^-(1)) + \sum_{v\in V\setminus\{1\}} \left(\hat{z}^i(\delta^+(v)) + \hat{s}_v^i - \hat{z}^i(\delta^-(v)) \right) && \text{(3c)}\\
&= \hat{z}^i(\delta^+(1)) + \hat{s}_1^i - \hat{z}^i(\delta^-(1)) && \text{(3d)}\\
&= \hat{z}^i(\delta^+(1)) + \hat{s}_1^i && \text{(3e)}\\
&\leq 1. && \text{(3f)}
\end{aligned}$$

Here, equality (3b) holds because the summation of all flows is zero in directed graph D_h. Equality (3d) holds by constraints (2c). Equality (3e) holds because the incoming flow to vertex one is zero in directed graph D_h. Finally, inequality (3f) holds by constraints (2b). This finishes the proof. □

3.1.2 MCF1

In this section, we propose an extended version of the FlowOCT formulation with same order in number of variables and constraints. In this formulation, we introduce a new binary variable q_v^i for every datapoint $i \in I$ and every vertex $v \in V'$: binary variable q_v^i equals one if datapoint i enters vertex v to reach terminal vertex $t \in V'$. Figure 3 illustrates directed graph D_2' and the corresponding flow variables of formulation MCF1.

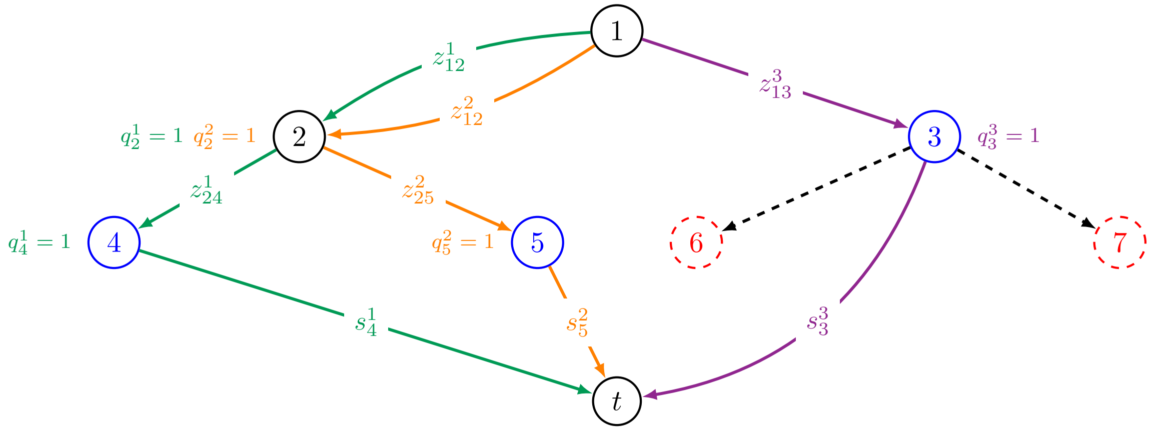


Figure 3: Directed decision tree $D_2 = (V', A)$ with decision variables of model MCF1.

$$
\text{(MCF1)}\quad
\begin{aligned}
& (p, s, b, w) \in \mathscr{P}_{\text{BASE}} && && \text{(4a)} \\
& z^i_{a(v),v} \le q^i_v && \forall v \in V \setminus \{1\},\ \forall i \in I && \text{(4b)} \\
& z^i_{a(v),v} = z^i_{v,\ell(v)} + z^i_{v,r(v)} + s^i_v && \forall v \in V \setminus \{1\},\ \forall i \in I && \text{(4c)} \\
& z^i_{1,\ell(1)} + z^i_{1,r(1)} + s^i_1 = q^i_t && \forall i \in I && \text{(4d)} \\
& q^i_{\ell(v)} \le \sum_{f \in F: x^i_f = 0} b_{vf} && && \\
& q^i_{r(v)} \le \sum_{f \in F: x^i_f = 1} b_{vf} && \forall v \in B,\ \forall i \in I && \text{(4e)} \\
& s \in \{0,1\}^{|I| \times n},\ b \in \{0,1\}^{n \times |F|} && && \\
& w \in \{0,1\}^{n \times |K|},\ z \in \mathbb{R}_+^{m \times |I|},\ q \in \{0,1\}^{|I| \times |V'|}. && && \text{(4f)}
\end{aligned}
$$

Here, constraints (4b) imply that if a datapoint goes through a directed edge $(a(v), v)$, then it reaches vertex v. Constraints (4c) imply flow conservation for each data point at every vertex $v \in V \setminus \{1\}$. Constraints (4d) imply that if datapoint $i \in I$ reaches terminal vertex t, then either the datapoint is correctly classified at vertex 1 or a flow of type i is generated at vertex 1. Constraints (4e) imply that if a vertex $v \in V \setminus \{1\}$ is reached by a datapoint, then its parent is branched on a feature.

Now, we define the polytope of the MCF1 model as follows.

$$
\mathscr{P}_{\text{MCF1}} := \Big\{ (p, s, b, w) \in \mathscr{P}_{\text{BASE}},\ z \in \mathbb{R}_+^{m \times |I|},\ q \in [0,1]^{|I| \times |V'|} : \\ (p, s, b, w, z, q) \text{ satisfies constraints (4b)-(4e)} \Big\}
$$

Remark 3. *The following constraints are implied in* $\mathscr{P}_{\text{MCF1}}$.

$$
\sum_{v \in V} s^i_v = q^i_t \qquad \forall i \in I. \tag{5}
$$

Proof. Let $(\hat{p}, \hat{s}, \hat{b}, \hat{w}, \hat{z}, \hat{q})$ be a point in $\mathscr{P}_{\text{MCF1}}$. For every datapoint $i \in I$, we have

$$
\begin{aligned}
\sum_{v \in V} \hat{s}^i_v &= \sum_{v \in V} \hat{s}^i_v + 0 && \text{(6a)} \\
&= \sum_{v \in V} \hat{s}^i_v + \Big(\sum_{v \in V} \Big(\hat{z}^i(\delta^+(v)) + \hat{s}^i_v - \hat{z}^i(\delta^-(v)) \Big) - \sum_{v \in V} \hat{s}^i_v \Big) && \text{(6b)} \\
&= \hat{z}^i(\delta^+(1)) + \hat{s}^i_1 - \hat{z}^i(\delta^-(1)) + \sum_{v \in V \setminus \{1\}} \Big(\hat{z}^i(\delta^+(v)) + \hat{s}^i_v - \hat{z}^i(\delta^-(v)) \Big) && \text{(6c)} \\
&= \hat{z}^i(\delta^+(1)) + \hat{s}^i_1 - \hat{z}^i(\delta^-(1)) && \text{(6d)} \\
&= \hat{q}^i_t. && \text{(6e)}
\end{aligned}
$$

Here, equality (6b) holds because the summation of all flows is zero in directed graph D_h. Equality (6d) holds by constraints (4c). Finally, equality (6e) holds by constraints (4d). This finishes the proof. □

3.1.3 MCF2

In this section, we propose a multi-commodity flow formulation in which flows are labeled by datapoints and their destination. Now, we define directed graph $D'_h := D_h - t$. For every datapoint $i \in I$, every destination vertex $v \in V \setminus \{1\}$, and every directed edge $(c, d) \in E(D'_h)$, decision variable z^{iv}_{cd} denotes the flow of type datapoint $i \in I$ emanating from the root vertex and heading to vertex $v \in V \setminus \{1\}$ on directed edge $(c, d) \in E$. Figure 4 illustrates an example of directed graph D'_2 and the corresponding flow variables of the MCF2 formulation.

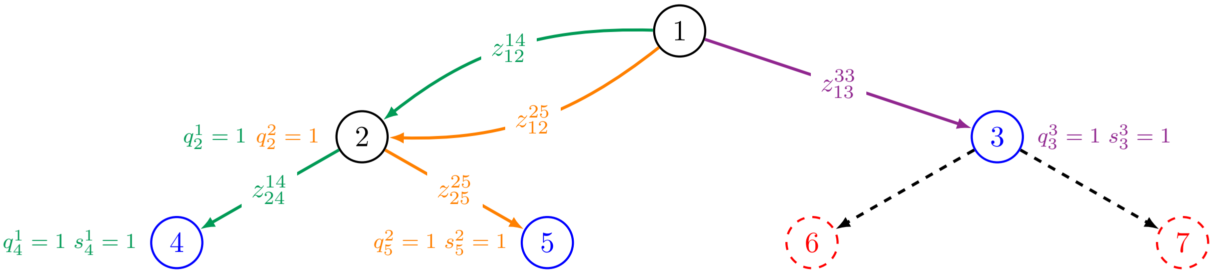


Figure 4: Directed decision tree $D'_2 = D_2 - t$ with decision variables of MCF2.

$$
\text{(MCF2)}\quad
\begin{aligned}
& (p, s, b, w) \in \mathscr{P}_{\text{BASE}} && && \text{(7a)}\\
& z^{iv}_{1,\ell(1)} + z^{iv}_{1,r(1)} = s^i_v && \forall v \in V \setminus \{1\},\ \forall i \in I && \text{(7b)}\\
& z^{iv}_{u,\ell(u)} + z^{iv}_{u,r(u)} - z^{iv}_{a(u),u} = 0 && \forall u \in V \setminus \{1, v\},\ \forall v \in V \setminus \{1\},\ \forall i \in I && \text{(7c)}\\
& \sum_{u \in V \setminus \{1\}} z^{iu}_{a(v),v} \le q^i_v && \forall v \in V \setminus \{1\},\ \forall i \in I && \text{(7d)}\\
& z^{iv}_{a(v),v} = s^i_v && \forall v \in V \setminus \{1\},\ \forall i \in I && \text{(7e)}\\
& q^i_{\ell(v)} \le \sum_{f \in F : x^i_f = 0} b_{vf} && && \\
& q^i_{r(v)} \le \sum_{f \in F : x^i_f = 1} b_{vf} && \forall v \in B,\ \forall i \in I && \text{(7f)}\\
& \sum_{v \in V} s^i_v \le 1 && \forall i \in I && \text{(7g)}\\
& s \in \{0,1\}^{|I| \times n}, && && \\
& b \in \{0,1\}^{n \times |F|},\ w \in \{0,1\}^{n \times |K|}, && && \\
& z \in \mathbb{R}^{m \times |I| \times n}_{+},\ q \in \{0,1\}^{|I| \times n}. && && \text{(7h)}
\end{aligned}
$$

Here, constraints (7b) imply that if datapoint $i \in I$ is correctly classified at vertex $v \in V \setminus \{1\}$, then a flow of type iv originates from vertex 1. Constraints (7c) imply the flow conservation of datapoint i heading towards vertex v at vertex $u \in V \setminus \{1, v\}$. Constraints (7d) imply that if a flow of datapoint $i \in I$ that is heading toward vertex u enters vertex v, then vertex v is selected on the $1, u$-path. Constraints (7e) imply that datapoint $i \in I$ is correctly classified at vertex $v \in V \setminus \{1\}$ if and only if a flow of type iv enters vertex v. Constraints (7f) imply that if the left (right) child of parent $v \in V$ is selected on the path of datapoint $i \in I$, then its parent is branched on a feature $f \in F$ for which $x^i_f = 0$ ($x^i_f = 1$). Constraints (7g) imply that each datapoint $i \in I$ can be correctly classified in at most one vertex.

Now, we define the polytope of the MCF2 model as follows.

$$\mathscr{P}_{\text{MCF2}} \coloneqq \Big\{(p,s,b,w) \in \mathscr{P}_{\text{BASE}},\ z \in \mathbb{R}_+^{m\times n\times |I|},\ q \in [0,1]^{|I|\times|V'|} :$$
$$(p,s,b,w,z,q) \text{ satisfies constraints (7b)-(7g)}\Big\}$$

Remark 4. *The following constraints are implied in $\mathscr{P}_{\text{MCF2}}$.*

$$z^{iv}_{a(v),v} - z^{iv}_{v,\ell(v)} - z^{iv}_{v,r(v)} = s^i_v \qquad \forall v \in V \setminus \{1\},\ \forall i \in I, \tag{8}$$
$$z^{iv}_{v,\ell(v)} + z^{iv}_{v,r(v)} = 0 \qquad \forall v \in V \setminus \{1\},\ \forall i \in I. \tag{9}$$

Proof. Let $(\hat{p},\hat{s},\hat{b},\hat{w},\hat{z},\hat{q})$ be a point that belongs to $\mathscr{P}_{\text{MCF2}}$. We are to show that the point satisfies constraints (8). For every vertex $v \in V \setminus \{1\}$ and every datapoint $i \in I$, we have

$$\begin{aligned}
\hat{z}^{iv}_{a(v),v} - \hat{z}^{iv}_{v,\ell(v)} - \hat{z}^{iv}_{v,r(v)} &= \hat{z}^{iv}(\delta^-(v)) - \hat{z}^{iv}(\delta^+(v)) - \sum_{u\in V}\big(\hat{z}^{iv}(\delta^-(u)) - \hat{z}^{iv}(\delta^+(u))\big) \\
&= -\sum_{u\in V\setminus\{v\}}\big(\hat{z}^{iv}(\delta^-(u)) - \hat{z}^{iv}(\delta^+(u)\big) \\
&= -\big(\hat{z}^{iv}(\delta^-(1)) - \hat{z}^{iv}(\delta^+(1)\big) = \hat{z}^{iv}_{1,\ell(1)} + \hat{z}^{iv}_{1,r(1)} = \hat{s}^i_v.
\end{aligned}$$

Here, the first equality holds because the summation of the net flows on all vertices of the directed graph D'_h is zero. The last equality holds by constraints (7b).

Now we are to show that point $(\hat{p},\hat{s},\hat{b},\hat{w},\hat{z},\hat{q})$ satisfies constraints (9). For every vertex $v \in V \setminus \{1\}$ and every datapoint $i \in I$, we have

$$\hat{z}^{iv}_{v,\ell(v)} + \hat{z}^{iv}_{v,r(v)} = \hat{z}^{iv}_{a(v),v} - \hat{s}^i_v = \hat{s}^i_v - \hat{s}^i_v = 0.$$

Here, the first equality holds by implied equality (8). The second equality holds by constraints (7e). This finishes the proof. □

3.2 Cut-based formulations

In this section, we propose two cut-based formulations. In both cut-based models, the connectivity constraints can be added on-the-fly. It needs more explanation!

3.2.1 CUT1

This formulation is the equivalent cut-based formulation for model MCF2. For every vertex $v \in V$, we define P_v as the set of non-root vertices on the $1, v$-path.

$$
\text{(CUT1)}\quad
\begin{aligned}
& (p, s, b, w) \in \mathscr{P}_{\text{BASE}} && && \text{(10a)}\\
& s_v^i \le q_c^i && \forall c \in V(P_v),\ \forall v \in V \setminus \{1\},\ \forall i \in I && \text{(10b)}\\
& q_{\ell(v)}^i \le \sum_{f \in F : x_f^i = 0} b_{vf} && && \\
& q_{r(v)}^i \le \sum_{f \in F : x_f^i = 1} b_{vf} && \forall v \in B,\ \forall i \in I && \text{(10c)}\\
& \sum_{v \in V} s_v^i \le 1 && \forall i \in I && \text{(10d)}\\
& s \in \{0,1\}^{|I| \times n},\ b \in \{0,1\}^{n \times |F|} && && \\
& w \in \{0,1\}^{n \times |K|},\ q \in \{0,1\}^{|I| \times n}. && && \text{(10e)}
\end{aligned}
$$

Here, constraints (10b) imply that if a datapoint $i \in I$ is correctly classified at vertex $v \in V \setminus \{1\}$, then all vertices on the path from 1 to v, excluding vertex 1, must be selected. We also note that there are $O(nh|I|)$ of these constraints in the model. Explanations of constraints (10c) and (10d) are similar to those of constraints (7f) and (7g), respectively.

Furthermore, we define the polytope of formulation (10) as follows.

$$\mathscr{P}_{\text{CUT1}} \coloneqq \Big\{(p, s, b, w) \in \mathscr{P}_{\text{BASE}}, q \in [0,1]^{|I| \times n} : (p, s, b, w, q) \text{ satisfies constraints (10b)-(10d)}\Big\}.$$

3.2.2 CUT2

We propose another cut-based formulation whose linear optimization relaxation is stronger than that of the CUT1 model. For every vertex $v \in V \setminus \{1\}$, we define

$$\text{CHILD}(v) \coloneqq \{u \in V \setminus \{v\} : u > v,\ \text{dist}_{D'}(v, u) < \infty\},$$

where $\text{dist}_{D'}(v, u)$ denotes the distance between vertices v and u in directed graph D'. For example in Figure 4, $\text{CHILD}(2) = \{4, 5\}$.

$$
\text{(CUT2)}\quad
\begin{aligned}
& (p, s, b, w) \in \mathscr{P}_{\text{BASE}} && && \text{(11a)}\\
& s_v^i + \sum_{u \in \text{CHILD}(v)} s_u^i \le q_c^i && \forall c \in V(P_v),\ \forall v \in V \setminus \{1\},\ \forall i \in I && \text{(11b)}\\
& q_{\ell(v)}^i \le \sum_{f \in F : x_f^i = 0} b_{vf} && && \\
& q_{r(v)}^i \le \sum_{f \in F : x_f^i = 1} b_{vf} && \forall v \in B,\ \forall i \in I && \text{(11c)}\\
& \sum_{v \in V} s_v^i \le 1 && \forall i \in I && \text{(11d)}\\
& s \in \{0,1\}^{|I| \times n},\ b \in \{0,1\}^{n \times |F|} && && \\
& w \in \{0,1\}^{n \times |K|},\ q \in \{0,1\}^{|I| \times n}. && && \text{(11e)}
\end{aligned}
$$

Here, constraints (11b) imply that if datapoint $i \in I$ is correctly classified at vertex v or one of its descendants, then the datapoint passes through every vertex on the path from 1 to v excluding vertex 1. Interpretations of constraints (11c) and (11d) are similar to those of constraints (7f) and (7g), respectively.

Furthermore, we define the polytope of formulation (11) as follows.

$$\mathscr{P}_{\mathrm{CUT2}} \coloneqq \Big\{(p,s,b,w) \in \mathscr{P}_{\mathrm{BASE}},\ q \in [0,1]^{|I|\times n} : (p,s,b,w,q) \text{ satisfies constraints (11b)-(11d)}\Big\}.$$

4 Theoretical comparison of formulations

In this section, we mainly prove the following theorem that compares the strength of different formulations discussed in Section 3.

Theorem 1 (Strength of Formulations)**.** *Let $X = (p,s,b,w,q)$. Then,*

$$\mathscr{P}_{\mathrm{CUT2}} \subseteq \mathscr{P}_{\mathrm{CUT1}} = \mathrm{proj}_X\, \mathscr{P}_{\mathrm{MCF2}} = \mathrm{proj}_X\, \mathscr{P}_{\mathrm{MCF1}} \subseteq \mathrm{proj}_X\, \mathscr{P}_{\mathrm{FlowOCT}}.$$

Proof. The proof follows by Lemmata (1), (2), (3), and (4). □

Lemma 1 shows that the CUT2 model is at least as strong as the CUT1 model.

Lemma 1. $\mathscr{P}_{\mathrm{CUT2}} \subseteq \mathscr{P}_{\mathrm{CUT1}}$*, and the inclusion can be strict.*

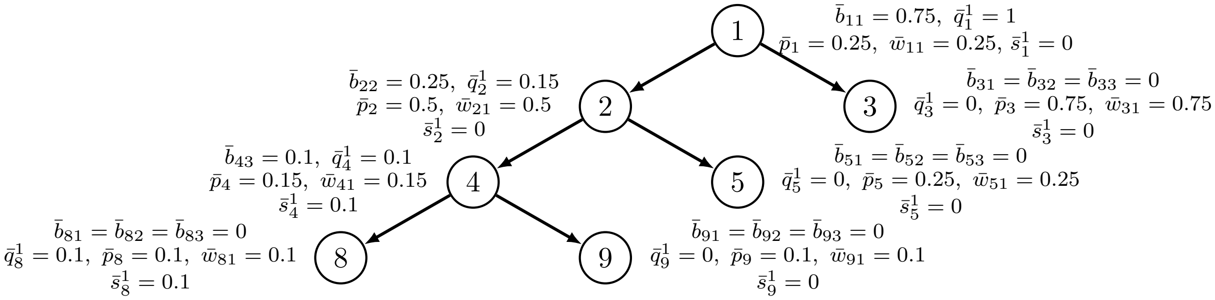
Proof. Let $(\hat{p},\hat{s},\hat{b},\hat{w},\hat{q})$ be a point that belongs to CUT2. We are to show that it also belongs to CUT1. It is clear that the point satisfies constraints (10a), (10c), and (10d). For every datapoint $i \in I$ and for every vertex $v \in V$ and for every cut vertex c on the path P_v, we show that point $(\hat{p},\hat{s},\hat{b},\hat{w},\hat{q})$ satisfies constraints (10b).

$$\hat{s}^i_v \leq \hat{s}^i_v + \sum_{u \in \mathrm{CHILD}(v)} \hat{s}^i_u \leq \hat{q}^i_c$$

The first inequality holds by the nonnegatviity of $\hat{s}^i_v$ for all $v \in V$ and $i \in I$. The second inequality holds by constraints (11b). This completes the proof.

Furthermore, Figure 5 illustrates an instance of the optimal binary decision tree problem that belongs to the $\mathscr{P}_{\mathrm{CUT1}}$ polytope, but not the $\mathscr{P}_{\mathrm{CUT2}}$ polytope. In this instance, we have one datapoint with three features and one class. Furthermore, we have $x^1_1 = x^1_2 = x^1_3 = 0$ and $y^1 = 1$. While the point belongs to the polytope $\mathscr{P}_{\mathrm{CUT1}}$, it does not belong to $\mathscr{P}_{\mathrm{CUT2}}$ because it violates constraints (11b) for vertex $v = 4$ and cut vertex $c = 2$. In better words,

$$\bar{s}^1_4 + \bar{s}^1_8 + \bar{s}^1_9 = 0.1 + 0.1 + 0 = 0.2 \not\leq 0.15 = q^1_2.$$



1
$\bar{b}_{11} = 0.75,\ \bar{q}^1_1 = 1$
$\bar{p}_1 = 0.25,\ \bar{w}_{11} = 0.25,\ \bar{s}^1_1 = 0$
2
$\bar{b}_{22} = 0.25,\ \bar{q}^1_2 = 0.15$
$\bar{p}_2 = 0.5,\ \bar{w}_{21} = 0.5$
$\bar{s}^1_2 = 0$
3
$\bar{b}_{31} = \bar{b}_{32} = \bar{b}_{33} = 0$
$\bar{q}^1_3 = 0,\ \bar{p}_3 = 0.75,\ \bar{w}_{31} = 0.75$
$\bar{s}^1_3 = 0$
4
$\bar{b}_{43} = 0.1,\ \bar{q}^1_4 = 0.1$
$\bar{p}_4 = 0.15,\ \bar{w}_{41} = 0.15$
$\bar{s}^1_4 = 0.1$
5
$\bar{b}_{51} = \bar{b}_{52} = \bar{b}_{53} = 0$
$\bar{q}^1_5 = 0,\ \bar{p}_5 = 0.25,\ \bar{w}_{51} = 0.25$
$\bar{s}^1_5 = 0$
8
$\bar{b}_{81} = \bar{b}_{82} = \bar{b}_{83} = 0$
$\bar{q}^1_8 = 0.1,\ \bar{p}_8 = 0.1,\ \bar{w}_{81} = 0.1$
$\bar{s}^1_8 = 0.1$
9
$\bar{b}_{91} = \bar{b}_{92} = \bar{b}_{93} = 0$
$\bar{q}^1_9 = 0,\ \bar{p}_9 = 0.1,\ \bar{w}_{91} = 0.1$
$\bar{s}^1_9 = 0$

Figure 5: A point that belongs to $\mathscr{P}_{\text{CUT1}}$, but not $\mathscr{P}_{\text{CUT2}}$.

□

Lemma 2. *Let $X = (p, s, b, w, q)$. Then, $\mathscr{P}_{\text{CUT1}} = \operatorname{proj}_X \mathscr{P}_{\text{MCF2}}$.*

Proof. ($\supseteq$) Let $(\hat{p}, \hat{s}, \hat{b}, \hat{w}, \hat{q}, \hat{z})$ be a point that belongs to polytope $\mathscr{P}_{\text{MCF2}}$. We are to show that point $(\hat{p}, \hat{s}, \hat{b}, \hat{w}, \hat{q})$ belongs to polytope $\mathscr{P}_{\text{CUT1}}$. It suffices to show that $(\hat{p}, \hat{s}, \hat{b}, \hat{w}, \hat{q})$ satisfies constraints (10b). For every datapoint $i \in I$, every vertex $v \in V \setminus \{1\}$, and every cut vertex c on the path P_v, let R be the set of reachable vertices from vertex 1 in tree $G - c$. Then, we have

$$\begin{align}
\hat{s}^i_v &= \hat{z}^{iv}_{1,\ell(1)} + \hat{z}^{iv}_{1,r(1)} \tag{12a}\\
&= \hat{z}^{iv}_{1,\ell(1)} + \hat{z}^{iv}_{1,r(1)} + \hat{z}^{iv}(\delta^+(R \setminus \{1\})) - \hat{z}^{iv}(\delta^-(R \setminus \{1\})) \tag{12b}\\
&\leq \hat{z}^{iv}(\delta^+(R)) - 0 \tag{12c}\\
&= \hat{z}^{iv}(\delta^-(c)) \tag{12d}\\
&\leq \sum_{u \in V \setminus \{1\}} \hat{z}^{iu}(\delta^-(c)) \tag{12e}\\
&\leq \hat{q}^i_c. \tag{12f}
\end{align}$$

Here, equality (12a) holds by constraints (7b). Equality (12b) holds by flow conservation constraints (7c). Inequality (12c) holds by nonnegativity of z variables. Equality (12d) holds because vertex c is a $1, v$-separator. Inequality (12e) holds by nonnegativity of z variables. Finally, inequality (12f) holds by constraints (7d).

($\subseteq$) Let $(\hat{b}, \hat{w}, \hat{s}, \hat{p}, \hat{q})$ be a point that belongs to $\mathscr{P}_{\text{CUT1}}$. We are to show that there exists $\bar{z}$ such that $(\hat{b}, \hat{w}, \hat{s}, \hat{p}, \hat{q}, \bar{z})$ belongs to $\mathscr{P}_{\text{MCF1}}$. It suffices to show that the point satisfies constraints (7b)-(7d). To construct $\bar{z}$, we solve the following max-flow problem for every datapoint $i \in I$ and every vertex $v \in V \setminus \{1\}$.

$$\begin{align}
&\max\ z^{iv}(\delta^+(1)) - z^{iv}(\delta^-(1)) \tag{13a}\\
&z^{iv}(\delta^+(u)) - z^{iv}(\delta^-(u)) = 0 && \forall u \in V \setminus \{1, v\},\ \forall i \in I \tag{13b}\\
&z^{iv}(\delta^-(v)) \leq \hat{s}^i_v \tag{13c}\\
&z^{iv}(\delta^-(u)) \leq \frac{\hat{q}^i_v}{n-2} && \forall u \in V \setminus \{1, v\},\ \forall i \in I \tag{13d}\\
&z^{iv}_a \geq 0 && \forall v \in V,\ \forall i \in I,\ \forall a \in E(G). \tag{13e}
\end{align}$$

Let $\bar{z}$ be an optimal solution obtained by solving the max-flow problem (13) for every datapoint $i \in I$ and every vertex $u \in V \setminus \{1\}$. It is clear that point $(\hat{b}, \hat{w}, \hat{s}, \hat{p}, \hat{q}, \bar{z})$ satisfies constraints (7c) because it satisfies constraints (13b). Furthermore, one can sum up both sides of constraints (13d) on vertices $V \setminus \{1, v\}$. This results in

$$\sum_{u \in V \setminus \{1,v\}} \bar{z}^{iv}(\delta^-(u)) \leq \sum_{u \in V \setminus \{1,v\}} \frac{\hat{q}_v^i}{n-2} = \hat{q}_v^i. \tag{14}$$

Hence, constraints (7d) are satisfied. Now we show that $(\hat{b}, \hat{w}, \hat{s}, \hat{p}, \hat{q}, \bar{z})$ satisfies constraints (7f). For every datapoint $i \in I$, every vertex $v \in V \setminus \{1\}$ and every $1, v$-separator $c \in P_v$, we have

$$\hat{s}_v^i \leq \hat{q}_c^i = \bar{z}^{iv}(\delta^+(1)) - \bar{z}^{iv}(\delta^-(1)) \leq \hat{s}_v^i.$$

Here, the first inequality holds by constraints (10b). The first equality holds by the max-flow/min-cut result of Ford Jr. and Fulkerson (1962). The last inequality holds because

$$\begin{aligned}
\bar{z}^{iv}(\delta^+(1)) - \bar{z}^{iv}(\delta^-(1)) &= \bar{z}^{iv}(\delta^+(1)) - \bar{z}^{iv}(\delta^-(1)) - \sum_{u \in V} \left(\bar{z}^{iv}(\delta^+(u)) - \bar{z}^{iv}(\delta^-(u))\right) \\
&= - \sum_{u \in V \setminus \{1\}} \left(\bar{z}^{iv}(\delta^+(u)) - \bar{z}^{iv}(\delta^-(u))\right) \\
&= - \left(\bar{z}^{iv}(\delta^+(v)) - \bar{z}^{iv}(\delta^-(v))\right) \\
&= \bar{z}^{iv}(\delta^-(v)) - \bar{z}^{iv}(\delta^+(v)) \leq \bar{z}^{iv}(\delta^-(v)) \leq \hat{s}_v^i.
\end{aligned}$$

Here, the first equality holds because $\sum_{u \in V} \left(\bar{z}^{iv}(\delta^+(u)) - \bar{z}^{iv}(\delta^-(u))\right) = 0$ for every datapoint $i \in I$ and every vertex $v \in V \setminus \{1\}$. The third equality holds by constraints (13b). The last inequality holds by constraints (13c). □

Lemma 3. *Let* $X = (p, s, b, w, q)$. *Then* $\operatorname{proj}_X \mathscr{P}_{\mathrm{MCF2}} = \operatorname{proj}_X \mathscr{P}_{\mathrm{MCF1}}$.

Proof. ($\subseteq$) Let $(\hat{p}, \hat{s}, \hat{b}, \hat{w}, \hat{q}, \hat{z})$ be a point that belongs to $\mathscr{P}_{\mathrm{MCF2}}$. We are to show that there exist $\bar{z}$ and $\bar{q}$ such that $(\hat{p}, \hat{s}, \hat{b}, \hat{w}, \bar{q}, \bar{z}) \in \mathscr{P}_{\mathrm{MCF1}}$. For every datapoint $i \in I$ and directed edge $(u, v) \in E(D'_h)$, we define

$$\bar{z}_{uv}^i := \sum_{j \in V \setminus \{1\}} \hat{z}_{uv}^{ij}. \tag{15}$$

For every datapoint $i \in I$ and every vertex $v \in V$, we define $\bar{q}_v^i := \hat{q}_v^i$. For every datapoint $i \in I$ and vertex $t \in V'$, we define

$$\bar{q}_t^i := \sum_{v \in V} \hat{s}_v^i. \tag{16}$$

We note that $\bar{q}_t^i \leq 1$ by constraints (7g). Further, it is clear that $(\hat{p}, \hat{s}, \hat{b}, \hat{w}, \bar{z}, \bar{q}) \in \mathscr{P}_{\mathrm{BASE}}$. So, it suffices to show that $(\hat{p}, \hat{s}, \hat{b}, \hat{w}, \bar{z}, \bar{q})$ satisfies constraints (4b)-(4e). For every datapoint $i \in I$ and every vertex $v \in V \setminus \{1\}$, we show that $(\hat{p}, \hat{s}, \hat{b}, \hat{w}, \bar{z}, \bar{q})$ satisfies constraints (4b).

$$\bar{z}_{a(v),v}^i = \sum_{u \in V \setminus \{1\}} \hat{z}_{a(v),v}^{iu} \leq \hat{q}_v^i = \bar{q}_v^i.$$

Here, the first equality holds by definition (15). The first inequality holds by constraints (7d). The last equality holds by the definition of $\bar{q}$.

For every datapoint $i \in I$ and for every vertex $v \in V \setminus \{1\}$, we are to show that $(\hat{p}, \hat{s}, \hat{b}, \hat{w}, \bar{z}, \bar{q})$ satisfies constraints (4c).

$$\begin{aligned}
\bar{z}^i_{a(v),v} - \bar{z}^i_{v,\ell(v)} - \bar{z}^i_{v,r(v)} &= \sum_{u \in V \setminus \{1\}} \hat{z}^{iu}_{a(v),v} - \sum_{u \in V \setminus \{1\}} \big(\hat{z}^{iu}_{v,\ell(v)} + \hat{z}^{iu}_{v,r(v)}\big) \\
&= \sum_{u \in V \setminus \{1,v\}} \hat{z}^{iu}_{a(v),v} - \sum_{u \in V \setminus \{1,v\}} \big(\hat{z}^{iu}_{v,\ell(v)} + \hat{z}^{iu}_{v,r(v)}\big) \\
&\quad + \hat{z}^{iv}_{a(v),v} - \big(\hat{z}^{iv}_{v,\ell(v)} + \hat{z}^{iv}_{v,r(v)}\big) \\
&= 0 + \hat{z}^{iv}_{a(v),v} - \big(\hat{z}^{iv}_{v,\ell(v)} + \hat{z}^{iv}_{v,r(v)}\big) \\
&= 0 + \hat{s}^i_v - 0 = \hat{s}^i_v.
\end{aligned}$$

Here, the first equality holds by definition (15). The third equality holds by constraints (7c). Finally, the last line holds by constraints (7e) and implied constraints (9).

For every datapoint $i \in I$, we show that point $(\hat{p}, \hat{s}, \hat{b}, \hat{w}, \bar{z}, \bar{q})$ satisfies constraints (4d).

$$\begin{aligned}
\bar{q}^i_t &= \sum_{v \in V} \hat{s}^i_v \\
&= \hat{s}^i_1 + \sum_{v \in V \setminus \{1\}} \hat{s}^i_v \\
&= \hat{s}^i_1 + \sum_{v \in V \setminus \{1\}} \left(\hat{z}^{iv}_{1,\ell(1)} + \hat{z}^{iv}_{1,r(1)}\right) \\
&= \hat{s}^i_1 + \bar{z}^i_{1,\ell(1)} + \bar{z}^i_{1,r(1)}.
\end{aligned}$$

Here, the first equality holds by definition (16). The third equality holds by constraints (7b). Finally, the last equality holds by definition (15). Further, it is clear that $(\hat{p}, \hat{s}, \hat{b}, \hat{w}, \bar{z}, \bar{q})$ satisfies constraints (4e) by the definition of $\bar{q}^i_v$ for every datapoint $i \in I$ and for every vertex $v \in V$; i.e., $\bar{q}^i_v := \hat{q}^i_v$.

($\supseteq$) As $\mathscr{P}_{\text{CUT1}} = \text{proj}_X \mathscr{P}_{\text{MCF2}}$ by Lemma 2, it suffices to show that $\mathscr{P}_{\text{CUT1}} \supseteq \text{proj}_X \mathscr{P}_{\text{MCF1}}$. Let $(\hat{p}, \hat{s}, \hat{b}, \hat{w}, \hat{q}, \hat{z})$ be a point that belongs to $\mathscr{P}_{\text{MCF1}}$. We are to show that $(\hat{p}, \hat{s}, \hat{b}, \hat{w}, \hat{q})$ belongs to the $\mathscr{P}_{\text{CUT1}}$ polytope. It suffices to show that $(\hat{p}, \hat{s}, \hat{b}, \hat{w}, \hat{q})$ satisfies constraints (10b) and (10d). For every datapoint $i \in I$, for every vertex $v \in V \setminus \{1\}$, for every cut vertex $c \in V(P_v)$, where P_v is the $1, v$-path including vertex v, we have,

$$\hat{s}^i_v \leq \hat{z}^i(\delta^-(v)) \leq \hat{z}^i(\delta^-(c)) \leq \hat{q}^i_c.$$

Here, the first and second inequalities hold by flow conservation constraints (4c). The last inequality holds by constraints (4b).

Now we show that the point satisfies constraints (10d).

$$\sum_{v \in V} \hat{s}^i_v = \hat{q}^i_t \leq 1.$$

Here, the first equality holds by implied constraints (5). The second inequality holds because $\hat{q}^i_t$ is a binary variable. □

Lemma 4. *Let $X = (p, s, b, w, q)$. Then* $\text{proj}_X \mathscr{P}_{\text{MCF1}} \subseteq \text{proj}_X \mathscr{P}_{\text{FlowOCT}}$.

Proof. Let $(\hat{p}, \hat{s}, \hat{b}, \hat{w}, \hat{q}, \hat{z})$ be a point that belongs to the $\mathscr{P}_{\text{MCF1}}$ polytope. We are to show that the point belongs to the polytope $\mathscr{P}_{\text{FlowOCT}}$. First, it is clear that $(\hat{p}, \hat{s}, \hat{b}, \hat{w}, \hat{q}, \hat{z}) \in \mathscr{P}_{\text{BASE}}$; so, it satisfies constraint (2a).

It suffices to show that $(\hat{p}, \hat{s}, \hat{b}, \hat{w}, \hat{q}, \hat{z})$ satisfies constraints (2b)-(2d). For every datapoint $i \in I$, we have

$$\hat{z}^i_{1,\ell(1)} + \hat{z}^i_{1,r(1)} + \hat{s}^i_1 = \hat{q}^i_t \leq 1.$$

Here, the equality holds by constraints (4d). The inequality holds because $\hat{q} \in [0,1]^{|I| \times |V'|}$. Hence, point $(\hat{p}, \hat{s}, \hat{b}, \hat{w}, \hat{q}, \hat{z})$ satisfies constraints (2b). Because $(\hat{p}, \hat{s}, \hat{b}, \hat{w}, \hat{z}, \hat{q})$ satisfies constraints (4c), the point satisfies constraints (2c). Finally for every datapoint $i \in I$ and every vertex $v \in B$, we have

$$\hat{z}^i_{v,\ell(v)} \leq \hat{q}^i_{\ell(v)} \leq \sum_{f \in F : x_{fi} = 0} \hat{b}_{vf}, \tag{17a}$$

$$\hat{z}^i_{v,r(v)} \leq \hat{q}^i_{r(v)} \leq \sum_{f \in F : x_{fi} = 1} \hat{b}_{vf}. \tag{17b}$$

Here, the first inequalities in (17a) and (17b) hold by constraints (4b). The second inequalities in (17a) and (17b) hold by constraints (4e). Hence, point $(\hat{p}, \hat{s}, \hat{b}, \hat{w}, \hat{z}, \hat{q})$ satisfies constraints (2d). This concludes the proof. □

To show the correctness, it suffices to prove the following statements.

1. Let $p^* \in$ FlowOCT be a feasible solution of formulation (2). Then p^* represents a optimal binary classification tree.

2. Suppose that a point p^* represents an optimal binary classification tree. Then, $p^* \in$ CUT2.

First, we will prove statement 1.

Proof. Let $p^* \in$ FlowOCT where X is defined by binary variables b, w, p, s, q. We are to show p^* represents an optimal binary classification tree. Variables b, w, p are defined on a vertex set $B \subseteq V$ and $L \subseteq V$ where $|V| = 2^{h+1}$ and $h \geq 1$ is integral and represents the maximal depth of a vertex in an unassigned binary decision tree. Further $B \cap L = \emptyset$. On each vertex $v \in V$ the left and right children $l(v), r(v)$ are defined as the left and right children and $P_{1,v}$ as the set of non-root $1, v$ vertices on the unique $1, v$ path of a vertex $v \in V$. Further decision variables b and w are defined binary encoded feature set F and discrete class set K, respectively, of given dataset I. Constraints (1c) to (1e) impose a valid tree structure through the following. For each $p_v = 1$, then $\sum_{k \in K} w_{vk} = 1$ by (1c). Thus exactly one $k \in K$ was chosen for vertex v (exactly one class was assigned to vertex v). Further $\sum_{f \in F} b_{vf} = \sum_{u \in V(P_{1,v} \setminus \{v\})} p_u = 0$. This implies, at vertex v it is not assigned any branching features and $p_u = 0$ for all of the ancestors u of v. Given that $p_u = 0$ for each ancestor u of v then $\sum_{f \in F} b_{uf} = 1$ by (1d). This implies each ancestor u of v is assigned exactly one branching feature. Then, again by (1c), $\sum_{k \in K} w_{uk} = 0$ so each ancestor u of v is not assigned any class. Then if we look at the children c of v where $p_v = 1$ by (1d) $p_c = \sum_{f \in F} b_{cf} = 0$. This implies all children c of v are not assigned a class or branching feature, and therefore pruned. If $v \in L$ then $\sum_{f \in F} b_{vf} = 0$ by (1e). This implies no leaf node is assigned to a branching feature and also assigned exactly one class. The above statements have shown that decision variables p implies that a vertex $v \in V$ is assigned exactly one class and all of its ancestors were assigned a single branching feature and that no leaf node can be assigned a branching feature. For each $b_{vf} = 1$, then at v, $\sum_{g\in F\setminus\{f\}} b_{vg} = p_v = \sum_{u\in V(P_{1,v}\setminus\{v\})} p_u = 0$ by (1d). Then for each ancestor u of v, $\sum_{f\in F} b_{uf} = 1$ by (1d). Thus, if a vertex v is assigned branching feature $f \in F$ then it is not pruned and each of its ancestors was assigned exactly one branching feature. Then if we look at the children of c of v then either $\sum_{f\in F} b_{cf} = 1$ or $p_c = 1$ by (1d), implying the children of branching vertex v are either branching or assigned a class. For each $w_{vk} = 1$ then $p_v = 1$ by (1c) and follows the implications previously defined. For each $s^i_v = 1$ then by (1f) $w_{vk} = 1$ where $k = y^i$. This implies a datapoint classified at vertex v is only correct if the vertex v is assigned it's given class k. Then by constraints (2b) and (2c) the datapoint flows $z^i_a = 1$ for $a \in E(G[V(P_{1,v})])$ along the $1, v$ path are connected and all other flows $z^i_{uv} \in E[G] = 0$. Thus a correctly classified datapoint found a successful $1, t$ flow in the tree. Then by (2d) the connected flows z^i_a along arcs $a \in G$ are branched such that the flow $z^i_{v,l(v)}$ ($z^i_{v,r(v)}$) along the arc connecting assigned branching vertex v to its left (right) child $l(v)$ ($r(v)$) meant vertex v was assigned some feature $f \in F$ such that $x^i_f = 0$ (1). This implies connected flows z^i_a of a datapoint $i \in I$ are such that branching is defined by the datapoint's feature set binary values. Lastly point p^* is an optimal classification tree because it satisfies (1a), and the maximal number of training datapoints were correctly classified by the tree. This complete the proof. □

Now we prove statement 2.

Proof. Let p^* be a point that represents an optimal binary classification tree trained by a binary encoded, discrete class dataset. Given that p^* is a binary tree starting at the root find the deepest classification vertex at depth h. We will assume the tree is of depth h. Enumerate nodes starting from the root of the tree until $2^{h+1} - 1$ ensuring the left, right children of vertex v labeled n are labeled $2n, 2n+1$, respectively. Further, if a vertex is the child of a classification vertex, assume it still exists and is pruned, rather than label only the existing vertices on the tree. Denote leaf set L as vertices whose label is in $[2^h - 1, 2^{h+1} - 1]$, all other vertices as branching set B, and $V = B \cup L$. For each classification vertex labeled v with class $k \in K$ denote binary variables $w_{vk} = p_v = 1$ and binary variables $w_{vk'} = b_{vf} = 0$ for every feature $f \in F$ where F is the set of all training features and $k' \in K \setminus \{k\}$, where K is the set of all training classes. Similarly for every vertex labeled v with branching feature f we denote binary variable $b_{vf} = 1$ and $w_{vk} = b_{vf'} = p_v = 0$ for every class $k \in K$ and $f' \in F \setminus \{f\}$. For every pruned vertex labeled v we denote binary variable $p_v = 0$ and $w_{vk} = b_{vf} = 0$ for every feature $f \in F$ and $k \in K$. Since p^* represents an optimal classification tree, for every datapoint i in training set I starting at the root with label 1 we denote binary variable $q^i_1 = 1$ then branch left right using the x^i_f binary value given to the feature of the branching vertex. From there we denote binary variables $q^i_v = 1$ for every vertex v visited by i and all other vertices $v' \in V \setminus \{v\}$ not visited have binary variables $q^i_{v'} = 0$. When datapoint i reaches a classification vertex, if it is correctly assigned we denote binary variable $s^i_v = 1$, else $s^i_v = 0$ and denote binary variables $s^i_{v'} = 0$ for every $v' \in V \setminus \{v\}$. Now we will show $p^* = (b, w, p, s, q)$ is in CUT2. For every vertex $v \in V$ with $w_{vk} = 1$ for some $k \in K$, since $w_{vk'} = b_{vf} = 0$ for every feature $k' \in K \setminus \{k\}$ and $f \in F$ (and $p_v = 1$), vertex v satisfies constraints (1c) and (1d). We assume p^* is a classification tree, thus for every vertex $v \in L$, v must be assigned some class $k \in K$ thus $w_{vk} = 1$ and we have the above. Further all such $v \in L$ satisfy constraints (1e). For every vertex $v \in V$ with $b_{vf} = 1$ for some $f \in F$, since $b_{vf'} = w_{vk} = p_v = 0$ for every $f' \in F \setminus \{f\}$ and $k \in K$, every such v satisfies constraints (1c) and (1d). There are no such $v \in L$ with $b_{vf} = 1$ for any $f \in F$ since p^* is a classification tree and thus constraints (1e) is also satisfied by every v with $b_{vf} = 1$ for some $f \in F$. Then for each datapoint $i \in I$ there will exist a connected $1, v$ path to some classification vertex v, denoted P_v, using the feature set of x^i; $q^i_c = 1$ for every $c \in P_v$. Further, since binary variables b, w, p satisfy constraints (1c)- (1e) then $q^i_{l(v)} \leq \sum_{f\in F: x^i_f=0} b_{vf}$, $q^i_{r(v)} \leq \sum_{f\in F: x^i_f=1} b_{vf}$ will be true because $\sum_{f\in F:x^i_f=0(1)} b_{vf} \leq \sum_{f\in F} b_{vf} = 1$. Thus, constraints (11c) are satisfied for every $i \in I$ and $v \in V$. At classification vertex v assigned class k, w_{vk}, if i is correctly classified, then $s^i_v = w_{vk} = 1$. Further, $k = y^i$ and $s^i_{v'} = 0$ for every $v' \in V \setminus \{v\}$, so constraints (1f) are satisfied for every $i \in I$ and $v \in V$. Classification vertex v was reached through a connected path, thus $s^i_v \leq q^i_c$ for every $c \in P_v$ because $s^i_v = q^i_c = 1$ for v and every $c \in P_v$. Thus, constraints (11b) are satisfied. From our labeling of binary variables $s^i_{v'} = 0$ for every $v' \in V \setminus \{v\}$ which implies $\sum_{v\in V} s^i_v \leq 1$, thereby satisfying constraints (11d). Lastly, because we assume our point p^* to be an optimal classification tree, features $f \in F$ and classes $k \in K$ were assigned to maximize the number of correctly classified datapoints. In turn $\sum_{i\in I}\sum_{v\in V} s^i_v$ is maximal satisfying objective (1a). This completes the proof. □

5 Computational Enhancements

In this section, we propose valid inequalities and heuristic procedures to improve the computational performance of the MILO formulations.

5.1 Valid Inequalities

We now propose two sets of valid inequalities to improve the computational performance of formulations (4), (7), (10) and (11). We first partition the vertex set $V \setminus \{1\}$ to left children (LC) and right children (RC). For example in Figure 1, sets LC and RC are $\{2, 4, 6\}$ and $\{3, 5, 7\}$, respectively.

Proposition 1. *The following sets of inequalities are valid.*

$$q^i_v + \sum_{u\in A(v)} b_{u,f} \leq 1 \qquad \forall v \in \mathrm{LC},\ \forall f \in F : x^i_f = 1,\ \forall i \in I \tag{18a}$$

$$q^i_v + \sum_{u\in A(v)} b_{u,f} \leq 1 \qquad \forall v \in \mathrm{RC},\ \forall f \in F : x^i_f = 0,\ \forall i \in I. \tag{18b}$$

Proof. Let $(\hat{p}, \hat{s}, \hat{b}, \hat{w}, \hat{q})$ be an integer point that represents a binary classification tree.

For every vertex $v \in V$, if $1 \equiv v \pmod 2$ then $q^i_v + b_{a(v),f} \leq 1$ can be rewritten as $q^i_{r(v')} + b_{v'f} \leq 1$, where $v' = a(v)$. Then the following statements are true at any vertex $v' \in V$ for all $i \in I$ and $f \in F : x^i_f \not\equiv v \pmod 2$,

$$\begin{aligned}
q^i_{r(v')} + b_{v'f} &\leq \sum_{f\in F: x^i_f=1} b_{v'f} + b_{v'f} \\
&\leq \sum_{f\in F: x^i_f=1} b_{v'f} + \sum_{f\in F: x^i_f=0} b_{v'f} \\
&= \sum_{f\in F} b_{v'f} \\
&\leq \sum_{f\in F} b_{v'f} + \sum_{u\in P_{1,v'}} p_u \\
&\leq 1
\end{aligned}$$

The first inequality holds by constraints (1d). The second inequality holds by $b \in \{0,1\}^{|V|\times|F|}$. The equality holds by the fact F is binary encoded. The last two inequalities holds by $p \in \{0,1\}^{|V|}$ and constraints (1d), respectively. Results are analogous when $0 \equiv v \pmod 2$ resulting in $q^i_{l(v')} + b_{v'f} \leq 1$. This completes the proof. □

The next sets of valid inequalities we consider are those valid to redefine the convex hull where solutions contain a predefined tree structure. One main goal of using binary decision trees is interpretability; at times, certain tree structures may be wanted to yield this interpretability. Traditional heuristic methods cannot guarantee a solution will contain such a structure with out post-processing methods. One main advantage of using MILO formulations for binary decision trees is the flexibility of constraints that can imply structure. However, this redefines the convex hull to be different from the ones described in MCF1 through CUT1. Further, this new convex hull may not contain the equivalent set of optimal solutions as before. The following sets of valid inequalities are two examples that define the convex hulls which contains solutions to optimal binary decision trees with predefined structures. Proposition 2 states a feature $f \in F$ can only be used at most $k \in \mathbb{N} \cup \{0\}$ times in the decision tree. This is intended to limit the number of times a feature is used.

Proposition 2. *Let F be the set of features. Then the following inequalities are valid.*

$$\sum_{v\in B} b_{v,f} \leq k \qquad\qquad \forall f \in F, k \in \mathbb{N} \cup \{0\}$$

Proof. If $|B| \leq k$ then the inequalities hold trivially because,

$$\sum_{v\in B} b_{vf} \leq \sum_{v\in B}\sum_{f\in F} b_{vf} \leq |B| \leq k.$$

The first and second inequalities hold by $b \in \{0,1\}^{|V|\times|F|}$. Assume without loss of generality that $k < |B|$. □

We also consider the original feature set of I since our dataset $I = \{x^i, y^i\}_{i\in\mathcal{I}}$ where $x^i \in \{0,1\}^{|F|}$ has encoded features. Let F be the encoded set used in models MCF1 through CUT2. Define $\mathcal{F} \in F$ as a super feature from the original feature set such that each $f \in \mathcal{F}$ are encoded from $\mathcal{F}$. Consider the following example. Let $[f_a, \ldots, f_k] \in F$ be features with the same 'super feature' $\mathcal{F}_s \in F$. For some vertex $v \in B$ let $b_{v,f_c} = 1$, for some $c \in [f_a, \ldots, f_k]$, then by Proposition 3 $b_{n,f_g} = 0$ for all $g \in \mathcal{F}_s \setminus \{c\}$ for left child of v. The intended purpose of 'super features' is to eliminate repeated use of related features in paths to classification vertices of the assigned decision tree in attempt to increase the interpretability of the final decision tree. An example of such a case is a decision tree used in political voting. Rather than having many features in a path all be related to the same political 'hot-topic', the 'super features' inequalities would build a tree that asks about each topic to a classification vertex and a more useful tree in understanding the voting habits of candidates.

Proposition 3. *Let $\mathcal{F}$ be a set of super features. Then, the following inequalities are valid.*

$$b_{v,f} + \sum_{u\in R(v)}\sum_{g\in F\setminus\{f\}} b_{u,g} \leq 1 \qquad\qquad \forall v \in V,\ \forall f \in \mathcal{F}. \tag{19}$$

Proof. □

For every level ℓ of the input tree, let $\mathrm{LS}(\ell)$ and $\mathrm{RS}(\ell)$ be the set of branching vertices at level ℓ on the left and right sides of the input tree, respectively. For example in Figure 1, sets $\mathrm{LS}(2)$ and $\mathrm{RS}(2)$ are $\{4, 5\}$ and $\{6, 7\}$, respectively.

Proposition 4. *At every level ℓ of the input tree, the following inequalities are super valid.*

$$\sum_{f \in F} b_{v,f} \leq \sum_{u \in \mathrm{RS}(\ell)} \sum_{g \in F} b_{u,g} \qquad \forall v \in \mathrm{LS}(\ell). \tag{20}$$

Proof. □

5.2 Our Heuristic

The heuristic methods such as CHAID, THAID, and CART employ probability distributions to build their respective decision trees. We employ binary decision variables to build our decision tree and instead elect to use our own heuristic methods for warm-starting our models. There are two different random methods we consider. The first assigns random features and classes to trees attempting to find feasible paths to a classification node. We fix the root node to a random feature. We then successively pick random unassigned vertices, assign them to random classes, assign all of its ancestors to random features and prune all its children. Each assignment of a class node builds a feasible path from the root to itself and cuts off all emanating paths, thus constructing a valid binary tree. Further each node in the tree is assigned exactly a single class, single branching feature, or pruned.

Algorithm 1 Random Path Tree

```
Require: $G_h, F, K$
Ensure: Valid Binary Decision Tree
nodelist ← vertices of $G_h$
Assign random feature $f \in F$ to root node $= 0$
nodelist ←nodelist−$\{0\}$
while nodelist not empty do
    pick random node $v \in$ nodelist
    assign random class $k \in K$ to $v$
    for node $n \in P_v$ do
        if $n \in$ nodelist then
            assign random feature $f \in F$ to $n$
            nodelist ← nodelist − $\{n\}$
        end if
    end for
    for node $p$ in CHILD[$v$] do
        assign node $p$ as pruned
        nodelist ← nodelist − $\{p\}$
    end for
    nodelist ← nodelist − $\{v\}$
end while
```

The second type of random tree we consider builds the tree through each level of the tree. We fix the root node to a random feature. Starting at depth $d = 1$, we assign a random feature or class to each node in the level. Iterate d and repeat this process with the following rules. If a node's ancestor is branching it can be assigned a random feature or class. If a node's ancestor is a class then it must be pruned. If a node is a leaf node then it must be assigned a class. This builds a tree such that each each node is assigned exactly one class, feature or pruned. All children of class vertices are pruned and all ancestors of class vertices are branching vertices.

Algorithm 2 Random Level Tree

```
Require: $G_h, F, K$
Ensure: Valid Binary Decision Tree
nodelist ← vertices of $G_h$
Assign random feature $f \in F$ to root node = 0
for $h \leftarrow d = 1$ do
    for vertices $v$ in depth $d$ do
        if direct ancestor of $v$ is branching node then
            Assign random feature $f \in F$ or class $k \in K$ to $v$
        else if direct ancestor of $v$ is class or pruned then
            assign node $v$ as pruned
        end if
    end for
end for
```

On the surface these two randomly assigned trees seem similar however they are fundamentally different. The first method fixes a feasible path a classification node independent of other feasible paths and builds the tree from the bottom up, whereas the second randomly assigns vertices from a top down approach, similar to traditional heuristic approaches.

6 Computational Experiments

We run all of our experiments on a Intel(R) Core(TM) i7-9800X CPU (3.8Ghz, 19.25MB, 165W) using 1 core, and 16GB RAM. Our code is written in Python 3.6 or above. The MILO formulations are solved using Gurobi 9.5. We consider a time limit of 3600 seconds for all of our experiments. All datasets are publicly available at `http://archive.ics.uci.edu/ml/index.php` and code is available at `https://github.com/brandalston/OBCT`.

We use 13 datasets from the UCI ML repository. For each dataset we create 5 random 75-25% train-test split. We do not consider the second objective of minimizing the number of branching vertices, except for the case of the pareto frontier. For height $h \in \{2, 3, 4, 5\}$ we train a decision tree using each model and report various metrics including out-of-sample accuracy, in-sample optimality gap, solution-time, and in-sample accuracy. No fixing procedures or warm starts are used unless otherwise indicated. FlowOCT and BendersOCT instances have a λ value of 0.

MILO Performance Summary. Table 2 summarizes the MILO models optimization performance. Listed are FlowOCT, MCF1, MCF2, CUT1, CUT2 and the Bender's decomposition of FlowOCT, BendersOCT. Comparing flow formulations FlowOCT, MCF1, and MCF2 we see models MCF1 and MCF2 outperform FlowOCT in 34 out of 52 possible instances in either solution time or optimality gap. Further, if we extend the comparisons of FlowOCT to cut-based models CUT1 and CUT2 we see an increase to 43 of 52 possible instances where our models outperform FlowOCT. The arithmetic average speedup (ratio of our model to FlowOCT) of our models over FlowOCT is 2.80X speedup where as FlowOCT over ours is only 1.28X. However, when we compare the Ben-

Dataset	$\lvert I\rvert$	$\lvert F\rvert$	$\lvert K\rvert$
soybean-small	47	45	4
monk3	122	15	2
monk1	124	15	2
hayes-roth	132	15	3
monk2	169	15	2
house-votes-84	232	16	2
spect	267	22	2
breast-cancer	277	38	2
balance-scale	625	20	3
tic-tac-toe	958	27	2
car_evaluation	1728	20	4
kr-vs-kp	3196	38	2
fico_binary	10459	19	2

Table 1: Datasets with size $|I|$, number of encoded features $|F|$, and classes $|K|$.

der's Decomposition we see that models our models perform best in 25 out of 52 possible times. Our models have an average speedup of 1.58X speedup over BendersOCT whereas Bender's has a 2.47X speedup. Complete tables for time and in-sample optimality gap are listed in Tables 6 and 7, respectively. Lastly we observe that for larger trees, ($h = 4, 5$), or larger datasets, ($|I| > 500$), we see that models CUT1 and CUT2 perform better highlighting the strength of separation constraints that can be added on the fly.

Tables 4 and 5 highlight introducing the separation constraints of models CUT1 and CUT2 lead to better solution time or in-sample optimality gap. However, as noted by Fischetti et al. (2017) adding too many fractional cuts may also lead to slower performance.

Machine Learning Performance Summary. Table 3 summarizes the MILO models out-of-sample accuracy performance. Listed are FlowOCT, MCF1, MCF2, CUT1, CUT2 and the Bender's decomposition of FlowOCT, BendersOCT. We see that FlowOCT outperforms our models in only 9 of 52 possible instances, with the maximum being a 8% difference and on average less than 2% variation. Extending comparisons to BendersOCT there are 6 additional instances in which FlowOCT or BendersOCT outperforms our models. Further the maximum difference increases to 20% and on average to roughly 4% due to BendersOCT finding a solution within the time limit and our models reporting gap. Additional results for in-sample accuracy are listed in Table 8.

Figure 6 shows the strength of taking a biobjective approach as providing an upper bound on the number of branching vertices, thereby reducing the size of the problem, results in comparable or better out-of-sample accuracy performance. For the same number of maximum branching vertices our models, on average, have better out-of-sample accuracy than FlowOCT. Further we see that as you increase the tree size models $\mathrm{MCF2}, \mathrm{CUT1}.$ and CUT2 perform well, as expected.

Table 2: Average solution time or in-sample optimality gap, (in parentheses), if the time limit (3600s) was reached. Best in **bold** (* if BendersOCT performed best).

Dataset	D_h	FlowOCT	MCF1	MCF2	CUT1	CUT2	BendersOCT
soybean-small	2	0.08	0.10	0.06	**0.04**	0.07	0.20
	3	0.16	0.08	0.12	**0.08**	**0.08**	0.20
	4	0.33	0.28	0.27	**0.22**	**0.22**	0.36
	5	0.65	0.52	0.66	**0.49**	0.50	0.49*
monk3	2	**0.51**	0.94	1.02	0.56	0.56	0.23*
	3	91.98	105.15	92.57	**66.25**	91.99	15.70*
	4	2210.76	2186.47	2197.56	**1733.84**	1924.94	2381.33
	5	187.22	219.00	142.03	**79.16**	136.59	42.62*
monk1	2	0.74	1.68	1.12	**0.67**	0.73	0.48*
	3	32.65	35.65	41.05	**25.44**	25.73	4.33*
	4	15.89	22.63	**8.92**	9.87	13.01	2.13*
	5	**8.46**	16.60	13.78	12.06	12.52	1.51*
hayes-roth	2	0.69	2.69	1.01	**0.62**	**0.62**	0.83
	3	**13.86**	59.02	31.5	40.76	26.83	7.86*
	4	2430.02	1819.18	1415.03	1512.81	**1270.11**	2422.33
	5	(8.86)	(9.53)	**(8.66)**	(9.12)	(9.82)	(9.06)
monk2	2	6.76	5.83	3.40	**1.87**	1.92	3.12
	3	2935.01	1898.05	1586.15	1014.42	**804.51**	1205.76
	4	**(17.23)**	(18.36)	(18.96)	(18.05)	(18.6)	(12.95)*
	5	(11.71)	(9.20)	(10.97)	(9.2)	**(8.45)**	(10.38)
house-votes-84	2	**0.76**	2.97	1.97	1.06	1.07	0.48*
	3	200.91	167.30	**127.05**	173.64	133.14	80.30*
	4	2882.22	2169.07	**1087.32**	1144.08	1209.22	2882.02
	5	471.55	**62.14**	76.84	115.90	116.2	555.25
spect	2	9.97	15.90	18.78	2.84	**2.63**	4.14
	3	2790.05	1760.49	1736.55	**942.73**	1385.65	2162.11
	4	(35.36)	(4.70)	(4.48)	(4.59)	**(4.36)**	(4.21)*
	5	(55.49)	(4.61)	(6.01)	(5.30)	**(4.17)**	(4.87)
breast-cancer	2	23.69	45.94	7.79	7.68	**7.43**	17.19
	3	(73.58)	(17.99)	(16.45)	**(12.98)**	(13.91)	(12.07)*
	4	(16.05)	(17.11)	(15.27)	(15.52)	**(15.02)**	(15.67)
	5	(11.68)	(10.37)	(11.69)	(10.15)	**(9.90)**	(11.34)
balance-scale	2	10.51	21.68	17.16	7.29	**7.23**	6.60*
	3	1863.60	1651.92	1384.10	1235.88	**1232.90**	895.25*
	4	(97.26)	(21.67)	(21.11)	**(18.36)**	(19.26)	(9.88)*
	5	(100)	(21.44)	**(21.12)**	(21.32)	(21.56)	(21.63)
tic-tac-toe	2	490.04	327.18	96.10	**93.31**	104.08	139.20
	3	(100)	(30.84)	(29.01)	**(28.45)**	(29.36)	(18.57)*
	4	(100)	(22.21)	(19.52)	**(19.20)**	(19.75)	(19.75)
	5	(100)	(13.04)	**(13.00)**	(16.27)	(14.81)	(14.24)
car_evaluation	2	83.46	111.03	86.05	38.99	**35.97**	32.08*
	3	(78.89)	(23.29)	(18.32)	(18.26)	**(17.5)**	(5.07)*
	4	(100)	(28.27)	**(21.44)**	(22.73)	(21.9)	(20.65)*
	5	(83.71)	(40.15)	**(32.01)**	(39.44)	(34.43)	(18.03)*
kr-vs-kp	2	**609.73**	3244.23	1049.04	1209.25	1220.4	420.01*
	3	(19.97)	(42.4)	(28.01)	**(15.56)**	(18.52)	(8.56)*
	4	**(27.98)**	(91.94)	(57.25)	(45.89)	(42.85)	(12.6)
	5	**(25.56)**	(57.43)	(45.86)	(34.53)	(43.39)	(5.36)*
fico_binary	2	944.53	(25.05)	**858.20**	955.18	1408.68	2414.30
	3	(98.31)	**(39.50)**	(39.51)	(41.47)	(41.43)	(39.95)
	4	(99.57)	(> 100)	**(60.95)**	(> 100)	(> 100)	(41.48)*
	5	**(40.71)**	(> 100)	(> 100)	(> 100)	(> 100)	(40.64)*

Table 3: Average out-of-sample accuracy (%). Best in **bold** (* if BendersOCT performed best).

Dataset	D_h	FlowOCT	MCF1	MCF2	CUT1	CUT2	BendersOCT
soybean-small	2	**100**	98.33	**100**	**100**	**100**	98.33
	3	**98.33**	**98.33**	96.67	**98.33**	**98.33**	95.00
	4	**98.33**	96.67	**98.33**	**98.33**	96.67	95.00
	5	98.33	**100**	96.67	96.67	**100**	90.00
monk3	2	**94.19**	**94.19**	**94.19**	**94.19**	**94.19**	94.19*
	3	**92.90**	92.26	92.26	**92.90**	**92.90**	92.26
	4	**92.26**	89.03	89.68	90.32	**92.26**	90.97
	5	88.39	89.03	88.39	**91.61**	87.74	87.10
monk1	2	**74.84**	**74.84**	**74.84**	**74.84**	**74.84**	74.84*
	3	84.52	**87.10**	85.81	86.45	**87.10**	87.74*
	4	98.06	**100**	**100**	**100**	**100**	100*
	5	**100**	98.71	**100**	**100**	**100**	100*
hayes-roth	2	**44.24**	**44.24**	**44.24**	**44.24**	**44.24**	44.24*
	3	**56.97**	56.36	55.76	56.36	56.36	56.36
	4	**67.27**	64.85	64.24	**67.27**	**67.27**	65.45
	5	70.91	64.24	70.91	73.33	**75.76**	68.48
monk2	2	54.42	55.81	55.81	55.81	**56.28**	56.28*
	3	68.84	64.65	67.44	69.3	**69.77**	66.05
	4	56.74	62.33	59.53	**65.58**	65.12	66.05*
	5	58.14	**65.58**	58.14	64.65	64.65	59.07
house-votes-84	2	**95.52**	**95.52**	**95.52**	**95.52**	**95.52**	95.52*
	3	**93.10**	92.76	92.41	**93.10**	92.76	92.41
	4	95.86	94.83	96.21	**96.55**	**96.55**	95.17
	5	93.79	93.79	95.17	**95.86**	95.52	93.79
spect	2	**77.01**	**77.01**	**77.01**	**77.01**	**77.01**	77.01*
	3	77.31	**77.61**	77.31	**77.61**	**77.61**	77.31
	4	**80.30**	77.61	78.81	79.70	80.00	79.40
	5	**81.19**	77.31	80.00	79.40	80.00	77.91
breast-cancer	2	**74.00**	72.00	73.14	72.57	72.57	72.00
	3	71.43	69.71	69.71	69.71	**72.00**	67.43
	4	68.57	69.14	69.43	**70.71**	**71.07**	68.86
	5	70.86	70.00	69.43	**71.43**	**71.43**	69.43
balance-scale	2	**64.20**	**64.20**	**64.20**	**64.20**	**64.20**	64.20*
	3	**67.64**	67.26	**67.64**	**67.64**	**67.64**	67.64*
	4	70.57	70.83	71.72	**72.48**	71.46	69.30
	5	73.12	72.48	71.34	**73.25**	**73.25**	69.30
tic-tac-toe	2	**68.58**	**68.58**	**68.58**	68.50	**68.58**	68.33
	3	73.17	74.08	**74.33**	74.08	**74.33**	73.75
	4	80.42	76.17	**80.92**	80.50	80.50	81.33*
	5	81.17	82.00	**84.92**	81.75	84.25	84.75
car_evaluation	2	**77.78**	**77.78**	**77.78**	**77.78**	**77.78**	77.78*
	3	78.43	78.19	78.19	**78.94**	**78.94**	78.52
	4	**81.85**	76.16	80.51	80.28	80.23	80.79
	5	**83.24**	71.67	75.60	71.25	75.00	82.41
kr-vs-kp	2	**86.78**	**86.78**	**86.78**	**86.78**	**86.78**	**86.78**
	3	83.65	72.19	78.87	**86.53**	83.88	92.17*
	4	**78.5**	52.54	64.86	67.98	69.56	89.04*
	5	**79.67**	64.26	67.93	74.64	69.31	94.72*
fico_binary	2	70.73	**70.80**	70.73	70.73	70.73	70.73
	3	**70.97**	70.88	**70.97**	69.76	69.77	70.65
	4	**70.20**	53.06	61.93	53.06	53.06	70.11
	5	68.82	53.06	61.93	53.06	53.06	70.02*

Figure 6 summarizes our experiments with all MILO formulations and *both* objective functions (1a) and (1b). In other words, we maximize objective function (1a) while objective function (1b) is considered as a constraint on the maximum number of branching decisions. For different values of the maximum number of branching decisions (on the x axis) and different values of accuracy percentage (on the y axis), and trend lines. In this figure, dominating and dominated points are illustrated in opaque and transparent colors, respectively. The solution for k braching vertices was a warm-start for $k+1$ branching vertices for all models.

Observe that MCF2, CUT1, CUT2 are the superior models when the maximum number of branching vertices is large. MCF2 generally performs well due to it's strong LO relaxation and the fact the entire model is fed upfront, compared to lazy constraints for CUT1 and CUT2. Further one can see our models achieve higher out of sample accuracy compared to FlowOCT with the same or fewer branching vertices (particularly in monk3 and house-votes-84). Also the Pareto frontiers confirm in MILO formulations of optimal binary decision trees that larger trees do not always result in better out-of-sample performance.

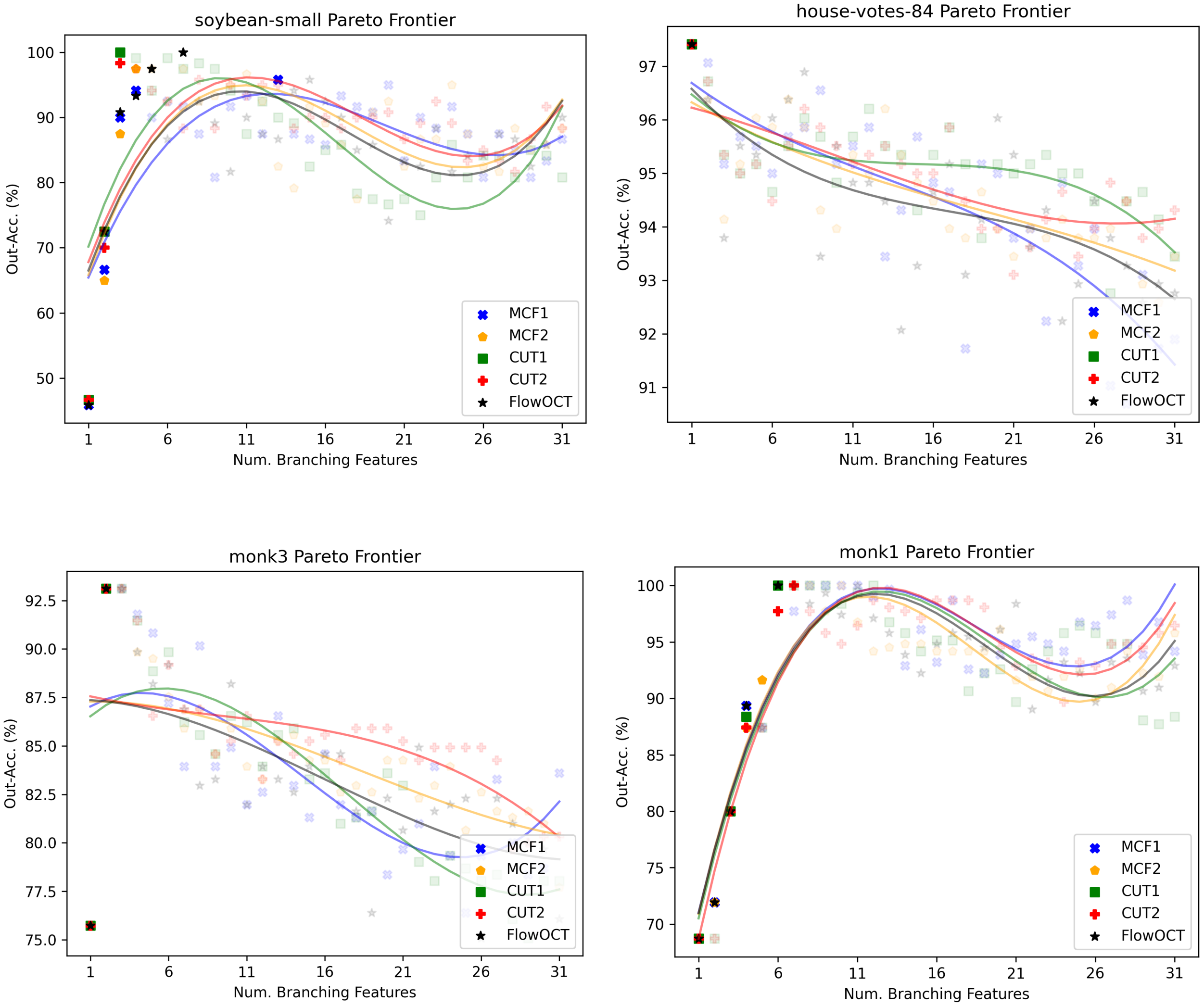
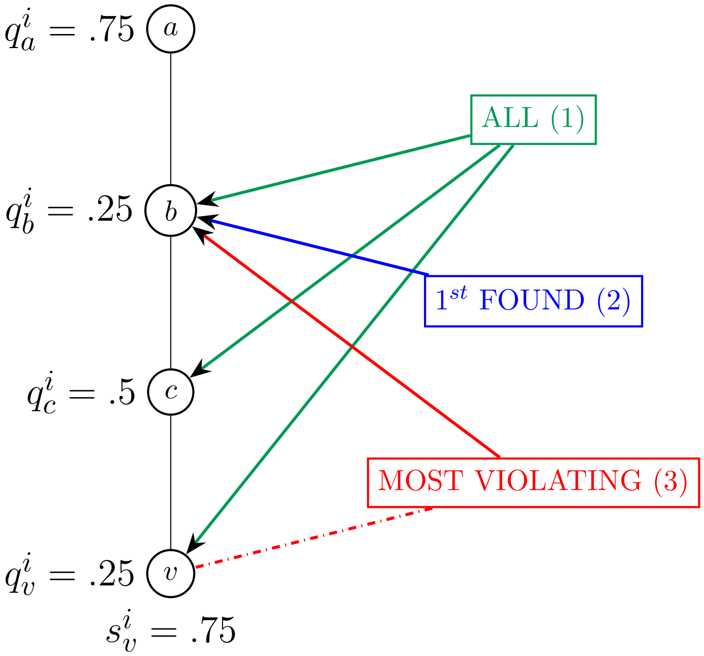


Figure 6: Pareto Frontiers for `monk1, monk3, house-votes-84` and `soybean-small` datasets. Warm start model for $k+1$ max features using model k max features solution.

For cut-based formulations CUT1 and CUT2, we consider a number of variations on implementing separation constraints (10b) and (11b). We implement integral separation constraints using Gurobi lazy parameter = 3, constraints that cut off the relaxation solution at the root node are also pulled in. We also introduce all integral separation constraints up front. Then we consider the separation constraints at fractional points in the branch and bound tree with three variations. The first type adds all violating cuts for a datapoint in the $1, v$ path of a terminal vertex v. The second type adds the first found violating cut in the $1, v$ path. The third type adds the most violating cut in the $1, v$ path (if more than one exists, only the cut closest to the root node of the DT is added). These user cuts are added in conjunction with integral constraints that cut-off relaxations at the root node of the branch and bound tree and use an $\epsilon = 10^{-4}$. We consider three variations on violating cuts due to Fischetti et al. (2017) who noted adding too many fractional cuts may slow down solution time. As a result we consider a heavy set of user cuts (all violating cuts) and two light sets of user cuts (first found and most violating). An illustration of the 3 types of cuts are shown in figure 7.

Figure 7: Let a, b, c, and v be vertices selected on the 1,v path of datapoint i. Consider fractional point with s_v^i and q_c^i for $c \in P_v$ be as defined. The 3 types of fractional user cuts are in solid colored arrows (the dotted line is a type 3 most violating cut considered but not added).



Tables 4 and 5 compare adding all separation constraints upfront vs introducing them at the root node of the branch and bound tree or with the fractional cuts outlined above. The INT-All column contains the solution time or gap of the CUT model with all integral separation constraints introduced upfront. The other columns represent solution time or gap ratios to INT-All. Decreases (increases) in solution time or gap are indicated in **bold** (light). Both tables highlight adding our separation constraints on the fly does help improve solution time, both with and without the three types of fractional cuts, over adding all separation constraints upfront. However, it is still feasible to introduce all separation constraints upfront with large datasets; and at times better to do so.

Table 4: CUT1 separation constraints comparisons. ALL represents integral constraints introduced upfront and their solution time (gap). Decreases (increases) in solution time or gap ratio are indicated in **bold** (light) for other columns. LAZY represents integral constraints introduced to cut off the relaxation solution at the root node of the branch and bound tree.

Dataset	D_h	INT-All	INT-Lazy	FRAC-(1)	FRAC-(2)	FRAC-(3)
soybean-small	2	0.04	1.83	2.23	2.21	2.23
	3	0.08	2.56	1.99	3.60	3.67
	4	0.22	2.11	2.73	2.34	2.35
	5	0.49	2.47	2.67	2.42	2.44
monk3	2	1.52	**0.37**	**0.38**	**0.38**	**0.38**
	3	66.25	1.22	1.13	1.86	1.85
	4	2222.62	**0.78**	**0.91**	**0.89**	**0.89**
	5	253.98	**0.48**	**0.31**	**0.50**	**0.51**
monk1	2	1.66	**0.45**	**0.41**	**0.41**	**0.41**
	3	34.63	**0.73**	**0.91**	1.00	1.00
	4	9.87	1.29	1.24	1.90	1.90
	5	13.42	0.90	**1.30**	0.98	0.98
hayes-roth	2	1.28	**0.48**	**0.50**	**0.50**	**0.50**
	3	42.02	1.09	1.07	**0.97**	**0.97**
	4	1512.81	1.84	1.62	1.65	1.94
	5	(9.12)	1.10	1.02	1.13	1.16
monk2	2	3.64	**0.53**	**0.52**	**0.52**	**0.51**
	3	1014.42	1.24	1.42	1.11	1.11
	4	(18.05)	1.06	1.02	1.12	1.13
	5	(10.38)	**0.95**	**0.97**	**0.89**	**0.92**
house-votes-84	2	2.89	**0.37**	**0.37**	**0.37**	**0.37**
	3	173.64	1.46	1.54	1.67	1.66
	4	1625.82	1.07	**0.70**	**0.80**	**0.80**
	5	115.90	1.37	1.22	1.07	1.07
spect	2	13.42	**0.21**	**0.23**	**0.24**	**0.24**
	3	1670.76	**0.73**	**0.56**	**0.62**	**0.77**
	4	(4.59)	1.15	1.15	1.10	1.12
	5	(5.30)	1.15	1.02	1.06	1.06
breast-cancer	2	18.72	**0.75**	**0.41**	**0.41**	**0.41**
	3	(12.98)	1.22	1.17	1.23	1.23
	4	(15.52)	1.03	1.07	1.01	1.01
	5	(10.98)	**0.92**	1.04	1.01	1.01
balance-scale	2	19.31	**0.38**	**0.40**	**0.40**	**0.40**
	3	1235.88	1.87	1.62	1.61	1.61
	4	(18.36)	1.21	1.13	1.11	1.18
	5	(21.32)	1.03	1.02	1.00	1.01
tic-tac-toe	2	283.51	**0.37**	**0.33**	**0.33**	**0.33**
	3	(28.45)	1.05	1.03	1.05	1.05
	4	(20.33)	**0.97**	1.02	**0.94**	**0.94**
	5	(16.27)	1.16	1.36	1.17	1.11
car_evaluation	2	38.99	1.23	1.27	1.27	1.26
	3	(19.46)	**0.96**	1.10	**0.95**	**0.94**
	4	(22.73)	1.09	1.10	1.12	1.12
	5	(39.44)	3.98	1.13	1.16	1.91
kr-vs-kp	2	2850.18	**0.42**	**0.70**	**0.69**	**0.77**
	3	(39.55)	**0.40**	**0.39**	**0.42**	**0.42**
	4	(46.16)	1.00	1.00	**0.99**	**.99**
	5	(34.53)	1.38	25.95	26.39	1.83
fico_binary	2	955.18	3.77	3.58	3.73	3.71
	3	(41.47)	5.30	6.70	7.35	7.35
	4	(106.89)	3.68	3.69	3.68	3.68
	5	(106.94)	60.45	91.69	52.28	35.67

Table 5: CUT2 separation constraints comparisons. ALL represents integral constraints introduced upfront and their solution time (gap). Decreases (increases) in solution time or gap ratio are indicated in **bold** (light) for other columns. LAZY represents integral constraints introduced to cut off the relaxation solution at the root node of the branch and bound tree.

Dataset	D_h	INT-All	INT-Lazy	FRAC-(1)	FRAC-(2)	FRAC-(3)
soybean-small	2	0.07	2.01	1.03	1.03	1.05
	3	0.08	2.12	2.93	2.88	2.89
	4	0.22	2.48	2.72	2.23	2.21
	5	0.50	2.99	3.30	2.71	2.71
monk3	2	1.28	**0.44**	**0.46**	**0.45**	**0.46**
	3	114.73	**0.80**	**0.92**	**0.97**	**0.97**
	4	2186.42	**0.88**	1.02	**0.90**	**0.90**
	5	383.06	**0.36**	**0.44**	**0.36**	**0.37**
monk1	2	1.54	**0.47**	**0.48**	**0.47**	**0.48**
	3	30.84	**0.83**	1.23	1.00	1.00
	4	17.14	**0.80**	1.70	**0.76**	**0.76**
	5	20.47	**0.63**	**0.61**	1.07	1.07
hayes-roth	2	1.24	**0.50**	**0.53**	**0.53**	**0.53**
	3	26.83	1.70	1.66	1.72	1.71
	4	1270.11	2.12	2.29	1.63	1.48
	5	(9.82)	1.05	1.07	1.05	1.00
monk2	2	4.01	**0.48**	**0.49**	**0.49**	**0.49**
	3	804.51	1.58	2.03	1.71	1.71
	4	(19.74)	1.00	**0.96**	**0.95**	**0.94**
	5	(8.45)	1.25	1.27	1.37	1.40
house-votes-84	2	2.65	**0.40**	**0.41**	**0.40**	**0.41**
	3	133.14	1.97	1.78	1.74	1.75
	4	1861.33	**0.93**	**0.65**	**0.71**	**0.71**
	5	116.20	1.39	1.76	1.95	1.96
spect	2	12.71	**0.21**	**0.22**	**0.24**	**0.24**
	3	1531.80	**0.97**	1.18	**0.92**	**0.90**
	4	(4.36)	1.22	1.17	1.33	1.21
	5	(4.17)	1.35	1.38	1.49	1.30
breast-cancer	2	65.27	**0.22**	**0.11**	**0.11**	**0.11**
	3	(15.74)	1.07	**0.89**	**0.88**	**0.88**
	4	(15.02)	1.08	1.09	1.09	1.08
	5	(10.66)	**0.93**	1.06	**0.99**	**0.94**
balance-scale	2	16.52	**0.44**	**0.46**	**0.46**	**0.46**
	3	1232.90	1.87	1.78	1.81	1.81
	4	(20.62)	1.02	**0.93**	**0.99**	**0.99**
	5	(21.82)	**0.99**	**0.99**	1.00	1.02
tic-tac-toe	2	281.79	**0.37**	**0.38**	**0.38**	**0.37**
	3	(29.71)	1.02	**0.99**	**0.99**	**0.99**
	4	(20.40)	**0.97**	1.01	1.02	1.01
	5	(14.81)	1.25	1.52	1.25	1.16
car_evaluation	2	35.97	1.33	1.42	1.42	1.42
	3	(17.50)	1.06	1.20	1.12	1.12
	4	(21.90)	1.09	1.31	1.17	1.16
	5	(34.43)	3.45	2.67	3.80	2.59
kr-vs-kp	2	3048.5	**0.40**	**0.50**	**0.51**	**0.49**
	3	(48.67)	**0.38**	**0.38**	**0.55**	**0.57**
	4	(42.85)	1.08	1.08	1.07	1.06
	5	(43.39)	1.10	1.07	1.25	1.25
fico_binary	2	1408.68	2.41	2.40	2.41	2.40
	3	(41.43)	4.10	3.35	4.93	4.93
	4	(106.73)	3.46	3.36	3.62	3.49
	5	(106.86)	5.34	21.95	21.95	19.98

7 Conclusion

In this paper we propose four new MILO formulations that improve upon the recent flow-based formulation FlowOCT of Aghaei et al. (2021). These novel models improve on the LO relaxation of FlowOCT, shown theoretically and empirically through experimental testing. This improvement in solution time or in-sample optimality gap is observed in 43 out of 52 instances. Further, our models are able to outperform a tailored Bender's decomposition of FlowOCT in 25 of 52 instances. We are also able to fix decision variables and warm start models using the original encoded dataset to improve solution time. Lastly, we improve upon Gurobi solution times by adding user fractional cuts at the root node of the branch and bound tree.

A few important things we would like to note for the use of MILO formulations of optimal binary classification trees over heuristic or probabilistic based methods. First and most importantly is the increasing need of optimal solutions for problems today. Many current techniques may move the solution space away from the true space through reliance on probabilistic distributions that are not true for the training dataset. Further, solutions may be too complex for understanding and thereby a black-box model. Lastly, many machine learning techniques need very large training datasets that may not be readily available.

We have shown MILO formulations allow for decision trees with interpretable structures, and through our valid inequalities and pareto frontiers can be built with predetermined structures that remain optimal. Further MILO solutions have good out-of-sample performance with both small and large training datasets.

Bibliography

S. Aghaei, M. J. Azizi, and P. Vayanos. Stong optimal classification trees. *CoRR*, 2021.

G. Aglin, S. Nijssen, and P. Schaus. Learning optimal decision trees using caching branch-and-bound search. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3146–3153, 2020.

H. Albaqami, G. M. Hassan, A. Subasi, and A. Datta. Automatic detection of abnormal eeg signals using wavelet feature extraction and gradient boosting decision tree. *Biomedical Signal Processing and Control*, 70:102957, 2021.

R. Balestriero. Neural decision trees. *ArXiv*, abs/1702.07360, 2017.

B. Balk and K. Elder. Combining binary decision tree and geostatistical methods to estimate snow distribution in a mountain watershed. *Water Resources Research*, 36(1):13–26, 2000.

A. P. Barata and C. J. Veenman. Fair tree learning. *CoRR*, abs/2110.09295, 2021.

K. P. Bennett and J. A. Blue. Optimal decision trees. *Rensselaer Polytechnic Institute Math Report*, 214:24, 1996.

D. Bertsimas and J. Dunn. Optimal classification trees. *Machine Learning*, 106(7):1039–1082, 2017.

D. Bertsimas and R. Shioda. Classification and regression via integer optimization. *Operations Research*, 55(2):252–271, 2007.

C. Bessiere, E. Hebrard, and B. O'Sullivan. Minimising decision tree size as combinatorial optimisation. In I. P. Gent, editor, *Principles and Practice of Constraint Programming - CP 2009*, pages 173–187, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

R. Bixby. A brief history of linear and mixed-integer programming computation. *Documenta Mathematica*, pages 107–121, 2012.

R. Blanquero, E. Carrizosa, C. Molero-Río, and D. R. Morales. Optimal randomized classification trees. *Computers & Operations Research*, 132:105281, 2021.

L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees.* CRC press, 1984.

P. Charlot and V. Marimoutou. On the relationship between the prices of oil and the precious metals: Revisiting with a multivariate regime-switching decision tree. *Energy Economics*, 44: 456–467, 2014.

I. I. Cplex. V12. 1: User's manual for cplex. *International Business Machines Corporation*, 46(53): 157, 2009.

S. Dash, O. Günlük, and D. Wei. Boolean decision rules via column generation. *arXiv preprint arXiv:1805.09901*, 2018.

D. Delen, C. Kuzey, and A. Uyar. Measuring firm performance using financial ratios: A decision tree approach. *Expert Systems with Applications*, 40(10):3970–3983, 2013.

M. Firat, G. Crognier, A. F. Gabor, C. A. Hurkens, and Y. Zhang. Column generation based heuristic for learning classification trees. *Computers & Operations Research*, 116:104866, 2020.

M. Fischetti, M. Leitner, I. Ljubić, M. Luipersbeck, M. Monaci, M. Resch, D. Salvagnin, and M. Sinnl. Thinning out Steiner trees: a node-based model for uniform edge costs. *Mathematical Programming Computation*, 9(2):203–229, 2017.

L. R. Ford Jr. and D. R. Fulkerson. *Flows in networks.* Princeton university press, 1962.

C. Gini. Variabilità e mutabilità. *Journal of the Royal Statistical Society*, 76:326–327, 1912.

O. Günlük, J. Kalagnanam, M. Li, M. Menickelly, and K. Scheinberg. Optimal decision trees for categorical data via integer programming. *Journal of Global Optimization*, pages 1–28, 2021.

L. Gurobi Optimization. Gurobi optimizer reference manual, 2021. URL `http://www.gurobi.com`.

L. Hyafil and R. L. Rivest. Constructing optimal binary decision trees is np-complete. *Inf. Process. Lett.*, 5(1):15–17, 1976.

G. V. Kass. An exploratory technique for investigating large quantities of categorical data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 29(2):119–127, 1980.

P. Kontschieder, M. Fiterau, A. Criminisi, and S. R. Bulò. Deep neural decision forests. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1467–1475, 2015.

S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

A. Kumar, M. Hanmandlu, and H. Gupta. Fuzzy binary decision tree for biometric based personal authentication. *Neurocomputing*, 99:87–97, 2013.

A. Land and A. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497–520, 1960.

G.-H. Lee and T. S. Jaakkola. Oblique decision trees from derivatives of relu networks. In *International Conference on Learning Representations*, 2020.

Z. Li, L. Wang, L.-s. Huang, M. Zhang, X. Cai, F. Xu, F. Wu, H. Li, W. Huang, Q. Zhou, et al. Efficient management strategy of covid-19 patients based on cluster analysis and clinical decision tree classification. *Scientific reports*, 11(1):1–13, 2021.

W.-Y. Loh and Y.-S. Shih. Split selection methods for classification trees. *Statistica sinica*, pages 815–840, 1997.

J. F. Magee. *Decision trees for decision making.* Harvard Business Review, 1964.

R. Manogna and A. K. Mishra. Measuring financial performance of indian manufacturing firms: application of decision tree algorithms. *Measuring Business Excellence*, 2021.

D. Maturana, D. Mery, and Á. Soto. Face recognition with decision tree-based local binary patterns. In *Computer Vision – ACCV 2010*, pages 618–629. Springer Berlin Heidelberg, 2011.

M. Mehta, R. Agrawal, and J. Rissanen. Sliq: A fast scalable classifier for data mining. In *International conference on extending database technology*, pages 18–32. Springer, 1996.

B. M. Menze, Bjoern H.and Kelm, D. N. Splitthoff, U. Koethe, and F. A. Hamprecht. On oblique random forests. In D. Gunopulos, T. Hofmann, D. Malerba, and M. Vazirgiannis, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 453–469, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

S. K. Murthy. Automatic construction of decision trees from data: A multi-disciplinary survey. *Data Mining and Knowledge Discovery*, 2:345–389, 2004.

S. K. Murthy, S. Kasif, and S. L. Salzberg. A system for induction of oblique decision trees. *J. Artif. Intell. Res.*, 2:1–32, 1994.

N. Narodytska, A. Ignatiev, F. Pereira, and J. Marques-Silva. Learning optimal decision trees with sat. In *Proceedings - 27th International Joint Conference on Artificial Intelligence, IJCAI 2018*, pages 1362–1368, United States of America, 2018. Association for the Advancement of Artificial Intelligence (AAAI).

C. Orsenigo and C. Vercellis. Multivariate classification trees based on minimum features discrete support vector machines. *Ima Journal of Management Mathematics*, 14:221–234, 2003.

J. R. Quinlan. Induction of decision trees. *Mach. Learn.*, 1(1):81–106, 1986.

J. R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.

J. Shafer, R. Agrawal, and M. Mehta. Sprint: A scalable parallel classifier for data mining. In *Vldb*, volume 96, pages 544–555. Citeseer, 1996.

C. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 1948.

A. Valdivia, J. Sánchez-Monedero, and J. Casillas. How fair can we go in machine learning? assessing the boundaries of accuracy and fairness. *International Journal of Intelligent Systems*, 36(4):1619–1643, Jan 2021. ISSN 1098-111X.

S. Verwer and Y. Zhang. Learning optimal classification trees using a binary linear program formulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1625–1632, 2019.

J. Wang, R. Fujimaki, and Y. Motohashi. Trading interpretability for accuracy: Oblique treed sparse additive models. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015.

D. C. Wickramarachchi, B. L. Robertson, M. Reale, C. J. Price, and J. Brown. Hhcart: An oblique decision tree. *Comput. Stat. Data Anal.*, 96:12–23, 2016.

Y. Yang, I. G. Morillo, and T. M. Hospedales. Deep neural decision trees. *CoRR*, abs/1806.06988, 2018.

S. H. Yoo, H. Geng, T. L. Chiu, S. K. Yu, D. C. Cho, J. Heo, M. S. Choi, I. H. Choi, C. Cung Van, N. V. Nhung, B. J. Min, and H. Lee. Deep learning-based decision-tree classifier for covid-19 diagnosis from chest x-ray imaging. *Frontiers in Medicine*, 7, 2020.

V. Zantedeschi, M. J. Kusner, and V. Niculae. Learning binary trees via sparse relaxation. *ArXiv*, abs/2010.04627, 2020.

W. Zhang and E. Ntoutsi. FAHT: an adaptive fairness-aware decision tree classifier. *CoRR*, abs/1907.07237, 2019.

H. Zhu, P. Murali, D. T. Phan, L. M. Nguyen, and J. R. Kalagnanam. A scalable MIP-based method for learning optimal multivariate decision trees. *Advances in Neural Information Processing Systems*, 33, 2020.

8 Appendix

Table 6: Average solution time. Best in **bold** (* if BendersOCT performed best).

Dataset	D_h	FlowOCT	MCF1	MCF2	CUT1	CUT2	BendersOCT
soybean-small	2	0.08	0.10	0.06	**0.04**	0.07	0.20
	3	0.16	**0.08**	0.12	**0.08**	**0.08**	0.20
	4	0.33	0.28	0.27	**0.22**	**0.22**	0.36
	5	0.65	0.52	0.66	**0.49**	0.50	0.49
monk3	2	**0.51**	0.94	1.02	0.56	0.56	0.23*
	3	91.98	105.15	92.57	**66.25**	91.99	15.70*
	4	2210.76	2186.47	2197.56	**1733.84**	1924.94	2381.33
	5	187.22	219.00	142.03	**79.16**	136.59	42.62
monk1	2	0.74	1.68	1.12	**0.67**	0.73	0.48*
	3	32.65	35.65	41.05	**25.44**	25.73	4.33
	4	15.89	22.63	**8.92**	9.87	13.01	2.13
	5	**8.46**	16.60	13.78	12.06	12.52	1.51
hayes-roth	2	0.69	2.69	1.01	**0.62**	**0.62**	0.83
	3	**13.86**	59.02	31.50	40.76	26.83	7.86*
	4	2430.02	1819.18	1415.03	1512.81	**1270.11**	2422.33
	5	**3600.00**	**3600.00**	**3600.00**	**3600.00**	**3600.00**	3600.00*
monk2	2	6.76	5.83	3.40	**1.87**	1.92	3.12
	3	2935.01	1898.05	1586.15	1014.42	**804.51**	1205.76
	4	**3600.00**	**3600.00**	**3600.00**	**3600.00**	**3600.00**	3600.00*
	5	**3600.00**	**3600.00**	**3600.00**	**3600.00**	**3600.00**	3600.00*
house-votes-84	2	**0.76**	2.97	1.97	1.06	1.07	0.48
	3	200.91	167.30	**127.05**	173.64	133.14	80.30*
	4	2882.22	2169.07	**1087.32**	1144.08	1209.22	2882.02
	5	471.55	**62.14**	76.84	115.90	116.20	555.25
spect	2	9.97	15.90	18.78	2.84	**2.63**	4.14
	3	2790.05	1760.49	1736.55	**942.73**	1385.65	2162.11
	4	**3600.00**	**3600.00**	**3600.00**	**3600.00**	**3600.00**	3600.00*
	5	**3600.00**	**3600.00**	**3600.00**	**3600.00**	**3600.00**	3600.00*
breast-cancer	2	23.69	45.94	7.79	7.68	**7.43**	17.19
	3	**3600.00**	**3600.00**	**3600.00**	**3600.00**	**3600.00**	3600.00*
	4	**3600.00**	**3600.00**	**3600.00**	**3600.00**	**3600.00**	3600.00*
	5	**3600.00**	**3600.00**	**3600.00**	**3600.00**	**3600.00**	3600.00*
balance-scale	2	10.51	21.68	17.16	7.29	**7.23**	6.60*
	3	1863.60	1651.92	1384.10	1235.88	**1232.90**	895.25*
	4	**3600.00**	**3600.00**	**3600.00**	**3600.00**	**3600.00**	3600.00*
	5	**3600.00**	**3600.00**	**3600.00**	**3600.00**	**3600.00**	3600.00*
tic-tac-toe	2	490.04	327.18	96.10	**93.31**	104.08	139.20
	3	**3600.00**	**3600.00**	**3600.00**	**3600.00**	**3600.00**	3600.00*
	4	**3600.00**	**3600.00**	**3600.00**	**3600.00**	**3600.00**	3600.00*
	5	**3600.00**	**3600.00**	**3600.00**	**3600.00**	**3600.00**	3600.00*
car_evaluation	2	83.46	111.03	86.05	38.99	**35.97**	32.08*
	3	**3600.00**	**3600.00**	**3600.00**	**3600.00**	**3600.00**	3600.00*
	4	**3600.00**	**3600.00**	**3600.00**	**3600.00**	**3600.00**	3600.00*
	5	**3600.00**	**3600.00**	**3600.00**	**3600.00**	**3600.00**	3600.00*
kr-vs-kp	2	**598.99**	3155.29	1029.96	1188.90	1199.33	420.01*
	3	**3600.00**	**3600.00**	**3600.00**	**3600.00**	**3600.00**	3600.00*
	4	**3600.00**	**3600.00**	**3600.00**	**3600.00**	**3600.00**	3600.00*
	5	**3600.00**	**3600.00**	**3600.00**	**3600.00**	**3600.00**	3600.00*
fico_binary	2	944.53	3600.00	**858.20**	955.18	1408.68	2414.30
	3	**3600.00**	**3600.00**	**3600.00**	**3600.00**	**3600.00**	3600.00*
	4	**3600.00**	**3600.00**	**3600.00**	**3600.00**	**3600.00**	3600.00*
	5	**3600.00**	**3600.00**	**3600.00**	**3600.00**	**3600.00**	3600.00*

Table 7: Average in-sample optimality gap (%). Best in **bold** (* if BendersOCT performed best).

Dataset	D_h	FlowOCT	MCF1	MCF2	CUT1	CUT2	BendersOCT
soybean-small	2	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00*
	3	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00*
	4	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00*
	5	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00*
monk3	2	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00*
	3	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00*
	4	1.81	1.35	0.67	**0.00**	0.22	1.35
	5	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00*
monk1	2	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00*
	3	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00*
	4	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00*
	5	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00*
hayes-roth	2	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00*
	3	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00*
	4	4.90	**0.00**	**0.00**	**0.00**	**0.00**	0.46
	5	8.86	9.53	**8.66**	9.12	9.82	9.06
monk2	2	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00*
	3	2.54	0.85	0.85	**0.00**	**0.00**	0.00*
	4	**17.23**	18.36	18.96	18.05	18.60	12.95*
	5	11.71	9.20	10.97	9.20	**8.45**	10.38
house-votes-84	2	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00*
	3	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.01
	4	80.00	0.23	**0.00**	**0.00**	0.12	0.46
	5	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00*
spect	2	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00*
	3	8.26	0.37	0.37	**0.00**	**0.00**	0.71
	4	35.36	4.70	4.48	4.59	**4.36**	4.21*
	5	55.49	4.61	6.01	5.30	**4.17**	4.87
breast-cancer	2	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00*
	3	73.58	17.99	16.45	**12.98**	13.91	12.07*
	4	16.05	17.11	15.27	15.52	**15.02**	15.67
	5	11.68	10.37	11.69	10.15	**9.90**	11.34
balance-scale	2	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00*
	3	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.01
	4	97.26	21.67	21.11	**18.36**	19.26	9.88*
	5	100.00	21.44	**21.12**	21.32	21.56	21.63
tic-tac-toe	2	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00*
	3	100.00	30.84	29.01	**28.45**	29.36	18.57*
	4	100.00	22.21	19.52	**19.20**	19.75	19.75*
	5	100.00	13.04	**13.00**	16.27	14.81	14.24*
car_evaluation	2	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.00*
	3	78.89	23.29	18.32	18.26	**17.50**	5.07*
	4	100.00	28.27	**21.44**	22.73	21.90	20.65*
	5	83.71	40.15	**32.01**	39.44	34.43	18.03*
kr-vs-kp	2	**0.00**	7.23	**0.00**	**0.00**	**0.00**	0.00*
	3	19.97	42.40	28.01	**15.56**	18.52	8.56*
	4	**27.98**	91.94	57.25	45.89	42.85	12.60*
	5	**25.56**	57.43	45.86	34.53	43.39	5.36*
fico_binary	2	**0.00**	25.05	**0.00**	**0.00**	**0.00**	0.00*
	3	98.31	**39.50**	39.51	41.47	41.43	39.95
	4	99.57	> 100	60.95	> 100	> 100	41.48*
	5	**40.71**	> 100	> 100	> 100	> 100	40.64*

Table 8: Average in-sample accuracy (%). Best in **bold** (* if BendersOCT performed best).

Dataset	D_h	FlowOCT	MCF1	MCF2	CUT1	CUT2	BendersOCT
soybean-small	2	**100**	**100**	**100**	**100**	**100**	100*
	3	**100**	**100**	**100**	**100**	**100**	100*
	4	**100**	**100**	**100**	**100**	**100**	100*
	5	**100**	**100**	**100**	**100**	**100**	100*
monk3	2	**93.77**	93.19	93.19	93.19	93.19	93.77*
	3	**95.74**	95.38	95.38	95.38	95.38	96.07*
	4	98.03	98.46	**98.68**	**98.68**	**98.68**	99.02*
	5	**100**	**100**	**100**	**100**	**100**	100*
monk1	2	**82.26**	81.51	81.51	81.51	81.51	82.26*
	3	91.29	**92.04**	**92.04**	**92.04**	**92.04**	90.97
	4	**100**	**100**	**100**	**100**	**100**	100*
	5	**100**	**100**	**100**	**100**	**100**	100*
hayes-roth	2	64.55	**65.25**	**65.25**	**65.25**	**65.25**	64.24
	3	**80.00**	79.60	79.60	79.60	79.60	80.00*
	4	**90.00**	88.89	88.89	88.89	88.89	89.09
	5	**92.12**	91.31	91.11	91.31	90.71	92.42*
monk2	2	67.62	**68.25**	**68.25**	**68.25**	**68.25**	67.62
	3	76.67	77.14	77.14	**77.30**	**77.30**	75.71
	4	81.43	83.17	83.02	**83.49**	83.17	82.38
	5	90.00	**91.59**	90.16	91.47	92.46	90.71
house-votes-84	2	96.72	**97.24**	**97.24**	**97.24**	**97.24**	96.72
	3	97.93	**98.39**	**98.39**	**98.39**	**98.39**	97.93
	4	99.31	**99.66**	**99.66**	**99.66**	**99.66**	99.31
	5	**100**	**100**	**100**	**100**	**100**	100*
spect	2	**79.85**	79.80	79.80	79.80	79.80	79.85*
	3	**82.56**	82.50	82.50	82.50	82.50	83.61*
	4	**88.12**	87.00	87.00	87.10	87.10	88.12*
	5	90.98	91.20	89.90	90.60	**91.30**	91.28
breast-cancer	2	**78.70**	78.45	78.45	78.45	78.45	78.99*
	3	**83.91**	81.84	81.74	82.32	82.42	83.04
	4	**87.20**	85.35	86.80	86.47	86.63	87.10
	5	90.22	89.86	89.37	**90.82**	89.86	91.74*
balance-scale	2	69.04	**69.49**	**69.49**	**69.49**	**69.49**	69.17
	3	75.06	**75.77**	**75.77**	**75.77**	**75.77**	75.13
	4	78.85	**79.10**	78.76	78.97	78.97	78.53
	5	81.47	82.35	**82.56**	82.44	82.26	82.24
tic-tac-toe	2	70.90	**70.97**	**70.97**	**70.97**	**70.97**	70.77
	3	77.83	76.46	77.52	**77.86**	77.33	77.62
	4	**84.38**	81.92	83.70	83.90	83.54	83.55
	5	86.76	88.52	**88.66**	86.10	87.16	88.10
car_evaluation	2	77.43	**77.78**	**77.78**	**77.78**	**77.78**	77.43
	3	80.81	80.23	80.97	**81.23**	81.22	80.81
	4	**82.55**	78.18	82.35	81.54	82.05	82.59
	5	**83.84**	72.35	75.91	71.82	74.58	84.58*
kr-vs-kp	2	**87.31**	86.97	86.97	86.97	86.97	87.31*
	3	84.91	72.50	78.87	**86.78**	84.85	92.33*
	4	**78.25**	52.22	64.10	68.54	70.10	89.11*
	5	**79.84**	64.07	68.58	74.70	70.16	95.16*
fico_binary	2	71.22	71.19	**71.27**	**71.27**	**71.27**	71.22
	3	**71.59**	71.44	71.45	70.48	70.51	70.97
	4	**70.89**	52.96	62.38	52.96	52.96	70.64
	5	**69.65**	52.96	61.93	52.96	52.96	71.11*