The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

Fairness-aware Edit of Thresholds in a Learned Decision Tree Using a Mixed Integer Programming Formulation

Kentaro Kanamori Hiroki Arimura

Graduate School of Infomation Science and Technology, Hokkaido University

Fairness in machine learning is an emerging topic in recent years. In this paper, we propose a post-processing method for editing a given decision tree to be fair according to a specified discrimination criterion by modifying its branching thresholds in internal nodes. We propose a mixed integer linear programming (MIP) formulation for the problem, which can deal with several other constraints flexibly and can be solved efficiently by any existing solver. By experiments, we confirm the effectiveness of our approach by comparing existing post-processing methods.

1. Introduction

Background and Motivation In the application of machine learning models to the actual decision making, problems other than their prediction accuracy, such as *interpretability* [10] and *fairness* [7], attract increasing attention. If their prediction results are unexplainable or discriminative, they are no longer usable in the actual decision making, even if they achieve high accuracy.

In this paper, we focus on *decision tree models* [4], and study a post-processing method [7] for decision trees. More specifically, we consider a problem of editing a decision tree by modifying its branching thresholds in internal nodes so as to satisfy fairness constraints w.r.t. a sensitive attribute such as gender and race, named *re-thresholding problem*. We formulate it as a *mixed integer linear programming (MIP)* problem, which we can obtain an optimal solution by powerful off-the-shelf solvers such as CPLEX[*1] and Gurobi[*2]. Our approach has the following advantages:

- **Interpretability:** Decision tree models are known as one of the interpretable machine learning models since their predictions are based on a set of rules that human can understand easily [10, 11].
- **Adaptivity:** Post-processing methods can deal with the situation that a sensitive attribute or fairness criterion is given after learning [6, 7, 9]. In the actual decision making, they are not always given in advance.
- **Flexibility:** Our MIP formulation can deal with additional constraints defined by users without implementing designated algorithms, if these constraints can be expressed as linear equations or inequalities [2, 3, 11].

Contribution Our contributions are as follows:

1. We formulate a post-processing problem of editing a given decision tree so as to satisfy fairness constraints by modifying its branching thresholds, named *re-thresholding problem*, as an MIP problem.
2. We formulate an edit distance [12] of a decision tree to measure dissimilarity between given and modified decision trees as edit limitation constraints.
3. By experiments on real datasets, we confirm the effectiveness of our proposed method by comparing other existing post-processing methods.

Contact: kanamori@ist.hokudai.ac.jp

∗1 https://www.ibm.com/analytics/cplex-optimizer

∗2 http://www.gurobi.com/

2. Preliminary

2.1 Notation

For $n \in \mathbb{N}$, we denote by $[n] = \{1, \ldots, n\}$. For a proposition ψ, $\mathbb{I}[\psi]$ denotes the indicator of ψ, i.e., $\mathbb{I}[\psi] = 1$ if ψ is true, and $\mathbb{I}[\psi] = 0$ if ψ is false.

In this paper, we consider a binary classification problem, and assume its input space is normalized to $[0,1]^D$ without loss of generality. Let a pair of an input and an output $(x, y) \in [0,1]^D \times \{0,1\}$ be an *example*, and $S = \{(x^{(j)}, y^{(j)})\}_{j=1}^N$ be a *dataset* with N examples. For a *prediction model* $h : [0,1]^D \to \{0,1\}$, the *empirical loss* on S is defined by $L(h \mid S) := \frac{1}{N} \sum_{j=1}^N \mathbb{I}\left[h(x^{(j)}) \neq y^{(j)}\right]$.

In addition, we consider a *sensitive attribute* $z \in \{0,1\}$, such as gender and race. Let $z^{(j)}$ be the sensitive attribute value w.r.t. j-th example $(x^{(j)}, y^{(j)}) \in S$, and $Z = \{z^{(j)}\}_{j=1}^N$ be the set of its values w.r.t. S.

2.2 Decision trees

The *decision tree* [4] is a prediction model that consists of a set of prediction rules expressed by the full binary ordered tree structure. It makes the prediction according to the label of the leaf node that the input x reaches, and the corresponding leaf node is determined by traversing the tree from the root. Each internal node has a pair of parameters $(d, b) \in [D] \times [0,1]$, where d is a *branching feature* and b is a *branching threshold*, and the input $x = (x_1, \ldots, x_D)$ is directed to one of two child nodes depending on whether the statement $x_d \leq b$ is true or not.

Then, the decision tree can be expressed as follows:

$$h(x) = \sum_{k=1}^{K} l_k \prod_{m \in a_k^{(\mathrm{L})}} \mathbb{I}[x_{d_m} \leq b_m] \prod_{m \in a_k^{(\mathrm{R})}} \mathbb{I}[x_{d_m} > b_m],$$

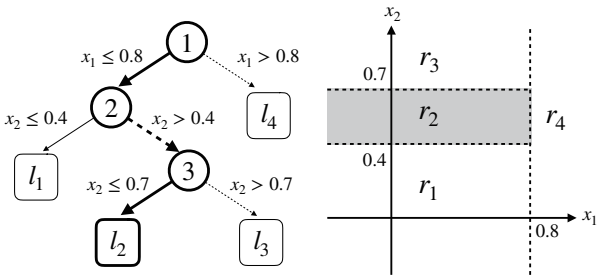

Figure 1: An illustration of a decision tree. For 2-nd leaf node, $a_2^{(\mathrm{L})} = \{1, 3\}$, $a_2^{(\mathrm{R})} = \{2\}$, and its corresponding region $r_2 = (-\infty, 0.8] \times (0.4, 0.7]$.

where $K \in \mathbb{N}$ is the total number of leaf nodes, $l_k \in \{0, 1\}$ is a predictive label of k-th leaf node, $(d_m, b_m) \in [D] \times [0, 1]$ is a branching rule in m-th internal node, and $a_k^{(\mathrm{L})}$ $(a_k^{(\mathrm{R})}) \subseteq [K-1]$ is a set of left(right)-branching internal node indexes on the pass from the root node to k-th leaf node. Note that a decision tree with K leaf nodes has $K-1$ internal nodes since it is expressed as the full binary ordered tree structure. We denote a region corresponding to k-th leaf node by $r_k \subseteq [0, 1]^D$. Then, a set of the regions $\{r_k\}_{k=1}^K$ expresses a partition of the input space. Figure 1 illustrates an example of a decision tree.

We denote a set of all possible decision trees by $\mathcal{H}$. In this paper, since we consider the problem of modifying branching thresholds $B := (b_1, \ldots, b_{K-1}) \in [0, 1]^{K-1}$ of a given decision tree, we denote a decision tree with B by $h_B \in \mathcal{H}$.

2.3 Discrimination scores

We use two major criteria named *demographic parity(DP)* [5] and *equal opportunity(EO)* [8] to evaluate the discrimination of the model. We denote an empirical probability on a dataset S and sensitive attribute Z by $\hat{P}$.

Definition 1 (DP score) *DP score of a model h on a dataset S w.r.t. a sensitive attribute Z is defined by*

$$\delta_{\mathrm{DP}}(h \mid S, Z) := |\hat{P}(h(x) = 1 \mid z = 1) - \hat{P}(h(x) = 1 \mid z = 0)|.$$

Definition 2 (EO score) *EO score of a model h on a dataset S w.r.t. a sensitive attribute Z is defined by*

$$\delta_{\mathrm{EO}}(h \mid S, Z) := |\hat{P}(h(x) = 1 \mid y = 1, z = 1) - \hat{P}(h(x) = 1 \mid y = 1, z = 0)|.$$

In this paper, we call DP and EO scores *discrimination score* together. These values approach 1 as the model h tends to make the predictions unfairly for z, while they approach 0 if the model makes the predictions fairly.

2.4 Problem formulation

Here, we define our problem named *re-thresholding problem*. We assume that a decision tree h_B with branching thresholds B is already given, and the goal is to reduce the discrimination score on a given dataset S by modifying branching thresholds in B without changing the given decision tree significantly.

Problem 1 (Re-thresholding) *Given a dataset S, sensitive attribute Z, decision tree h_B with branching thresholds $B = (b_1, \ldots, b_{K-1})$, discrimination score $disc \in \{\mathrm{DP}, \mathrm{EO}\}$, discrimination threshold $t \in [0, 1]$, dissimilarity measure of decision trees $\Delta : \mathcal{H} \times \mathcal{H} \to \mathbb{R}_{\ge 0}$, and parameters $\lambda \ge 0$ and $N_{\mathrm{minsup}} \in [|S|]$, re-thresholding problem is defined as follows:*

$$\begin{aligned} \underset{\boldsymbol{\theta} \in [0,1]^{K-1}}{\text{minimize}} \quad & L(h_{\boldsymbol{\theta}} \mid S) + \lambda \cdot \Delta(h_B, h_{\boldsymbol{\theta}}) \\ \text{subject to} \quad & \delta_{disc}(h_{\boldsymbol{\theta}} \mid S, Z) \le t \\ & \forall k \in [K] : \ |\{(x, y) \in S \mid x \in r_k\}| \ge N_{\mathrm{minsup}} \end{aligned}$$

Note that we modify only branching thresholds, and do not modify branching features and predictive labels in leaf nodes. As a dissimilarity measure of decision trees Δ in Problem 1, we propose an *edit distance of a decision tree* Δ_{ED}, which will be defined in the next section.

3. Proposed Method

Our formulation is based on *OCT*, the MIP framework for learning optimal classification trees proposed by Bertsimas and Dunn [3]. In order to adapt it to our editing problem for decision trees, we formulate an edit distance of decision trees and fairness constraints.

3.1 Program variables

For $m \in [K-1]$, $j \in [N]$, $k \in [K]$, we introduce some variables for formulating Problem 1 as follows:

- $\theta_m \in [0, 1]$ is a modified branching threshold in m-th internal node.
- $\phi_{j,k} \in \{0, 1\}$ indicates whether j-th input $x^{(j)}$ reaches k-th leaf node, i.e., $\phi_{j,k} = \mathbb{I}\left[x^{(j)} \in r_k\right]$.
- $\psi_k \in \{0, 1\}$ indicates whether some example reaches k-th leaf node, i.e., $\psi_k = \mathbb{I}\left[\exists j \in [N] : x^{(j)} \in r_k\right]$.
- $\epsilon_m^{(\mathrm{L})}$ $(\epsilon_m^{(\mathrm{R})}) \in \{0, 1\}$ indicates whether no example reaches leaf nodes in the left (right) subtree of m-th internal node $c_m^{(\mathrm{L})}$ $(c_m^{(\mathrm{R})}) \subseteq [K]$, i.e., $\epsilon_m^{(\mathrm{L})} = \mathbb{I}\left[\forall k \in c_m^{(\mathrm{L})} : \psi_k = 0\right]$ $\left(\epsilon_m^{(\mathrm{R})} = \mathbb{I}\left[\forall k \in c_m^{(\mathrm{R})} : \psi_k = 0\right]\right)$.
- $\xi_m \in [0, 2]$ expresses the cost corresponding to edit operations for m-th internal node.

$\phi_{j,k}$, ψ_k, $\epsilon_m^{(\mathrm{L})}$, $\epsilon_m^{(\mathrm{R})}$ and ξ_m are auxiliary variables. The total number of the variables is $\mathcal{O}(NK)$.

3.2 Edit distance of decision trees

We define an *edit distance of a decision tree* based on the standard tree edit distance [12]. For two ordered trees T and T', the tree edit distance between them is defined as the minimal length of the sequence of editing operations to transform T into T'. Available edit operations are insertion, deletion, and relabeling.

In our problem, an explicit edit operation is relabeling threshold values in internal nodes. We define $|b_m - \theta_m| \in$

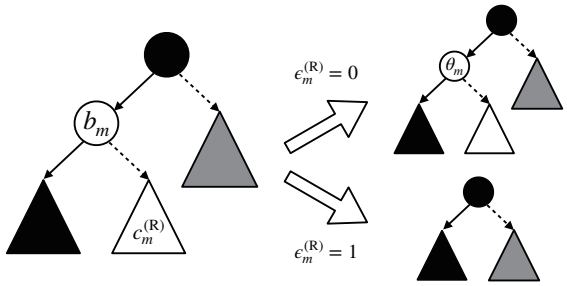

Figure 2: An example of deleting and relabeling operations. If no example reaches leaf nodes in the right subtree of m-th internal node ($\epsilon_m^{(\mathrm{R})} = 1$), m-th internal node and its right subtree are deleted from the tree, and the total number of these edit operations are measured as an edit distance. Otherwise ($\epsilon_m^{(\mathrm{R})} = 0$), $|b_m - \theta_m|$ occurs as a cost for the relabeling operation.

$[0, 1]$ as the cost of the relabeling operation for m-th internal node. However, when a modified branching threshold exceeds a specific value, no examples in S reaches its child leaf node, which is denoted by $\psi_k = 0$. By deletion of all leaf nodes with $\psi_k = 0$ and their parent internal nodes in the decision tree $h_{\boldsymbol{\theta}}$, we can obtain a unique tree structure whose prediction results on S are equivalent to $h_{\boldsymbol{\theta}}$.

Then, we define our edit distance Δ_{ED} between $h_{\boldsymbol{\theta}}$ and h_B by the total number of these delete operations and relabeling costs. For m-th internal node, if no example reaches leaf nodes in its left or right subtree, i.e., $\epsilon_m^{(\mathrm{L})} = 1$ or $\epsilon_m^{(\mathrm{R})} = 1$, the editing cost ξ_m is 2 because m-th internal node and its subtree are deleted. Note that if $\epsilon_m^{(\mathrm{L})} = 1$ ($\epsilon_m^{(\mathrm{R})} = 1$), $\xi_{m'} = 2$ holds for any m'-th internal nodes included in left (right) subtree of m-th internal node because $\psi_k = 0$ for any $k \in c_{m'}^{(\mathrm{L})}$ ($c_{m'}^{(\mathrm{R})}$) and $\epsilon_{m'}^{(\mathrm{L})} = 1$ or $\epsilon_{m'}^{(\mathrm{R})} = 1$ hold, and sum of these $\xi_{m'}$ is equivalent to the cost of delete operations for the subtree. It can be expressed as follows:

$$\Delta_{\mathrm{ED}}(h_B, h_{\boldsymbol{\theta}}) = \sum_{m=1}^{K-1} \max\{|b_m - \theta_m|, 2\epsilon_m^{(\mathrm{L})}, 2\epsilon_m^{(\mathrm{R})}\}.$$

Figure 2 shows an example of edit operations in our problem. By using ξ_m, ψ_m, $\epsilon_m^{(\mathrm{L})}$, and $\epsilon_m^{(\mathrm{R})}$, it can be expressed as a linear function and constraints.

3.3 Fairness constraints

We can express both DP and EO scores by using variables $\phi_{j,k}$ as follows:

$$\delta_{disc}(h_{\boldsymbol{\theta}} \mid S, Z) = \sum_{k=1}^{K} l_k \sum_{j=1}^{N} d_{j,k}^{(disc)} \phi_{j,k}.$$

$d_{j,k}^{(disc)}$ is a constant value determined automatically when S and Z are given such that

$$d_{j,k}^{(disc)} = \begin{cases} \frac{z^{(j)}}{|S_1^{(\mathrm{DP})}|} - \frac{1-z^{(j)}}{|S_0^{(\mathrm{DP})}|} & (disc = \mathrm{DP}), \\ \frac{y^{(j)}z^{(j)}}{|S_1^{(\mathrm{EO})}|} - \frac{y^{(j)}(1-z^{(j)})}{|S_0^{(\mathrm{EO})}|} & (disc = \mathrm{EO}), \end{cases}$$

where $S_z^{(\mathrm{DP})} := \{(x^{(j)}, y^{(j)}) \in S \mid z^{(j)} = z\}$ and $S_z^{(\mathrm{EO})} := \{(x^{(j)}, y^{(j)}) \in S \mid y^{(j)} = 1 \wedge z^{(j)} = z\}$ for $z \in \{0, 1\}$.

3.4 Overall formulation

Now, we can formulate Problem 1 as the following MIP problem:

$$\text{minimize} \quad \frac{1}{N} \sum_{k=1}^{K} \sum_{j=1}^{N} c_{j,k} \phi_{j,k} + \lambda \sum_{m=1}^{K-1} \xi_m \tag{1}$$

$$\text{subject to} \quad \sum_{k=1}^{K} \phi_{j,k} = 1, \forall j \in [N] \tag{2}$$

$$\phi_{j,k} \leq \psi_k, \forall k \in [K], j \in [N] \tag{3}$$

$$\sum_{j=1}^{N} \phi_{j,k} \geq N_{\mathrm{minsup}} \cdot \psi_k, \forall k \in [K] \tag{4}$$

$$x_{d_m}^{(j)} \leq \theta_m + (1 - \phi_{j,k}), \quad \forall k \in [K], j \in [N], m \in a_k^{(\mathrm{L})} \tag{5}$$

$$x_{d_m}^{(j)} - e_m \geq \theta_m - (1 + e_{\max})(1 - \phi_{j,k}), \quad \forall k \in [K], j \in [N], m \in a_k^{(\mathrm{R})} \tag{6}$$

$$-\xi_m \leq b_m - \theta_m, \forall m \in [K-1] \tag{7}$$

$$b_m - \theta_m \leq \xi_m, \forall m \in [K-1] \tag{8}$$

$$\xi_m \geq 2\epsilon_m^{(\mathrm{L})}, \forall m \in [K-1] \tag{9}$$

$$\xi_m \geq 2\epsilon_m^{(\mathrm{R})}, \forall m \in [K-1] \tag{10}$$

$$1 - \epsilon_m^{(\mathrm{L})} \leq \sum_{k \in c_m^{(\mathrm{L})}} \psi_k, \forall m \in [K-1] \tag{11}$$

$$\sum_{k \in c_m^{(\mathrm{L})}} \psi_k \leq (1 - \epsilon_m^{(\mathrm{L})})|c_m^{(\mathrm{L})}|, \forall m \in [K-1] \tag{12}$$

$$1 - \epsilon_m^{(\mathrm{R})} \leq \sum_{k \in c_m^{(\mathrm{R})}} \psi_k, \forall m \in [K-1] \tag{13}$$

$$\sum_{k \in c_m^{(\mathrm{R})}} \psi_k \leq (1 - \epsilon_m^{(\mathrm{R})})|c_m^{(\mathrm{R})}|, \forall m \in [K-1] \tag{14}$$

$$\sum_{k=1}^{K} l_k \sum_{j=1}^{N} d_{j,k}^{(disc)} \phi_{j,k} \leq t \tag{15}$$

$$-\sum_{k=1}^{K} l_k \sum_{j=1}^{N} d_{j,k}^{(disc)} \phi_{j,k} \leq t \tag{16}$$

where $e_m = \min\{|x_{d_m}^{(i)} - x_{d_m}^{(j)}| \mid i, j \in [N], x_{d_m}^{(i)} \neq x_{d_m}^{(j)}\}$, $e_{\max} = \max\{e_m\}_{m=1}^{K-1}$, and $c_{j,k} := l_k(1 - y^{(j)}) + (1 - l_k)y^{(j)}$. We can express $L(h_{\boldsymbol{\theta}} \mid S)$ and $\Delta_{\mathrm{ED}}(h_B, h_{\boldsymbol{\theta}})$ by $\sum_{k=1}^{K} \sum_{j=1}^{N} c_{j,k}\phi_{j,k}$ and $\sum_{m=1}^{K-1} \xi_m$, respectively. Equation (2) and inequalities (3-6) are the same constraints with OCT [3]. Inequalities (7-15) and (16,17) are constraints for expressing our edit distance $\Delta_{\mathrm{ED}}(h_B, h_{\boldsymbol{\theta}})$ and fairness constraint $\delta_{disc}(h_{\boldsymbol{\theta}} \mid S, Z) \leq t$.

4. Experiments

Experimental setup We used the **COMPAS** dataset [1]. It has $N = 6172$ examples and the total number of features is $D = 9$. Its output label $y^{(j)} \in \{0, 1\}$ indicates whether j-th person recidivates within two years. We use the attribute "African_American" as its sensitive attribute $z^{(j)} \in \{0, 1\}$.

Table 1: Experimental results averaged over 5 trials. "Loss" anf "DP" denote the average empirical loss with its standard deviation and DP score for training and test datasets. "time" denotes the average running times.

	Training		Test		
method	**Loss**	**DP**	**Loss**	**DP**	**time[s]**
NB-Modified	0.387 ± 0.022	0.103	0.384 ± 0.005	0.118	2.028
Relabeling	0.412 ± 0.018	0.044	0.412 ± 0.028	0.063	0.704×10^{-4}
MIP (proposed)	0.396 ± 0.011	0.095	0.389 ± 0.012	0.112	316.72

We compared our method (**MIP**) with the existing two post-processing methods: (1) modifying naive Bayes (**NB-Modified**) [6] , and (2) relabeling for decision trees (**Relabeling**) [9]. In our experiments, we randomly split the dataset into training (50%) and test (50%) datasets, and report the average statistics over 5 trials. We obtained initial prediction models and applied each methods by using training datasets. Decision trees were learned by CART [4] with the constraint on these height less than 3. We used the threshold value of the fairness constraint $t = 0.1$ for all methods, and $\lambda = 0.1$ and $N_{\text{minsup}} = 100$ for Problem 1. All codes were implemented in Python 3.6 with scikit-learn*3 and IBM ILOG CPLEX Optimization Studio v12.8. All experiments were conducted on 64-bit macOS Sierra 10.12.6 with Intel Core i5 2.90GHz CPU and 8GB Memory.

Experimental results Table 1 shows the experimental results for DP scores. The DP scores of naive Bayes and decision trees before modification were 0.462 and 0.226, respectively. Our method maintained slightly lower loss than the relabeling method for decision trees on both training and test datasets, and comparable accuracy with the modifying naive Bayes method. We note that the average DP score attained by the modifying naive Bayes method exceeds $t = 0.1$ since it sometimes failed to obtain a model satisfying the fairness constraint. On the other hands, the running time of our method was longer than other two methods. This result implies that we need to improve our formulation, e.g., reducing program variables and constraints.

5. Conclusion and Discussion

We studied a fairness-aware post-processing method for decision trees, and proposed an MIP formulation of the rethresholding problem, which makes a given decision tree fair by modifying their branching thresholds. Also, we formulate an edit distance of a decision tree so as to avoid that a learned model is changed significantly. By experiments on real datasets, we confirmed the effectiveness of our methods by comparing with the existing post-processing methods.

As future work, we will try to extend our framework so as to modify branching features, and deal with user-defined constraints more flexibly. It is important that maintaining user's prior knowledge contained in the model, e.g., order of branching features on a pass from its root node to a leaf node, while improving fairness by editing operations.

Acknowledgements This work was partially supported by JSPS KAKENHI(S) 15H05711 and JSPS KAKENHI(A) 16H01743.

*3 `https://scikit-learn.org/`

References

[1] J. Adebayo. FairML: Auditing black-box predictive models. `https://github.com/adebayoj/fairml`, 2018.

[2] S. Aghaei, M. J. Azizi, and P. Vayanos. Learning optimal and fair decision trees for non-discriminative decision-making. In *Proc. AAAI 2019, Hawaii*, 2019 (to appear).

[3] D. Bertsimas and J. Dunn. Optimal classification trees. *Mach. Learn.*, pages 1039–1082, 2017.

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

[5] T. Calders, F. Kamiran, and M. Pechenizkiy. Building classifiers with independency constraints. In *IEEE ICDM 2009 Workshops*, pages 13–18, Dec 2009.

[6] T. Calders and S. Verwer. Three naive bayes approaches for discrimination-free classification. *Data Min. Knowl. Discov.*, pages 277–292, 2010.

[7] S. Hajian, F. Bonchi, and C. Castillo. Algorithmic bias: From discrimination discovery to fairness-aware data mining. In *Proc. ACM KDD 2016, San Francisco*, pages 2125–2126, 2016.

[8] M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In *NIPS 2016, Barcelona*, pages 3315–3323.

[9] F. Kamiran, T. Calders, and M. Pechenizkiy. Discrimination aware decision tree learning. In *IEEE ICDM 2010*, pages 869–874, 2010.

[10] M. T. Ribeiro, S. Singh, and C. Guestrin. "Why should I trust you?": Explaining the predictions of any classifier. In *Proc. ACM KDD 2016, San Francisco*, pages 1135–1144, 2016.

[11] C. Rudin and Ş. Ertekin. Learning customized and optimized lists of rules with mathematical programming. *Mathematical Programming Computation*, 10(4):659–702, Dec 2018.

[12] K. Zhang and D. Shasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal on Computing*, 18(6):1245–1262, 1989.