NONLINEAR OPTIMIZATION OVER TREES WITH BINARY COUPLING DECISIONS

LIANGJU LI AND SHAONING HAN

Abstract. Mixed integer nonlinear programs with binary coupling decisions naturally model selective coordination tasks where a fixed penalty is incurred whenever adjacent continuous variables differ. A prominent example is the classical Potts model, which is widely used in statistical inference. However, exact solvability remains theoretically challenging since the problem is NP hard on general graphs, and generic global optimization approaches such as branch and bound methods often lack worst case guarantees. In this paper, we resolve this computational bottleneck for arbitrary tree topologies. We propose an exact parametric dynamic programming algorithm that achieves polynomial oracle complexity when the node cost functions belong to a fixed degree weak Chebyshev space, a broad functional class that includes piecewise polynomials. For the widely used quadratic cost setting, we perform a refined analysis which improves the complexity exponent to a near quadratic bound.

Keywords. Mixed-integer nonlinear programming, Parametric dynamic programming, Potts model, Trees, Weak Chebyshev spaces, Piecewise polynomials

September 2026

1. Introduction

Given a tree $\mathcal{T} = (\mathcal{V}, \mathcal{E})$, we study the following mixed-integer nonlinear optimization problem

$$\min_{x,z} \sum_{i\in\mathcal{V}} f_i(x_i) + \sum_{e\in\mathcal{E}} c_e z_e \tag{1a}$$

$$\text{s.t. } (x_i - x_j)(1 - z_e) = 0,\ z_e \in \{0,1\} \qquad \forall e = (i,j) \in \mathcal{E} \tag{1b}$$

$$\ell \le x_i \le u, \qquad \forall i \in \mathcal{V}. \tag{1c}$$

L. Li, S. Han: Department of Mathematics, National University of Singapore, Singapore 119076. e1583542@u.nus.edu, shaoninghan@nus.edu.sg.

where each $f_i : \mathbb{R} \to \mathbb{R}$ is a continuous function with favorable properties (e.g. polynomial), $c_e \geq 0$ for all $e \in \mathcal{E}$, and $\ell, u \in \mathbb{R} \cup \{\pm\infty\}$. We assume that $\mathcal{V} = [n]$ and that the optimum of (1) can be attained.

For an edge $e = (i, j) \in \mathcal{E}$, if $z_e = 0$, the complementarity constraint (1b) forces $x_i = x_j$; if $z_e = 1$, then x_i and x_j are decoupled. Therefore, problem (1) can model selective coordination with a fixed charge. Specifically, each entity $i \in \mathcal{V}$ selects a continuous decision x_i and incurs a local cost $f_i(x_i)$. For each prescribed relationship encoded by $\mathcal{E}$, the binary variable z_e controls whether the two entities maintain a common decision or may select separate decisions at a fixed cost c_e; we refer to z_e as a *binary coupling decision*. The optimization then determines whether the benefit of permitting separate decisions justifies this fixed charge. Unlike a smooth penalty, c_e is incurred once the decisions differ, regardless of the magnitude of their difference.

This fixed-charge mechanism with binary coupling decisions arises in many applications. In multiprocessor module assignment, each module has a processor-dependent execution cost, while assigning interacting modules to different processors incurs an interprocessor communication cost [35]. In flexible manufacturing, each operation is assigned to an eligible multipurpose machine, and transferring a part between consecutive operations assigned to different machines incurs a cost [1]. In technology adoption and standard coordination, an agent incurs a cost when its chosen technology or standard is incompatible with one adopted by a related agent [10]. In dynamic pricing with costly price adjustments, a firm selects price levels and change times to balance revenue against the fixed adjustment costs incurred whenever it revises the price [14]. Across these applications, the fixed charge is incurred due to communication, transfer, incompatibility, or adjustment.

A prominent instance of problem (1) arises in statistical inference for signals with abrupt changes. Here, x_i represents a fitted signal or true parameter associated with a noisy observation $y_i \in \mathbb{R}$. The function $f_i(x_i)$ measures its fidelity to the local data and typically takes the form $f_i(x_i) = a_i(x_i - y_i)^p$, where $p \geq 1$ and $a_i \geq 0$. Concurrently, the edge charges penalize changes between related fitted values. In this context, problem (1) reduces to the renowned *Potts functional* [32] in statistics

$$\min_{\ell \leq x_i \leq u} \sum_{i \in \mathcal{V}} a_i(x_i - y_i)^p + \mu \sum_{(i,j) \in \mathcal{E}} \mathbb{1}_{\{x_i \neq x_j\}}, \tag{Potts}$$

where $\mathbb{1}_{\{x_i \neq x_j\}} = 1$ if $x_i \neq x_j$ and 0 otherwise. The scalar $\mu \geq 0$ is the regularity parameter that controls the number of abrupt changes, thereby determining the complexity of the statistical model. Potts model is prevalent across various domains, including change-point detection [7, 37], network anomaly detection [17], image segmentation and denoising [28, 34], genomic copy-number analysis [31], and physics [38].

Branch-and-bound methods provide a general-purpose approach for solving problem (1). For instance, Shen et al. [34] employ this approach to solve an application of the model (Potts) in image segmentation. However, these generic methods typically lack favorable worst-case complexity guarantees and often become computationally prohibitive for large instances. In fact, as we prove in Section 2.1, problem (1) is NP-hard on general graphs even for convex quadratic f_i. For a path graph with $f_i(x_i) = a_i|x_i - y_i|$ and $c_e = \mu$ in (Potts), Weinmann et al. [37] propose an exact dynamic program with running time $\mathcal{O}\left(n^2\right)$. For convex quadratic costs on general graphs, Fan and Guan [17] discretize the continuous variables x_i and apply α-expansion to compute approximate solutions with statistical risk guarantees. Additionally, Landrieu and Obozinski [28] develop an ℓ_0-cut-pursuit algorithm that converges to a local minimum. However, neither of these recent methods can certify global optimality.

1.1. **Contributions and outline.** In this paper, we develop an exact polynomial time algorithm for problem (1) on arbitrary trees under mild regularity and computational assumptions on the node costs, which are satisfied particularly by polynomials of a fixed degree. To the best of our knowledge, no such algorithm was previously known even for convex quadratic node costs. The rest of the paper is organized as follows.

The remainder of Section 1 is devoted to the discussion of related works and the adopted throughout the paper. In Section 2, we establish a complexity boundary: problem (1) is NP-hard on general graphs, whereas on path graphs it admits a simple dynamic programming (DP) algorithm with oracle complexity $\mathcal{O}\left(n^2\right)$. We then formally specify the assumptions on the node costs and discuss their broad applicability. In Section 3, we develop an exact parametric DP algorithm for solving (1) over arbitrary trees and prove its correctness. Following that, we derive an equivalent formulation of the DP recursion in an extended space, which provides the foundation for our complexity analysis. Section 4 establishes the polynomial complexity claimed above and makes its dependence on the node-cost class explicit. For quadratic node costs, Section 5 strengthens the analysis and establishes a nearly quadratic complexity bound. Section 6 concludes the paper.

1.2. **Related work.** We review two closely related streams of research. One stream investigates the structurally tractable classes of quadratic optimization with indicator variables. The other focuses on the application of parametric dynamic programming to related models.

1.2.1. *Quadratic optimization with indicator variables.* For node costs $f_i(t) = a_i t^2 + b_i t$ with $a_i > 0$ and $b_i \in \mathbb{R}$, problem (1) is closely related to convex quadratic optimization with indicator variables. To illustrate the connection, assume $\ell = -\infty$ and $u = \infty$ for simplicity. We root the tree $\mathcal{T}$ at a fixed $r \in \mathcal{V}$ to establish a topological order. This orientation allows us to introduce the variable transformation $y_r = x_r$ and $y_i = x_i - x_{\mathsf{par}(i)}$ for each $i \in \mathcal{V} \setminus \{r\}$, where $\mathsf{par}(i)$ denotes the parent of i. Under the linear mapping, the objective (1a) remains quadratic in y. Thus, by relabeling $c_i = c_e$ and $z_i = z_e$ for all $e = (\mathsf{par}(i), i) \in \mathcal{E}$, problem (1) reformulates into the form

$$\min_{y,z} \left\{ y^\top Q y + q^\top y + c^\top z : \ y_i(1 - z_i) = 0, \ z_i \in \{0,1\}, \ \forall i \in \mathcal{V} \setminus \{r\} \right\}, \tag{MIQP}$$

where matrix $Q \succeq 0$ and vector q are determined by the variable changing.

Because (MIQP) is generally NP-hard [8, 23], extensive research focuses on identifying structural classes of $Q \succeq 0$ that render the problem theoretically tractable. Exact polynomial solvability is established under a few specific conditions. (MIQP) reduces to submodular minimization when Q is a Stieltjes matrix [22]. If Q possesses a small but fixed rank, (MIQP) can be solved as a convex conic quadratic program [36, 21, 33]. Furthermore, when the sparsity pattern of Q forms a path graph [30] or a tree [2], it admits an exact dynamic programming algorithm running in $\mathcal{O}\left(n^2\right)$ time. Beyond these exact solutions, strong approximation guarantees exist when Q has a small bandwidth [19], its sparsity pattern has a small treewidth [3], or its inverse has a tree-structure sparsity pattern [11]. However, the variable transformation applied in (MIQP) yields a full-rank dense matrix Q that does not fall into any of these tractable classes.

1.2.2. *Parametric dynamic programming on trees.* Parametric dynamic programming is a powerful tool for solving optimization problems on trees. Since our proposed algorithm also belongs to this methodological category, we provide a brief review of closely related works.

Kolmogorov et al. [26] and Kuric et al. [27] develop parametric DP algorithms for total variation and truncated total variation models on trees. In their formulations, the fixed pairwise regularizer $\mathbb{1}_{\{x_i \neq x_j\}}$ is replaced by a continuous piecewise quadratic function $f_{ij}(x_i - x_j)$. The proposed algorithm achieves an $\mathcal{O}\left(n^2\right)$ running time when all node and edge functions are convex. However, the worst-case performance degrades to an exponential bound for nonconvex variants. For the specific fixed-charge penalty in (Potts), they establish a polynomial running time restricted to path graphs.

Very recently, Chen et al. [9] apply parametric dynamic programming to isotonic optimization with fixed costs and convex node costs. Their model differs from problem (1) by additionally enforcing an additional isotonic constraint $x_{\mathsf{par}(i)} \le x_i$ along each edge. When the node costs f_i are convex piecewise linear functions with a bounded number of pieces, the proposed approach yields a quadratic-time algorithm on arbitrary trees. The authors further establish a general complexity bound which is polynomial in the problem size n but exponential in the Horton-Strahler number [4], a metric quantifying the branching complexity of $\mathcal{T}$.

1.3. **Notation.** Given $x \in \mathbb{R}^n$ and $\alpha \subseteq [n]$, we denote by x_α the subvector of x indexed by α. For a matrix $P \in \mathbb{R}^{n_1 \times n_2}$, we denote by P_i the i-th column of P. Given a positive integer n, let $[n] := \{1, \ldots, n\}$. For a finite set $\mathcal{S}$, we denote by $|\mathcal{S}|$ its cardinality and by $\operatorname{conv}(\mathcal{S})$ its convex hull. For subsets $\mathcal{S}_1$ and $\mathcal{S}_2$, we denote the Minkowski sum of $\mathcal{S}_1$ and $\mathcal{S}_2$ by $\mathcal{S}_1 + \mathcal{S}_2 := \{x + y : x \in \mathcal{S}_1,\ y \in \mathcal{S}_2\}$. For a condition A, $\mathbb{1}_{\{A\}}$ equals one if A holds and zero otherwise. Given finitely many functions $g_1, \ldots, g_k : \mathbb{R} \to \mathbb{R}$, we denote the functional space $\operatorname{span}\{g_1, \ldots, g_2\} := \left\{\sum_{i=1}^k \lambda_i g_i :\ \lambda_i \in \mathbb{R},\ i \in [k]\right\}$.

Throughout the paper, $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ denotes an undirected tree whose vertices are indexed so that $\mathcal{V} = [n]$. Unless otherwise stated, we root $\mathcal{T}$ at a fixed vertex $r \in \mathcal{V}$, which also induces the topological order of $\mathcal{T}$. For each $v \in \mathcal{V} \setminus \{r\}$, we denote its parent by $\mathsf{par}(v)$ and its parent edge by $e_v = (\mathsf{par}(v), v)$. Following this and for convenience, we rewrite $z_v = z_{e_v}$ and $c_v = c_{e_v}$ for $v \in \mathcal{V} \setminus \{r\}$ and $c_r = 0$. We let $\mathsf{ch}(v)$ be the set of children of v, and write $\mathcal{T}_v = (\mathcal{V}_v, \mathcal{E}_v)$ for the subtree rooted at v, where $\mathcal{V}_v$ consists of v and all its descendants. In particular, $\mathcal{T}_r = \mathcal{T}$. For any vertex $v \in \mathcal{V}$, we denote by $\mathfrak{T}_v$ the set of subtrees obtained by deleting v from $\mathcal{T}$; moreover, we denote by $\mathcal{T}_v^{\uparrow} \in \mathfrak{T}_v$ the component containing r. We call $\mathcal{T}_v^{\uparrow}$ the parent component of v and other components in $\mathfrak{T}_v$ the child components of v. We call $\mathfrak{T}_v$ a $(n_0, n_1, \ldots, n_k)$-*decomposition* of $\mathcal{T}$ if $|\mathcal{T}_v^{\uparrow}| = n_0$ and the size of the child components are $n_1, \ldots, n_k$ respectively, where $|\mathfrak{T}_v| = k + 1$ if $v \ne r$. For simplicity, we adopt the convention that $n_0 = 0$ if $v = r$.

We write $\overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty, +\infty\}$ and, for $\ell, u \in \overline{\mathbb{R}}$ with $\ell \le u$, let $\mathcal{I} = \{t \in \mathbb{R} : \ell \le t \le u\}$. For convenience, we denote

$$\mathcal{M} = \left\{(x_1, x_2, z) \in \mathbb{R}^3 : \begin{array}{c} (x_1 - x_2)(1 - z) = 0, \\ x_1, x_2 \in \mathcal{I},\ z \in \{0, 1\} \end{array}\right\},$$

which captures the key coupling structure (1b). We denote by $\mathcal{D}_{\mathcal{T}}$ the feasible region of (1) associated with a tree $\mathcal{T}$. Following this notation, $\mathcal{D}_{\mathcal{T}_v}$ represents the feasible region associated with a rooted subtree $\mathcal{T}_v$ in the counterpart problem, that is,

$$\mathcal{D}_{\mathcal{T}_v} = \{(x, z) : (x_i, x_j, z_e) \in \mathcal{M},\ \forall e = (i, j) \in \mathcal{E}_v\}.$$

We denote by τ_{MIP} the optimal value of (1).

2. Preliminary

In this section, we introduce two preliminary components necessary for understanding the scope and mechanics of our study. We begin by examining how the topology of the underlying graph affects the theoretical tractability of problem (1). Specifically, we demonstrate that the problem is NP-hard on general graphs while admitting a simple dynamic programming algorithm on path graphs. This disparity highlights the theoretical significance of extending exact solvability to arbitrary trees. In the second part, we shift our focus to the continuous aspect of the formulation. We introduce weak Chebyshev spaces to formalize the necessary regularity requirements on the node functions and illustrate its broad applicability.

2.1. **Complexity boundary.** It has long been known that if the variables x_i are restricted to finitely many prescribed values, the resulting variant of problem (1) is NP-hard on general graphs [6]. Its proof relies on a reduction from the three-terminal cut problem. We adapt a similar reduction strategy to extend the complexity result to node functions f_i with a continuous domain. In particular, we prove that problem (1) is NP-hard on general graphs even under restricted convex quadratic costs.

Theorem 1. *If the tree in problem* (1) *is replaced by an arbitrary graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, *the resulting problem is NP-hard under*

$$c_e = 1,\ \forall e \in \mathcal{E} \quad \textit{and} \quad f_i(x) = a_i(x - b_i)^2,\ \forall i \in \mathcal{V},\ x \in \mathcal{I} = \mathbb{R},$$

where the graph $\mathcal{G}$ *and* $a_i, b_i \in \mathbb{Z}_+, \forall i \in \mathcal{V}$ *are given as problem data.*

Proof. We show that under the assumptions of the theorem, problem (1) includes as a special case the following unweighted multiterminal cut problem with three specified terminals, which is known to be NP-hard [13]: Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and three specified vertices or *terminals* $v_0, v_1, v_2 \in \mathcal{V}$, find a minimum-cardinality set $\mathcal{E}' \subseteq \mathcal{E}$ such that removing $\mathcal{E}'$ from $\mathcal{E}$ disconnects each terminal from the other two. Denote this minimum by τ_{cut}.

To build the reduction, we set $m := |\mathcal{E}|$ and

$$f_i(x) = \begin{cases} 2(m+1)(x-k)^2 & \text{if } i = v_k,\ k = 0, 1, 2, \\ 0 & \text{if } i \in \mathcal{V} \setminus \{v_0, v_1, v_2\}. \end{cases} \tag{2}$$

We aim to show $\tau_{\mathsf{cut}} = \tau_{\mathsf{MIP}}$.

We first prove $\tau_{\mathsf{MIP}} \le \tau_{\mathsf{cut}}$. For any multiterminal cut $\mathcal{E}' \subseteq \mathcal{E}$, let $\mathcal{V}_i$ be the vertex set of the component of $\mathcal{G} - \mathcal{E}'$ containing i, and set

$$\bar{x}_i = \sum_{k=0}^{2} k\, \mathbb{1}_{\{v_k \in \mathcal{V}_i\}},\ i \in \mathcal{V}, \quad \bar{z}_e = \mathbb{1}_{\{e \in \mathcal{E}'\}},\ e \in \mathcal{E}. \tag{3}$$

We cliam that such a $(\bar{x}, \bar{z})$ is feasible for (1). Indeed, any $e = (i,j) \notin \mathcal{E}'$ implies $\bar{z}_e = 0$ and $\mathcal{V}_i = \mathcal{V}_j$. The latter further implies $\bar{x}_i = \bar{x}_j$ and thus $(\bar{x}_i, \bar{x}_j, \bar{z}_e) \in \mathcal{M}$, which proves the feasibility of $(\bar{x}, \bar{z})$. Moreover, since the removal of $\mathcal{E}'$ separates the terminals, one has $\bar{x}_{v_k} = k$ for $k = 0, 1, 2$. It follows that the objective of (1) at $(\bar{x}, \bar{z})$ is

$$\sum_{i \in \mathcal{V}} f_i(\bar{x}_i) + \sum_{e \in \mathcal{E}} \bar{z}_e = |\mathcal{E}'|,$$

which yields $\tau_{\mathsf{MIP}} \le \tau_{\mathsf{cut}} \le m$.

We next prove the opposite direction $\tau_{\mathsf{MIP}} \ge \tau_{\mathsf{cut}}$. Assume $(x^\star, z^\star)$ is the optimal solution to (1). Define $\mathcal{E}^\star = \{e \in \mathcal{E} : z_e^\star = 1\}$. Then for any connected component $\tilde{\mathcal{V}} \subseteq \mathcal{V}$ obtained from the removal of $\mathcal{E}^\star$, the coupling constraints (1) imply $x_i^\star = x_j^\star$ for any $i, j \in \tilde{\mathcal{V}}$. To show $\mathcal{E}^\star$ is a valid multiterminal cut, we assume for contradiction that two terminals v_k and $v_{k'}$ belong to the same component with $k, k' \in \{0,1,2\}$ but $k \neq k'$. Hence, one has $x_{v_k}^\star = x_{v_{k'}}^\star =: \bar{t}$ for a certain $\bar{t} \in \mathbb{R}$ and

$$\begin{aligned} \tau_{\mathsf{MIP}} \ge& f_{v_k}(x_{v_k}^\star) + f_{v_{k'}}(x_{v_{k'}}^\star) = 2(m+1)\left[(\bar{t} - k)^2 + (\bar{t} - k')^2\right] \\ \ge& (m+1)(k - k')^2 \ge m + 1, \end{aligned} \tag{4}$$

which contradicts $\tau_{\mathsf{MIP}} \le m$. Thus, $\mathcal{E}^\star$ is a multiterminal cut, and

$$\tau_{\mathsf{cut}} \le |\mathcal{E}^\star| \le \sum_{i \in \mathcal{V}} f_i(x_i^\star) + \sum_{e \in \mathcal{E}} z_e^\star = \tau_{\mathsf{MIP}}. \tag{5}$$

Therefore, we conclude $\tau_{\mathsf{MIP}} = \tau_{\mathsf{cut}}$ and finish the proof. □

Next, we demonstrate that (1) becomes tractable when $\mathcal{T}$ is a path graph. Without loss of generality (WLOG), we assume $\mathcal{E} = \{e_i := (i-1, i) : 2 \le i \le n\}$. For any feasible solution (x, z) of (1), the zero-valued edges naturally form consecutive blocks. Conversely, given any $k \in [n]$ and breakpoints

$$\underline{b}_1 \le \bar{b}_1 < \underline{b}_2 \le \bar{b}_2 < \cdots < \underline{b}_k \le \bar{b}_k \tag{6}$$

with each $\underline{b}_i, \bar{b}_i \in [n]$, one can define blocks $B(\underline{b}_i, \bar{b}_i) \overset{\text{def}}{=} \{j \in [n] : \underline{b}_i < j \le \bar{b}_i\}$ alongside an associated vector $z \in \{0,1\}^{n-1}$ such that

$$z_j = \begin{cases} 0 & \text{if } j \in \cup_{i=1}^{k} B(\underline{b}_i, \bar{b}_i) \\ 1 & \text{otherwise.} \end{cases} \tag{7}$$

By construction, the coupling constraints (1b) enforce $x_j = x_{j'}$ for any $\underline{b}_i \le j, j' \le \bar{b}_i$, $\forall i \in [k]$. Thus, minimizing x over each block yields

$$\tau(\underline{b}_i, \bar{b}_i) \stackrel{\text{def}}{=} \min_{t \in \mathcal{I}} \sum_{j=\underline{b}_i}^{\bar{b}_i} f_j(t) - \sum_{j \in B(\underline{b}_i, \bar{b}_i)} c_j, \tag{8}$$

where the second sum deducts the fixed charges associated with the zero-valued edges within the block. Consequently, the minimum of (1) for such an arbitrary but fixed z is given by

$$\tau_{\mathsf{MIP}} = \sum_{e \in \mathcal{E}} c_e + \sum_{i=1}^{k} \tau(\underline{b}_i, \bar{b}_i). \tag{9}$$

Above discussion demonstrates that solving (1) amounts to minimizing (9) subject to all feasible specification of blocks (6). The latter problem can be solved recursively and formalized in Proposition 1 below. Let $\bar{\tau}_0 = 0$ and for $j \in [n]$, we define in a bottom-up manner

$$\bar{\tau}_j = \min_{1 \le i \le j \le n} \left\{ \bar{\tau}_{i-1} + \tau(i, j) \right\}. \tag{10}$$

Proposition 1. *Assume $\mathcal{T}$ is a path graph. If problem (8) defining $\tau(i,j)$ can attain its minimum and can be solved in $\mathsf{T}_{\text{block}}$ time for all $1 \le i < j \le n$, then the optimal value of (1) is exactly $\bar{\tau}_n + \sum_{e \in \mathcal{E}} c_e$, which can be computed in $\mathcal{O}\left(n^2 \cdot \mathsf{T}_{\text{block}}\right)$ time.*

Proof. By (10), computing $\bar{\tau}_n$ amounts to computing $n(n+1)/2$ values τ_{ij} for all $1 \le i \le j \le n$, which costs $\mathcal{O}\left(n^2 \cdot \mathsf{T}_{\text{block}}\right)$ in total. Thus, it suffices to prove the first conclusion by induction on n. In the base case where $n = 1$, one has $\bar{\tau}_1 = \bar{\tau}_0 + \tau(1,1) = \min_{x_1 \in \mathcal{I}} f_1(x)$ and the conclusion holds trivially.

Assume the conclusion holds for any path with size less than n. For simplicity, we define a dummy variable $z_1 = 1$. For each $s \in [n]$, we denote by $\tau_s^\star$ the optimal objective value of (1) conditioning on $z_s = 1$ and $z_i = 0$ for all $i > s$. Then under such conditioning, the problem (1) is decomposed into two subproblems: one defined over a smaller path $\{1, \dots, s-1\}$, the other defined over the remaining nodes $\{s, \ldots, n\}$ with $x_s = \cdots = x_n$. Thus,

$$\begin{aligned}
\tau_s^\star =& \min\left\{\sum_{i=1}^{s-1} f_i(x_i) + \sum_{i=2}^{s-1} c_i z_i : (x_i, x_{i+1}, z_{i+1}) \in \mathcal{M}, \forall i \in [s-2]\right\} \\
&+ c_s + \min\left\{\sum_{i=s}^{n} f_i(t) : t \in \mathcal{I}\right\} \\
=& \left(\bar{\tau}_{s-1} + \sum_{i=2}^{s-1} c_i\right) + c_s + \left(\tau(s,n) + \sum_{i=s+1}^{n} c_i\right) \\
=& \bar{\tau}_{s-1} + \tau(s,n) + \sum_{e\in\mathcal{E}} c_e,
\end{aligned}$$

where the second equality is due to our induction hypothesis and the definition of $\tau(s,n)$ in (8). Minimizing over all choices $s \in [n]$ gives

$$\tau_{\mathsf{MIP}} = \min_{1\le s\le n} \{\bar{\tau}_{s-1} + \tau(s,n)\} + \sum_{e\in\mathcal{E}} c_e = \bar{\tau}_n + \sum_{e\in\mathcal{E}} c_e.$$

This completes the proof. □

2.2. **Weak Chebyshev spaces.** In this subsection, we specify the requirements on node functions $\{f_i\}_{i\in\mathcal{V}}$, which relies on the notion of weak Chebyshev space. Chebyshev spaces originate in approximation theory, where bounds on the zeros of a function underlie classical results on interpolation and approximation [20, 25]. The weak notion replaces the zero-count condition by a bound on strict sign changes [15]. For a nonzero continuous function g on an interval $\mathcal{J}$, let $\mathsf{sc}_{\mathcal{J}}(g)$ be the maximal integer $k \ge 0$ for which there exist $t_0 < \cdots < t_k$ in J such that

$$g(t_{j-1})g(t_j) < 0, \qquad j = 1, \ldots, k.$$

Definition 1 (Weak Chebyshev space)**.** Given an interval $\mathcal{J} \subseteq \mathbb{R}$, we call a finite-dimension functional space $\mathbb{U}$ of continuous functions a *weak Chebyshev space* with degree d on $\mathcal{J}$ if $\mathsf{sc}_J(g) \le d$ for every $g \in \mathbb{U}$.

We now impose this structure on the node costs.

Assumption 1 (Weak Chebyshev regularity)**.** The functional space $\bar{\mathbb{U}} \overset{\text{def}}{=} \text{span}\{1, f_1, \ldots, f_n\}$ is a weak Chebyshev space with degree d on $\mathcal{I}$ and admits a basis $\boldsymbol{\phi} = (\phi_0, \phi_1, \ldots, \phi_\nu)^\top$, where each $\phi_i : \mathcal{J} \to \mathbb{R}$ is a function and $\phi_0 \equiv 1$.

Weak Chebyshev spaces include many standard function classes. The most prominent instance is polynomial space $\mathbb{P}_d \overset{\text{def}}{=} \text{span}\{1, t, \ldots, t^d\}$. Indeed, since any polynomial $g \in \mathbb{P}_d$ has at most d roots, one has $\mathsf{sc}_J(g) \le d$ and thus, $\mathbb{P}_d$ is a weak Chebyshev space. Besides polynomial spaces, given fixed knots $b_1 < \cdots < b_K$, the space of continuous piecewise polynomials of degree at most p is also a weak Chebyshev space. To see this, this space has the truncated-power basis

$$\{1, t, \ldots, t^p\} \ \cup \ \big\{(t - b_k)_+^j : k = 1, \ldots, K, \ j = 1, \ldots, p\big\},$$

where $(a)_+ := \max\{a, 0\}$, which is a weak Chebyshev space with degree $(K+1)p$. In particular, it includes piecewise-linear and piecewise-quadratic functions such as hinge, absolute-deviation, quantile, and Huber losses when their breakpoints belong to the prescribed knot set.

The degree d quantifies how frequently a member $g \in \mathbb{U}$ can reverse its sign and plays a central role in the complexity analysis presented in subsequent sections. From this perspective, Assumption 1 can be viewed as a regularity condition that excludes functions with arbitrarily frequent oscillations. Thus, it maintains strong modeling power and encompasses a broad spectrum of commonly used elementary functions beyond standarad and piecewise polynomials. For example, $\operatorname{span}\{1, e^t, t\log(t), \sin(t)\}$ can be readily verified to be a weak Chebyshev space over any fixed compact interval in $\mathbb{R}_{++}$. We also note that the subspace of a weak Chebyshev space remains weak Chebyshev.

To isolate the algebraic details of specific function classes and focus on the essential discrete structure of the problem, we assume that univariate minimization problems are solvable and the changing points of a function $g \in \bar{\mathbb{U}}$ is computable, which we formalize below.

Assumption 2 (Computable univariate operations)**.** For every $g \in \bar{\mathbb{U}}$, there exists an oracle such that

- for any closed $\mathcal{I}' \subseteq \mathcal{I}$, one can solve the univariate minimization $\min\limits_{t \in \mathcal{I}'} g(t)$;
- one can compute the *changing points* of g, that is, the set of $k = \mathsf{sc}_{\mathcal{I}}(g)$ points $\ell < b_1 < \cdots < b_k < u$ such that $g(x)g(y) \geq 0$ for all $x, y \in [b_{i-1}, b_i]$ and $i \in [k+1]$.

The worst-case oracle time for each operation is denoted by TO.

Assumption 2 is practically realizable. For example, when $\bar{\mathbb{U}}$ consists of all polynomials with degree up to d, both the minimization and the changing-point problems reduce to finding the roots of a univariate polynomial. These roots can be computed by evaluating the eigenvalues of the associated companion matrix in polynomial time; we refer readers to [5] and [16] for more details. The parameter TO allows the subsequent complexity bounds to apply uniformly to other function classes for which the same two operations are available. Therefore, the computational model introduced in Assumption 2 separates the complexity of solving the continuous subproblems from the overall combinatorial complexity. Unless stated otherwise, Assumptions 1 and 2 are maintained throughout the paper.

3. Parametric Dynamic Programming and Lifted Representation

In this section, we develop the main algorithmic framework for solving problem (1) on arbitrary trees. We first construct an exact parametric dynamic programming algorithm and prove its correctness. We then derive an equivalent formulation of the DP recursion in an extended space, which provides the mathematical foundation for the complexity analysis conducted in the subsequent sections.

3.1. A parametric dynamic programming algorithm.

We begin by defining the value functions required for our recursive algorithm. For any tree $\mathcal{T}$ rooted at r and any $t \in \mathcal{I}$, consider the value function

$$\Phi_{\mathcal{T}}(t) \overset{\text{def}}{=} \min_{x,z} \left\{ \sum_{i \in \mathcal{V}} f_i(x_i) + \sum_{e \in \mathcal{E}} c_e z_e : \ x_r = t, \ (x, z) \in \mathcal{D}_{\mathcal{T}} \right\}. \tag{11}$$

One can similarly define $\Phi_{\mathcal{T}_v}$ by assigning value t to the root of $\mathcal{T}_v$. Define

$$\beta_v \overset{\text{def}}{=} c_v + \min_{t \in \mathcal{I}} \Phi_{\mathcal{T}_v}(t) \quad \forall v \in \mathcal{V}. \tag{12}$$

The next proposition demonstrates how to compute these value functions recursively from the leaves to the root.

Proposition 2. *For every $v \in \mathcal{V}$ and $t \in \mathcal{I}$,*

$$\Phi_{\mathcal{T}_v}(t) = f_v(t) + \sum_{w \in \mathsf{ch}(v)} \min \left\{ \Phi_{\mathcal{T}_w}(t), \beta_w \right\}. \tag{13}$$

Moreover, $\tau_{\mathsf{MIP}} = \min_{t \in \mathcal{I}} \Phi_{\mathcal{T}}(t)$.

Proof. Fix $v \in \mathcal{V}$ and $t \in \mathcal{I}$. Once $x_v = t$ is fixed, the problems associated with each $\Phi_{\mathcal{T}_v}(t)$ for all $w \in \mathsf{ch}(v)$ are mutually independent, which implies

$$\Phi_{\mathcal{T}}(t) = f_v(t) + \sum_{w \in \mathsf{ch}(v)} \underbrace{\min_{x_w, z_w} \left\{ \Phi_{\mathcal{T}_w}(x_w) + c_w z_w : \ (t, x_w, z_w) \in \mathcal{M} \right\}}_{\overset{\text{def}}{=} A_w}.$$

For a child $w \in \mathsf{ch}(v)$, if $z_w = 0$, then $x_w = t$ and $A_w = \Phi_{\mathcal{T}_w}(t)$. On the other hand, if $z_w = 1$, then x_w and $x_v = t$ are decoupled. In this case, x_w can take any value in $\mathcal{I}$, implying $A_w = c_w + \min_{x_w} \Phi_{\mathcal{T}_w}(x_w) = \beta_w$ in this case. Minimizing these two alternatives independently for all $w \in \mathsf{ch}(v)$ yields (13). $\square$

Functional recursion (13) can introduce points at which the member of $\bar{\mathbb{U}}$ representing the value function changes. A continuous function $\psi : \mathcal{I} \to \mathbb{R}$ is *piecewise* $\bar{\mathbb{U}}$ if there exist an integer $s \geq 1$, ordered breakpoints

$$\ell = b_0 < b_1 < \cdots < b_s = u, \qquad b_0, \ldots, b_s \in \overline{\mathbb{R}},$$

and functions $g_1, \ldots, g_s \in \bar{\mathbb{U}}$ such that $\psi(t) = g_j(t),\ \forall t \in (b_{j-1}, b_j)$.

Assume $\psi : \mathcal{I} \to \mathbb{R}$ is a piecewise $\bar{\mathbb{U}}$ function with s pieces. Then we can represent ψ by a list of endpoint–function triples

$$\mathsf{LIST}\,(\psi) \overset{\text{def}}{=} \{(b_{j-1}, b_j, g_j)\}_{j=1}^{s}\,, \tag{14}$$

where $\ell = b_0 < b_1 < \cdots < b_s = u$, $g_j \in \bar{\mathbb{U}}$ for each $j \in [s]$ and

$$\psi(t) = g_j(t),\ \forall t \in [b_{j-1}, b_j],\ j \in [s]. \tag{15}$$

In particular, for a scalar $a \in \mathbb{R}$, it is understood that $\mathsf{LIST}\,(a) = \{(\ell, u, a)\}$, where we treat a as a constant function.

Assume $\psi_1, \psi_2 : \mathcal{I} \to \mathbb{R}$ are two piecewise $\bar{\mathbb{U}}$ functions. We now discuss how to obtain an triple-list representation of $\psi_1 + \psi_2$ and $\min\{\psi_1, \psi_2\}$. For convenience, we introduce two basic operations ”+” and ”$\vee$” on triple lists. More specifically, given the representation of ψ_1 and ψ_2

$$\mathsf{LIST}\,(\psi_1) \overset{\text{def}}{=} \left\{(\bar{b}^1_{j-1}, \bar{b}^1_j, g^1_j)\right\}_{j=1}^{\bar{s}_1}, \quad \mathsf{LIST}\,(\psi_2) \overset{\text{def}}{=} \left\{(\bar{b}^2_{j-1}, \bar{b}^2_j, g^2_j)\right\}_{j=1}^{\bar{s}_2},$$

let $\ell = b_0 < b_1 < \cdots < b_K = u$ be the ordered distinct elements of $\{\bar{b}^1_j\}_{j=0}^{\bar{s}_1} \cup \{\bar{b}^2_j\}_{j=0}^{\bar{s}_2}$. For $i = 1, 2$ and $k \in [K]$, let $\bar{g}^i_k \in \bar{\mathbb{U}}$ be the member function carried by $\mathsf{LIST}\,(\psi_i)$ on (b_{k-1}, b_k), that is, $\bar{g}^i_k(t) = \psi(t),\ \forall t \in (b_{k-1}, b_k)$. Then we define

- <u>Summation</u>:
$$\mathsf{LIST}\,(\psi_1) + \mathsf{LIST}\,(\psi_2) \overset{\text{def}}{=} \left\{(b_{k-1}, b_k, \bar{g}^1_k + \bar{g}^2_k)\right\}_{k=1}^{K}. \tag{16}$$
- <u>Join</u>: for every $k \in [K]$, compute the changing points of $\bar{g}^1_k - \bar{g}^2_k$ on $[b_{k-1}, b_k]$ and write
$$b_{k-1} = b_{k,0} < b_{k,1} < \cdots < b_{k,s_k} = b_k \qquad (s_k \leq d). \tag{17}$$
For each $j \in [s_k]$, choose $g_{k,j} \in \{\bar{g}^1_k, \bar{g}^2_k\}$ such that
$$g_{k,j}(t) = \min\{\bar{g}^1_k(t), \bar{g}^2_k(t)\}, \qquad t \in [b_{k,j-1}, b_{k,j}].$$
Then $g_{k,j} \in \bar{\mathbb{U}}$. The join of $\mathsf{LIST}\,(\psi_1)$ and $\mathsf{LIST}\,(\psi_2)$ is defined as
$$\mathsf{LIST}\,(\psi_1) \vee \mathsf{LIST}\,(\psi_2) \overset{\text{def}}{=} \{(b_{k,j-1}, b_{k,j}, g_{k,j})\}_{k\in[K], j\in[s_k]}\,. \tag{18}$$

By construction, $\mathsf{LIST}\,(\psi_1) + \mathsf{LIST}\,(\psi_2)$ and $\mathsf{LIST}\,(\psi_1) \vee \mathsf{LIST}\,(\psi_2)$ are indeed valid triple-list representations for $\psi_1 + \psi_2$ and $\min\{\psi_1, \psi_2\}$, respectively. The list-sum operation can be extended to any finite number of piecewise functions directly.

With the sum and join operations defined for the triple-list representations, we formally present the parametric dynamic programming procedure in Algorithm 1. The algorithm traverses the tree in a bottom-up manner, iteratively applying the sum and join operations to construct the value functions and ultimately returning the global minimum of problem (1).

Algorithm 1 Parametric dynamic programming on a rooted tree

Input: A tree $\mathcal{T} = (\mathcal{V}, \mathcal{E})$, root r, an interval $\mathcal{I} = [\ell, u]$, functions $(f_v)_{v\in\mathcal{V}}$, and charges $(c_e)_{e\in\mathcal{E}}$

Output: The optimal value of (1)

1: **for** each leaf v **do**

$$\mathsf{LIST}\,(\Phi_{\mathcal{T}_v}) \leftarrow \{(\ell, u, f_v)\},\ \tau_v \leftarrow \min_{t\in\mathcal{I}} f_v(t),\ \beta_v \leftarrow c_v + \tau_v$$

2: **end for**

3: **for** each nonleaf vertex $v \in \mathcal{V}$ in postorder **do**

4:     **for** each $w \in \mathsf{ch}(v)$ **do**

$$\mathsf{LIST}\,(\min\{\Phi_{\mathcal{T}_w}, \beta_w\}) \leftarrow \mathsf{LIST}\,(\Phi_{\mathcal{T}_w}) \vee \{(\ell, u, \beta_w)\}$$

5:     **end for**

6:     Obtain

$$\mathsf{LIST}\,(\Phi_{\mathcal{T}_v}) \leftarrow \{(\ell, u, f_v)\} + \sum_{w\in\mathsf{ch}(v)} \mathsf{LIST}\,(\min\{\Phi_{\mathcal{T}_w}, \beta_w\})$$

7:     Solve the continuous programs on the pieces of $\mathsf{LIST}\,(\Phi_{\mathcal{T}_v})$ and set

$$\tau_v \leftarrow \min_{(a,b,g)\in\mathsf{LIST}(\Phi_{\mathcal{T}_v})} \min_{t\in[a,b]} g(t), \qquad \beta_v \leftarrow c_v + \tau_v$$

8: **end for**

9: **return** $\tau_{\mathsf{MIP}} \leftarrow \tau_r$

Proposition 3. *Under Assumption 2, Algorithm 1 can be implemented in*

$$\mathcal{O}\left((\mathsf{T} + d) \sum_{v\in\mathcal{V}} |\,\mathsf{LIST}\,(\Phi_{\mathcal{T}_v})\,|\right) \tag{19}$$

time.

Proof. Note that the running cost of Algorithm 1 primarily consists of three parts, that is, the costs of join operations $T_\vee$ incurred in Line 1 and Line 4, the list-sum operations T_+ incurred in Line 6, and solving continuous minimizations $T_{\min}$ in Line 7. Next, we estimate $T_\vee$, T_+ and $T_{\min}$ separately.

For any $v \in \mathcal{V} \setminus \{r\}$, in view of (17) and Line 4, each triple (a, b, g) in $\mathsf{LIST}\,(\Phi_{\mathcal{T}_v})$ requires one computation of the changing points of $g - \beta_v$ on

(a,b), which costs T and produces at most $d+1$ new triples. Hence,

$$T_{\vee} = \mathcal{O}\left([\mathsf{T} + (d+1)] \sum_{v\in\mathcal{V}\setminus\{r\}} |\,\mathsf{LIST}\,(\Phi_{\mathcal{T}_v})\,|\right). \tag{20}$$

Since each $\mathsf{LIST}\,(\min\{\Phi_{\mathcal{T}_w}, \beta_w\})$ consists of $(d+1)$ triples, Line 6 requires merging them together as in (16), which leads to

$$T_{+} = \mathcal{O}\left((d+1) \sum_{v\in\mathcal{V}\setminus\{r\}} |\,\mathsf{LIST}\,(\Phi_{\mathcal{T}_v})\,|\right).$$

Finally, in view of 7, every triple in $\mathsf{LIST}\,(\Phi_{\mathcal{T}_v})$ gives one continuous minimization. Thus,

$$T_{\min} = \mathcal{O}\left(\sum_{v\in\mathcal{V}} |\,\mathsf{LIST}\,(\Phi_{\mathcal{T}_v})\,| \cdot \mathsf{T}\right). \tag{21}$$

The oracle complexity $\mathcal{O}\left((\mathsf{T}+d)\sum_{v\in\mathcal{V}} |\Phi_{\mathcal{T}_v}|\right)$ is given by combining the three bounds. $\square$

Given a specific triple-list representation $\mathsf{LIST}\,(\psi)$ of ψ in (14), if $g_j(t) = g_{j+1}(t)$ for all $t \in (b_j, b_{j+1})$, then one can replace two pieces (b_{j-1}, b_j, g_j) and (b_j, b_{j+1}, g_{j+1}) by a merged piece (b_{j-1}, b_{j+1}, g_j). Moreover, testing whether g_j and g_{j+1} are identical over an interval (b_j, b_{j+1}) can be accomplished by testing if (b_j, b_{j+1}) contains a changing point of $g_j - g_{j+1}$, which is implementable. For convenience, for any piecewise $\bar{\mathbb{U}}$ function ψ, we always treat $\mathsf{LIST}\,(\psi)$ as a minimal triple-list representation of ψ in the rest of this paper; otherwise, one can always get a minimal triple-list representation by sequentially applying such merging operations. This motivates us to introduce *piece complexity* $\mathsf{pc}\,(\psi)$, which is defined as the minimum size of valid triple-list representation of ψ. By above discussion, we can resort the complexity analysis of Algorithm 1 to the analysis of piece complexity of $\Phi_{\mathcal{T}}$. If $\mathsf{pc}\,(\Phi_{\mathcal{T}})$ is polynomial in problem dimensions n, then Algorithm 1 is also a polynomial oracle algorithm.

We conclude this section with the calculus rules for upper bounding the piece complexity.

Proposition 4 (Piece calculus)**.** *Let $g_1, \ldots, g_k$ be piecewise $\bar{\mathbb{U}}$ functions. Then $\sum_{i=1}^k g_i$ and $\min\{g_1, g_2\}$ are also piecewise $\bar{\mathbb{U}}$ functions, and*

$$\mathsf{pc}\left(\sum_{i=1}^k g_i\right) \leq 1 + \sum_{i=1}^k (\mathsf{pc}\,(g_i) - 1),$$
$$\mathsf{pc}\,(\min\{g_1, g_2\}) \leq (d+1)(\mathsf{pc}\,(g_1) + \mathsf{pc}\,(g_2) - 1).$$

In particular, $\Phi_{\mathcal{T}_v}$ is a piecewise $\bar{\mathbb{U}}$ function for all $v \in \mathcal{V}$.

Proof. Given minimal triple-list representations of g_i's, one can compute a valid triple-list representation of $\sum_{i=1}^k g_i$ and $\min\{g_1, g_2\}$ by sum- and join-list operations, respectively. The size of the resulting triple-list can be bounded directly by using (16)-(18), which concludes the proof. $\square$

3.2. **Linearized representation in a lifted space.** We now develop an alternative representation of the value functions $\Phi_{\mathcal{T}_v}$ to facilitate our complexity analysis. Assume

$$\boldsymbol{\phi}(t) \overset{\text{def}}{=} (\phi_0(t), \ldots, \phi_\nu(t))^\top$$

is the basis of the Chebyshev space $\mathbb{U}_d$, where $\phi_0(t) \equiv 1$. Then for each $i \in \mathcal{V}$, there exists a vector q^i such that $f_i(t) = \boldsymbol{\phi}(t)^\top q^i$. Let

$$e^0 \overset{\text{def}}{=} (1, 0, \ldots, 0)^\top \in \mathbb{R}^{\nu+1} \quad \text{and} \quad \bar{q}^i \overset{\text{def}}{=} \beta_i e^0, \quad i \in \mathcal{V} \setminus \{r\}. \tag{22}$$

For every $v \in \mathcal{V}$, define the finite coefficient set recursively in a postorder by

$$\mathcal{Q}_v \overset{\text{def}}{=} \{q^v\} + \sum_{w \in \mathsf{ch}(v)} \big(\mathcal{Q}_w \cup \{\bar{q}^w\}\big). \tag{23}$$

For a leaf v, the sum is empty and hence $\mathcal{Q}_v = \{q^v\}$.

Next, we express the value function as a linear minimization over these recursive coefficient sets.

Lemma 1 (Linearization)**.** *For every $v \in \mathcal{V}$ and $t \in \mathcal{I}$,*

$$\Phi_{\mathcal{T}_v}(t) = \min_{q \in \mathcal{Q}_v} \boldsymbol{\phi}(t)^\top q. \tag{24}$$

Proof. We prove (24) by induction. In the base case where v is a leaf, one has $\mathcal{Q}_v = \{q^v\}$ and $\Phi_{\mathcal{T}_v}(t) = f_v(t) = \boldsymbol{\phi}(t)^\top q^v$. Suppose the claim holds for all $\Phi_{\mathcal{T}_w}$ such that $w \in \mathsf{ch}(v)$. Substitution in (13) gives

$$\begin{aligned}\Phi_{\mathcal{T}_v}(t) &= \boldsymbol{\phi}(t)^\top q^v + \sum_{w \in \mathsf{ch}(v)} \min\left\{\min_{q \in \mathcal{Q}_w} \boldsymbol{\phi}(t)^\top q, \boldsymbol{\phi}(t)^\top \bar{q}^w\right\} \\ &= \boldsymbol{\phi}(t)^\top q^v + \sum_{w \in \mathsf{ch}(v)} \min_{q \in \mathcal{Q}_w \cup \{\bar{q}^w\}} \boldsymbol{\phi}(t)^\top q.\end{aligned}$$

Because for any two sets $\mathcal{S}_1$ and $\mathcal{S}_2$ and any vector a, one always has $\min_{x \in \mathcal{S}_1} a^\top x + \min_{y \in \mathcal{S}_2} a^\top y = \min_{x \in \mathcal{S}_1 + \mathcal{S}_2} a^\top z$, one can deduce that

$$\Phi_{\mathcal{T}_v}(t) = \min\left\{\boldsymbol{\phi}(t)^\top q : q \in \{a^v\} + \sum_{w \in \mathsf{ch}(v)} (\mathcal{Q}_w \cup \{\bar{q}^w\})\right\} = \min_{q \in \mathcal{Q}_v} \boldsymbol{\phi}(t)^\top q.$$

This completes the proof. $\square$

For $v \in \mathcal{V}$, define the rooted choice polytope

$$\bar{\Lambda}_v \overset{\text{def}}{=} \left\{\lambda \in \mathbb{R}^{\mathcal{V}_v} : \lambda_v = 1,\ 0 \le \lambda_i \le \lambda_{\mathsf{par}(i)},\ \forall i \in \mathcal{V}_v\{v\}\right\} \tag{25}$$

and its binary counterpart by

$$\Lambda_v \overset{\text{def}}{=} \bar{\Lambda}_v \cap \{0,1\}^{\mathcal{V}_v}.$$

Using disjunctive programming techniques, we can describe the convex hull of $\mathcal{Q}_v$ in a lifted space.

Lemma 2 (Convexification)**.** *For every $v \in \mathcal{V}$,*

$$\operatorname{conv}(\mathcal{Q}_v) = \left\{ q \in \mathbb{R}^{\nu+1} : q = \sum_{i \in \mathcal{V}_v} \lambda_i q^i + \sum_{i \in \mathcal{V}_v \setminus \{v\}} (\lambda_{\mathsf{par}(i)} - \lambda_i) \bar{q}^i,\ \lambda \in \bar{\Lambda}_v \right\}. \tag{26}$$

Proof. We prove the conclusion by induction. In the base case where v is a leaf node, $\mathcal{Q}_v = \{q^v\}$ is a singleton, implying (26) holds trivially. Suppose (26) holds for any $w \in \mathsf{ch}(v)$. Applying Conforti et al. [12, Theorem 4.39] to the disjunction between $\mathcal{P}_w$ and $\{\bar{q}^w\}$, one has for each $w \in \mathsf{ch}(v)$ that

$$\operatorname{conv}\left(\mathcal{Q}_w \cup \{\bar{q}^w\}\right) = \left\{ q \in \mathbb{R}^{\nu+1} : \begin{array}{c} q = p + (1-\alpha_w)\bar{q}^w,\ 0 \le \alpha_w \le 1, \\ p = \displaystyle\sum_{i \in \mathcal{V}_w} \lambda_i q^i + \sum_{i \in \mathcal{V}_w \setminus \{w\}} (\lambda_{\mathsf{par}(i)} - \lambda_i)\bar{q}^i \\ \lambda_w = \alpha_w,\ 0 \le \lambda_i \le \lambda_{\mathsf{par}(i)},\ \forall i \in \mathcal{V}_w \setminus \{w\} \end{array} \right\}$$

$$= \left\{ q \in \mathbb{R}^{\nu+1} : \begin{array}{c} q = \displaystyle\sum_{i \in \mathcal{V}_w} \lambda_i q^i + \sum_{i \in \mathcal{V}_w} (\lambda_{\mathsf{par}(i)} - \lambda_i)\bar{q}^i \\ 0 \le \alpha_w \le \lambda_v = 1,\ 0 \le \lambda_i \le \lambda_{\mathsf{par}(i)},\ \forall i \in \mathcal{V}_w \setminus \{w\} \end{array} \right\}, \tag{27}$$

where we substitute out p and α_w and introduce $\lambda_v = 1$. Since $\operatorname{conv}(S_1 + S_2) = \operatorname{conv}(S_1) + \operatorname{conv}(S_2)$ holds for any two sets S_1 and S_2, one has by the definition of $\mathcal{Q}_v$ in (23) that

$$\operatorname{conv}(\mathcal{Q}_v) = \{q^v\} + \sum_{w \in \mathsf{ch}(v)} \operatorname{conv}\left(\mathcal{Q}_w \cup \{\bar{q}^w\}\right).$$

The conclusion follows by substituting out each $\operatorname{conv}\left(\mathcal{Q}_w \cup \{\bar{q}^w\}\right)$ using (27). $\square$

For $v \in \mathcal{V}$, let $G^v \in \mathbb{R}^{(\nu+1)\times n_v}$ be the matrix for which each column $i \in \mathcal{V}_v$ is given by

$$G_i \overset{\text{def}}{=} q^i + \sum_{w \in \mathsf{ch}(i)} \bar{q}^w - \mathbb{1}_{\{i \ne v\}} \bar{q}^i.$$

Combining this convex hull description with our linear representation yields the following formulation for the value function.

Proposition 5. *For every $v \in \mathcal{V}$ and $t \in \mathcal{I}$,*

$$\Phi_{\mathcal{T}_v}(t) = \min_{\lambda \in \Lambda_v} \phi(t)^\top G^v \lambda. \tag{28}$$

Proof. Rearranging the two sums in (26) gives $\mathrm{conv}(\mathcal{Q}_v) = \{G^v \lambda : \lambda \in \bar{\Lambda}_v\}$. Because the objective of (24) is linear, one can replace the feasible region $\mathcal{Q}_v$ with its convex hull and obtain

$$\Phi_{\mathcal{T}_v}(t) = \min_{q \in \mathrm{conv}(\mathcal{Q}_v)} \phi(t)^\top q = \min_{\lambda \in \bar{\Lambda}} \phi(t)^\top G^v \lambda.$$

Since the constraint matrix defining $\bar{\Lambda}_v$ is totally unimodular and its right-hand side is integral, the preceding minimization always admits an integral optimal solution. The conclusion follows. □

4. Complexity Analysis

Building upon the extended formulation established in the previous section, we now analyze the piece complexity of our dynamic programming algorithm.

4.1. **Power centroid.** To analyze the piece complexity of our dynamic programming recursion, we require a systematic tree balancing approach. Centroid decomposition is a classical node balancing technique in combinatorial optimization [24, 18]. More specifically, a vertex v is called a *centroid* of a tree $\mathcal{T}$ if each component in $\mathfrak{T}_v$ has no more than $|\mathcal{T}|/2$ nodes. Other variants such as the weighted centroid and the branch-weight centroid also exist in the literature; see Lin and Shang [29] and the references therein. In this subsection, we introduce a new variant termed the *power centroid*.

Definition 2 (Power centroid)**.** Let $A > C > 1$ and $\gamma > 1$. A vertex $v \in \mathcal{V}$ is an (A, C, γ)-power centroid of $\mathcal{T}$ if $\mathfrak{T}_v$ is a $(m, n_1, \dots, n_k)$-decomposition satisfying

$$Am^\gamma + C \sum_{j=1}^{k} n_j^\gamma \le n^\gamma. \tag{29}$$

The following lemma establishes the existence of a balanced node that satisfies specific component size bounds.

Lemma 3. *For a n-size tree and any number $\theta \in (0, 1/2]$, there exists a node v such that $\mathfrak{T}_v$ is a $(m, n_1, \dots, n_k)$-decomposition satisfying $m \le \theta n$ and $n_j \le (1-\theta)n$ for all $j \in [k]$.*

Proof. Starting from the root, repeatedly move to a child whose rooted subtree has more than $(1-\theta)n$ vertices, and stop when no such child exists. Such a stopping child v is unique because $1 - \theta > 1/2$. If v is not the root of the tree, then $\mathcal{T}_v$ has more than $(1-\theta)n$ vertices, implying its parent component has size $m \leq \theta n$. The conclusion holds trivially if v is the root. □

The following lemma establishes a general bound for sums of powers, which will be instrumental in our subsequent complexity analyses.

Lemma 4. *Let $k \in \mathbb{Z}_+$, $\gamma \geq 1$, and $U, S, n_1, \ldots, n_k \in \mathbb{R}_+$. If $\sum_{i=1}^k n_i \leq S$, then*

$$\sum_{i\in[k]} n_i^\gamma \leq S^\gamma. \tag{30}$$

If additionally $n_i \leq U$ for all $i \in [k]$ and $U \leq S \leq 2U$, then

$$\sum_{i\in[k]} n_i^\gamma \leq U^\gamma + (S-U)^\gamma. \tag{31}$$

Proof. Because (30) can be viewed as a special case of (31) by taking $U = S$, it is sufficient to prove only (31). Define $h(t) = t^\gamma$ and consider

$$\begin{aligned} \max_{t_1,\ldots,t_k} \quad & \sum_{i=1}^k h(t_i) \\ \text{s.t.} \quad & \sum_{i=1}^k t_i \leq S \\ & 0 \leq t_i \leq U \quad \forall i \in [k]. \end{aligned}$$

Since $h(\cdot)$ is convex, the maximum can be attained at an extreme point of the feasible region. Because $U \leq S \leq 2U$, at most two components t_i of such an optimal extreme point can attain the upper bound U. Consequently, the solution is determined by making one upper-bound constraint and the sum constraint tight. Therefore,

$$\sum_{i\in[k]} h(t_i) \leq h(U) + h(S-U).$$

The conclusion follows by specifying a feasible solution $t_i = x_i$, $\forall i \in [n]$. □

Building upon above auxiliary results, the following proposition characterize the precise conditions under which a power centroid is guaranteed to exist.

Proposition 6. *Let $A > C > 1$ be fixed. There exists an (A, C, γ)-centroid for any finite rooted tree if and only if there exists a $\theta \in (0, 1/2]$ such that $A\theta^\gamma + C(1-\theta)^\gamma \leq 1$.*

Proof. We first assume the inequality holds true. Let v, m, n, $n_1, \ldots, n_k$ be defined in Lemma 3. Consider the function $h(t) = At^\gamma + C(1-\theta)^\gamma + C(\theta - t)^\gamma$. Since $h(t)$ is a convex function, $\max_{t\in[0,\theta]} h(t)$ is attained at either the endpoint 0 or θ. Because $h(\theta)-h(0)=(A-C)\theta^\gamma>0$ and $m/n\in[0,\theta]$, one has that $h(m/n)\le h(\theta)=A\theta^\gamma+C(1-\theta)^\gamma\le 1$, which leads to

$$n^\gamma h(m/n)=Am^\gamma+C[(1-\theta)n]^\gamma+C\,(\theta n-m)^\gamma\le n^\gamma. \tag{32}$$

Moreover, because $0<\theta\le 1/2$, by setting $U=(1-\theta)n$ and $S=n-m$, one has $\sum_{i\in[k]} n_i\le S$, $U\le S\le 2U$, and $n_i\le U$ for all $i\in[k]$. Thus, Lemma 4 implies

$$\sum_{i=1}^{k} n_i^\gamma\le(\theta n-m)^\gamma+[(1-\theta)n]^\gamma. \tag{33}$$

Combining (32) and (33) yields (29), which shows v is a (A,C,γ)-centroid.

Next, we prove the other direction. Assume that every finite rooted tree has an (A,C,γ)-power centroid. For each $n\ge 2$, consider an n-vertex path rooted at an endpoint, and let v_n be a power centroid. If m_n is the size of the parent component of v_n, then the other component has size $n-1-m_n$, and therefore

$$A\left(\frac{m_n}{n}\right)^\gamma+C\left(1-\frac{1}{n}-\frac{m_n}{n}\right)^\gamma\le 1.$$

Since $m_n/n\le 1$, we assume the limit $t:=\lim_{n\to\infty} m_n/n\in[0,1]$ exists; otherwise we pass the arguments to one convergent subsequence of $\{m_n/n\}_n$. This gives

$$At^\gamma+C(1-t)^\gamma\le 1.$$

Since $A,C>1$, one has $t\in(0,1)$. If $t\le 1/2$, set $\theta=t$. Otherwise, set $\theta=1-t$ and observe that

$$A(1-t)^\gamma+Ct^\gamma<At^\gamma+C(1-t)^\gamma,$$

where the strict inequality follows from $A>C$ and $t>1/2$. In both cases, $\theta\in(0,1/2]$ and $A\theta^\gamma+C(1-\theta)^\gamma\le 1$. This completes the proof. □

Let $\gamma(A,C)\overset{\text{def}}{=}\min\{\gamma>1:\ A\theta^\gamma+C(1-\theta)^\gamma\le 1,\,\theta\in(0,1/2]\}$. Using KKT conditions, it is not difficult to prove that the optimal $\gamma(A,C)$ is given by the unique root of the equation

$$A^{-\frac{1}{\gamma-1}}+C^{-\frac{1}{\gamma-1}}=1. \tag{34}$$

4.2. **Fixed-degree complexity.** We now present a new recursion induced by a path. For any subtree $\mathcal{B}$ of $\mathcal{T}$, its root is uniquely determined by the orientation of $\mathcal{T}$ and is denoted by $r_{\mathcal{B}}$. In view of (28), we extend the definition of $\Phi_{\mathcal{T}_v}$ to a general subtree $\mathcal{B}$ of $\mathcal{T}$ as

$$\begin{aligned}\Phi_{\mathcal{B}}(t)=\min_{\lambda}\ &\sum_{i\in\mathcal{V}(\mathcal{B})}\lambda_i\phi(t)^\top G_i\\ \text{s.t. }&\lambda_{r_{\mathcal{B}}}=1,\ \lambda_i\in\{0,1\},\ \lambda_i\le\lambda_{\mathsf{par}(i)},\ \forall i\in\mathcal{V}(\mathcal{B})\setminus\{r_{\mathcal{B}}\}.\end{aligned} \tag{35}$$

Let $\mathcal{P}_s$ be the path from the root r of $\mathcal{T}$ to node $s \in \mathcal{V}$. Let $\mathfrak{B}_s$ be the family of components of $\mathcal{T} \setminus \mathcal{P}_s$. Root each branch $\mathcal{B} \in \mathfrak{B}_s$ at its unique vertex adjacent to $\mathcal{P}_s$, and let $a_{\mathcal{B}} \in \mathcal{P}_s$ be its attachment vertex. Accordingly, we partition $\mathfrak{B}_s = \mathfrak{B}_s^{\uparrow} \cup \mathfrak{B}_s^{\downarrow}$ by attachment location:

$$\mathfrak{B}_s^{\uparrow} \overset{\text{def}}{=} \{\mathcal{B} \in \mathfrak{B}_s : a_{\mathcal{B}} \in \mathcal{P}_s \setminus \{s\}\}, \quad \mathfrak{B}_s^{\downarrow} \overset{\text{def}}{=} \{\mathcal{B} \in \mathfrak{B}_s : a_{\mathcal{B}} = s\}. \tag{36}$$

For any subtree $\mathcal{B}$ of $\mathcal{T}$, define $\Phi_{\mathcal{B}}^{\mathrm{o}}(t) \overset{\text{def}}{=} \min\{0, \Phi_{\mathcal{B}}(t)\}$. We now prove a key decomposition lemma by conditioning on the binary choice along the root path.

Lemma 5 (Root-path disjunction)**.** *For every $s \in \mathcal{T}$,*

$$\Phi_{\mathcal{T}}(t) = \min\left\{\Phi_{\mathcal{T}_s^{\uparrow}}(t),\ \boldsymbol{\phi}(t)^{\top} \sum_{i \in \mathcal{P}_s} G_i + \sum_{\mathcal{B} \in \mathfrak{B}_s} \Phi_{\mathcal{B}}^{\mathrm{o}}(t)\right\}, \tag{37}$$

where the first entry is omitted when $\mathcal{T}_s^{\uparrow}$ is empty.

Proof. We prove (37) by conditioning on the binary value of λ_s in (28):

- If $\lambda_s = 0$, then the order constraints $\lambda_i \leq \lambda_{\mathsf{par}(i)}$ force every λ variable indexed by a descendant of s to zero. In this case, the value $\Phi_{\mathcal{T}}(t)$ reduces to $\Phi_{\mathcal{T}_s^{\uparrow}}(t)$.
- If $\lambda_s = 1$, then the ordering constraints $\lambda_i \leq \lambda_{\mathsf{par}(i)}$ gives $\lambda_i = 1$ for every $i \in \mathcal{P}_s$. After these variables are fixed, the remaining variables and objective coefficients decompose over the components in $\mathfrak{B}_s$. This results in the second term in (37).

The conclusion follows. ☐

Building on this structural decomposition, we can bound the resulting piece complexity growth as in the following lemma.

Lemma 6. *For every $s \in \mathcal{T}$,*

$$\mathsf{pc}\left(\Phi_{\mathcal{T}}\right) \leq (d+1)\left[\mathsf{pc}\left(\Phi_{\mathcal{T}_s^{\uparrow}}\right) + \sum_{\mathcal{B} \in \mathfrak{B}_s} \mathsf{pc}\left(\Phi_{\mathcal{B}}^{\mathrm{o}}\right) - |\mathfrak{B}_s|\right], \tag{38}$$

$$\mathsf{pc}\left(\Phi_{\mathcal{T}}^{\mathrm{o}}\right) \leq (d+1)\left[\mathsf{pc}\left(\Phi_{\mathcal{T}_s^{\uparrow}}^{\mathrm{o}}\right) + \sum_{\mathcal{B} \in \mathfrak{B}_s} \mathsf{pc}\left(\Phi_{\mathcal{B}}^{\mathrm{o}}\right) - |\mathfrak{B}_s|\right]. \tag{39}$$

Proof. The recursive bound (38) follows directly by applying two calculus rules in Proposition 4 to (37). To prove (39), we notice that Lemma 5 implies the simple fact

$$\Phi_{\mathcal{T}}^{\mathrm{o}}(t) = \min\left\{\Phi_{\mathcal{T}_s^{\uparrow}}^{\mathrm{o}}(t),\ \boldsymbol{\phi}(t)^{\top} \sum_{i \in \mathcal{P}_s} G_i + \sum_{\mathcal{B} \in \mathfrak{B}_s} \Phi_{\mathcal{B}}^{\mathrm{o}}(t)\right\}.$$

Then (39) follows also from Proposition 4. ☐

For any fixed $d \geq 1$, let $\gamma_d = \gamma(A, C)$ with specification $A = 2(d+1)$ and $C = d+1$. Combining the properties of the power centroid with our recursive piece bounds yields the main complexity theorem for fixed degrees.

Theorem 2. *Under Assumption 1, every rooted tree $\mathcal{T}$ with n nodes satisfies*

$$\mathsf{pc}\left(\Phi_{\mathcal{T}}\right) \leq (d+1)n^{\gamma_d}.$$

Proof. We prove by induction on the size of trees a stronger version that

$$\max\left\{\mathsf{pc}\left(\Phi_{\mathcal{T}}\right), \mathsf{pc}\left(\Phi_{\mathcal{T}}^{\mathrm{o}}\right)\right\} \leq (d+1)n^{\gamma_d}. \tag{40}$$

In the base case where $n = 1$, one trivially has (40) holds true since $\mathsf{pc}\left(\Phi_{\mathcal{T}}\right) = 1$ and $\mathsf{pc}\left(\Phi_{\mathcal{T}}^{\mathrm{o}}\right) = \mathsf{pc}\left(\min\left\{\Phi_{\mathcal{T}}, 0\right\}\right) \leq d+1$, where the inequality follows from Proposition 4.

Consider an arbitrary rooted tree with n vertices, and assume the claim holds for all smaller trees. By Proposition 6, $\mathcal{T}$ has a $(2(d+1), d+1, \gamma_d)$-power centroid s satisfying

$$2(d+1)m^{\gamma_d} + (d+1)\sum_{j=1}^{k} n_j^{\gamma_d} \leq n^{\gamma_d}, \tag{41}$$

where $m = |\mathcal{T}_s^{\uparrow}|$, $k = |\mathfrak{B}_s^{\downarrow}|$, and $n_1, \ldots, n_k$ are the sizes of branches in $\mathfrak{B}_s^{\downarrow}$. Because all nonempty trees in (37) have fewer than n vertices, one can deduce from Lemma 6 and our induction hypothesis that

$$\mathsf{pc}\left(\Phi_{\mathcal{T}}\right) \leq (d+1)^2 \left[m^{\gamma} + \sum_{\mathcal{B} \in \mathfrak{B}_s^{\uparrow}} |\mathcal{B}|^{\gamma_d} + \sum_{j=1}^{k} n_j^{\gamma_d} \right].$$

Moreover, since $\sum_{\mathcal{B} \in \mathfrak{B}_s^{\uparrow}} |\mathcal{B}| \leq m$, Lemma 4 implies $\sum_{\mathcal{B} \in \mathfrak{B}_s^{\uparrow}} |\mathcal{B}|^{\gamma_d} \leq m^{\gamma_d}$. Thus, one has

$$\mathsf{pc}\left(\Phi_{\mathcal{T}}\right) \leq (d+1)^2 \left(2m^{\gamma_d} + \sum_{j=1}^{k} n_j^{\gamma_d} \right) \leq (d+1)n^{\gamma_d},$$

where the last inequality is due to (41). Similarly, one can invoke (39) to prove $\mathsf{pc}\left(\Phi_{\mathcal{T}}^{\mathrm{o}}\right) \leq (d+1)n^{\gamma_d}$. This closes the induction. ☐

We now give an estimate of γ_d. Substituting $A = 2(d+1)$ and $C = d+1$ into (34) and factoring gives

$$(d+1)^{-\frac{1}{\gamma_d - 1}} \left(1 + 2^{-\frac{1}{\gamma_d - 1}}\right) = 1.$$

Taking logarithms and using $1 + e^{-t} = 2e^{-t/2}\cosh(t/2)$, one obtains

$$\log_2(d+1) = \frac{\gamma_d - 1}{\ln 2} \ln\left(1 + 2^{-\frac{1}{\gamma_d - 1}}\right) = \gamma_d - \frac{3}{2} + \frac{\gamma_d - 1}{\ln 2} \ln\cosh\left(\frac{\ln 2}{2(\gamma_d - 1)}\right).$$

Since $\ln\cosh(z) \geq 0$ for all $z \in \mathbb{R}$, it follows that

$$\gamma_d \leq \frac{3}{2} + \log_2(d+1).$$

Observe that each piecewise $\bar{\mathbb{U}}$-function can be embedded into a larger weak Chebyshev space. Although a straightforward complexity bound follows from applying Theorem 2 to this expanded space, Proposition 7 delivers a significantly sharper result.

Proposition 7. *If each node cost function f_i is a piecewise $\bar{\mathbb{U}}$-function with at most M pieces, then one has*

$$\mathsf{pc}\left(\Phi_{\mathcal{T}}\right) = \mathcal{O}\left(Mdn^{\gamma_d+1}\right).$$

Proof. Let $\mathcal{K}$ be the union of the knots of all node cost functions. Since every f_i has at most M pieces, one has $|\mathcal{K}| \leq n(M-1)$. The points in $\mathcal{K}$ partition the domain $\mathcal{I}$ into at most $n(M-1)+1$ intervals. On each such interval, Theorem 2 gives at most $(d+1)n^{\gamma_d}$ pieces for the restriction of $\Phi_{\mathcal{T}}$ to that interval. Concatenating these representations over all such intervals yields $\mathsf{pc}\left(\Phi_{\mathcal{T}}\right) \leq \big(n(M-1)+1\big)(d+1)n^{\gamma_d} = \mathcal{O}\left(Mdn^{\gamma_d+1}\right)$. □

5. Improved complexity for quadratic node costs

Throughout this section, we assume that every f_i is a quadratic polynomial on $\mathcal{I}$ and set the common basis for the lifted representation to $\boldsymbol{\phi}(t) = (1, t, t^2)^\top$. Under this quadratic setting, the generic piece complexity derived in Section 4 yields an exponent of roughly $\gamma_2 \approx 3.04$. We demonstrate that this bound can be sharpened significantly for quadratic objectives through a refined analysis and a specialized node balancing technique.

5.1. Active Quadratic Calculus.

In view of (35), each feasible binary λ defines a quadratic polynomial on $\mathcal{I}$. Hence, $\Phi_{\mathcal{B}}$ is a lower envelope of a finite family generated by the lifted formulation. For a general finite lower envelope g, define its *number of active quadratics* by

$$\mathsf{aq}\left(g\right) \stackrel{\text{def}}{=} \min\left\{ k \in \mathbb{Z}_+ : \begin{array}{ll} g_j : \mathcal{I} \to \mathbb{R} \text{ is a quadratic}, & \forall j \in [k] \\ g(t) = \min\limits_{j\in[k]} g_j(t), & \forall t \in \mathcal{I} \end{array} \right\},$$

We next present calculus rules for $\mathsf{aq}\left(\cdot\right)$.

Lemma 7. *If g is the lower envelope of finitely many quadratics, then*

$$\mathsf{aq}\left(g\right) \leq \mathsf{pc}\left(g\right) \leq 2\,\mathsf{aq}\left(g\right) - 1. \tag{42}$$

For any two such lower envelopes g and h,

$$\mathsf{aq}\left(\min\{g,h\}\right) \leq \mathsf{aq}\left(g\right) + \mathsf{aq}\left(h\right), \tag{43}$$

$$\mathsf{aq}\left(g+h\right) \leq 2(\mathsf{aq}\left(g\right) + \mathsf{aq}\left(h\right)) - 1 \tag{44}$$

Proof. We first note that (43) is trivial by definition, and (44) follows directly from (43). It remains to prove (42).

Let $m = \mathsf{aq}\,(g)$ and fix an irredundant representation $g = \min_{j\in[m]} g_j$. Define $\mathcal{R}_j = \{t \in \mathcal{I} : g_j(t) \le g_k(t),\ \forall k \in [m]\backslash\{j\}\}$, over which g_j agree with g. By construction, the set $\mathcal{R}_j$ has a nonempty interior and thus contains at least one interval piece of g; otherwise, g_j can be deleted from the expression of g. Consequently, one has $\mathsf{aq}\,(g) \le \mathsf{pc}\,(g)$.

Next we prove $\mathsf{pc}\,(g) \le 2\,\mathsf{aq}\,(g) - 1$ by induction. The base case where $m = 1$ is trivial. We assume the conclusion holds for all quadratic lower envelopes $\tilde{g}$ with $\mathsf{aq}\,(\tilde{g}) \le m - 1$ and consider the case $\mathsf{aq}\,(g) = m$. WLOG we suppose that each $g_j(t) = q_2^j t^2 + q_1^j t + q_0^j$ for certain $q_2^j, q_1^j, q_0^j \in \mathbb{R}$ with $q_2^j \le \cdots \le q_2^m$. Let $\tilde{g}(t) = \min_{j\in[m-1]} g_j(t)$. By adding function g_m to the lower envelop, the newly induced breakpoints of $g(t) = \min\{g_m(t), \tilde{g}(t)\}$ are given by the roots of $g_m(t) = \tilde{g}(t)$.

We claim that the equation $g_m(t) = \tilde{g}(t)$ has at most two roots. Indeed, the roots can only come from the boundary points of the set

$$
\begin{aligned}
\bar{\mathcal{I}} &\stackrel{\text{def}}{=} \{t \in \mathcal{I} :\ g_m(t) \le \tilde{g}(t)\} \\
&= \bigcap_{j\in[m-1]} \{t \in \mathcal{I} :\ g_m(t) \le g_j(t)\} \\
&= \bigcap_{j\in[m-1]} \left\{t \in \mathcal{I} :\ (q_2^m - q_2^j)t^2 + (q_1^m - q_1^j)t + (q_0^m - q_0^j) \le 0\right\}.
\end{aligned}
$$

Because $q_2^m \ge q_2^j,\ \forall j \in [m-1]$, $\bar{\mathcal{I}}$ is the intersection of $m-1$ convex sets in $\mathbb{R}$ and thus is an interval. This proves the claim. Consequently, $\mathsf{pc}\,(g) \le \mathsf{pc}\,(\tilde{g}) + 2$. By the induction hypothesis, one has $\mathsf{pc}\,(\tilde{g}) \le 2(\mathsf{aq}\,(\tilde{g})) - 1 = 2\,\mathsf{aq}\,(g) - 3$. It follows that $\mathsf{pc}\,(g) \le 2\,\mathsf{aq}\,(g) - 1$. □

Building upon the calculus rules in Lemma 7, we now establish a recursive bound for active quadratics resulting from a root-path disjunction.

Lemma 8. *For every $s \in \mathcal{T}$,*

$$
\mathsf{aq}\,(\Phi_{\mathcal{T}}) \le \mathsf{aq}\left(\Phi_{\mathcal{T}_s^{\uparrow}}\right) + 2\sum_{\mathcal{B}\in\mathfrak{B}_s} \mathsf{aq}\,(\Phi_{\mathcal{B}}) + 1. \tag{45}
$$

Proof. For every $\mathcal{B} \in \mathfrak{B}_s$, invoking Proposition 4 and Lemma 6, one gets

$$\begin{aligned}
\mathsf{aq}\left(\Phi_{\mathcal{T}}\right) &\leq \mathsf{aq}\left(\Phi_{\mathcal{T}_s^{\uparrow}}\right) + \mathsf{aq}\left(\phi(\cdot)^{\top} \sum_{i \in \mathcal{P}_s} G_i + \sum_{\mathcal{B} \in \mathfrak{B}_s} \Phi_{\mathcal{B}}^{\circ}\right) && \text{(by (43))} \\
&\leq \mathsf{aq}\left(\Phi_{\mathcal{T}_s^{\uparrow}}\right) + \mathsf{pc}\left(\sum_{\mathcal{B} \in \mathfrak{B}_s} \Phi_{\mathcal{B}}^{\circ}\right) && \text{(by (42))} \\
&\leq \mathsf{aq}\left(\Phi_{\mathcal{T}_s^{\uparrow}}\right) + \sum_{\mathcal{B} \in \mathfrak{B}_s} \left(\mathsf{pc}\left(\Phi_{\mathcal{B}}^{\circ}\right) - 1\right) + 1 && \text{(Proposition 4)} \\
&\leq \mathsf{aq}\left(\Phi_{\mathcal{T}_s^{\uparrow}}\right) + 2 \sum_{\mathcal{B} \in \mathfrak{B}_s} \left(\mathsf{aq}\left(\Phi_{\mathcal{B}}^{\circ}(t)\right) - 1\right) + 1. && \text{(by (44))}
\end{aligned}$$

The conclusion (45) follows from $\mathsf{aq}\left(\Phi_{\mathcal{B}}^{\circ}(t)\right) \leq \mathsf{aq}\left(\Phi_{\mathcal{B}}(t)\right) + 1$ by (44). $\square$

Lemma 8 converts the active-quadratic representation into a recurrence over rooted components. The off-path components carry a factor of two and may attach anywhere along $\mathcal{P}_s$. Since a centroid controls the components incident to one vertex, but does not control this attachment pattern, we introduce a new node-balancing technique in the next subsection.

5.2. **Complexity results.** Note that every $\mathcal{B} \in \mathfrak{B}_s^{\uparrow}$ is contained in the parent component $\mathcal{T}_s^{\uparrow}$ and also appears in the branch sum in (45). The power-centroid argument in Section 4 does not exploit where this overlap occurs. We instead use the ordinary centroid unless one member of $\mathfrak{B}_s^{\uparrow}$ is too large. In that event, moving to its attachment vertex yields the required balance; see Figure 1 for illustration.

Proposition 8. *Let $\mathcal{T}$ be a rooted tree with n vertices. Suppose that $\alpha \geq 2$ and $h \in [1/4, 1/2]$ satisfy*

$$2\big[h^{\alpha} + (1-h)^{\alpha}\big] \leq 1, \tag{46}$$

$$3 \cdot 2^{-\alpha} + 2\big[h^{\alpha} + (1/2-h)^{\alpha}\big] \leq 1. \tag{47}$$

Then there exists $s \in \mathcal{T}$ such that

$$|\mathcal{T}_s^{\uparrow}|^{\alpha} + 2 \sum_{\mathcal{B} \in \mathfrak{B}_s} |\mathcal{B}|^{\alpha} \leq n^{\alpha}. \tag{48}$$

Proof. Apply Lemma 3 with $\theta = 1/2$, and let z be the resulting centroid. Then one has

$$|\mathcal{T}_z^{\uparrow}| \leq n/2, \qquad |\mathcal{B}| \leq n/2, \quad \forall \mathcal{B} \in \mathfrak{B}_z^{\downarrow}. \tag{49}$$

Set $\bar{b} \overset{\text{def}}{=} \max\left\{|\mathcal{B}| : \mathcal{B} \in \mathfrak{B}_z^{\uparrow}\right\}$. We consider two cases based on whether $\bar{b} \leq hn$ or $\bar{b} > hn$.

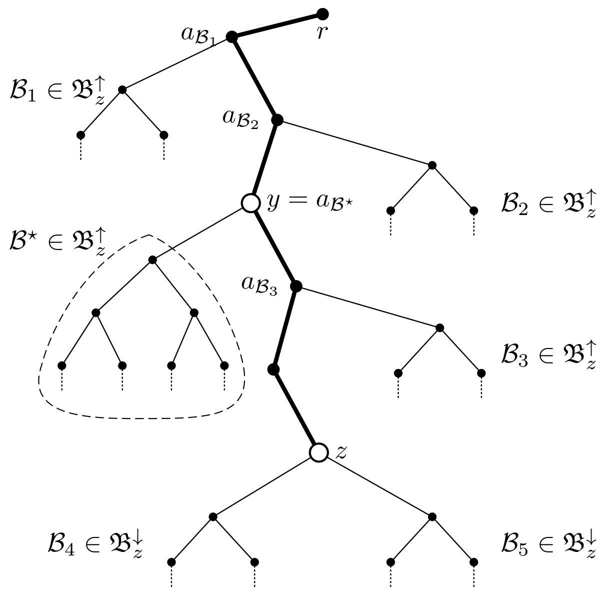

FIGURE 1. Tree decomposition along a root path

<u>*Case 1:*</u> $\bar{b} \le hn$. Set $x := |\mathcal{T}_z^{\uparrow}|/n$. By (49), one has

$$\sum_{\mathcal{B} \in \mathfrak{B}_z^{\uparrow}} |\mathcal{B}| \le xn, \quad |\mathcal{B}| \le hn, \ \forall \mathcal{B} \in \mathfrak{B}_z^{\uparrow}, \tag{50}$$

$$\sum_{\mathcal{B} \in \mathfrak{B}_z^{\downarrow}} |\mathcal{B}| \le (1-x)n, \quad |\mathcal{B}| \le n/2, \ \forall \mathcal{B} \in \mathfrak{B}_z^{\downarrow}. \tag{51}$$

There are two subcases:

- If $0 \le x \le h$, applying Lemma 4 to (50) and (51) yields

$$\begin{aligned} \sum_{\mathcal{B} \in \mathfrak{B}_z^{\uparrow}} |\mathcal{B}|^\alpha &\le x^\alpha n^\alpha, \\ \sum_{\mathcal{B} \in \mathfrak{B}_z^{\downarrow}} |\mathcal{B}|^\alpha &\le \left[2^{-\alpha} + (1/2 - x)^\alpha\right] n^\alpha. \end{aligned} \tag{52}$$

  This implies

$$|\mathcal{T}_z^{\uparrow}|^\alpha + 2 \sum_{\mathcal{B} \in \mathfrak{B}_z} |\mathcal{B}|^\alpha \le g_1(x) n^\alpha, \tag{53}$$

where $g_1(x) \overset{\text{def}}{=} 3x^\alpha + 2^{1-\alpha} + 2(1/2-x)^\alpha$. Because $g_1(x)$ is convex in x and $x \in [0,h]$, one can deduce that

$$\begin{aligned} g_1(x) &\le \max_{t\in[0,h]} g(t) = \max\{g_1(0), g_1(h)\} \\ &= \max\left\{2^{2-\alpha}, 2^{1-\alpha} + 3h^\alpha + 2(1/2-h)^\alpha\right\} \qquad (h \le 1/2) \\ &\le \max\left\{2^{2-\alpha}, 3\cdot 2^{-\alpha} + 2\big[h^\alpha + (1/2-h)^\alpha\big]\right\} \le 1, \end{aligned}$$

where the last inequality is due to $\alpha \ge 2$ and (47). In view of (53), it proves (48) in this subcase.

- If $h \le x \le 1/2$, then $hn \le xn \le 2hn$. Applying (31) to (50) gives

$$\sum_{\mathcal{B}\in\mathfrak{B}_z^\uparrow} |\mathcal{B}|^\alpha \le \big(h^\alpha + (x-h)^\alpha\big)n^\alpha.$$

Combining this inequality with the downstream estimate (52) gives

$$|\mathcal{T}_z^\uparrow|^\alpha + 2\sum_{\mathcal{B}\in\mathfrak{B}_z} |\mathcal{B}|^\alpha \le g_2(x)n^\alpha, \tag{54}$$

where $g_2(x) = x^\alpha + 2h^\alpha + 2(x-h)^\alpha + 2^{1-\alpha} + 2(1/2-x)^\alpha$. Since $g_2(x)$ is convex in $x \in [h, 1/2]$, one can deduce that

$$\begin{aligned} g_2(x) &\le \max_{t\in[h,1/2]} g(t) = \max\{g_2(1/2), g_2(h)\} \\ &= \max\left\{3\cdot 2^{-\alpha} + 2h^\alpha + 2(1/2-h)^\alpha, 3h^\alpha + 2^{1-\alpha} + 2(1/2-h)^\alpha\right\} \le 1, \end{aligned}$$

where the last inequality is due to $h \le 1/2$ and (47). In view of (54), we find (48) holds with $s = z$ whenever $\bar{b} \le hn$.

<u>*Case 2:*</u> $\bar{b} > hn$. In this case, choose $\mathcal{B}^\star \in \mathfrak{B}_z^\uparrow$ with $|\mathcal{B}^\star| = \bar{b}$, set $y \overset{\text{def}}{=} a_{\mathcal{B}^\star}$, and let $m \overset{\text{def}}{=} |\mathcal{T}_y^\uparrow|$. The sets $\mathcal{T}_y^\uparrow$ and $\mathcal{B}^\star$ are disjoint subsets of $\mathcal{T}_z^\uparrow$, and therefore

$$m + \bar{b} \le |\mathcal{T}_z^\uparrow| \le n/2. \tag{55}$$

Then by construction, one can see

$$\sum_{\mathcal{B}\in\mathfrak{B}_y^\uparrow} |\mathcal{B}| \le m, \qquad \sum_{\mathcal{B}\in\mathfrak{B}_y^\downarrow\setminus\{\mathcal{B}^\star\}} |\mathcal{B}| \le n - m - \bar{b}.$$

Thus, (30) gives

$$m^\alpha + 2\sum_{\mathcal{B}\in\mathfrak{B}_y} |\mathcal{B}|^\alpha \le g_3(m,\bar{b}) \overset{\text{def}}{=} 3m^\alpha + 2b^\alpha + 2(n-m-\bar{b})^\alpha. \tag{56}$$

Because $(m,\bar{b}) \in \Delta \overset{\text{def}}{=} \{(t_1,t_2) :\ t_1 \ge 0,\ t_2 \ge hn,\ t_1+t_2 \le n/2\}$ and $g_3(t_1,t_2)$ is convex over Δ, one obtains

$$\begin{aligned} g_3(m,\bar{b}) &\le \max_{(t_1,t_2)\in\Delta} g_3(t_1,t_2) = \max\{g_3(0,hn),\ g_3(0,n/2),\ g_3((1/2-h)n,hn)\} \\ &\le \max\Big\{2\big(h^\alpha + (1-h)^\alpha\big), 2^{2-\alpha}, 3\cdot 2^{-\alpha} + 2\big(h^\alpha + (1/2-h)^\alpha\big)\Big\}n^\alpha \le 1, \end{aligned}$$

where the bound on $g_3((1/2-h)n, hn)$ also uses $(1/2-h)^\alpha \le 2^{-\alpha}$. Consequently, (48) holds with $s = y$. This finishes the entire proof. $\square$

The weighted balance in (48) matches the coefficients in (45); it therefore supplies the induction step that the power centroid provides for the generic piece recurrence. Define

$$\alpha_\star \stackrel{\text{def}}{=} \min \left\{ \alpha \ge 2 : \begin{array}{c} \text{there exists } h \in [1/4, 1/2] \text{ such that} \\ 2\bigl(h^\alpha + (1-h)^\alpha\bigr) \le 1, \\ 3 \cdot 2^{-\alpha} + 2\bigl(h^\alpha + (1/2-h)^\alpha\bigr) \le 1 \end{array} \right\}. \tag{57}$$

It is not difficult to show that the minimum of (57) is attained when both inequalities are tight. Numerically, $\alpha_\star \approx 2.086$ and $h_\star \approx 0.383$. We now state the complexity result for quadratic node functions.

Theorem 3. *Every rooted tree $\mathcal{T}$ with n vertices satisfies*

$$\mathsf{aq}\left(\Phi_{\mathcal{T}}\right) \le 2n^{\alpha_\star} - 1. \tag{58}$$

Proof. We prove the conclusion by induction on n. If $n = 1$, then $\Phi_{\mathcal{T}}$ is one quadratic, and (58) holds with equality.

Consider an arbitrary rooted tree $\mathcal{T}$ with $n \ge 2$ vertices, and assume that (58) holds for every smaller rooted tree. Apply Proposition 8 with $(\alpha, h) = (\alpha_\star, h_\star)$, and let s be the associated separating vertex. Then Every nonempty $\mathcal{T}_s^\uparrow$ and $\mathcal{B} \in \mathfrak{B}_s$ is a rooted subtree in the sense of (35) and has fewer than n vertices.

If $\mathfrak{B}_s = \emptyset$, then $\mathcal{T} = \mathcal{P}_s$ is a path and $|\mathcal{T}_s^\uparrow| = n-1$. Lemma 8 and the induction hypothesis give

$$\mathsf{aq}\left(\Phi_{\mathcal{T}}\right) \le \mathsf{aq}\left(\Phi_{\mathcal{T}_s^\uparrow}\right) + 1 \le 2(n-1)^{\alpha_\star} \le 2n^{\alpha_\star} - 2.$$

Suppose now that $|\mathfrak{B}_s| \ge 1$. In this case, Lemma 8 and the induction hypothesis imply

$$\begin{aligned}
\mathsf{aq}\left(\Phi_{\mathcal{T}}\right) &\le \left(2|\mathcal{T}_s^\uparrow|^{\alpha_\star} - 1\right) + 2\sum_{\mathcal{B} \in \mathfrak{B}_s} \left(2|\mathcal{B}|^{\alpha_\star} - 1\right) + 1 \\
&= 2\left(|\mathcal{T}_s^\uparrow|^{\alpha_\star} + 2\sum_{\mathcal{B} \in \mathfrak{B}_s} |\mathcal{B}|^{\alpha_\star}\right) - 2|\mathfrak{B}_s| \\
&\le 2n^{\alpha_\star} - 2|\mathfrak{B}_s| \le 2n^{\alpha_\star} - 1,
\end{aligned}$$

where the second inequality follows from (48). This proves (58) and hence the theorem. $\square$

Theorem 3 improves the exponent $\gamma_2 \approx 3.04$ from the generic piece bound to $\alpha_\star \approx 2.08$ for quadratic functions. This refinement stems from tracking active quadratics through envelope geometry, which is a structural property unique to quadratic polynomials.

6. Conclusion

In this paper, we study optimization over trees with binary coupling decisions, where a fixed charge permits neighboring continuous decisions to differ. Although the problem is NP-hard on general graphs even for convex quadratic node costs, we develop an exact parametric dynamic programming algorithm for arbitrary trees. The proposed algorithm runs in oracle time polynomial in the problem size for a fixed degree of the underlying weak Chebyshev space. Our complexity analysis relies on a reformulation of the DP recursion in a lifted space and a novel power centroid decomposition to bound the piece complexity of the value functions. For quadratic polynomial node functions, we improve this bound from nearly cubic to nearly quadratic. As a direct implication, these results establish polynomial oracle complexity for the Potts model on arbitrary trees.

References

[1] Alfieri, A., Nicosia, G., and Pacifici, A. (2006). Exact algorithms for a discrete metric labeling problem. *Discrete Optimization*, 3(3):181–194.

[2] Bhathena, A., Fattahi, S., Gómez, A., and Küçükyavuz, S. (2025). A parametric approach for solving convex quadratic optimization with indicators over trees. *Mathematical Programming*.

[3] Bhathena, A., Fattahi, S., Gómez, A., and Küçükyavuz, S. (2026). Solving Convex Quadratic Optimization with Indicators Over Structured Graphs.

[4] Biedl, T. (2022). Horton-strahler number, rooted pathwidth and upward drawings of trees. *Information Processing Letters*, 175:106230.

[5] Boyd, J. P. (2006). Computing the zeros, maxima and inflection points of Chebyshev, Legendre and Fourier series: Solving transcendental equations by spectral interpolation and polynomial rootfinding. *Journal of Engineering Mathematics*, 56:203–219.

[6] Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11):1222–1239.

[7] Boysen, L., Kempe, A., Liebscher, V., Munk, A., and Wittich, O. (2009). Consistencies and rates of convergence of jump-penalized least squares estimators. *The Annals of Statistics*, 37(1):157–183.

[8] Chen, X., Ge, D., Wang, Z., and Ye, Y. (2014). Complexity of unconstrained minimization. *Mathematical Programming*, 143(1):371–383.

[9] Chen, X., Li, S., and Xie, W. (2025). Isotonic optimization with fixed costs.

[10] Chierichetti, F., Kleinberg, J., and Oren, S. (2018). On discrete preferences and coordination. *Journal of Computer and System Sciences*, 93:11–29. An open conference version appeared at EC 2013.

[11] Choi, S., Fattahi, S., Gomez, A., Han, S., and Lozano, L. (2026). Convexification of mixed-integer quadratic optimization via decision diagrams. *arXiv preprint arXiv:2608.22815*.

[12] Conforti, M., Cornuéjols, G., Zambelli, G., et al. (2014). *Integer programming*, volume 271. Springer.

[13] Dahlhaus, E., Johnson, D. S., Papadimitriou, C. H., Seymour, P. D., and Yannakakis, M. (1994). The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23(4):864–894.

[14] Dasci, A. and Huang, R. (2017). A continuous approximation method for dynamic pricing problem under costly price modifications. *Omega*, 72:38–49.

[15] Deutsch, F., Nürnberger, G., and Singer, I. (1980). Weak chebyshev subspaces and alternation. *Pacific Journal of Mathematics*, 89(1):9–31.

[16] Edelman, A. and Murakami, H. (1995). Polynomial roots from companion matrix eigenvalues. *Mathematics of Computation*, 64(210):763–776.

[17] Fan, Z. and Guan, L. (2018). Approximate ℓ_0-penalized estimation of piecewise-constant signals on graphs. *The Annals of Statistics*, 46(6B):3217–3245.

[18] Fernández-Baca, D. and Slutzki, G. (1989). Solving parametric problems on trees. *Journal of algorithms*, 10(3):381–402.

[19] Gómez, A., Han, S., and Lozano, L. (2024). Real-time solution of quadratic optimization problems with banded matrices and indicator variables.

[20] Haar, A. (1917). Die Minkowskische Geometrie und die Annäherung an stetige Funktionen. *Mathematische Annalen*, 78:294–311.

[21] Han, S. and Gómez, A. (2025a). Compact extended formulations for low-rank functions with indicator variables. *Mathematics of Operations Research*.

[22] Han, S. and Gómez, A. (2025b). Convex submodular minimization with indicator variables.

[23] Han, S., Gómez, A., and Atamtürk, A. (2023). 2×2-convexifications for convex quadratic optimization with indicator variables. *Mathematical Programming*, 202:95–134.

[24] Jordan, C. (1869). Sur les assemblages de lignes.

[25] Karlin, S. and Studden, W. J. (1966). *Tchebycheff Systems: With Applications in Analysis and Statistics*, volume 15 of *Pure and Applied Mathematics*. Interscience Publishers, New York.

[26] Kolmogorov, V., Pock, T., and Rolinek, M. (2016). Total variation on a tree. *SIAM Journal on Imaging Sciences*, 9(2):605–636.

[27] Kuric, M., Ahmetspahic, J., and Pock, T. (2024). Total Generalized Variation on a Tree. *SIAM Journal on Imaging Sciences*, 17(2):1040–1077.

[28] Landrieu, L. and Obozinski, G. (2017). Cut pursuit: Fast algorithms to learn piecewise constant functions on general weighted graphs. *SIAM Journal on Imaging Sciences*, 10(4):1724–1766.

[29] Lin, C. and Shang, J.-L. (2009). Statuses and branch-weights of weighted trees. *Czechoslovak Mathematical Journal*, 59(4):1019–1025.

[30] Liu, P., Fattahi, S., Gómez, A., and Küçükyavuz, S. (2022). A graph-based decomposition method for convex quadratic optimization with indicators. *Mathematical Programming*, pages 1–33.

[31] Picard, F., Robin, S., Lavielle, M., Vaisse, C., and Daudin, J.-J. (2005). A statistical approach for array CGH data analysis. *BMC Bioinformatics*, 6:27.

[32] Potts, R. B. (1952). Some generalized order-disorder transformations. In *Mathematical proceedings of the cambridge philosophical society*, volume 48, pages 106–109. Cambridge University Press.

[33] Shafiee, S. and Kılınç-Karzan, F. (2024). Constrained optimization of rank-one functions with indicator variables: S. shafiee, f. kılınç-karzan. *Mathematical Programming*, 208(1):533–579.

[34] Shen, R., Reinelt, G., and Canu, S. (2018). A first derivative Potts model for segmentation and denoising using ILP. In *Operations Research Proceedings 2017*, pages 53–59. Springer.

[35] Stone, H. S. (1977). Multiprocessor scheduling with the aid of network flow algorithms. *IEEE Transactions on Software Engineering*, SE-3(1):85–93.

[36] Wei, L., Gómez, A., and Kucukyavuz, S. (2020). Ideal formulations for constrained convex optimization problems with indicator variables. *arXiv preprint arXiv:2007.00107*.

[37] Weinmann, A., Storath, M., and Demaret, L. (2015). The L^1-Potts functional for robust jump-sparse reconstruction. *SIAM Journal on Numerical Analysis*, 53(1):644–673.

[38] Wu, F.-Y. (1982). The Potts model. *Reviews of modern physics*, 54(1):235.