arXiv:2304.00244v1 [math.OC] 1 Apr 2023

一种用于求解具有广义序约束的凸保序回归的基于活动集的递归方法

Xuyu Chen*，Xudong Li†，杨风苏‡

2023年4月4日

摘要

本文研究由有向树诱导的具有广义序约束的凸保序回归。所提出的模型涵盖了各种有趣的带形状或序约束的优化问题，包括广义近似保序优化和树上的全变分。受相邻池违例者算法及其活动集解释的成功启发，我们提出了一种基于活动集的递归方法来求解该模型。与需要遍历指数数量的可能活动集组合的蛮力方法不同，在温和假设下，我们的算法具有多项式时间计算复杂度。

关键词：活动集方法；凸保序回归；广义序约束 AMS主题分类：90C25，90C30

1 引言

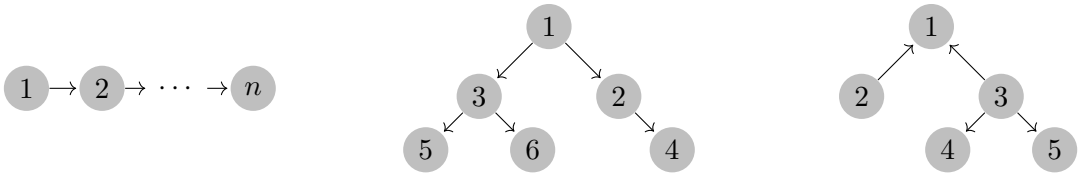
给定一棵有向树 $G=(V,E)$，我们考虑如下带有广义序约束的凸保序回归问题：

$$\min_{x\in\Re^{|V|}} \sum_{i\in V} f_i(x_i) + \sum_{(i,j)\in E} \lambda_{i,j}(x_i-x_j)_+ + \sum_{(i,j)\in E} \mu_{i,j}(x_j-x_i)_+, \tag{1}$$

其中，对于每个 $i\in V$, $f_i:\ \Re\to\Re$ 是凸损失函数；$\lambda_{i,j}$ 和 $\mu_{i,j}$ 对于 $(i,j)\in E$ 是可能取无穷大的非负标量，即 $0\le\lambda_{i,j},\mu_{i,j}\le+\infty$；而 $(x)_+=\max(0,x)$ 是任意 $x\in\Re$ 的 x 的非负部分。在 (1) 中，当 $\lambda_{i,j}=+\infty$ （分别为 $\mu_{i,j}=+\infty$）时，相应项 $\lambda_{i,j}(x_i-x_j)_+$ （分别为 $\mu_{i,j}(x_j-x_i)_+$）应理解为指示函数$\delta(x_i,x_j\mid x_i-x_j\le 0)$（分别为 $\delta(x_i,x_j\mid x_i-x_j\ge 0)$），或者等价地，理解为约束 $x_i-x_j\le 0$（分别为 $x_i-x_j\ge 0$）。图1展示了一些简单的有向树示例。

可以看出，引入有向树 G 使问题 (1) 成为一个相当一般的模型，其中包含许多有趣的特例。为简便起见，这里仅介绍

*复旦大学数学科学学院，中国上海市，邮编 200433，chenxy18@fudan.edu.cn
†复旦大学数据科学学院，中国上海市，邮编 200433，lixudong@fudan.edu.cn
‡复旦大学数学科学学院，中国上海市，邮编 200433，yfsu@fudan.edu.cn



(a) 链 (b) 树形图 (c) 一般有向树

图1：有向树的示例。有向树是底层图为树的有向图，有向树也称为有向无环图。

其中两个。第一个是 [26] 提出的广义近似保序优化（GNIO）问题:

$$\min_{x\in\Re^n} \sum_{i=1}^{n} f_i(x_i) + \sum_{i=1}^{n-1} \lambda_i (x_i - x_{i+1})_+ + \sum_{i=1}^{n-1} \mu_i (x_{i+1} - x_i)_+, \tag{2}$$

这显然是式（1）的一个特例，其中 G选为一条链，如图1a所示。正如 [26], 所述，模型（2）将许多形状约束统计回归中的经典问题作为特例加以求解，包括保序回归 [6, 7], 、单峰回归 [12, 21], 和近似保序回归 [22]。第二个问题是 [15]中研究的树上的全变分:

$$\min_{x\in\Re^{|V|}} \sum_{i\in V} f_i(x_i) + \sum_{(i,j)\in E} w_{i,j}|x_i - x_j|, \tag{3}$$

其中 $G = （V， E）$是一棵有向树，并假设每个 f_i 均为分段线性或分段二次函数。文献还研究了模型（1）的其他特例。例如，[8, 25] 研究了由树形图所诱导的偏序约束下的保序回归问题。这些特例及其在统计推断 [20], 、运筹学 [1], 、信号处理 [17, 9], 、医学预后 [19], 以及交通与气候数据分析[16, 24], 中的应用，揭示了研究模型（1）的重要性和必要性。

据我们所知，目前尚无高效算法可直接求解一般模型 (1)。不过，该模型的某些特殊情形可由现有算法求解。例如，可采用 [26] 中设计的动态规划方法，高效求解 GNIO 问题 (2)。此外，在假设决策变量有界的情况下， [13] 中提出的基于 KKT 的快速算法也可求解 GNIO 问题。然而，这两种算法都严重依赖底层的链结构，因此无法用于求解涉及有向树的一般模型 (1)。若 G 是一条链，且每个 f_i 都是二次函数，则全变差问题 (3) 可化为著名的 ℓ_2 全变差去噪问题，该问题已在信号处理领域得到广泛研究 [10, 14]。直接算法 [10] 和绷紧弦算法 [2] 被认为是求解 ℓ_2 全变差去噪问题的最先进方法。同时，若假设 G 是一棵有向树，且在 (3) 中每个 f_i 都是连续的分段线性或分段二次函数，并且只有有限个断点，则可采用 [15] 中研究的消息传递算法。不过，这些算法无法处理涉及一般凸损失函数 f_i 的问题 (1)。

此外，还有另一类研究致力于求解问题（1）的特殊情形。20世纪50年代，Ayer 在 [1] 提出了著名的相邻池违例者算法（PAVA），用于求解

以下保序回归问题：

$$\begin{aligned} \min_{x\in\Re^n} \quad & \frac{1}{2}\sum_{i=1}^{n}(x_i-y_i)^2, \\ \text{s.t.} \quad & x_1 \le x_2 \le \ldots \le x_n, \end{aligned} \tag{4}$$

这显然是问题（1）的一个特殊情形。自问世以来，PAVA 一直被视为求解保序回归问题的最先进技术。后来，在 [4]，Best 和 Chakravarti 发现，PAVA 实际上是一种用于求解（4）的对偶可行活动集方法。在 [5]，，PAVA 得到推广，可用于求解将最小二乘目标函数替换为一般可分凸损失函数后的（4）。在 [25]，，Yu 和 Xing 进一步推广了 PAVA，使其能够求解由树形图诱导序约束的凸可分最小化问题。然而，模型（1）目标函数中包含的广义正则项在 [25] 尚未得到研究。据我们所知，目前仍不清楚能否采用 PAVA 背后的思路求解更一般的模型（1）。

受 PAVA 及其变体在求解广义凸保序回归问题（1）的特殊情形中取得成功的启发，本文提出一种新颖的基于活动集的算法。我们的方法不同于暴力法，后者需要探索数量可能呈指数级增长的不同活动集。相反，我们提出一种递归方法，以加快对所需活动集的搜索。我们证明，问题（1）可以通过递归求解一系列规模更小的子问题来处理。针对这些子问题，我们仔细考察了相应 Karush-Kuhn-Tucker（KKT）条件的特殊递归结构，并据此设计了一种新颖的基于活动集的递归方法（ASRA）。具体而言，该方法使我们能够推导出上述递归子问题最优解的半闭式公式。在适度的假设下，我们进一步证明，ASRA 求解问题（1）具有多项式时间计算复杂度。

本文后续各节安排如下。第 2 节介绍与问题（1）相关的必要预备知识，包括图论的基本概念及相应的 KKT 条件。此外，我们还介绍一种求解（1）的朴素活动集方法。第 3 节详细介绍我们的递归方法 ASRA。最后，我们在第 4 节总结全文。附录给出一个示例，说明如何应用 ASRA 求解问题（1）的一个简单实例。

2 预备知识

我们先介绍一些图论中的相关预备知识。有向树 $G=(V,E)$ 是指底层图为树的有向图，而树形图（也称为有根有向树）[11, 23] 是恰有一个入度为零的节点的有向树。该节点也称为树形图的根节点。设 $G=(V,E)$ 和 $B=(V_B,E_B)$ 是两棵有向树。若 $V_B\subseteq V$ 且 $E_B\subseteq E$，则称 B 为 G的子树，记作 $B\subset G$。若两个子树的节点集不相交，则称它们互不相交。给定某棵有向树 $G=(V,E)$ 的一组互不相交子树 $P=\{B_k\}_{k=1}^K$，若 $V=\cup_{k=1}^K V_{B_k}$，则称 P 是 G 的一个划分。

对于给定的有向树 $G=(V,E)$，我们可以选择任意节点 $l\in V$作为 G的祖先。此时，对于任意 $i,j\in V$，如果连接 l和 i 的无向路径严格包含在连接 l 和 j的无向路径中，我们就称 j是 i的子节点，记作 $j\lhd i$。例如，如果我们选择节点 2 作为

图1c所示有向树中的祖先，那么有 $3 \lhd 1 \lhd 2$。现在，令 $D \in \Re^{|V|\times|E|}$为与 G相关联的节点-弧关联矩阵。由 [3] 可知，$\mathrm{rank}(D) = |E|$ ，且从 D 中删除第 l行后得到的矩阵 $\tilde{D}_l \in \Re^{|E|\times|E|}$ 可逆。给定一个向量$b \in \Re^{|E|}$，我们在下面的引理中给出线性˜系统 $D_l z = b$的解的闭式公式。

引理 1. 对于任意给定的 $b \in \Re^{|E|}$，线性方程组$D_l z = b$ 的唯一解 $z^* = (z_{i,\ j})_{(i,j)\in E} \in \Re^{|E|}$ 具有如下形式：

$$z^*_{i,j} = \begin{cases} \sum\limits_{k\in C_i} b_k, \ \textit{if } i \lhd j, \\ -\sum\limits_{k\in C_j} b_k, \ \textit{if } j \lhd i, \end{cases} \qquad \forall\, (i,j) \in E,$$

其中，对于任意节点 i， C_i由 i及其所有子节点组成，即 $C_i := \{j \in V \mid j \lhd i\} \cup \{i\}$。

证明. 该结果直接源于节点-弧关联矩阵的特殊结构，也可通过直接验证得到。 □

接下来，我们陈述损失函数 $f_i, i \in V$ 的全面假设，并推导与问题 (1) 相关的 KKT 条件。为清晰地阐述主要思想，我们对 f_i 作出较强的假设，例如强凸性和可微性。然而可以看出，若采用更精细的分析，这些较强的假设可以去除。

Assu假设 1. (1) 中的每个 $f_i : \Re \to \Re, i \in V$ 都是可微且强凸的 .

由于每个 f_i 都是强凸的，我们知道问题 (1) 中的目标函数也具有强凸性，因此其水平集有界。此外，根据 [18, 定理 27.1 和 27.2]，问题 (1) 有唯一解。我们还注意到，在一些统计学和机器学习问题中，假设 1 成立 [4, 10, 22]。在假设 1 下，根据 [18] ，我们知道每个 f_i^* 也都是强凸可微函数。此外， f_i' 和 $(f_i^*)'$ 在 $\Re$ 上都严格递增，并且对于任意给定的 x， $y \in \Re$， $y = f_i'(x)$ 当且仅当 $x = (f_i^*)'(y)$。

现在，我们准备写出与问题 (1) 相关的 KKT 条件。对于$0 \le \lambda$， $\mu \le +\infty$，令

$$\begin{cases} h_\lambda^-(x) := \delta(x \mid x \ge 0), & \text{if } \lambda = +\infty, \\ h_\lambda^-(x) := \begin{cases} -\lambda x, & x < 0, \\ 0, & x \ge 0, \end{cases} & \text{if } 0 \le \lambda < +\infty, \end{cases} \quad \text{and} \quad \begin{cases} h_\mu^+(x) := \delta(x \mid x \le 0), & \text{if } \mu = +\infty, \\ h_\mu^+(x) := \begin{cases} 0, & x \le 0, \\ \mu x, & x > 0, \end{cases} & \text{if } 0 \le \mu < +\infty. \end{cases}$$

对于 $(i,\ j) \in E$，我们定义 $h_{i,\ j} : \Re \to [0, +\infty]$ 如下

$$h_{i,j}(x) := h^-_{\lambda_{i,j}}(x) + h^+_{\mu_{i,j}}(x), \quad \forall\, x \in \Re.$$

显然，对于每个（$i \quad j$) $\in E$, $h_{i,j}$ 是凸函数，其在 $x \in \Re$ 处的次微分具有以下形式 :

$$\partial h_{i,j}(x) = \begin{cases} \{-\lambda_{i,j}\}, & \text{if } x < 0, \\ [-\lambda_{i,j}, \mu_{i,j}], & \text{if } x = 0, \\ \{\mu_{i,j}\}, & \text{if } x > 0. \end{cases} \tag{5}$$

这里，$\partial h_{i,j}(x) = \{+\infty\}$或$\partial h_{i,j}(x) = \{-\infty\}$应理解为$\partial h_{i,j}(x) = \emptyset$。我们还采用 (5) 中的约定，即对于某个 $\alpha \in \Re$，$[-\infty, +\infty] = (-\infty, +\infty)$、$[-\infty, \alpha] = (-\infty, \alpha]$和 $[\alpha, +\infty] = [\alpha, +\infty)$。

定义 $H(z) := \sum_{(i,j)\in E} h_{i,j}(z_{i,j})$，其中 $z \in \Re^{|E|}$；并定义 $F(x) := \sum_{i\in V} f_i(x_i)$，其中 $x \in \Re^{|V|}$。令$M = -D^T \in \Re^{|E|\times|V|}$，其中 D 是与 G相关联的节点-弧关联矩阵。也就是说，对于 $e = (i, j) \in E$，$M(e, i) = -1$，以及 $M(e, j) = 1$，M的所有其他元素均为零。令$H_M(x) := H(Mx)$，其中 $x \in \Re^{|V|}$。于是，很容易验证，问题 (1) 可等价地改写为

$$\min_{x\in\Re^{|V|}} \; F(x) + H_M(x).$$

接下来给出一个关于问题 (1) 的 KKT 条件的引理。

引理 2. 问题 (1) 存在唯一的极小化解 $x^* \in \Re^{|V|}$。此外，x^* 是问题 (1) 的解，当且仅当存在唯一的乘子 $z^* \in \Re^{|E|}$，使得 (x^*, z^*) 满足以下 KKT 系统：

$$\begin{aligned}
&\sum_{k:(i,k)\in E} z^*_{i,k} - \sum_{k:(k,i)\in E} z^*_{k,i} = f_i'(x_i^*), \quad \forall\, i \in V,\\
&z^*_{i,j} \in \begin{cases} \{-\lambda_{i,j}\}, & \text{if } x_i^* > x_j^*,\\ [-\lambda_{i,j}, \mu_{i,j}], & \text{if } x_i^* = x_j^*, \quad \forall\, (i,j) \in E.\\ \{\mu_{i,j}\}, & \text{if } x_i^* < x_j^*, \end{cases}
\end{aligned} \tag{6}$$

证明。问题 (1) 的最优解存在且唯一，这是由 F 的强凸性得出的。由于 F 可微，根据 [18, 定理 23.8] ，我们知道

$$0 \in F'(x^*) + \partial H_M(x^*).$$

由 $[18, \text{定理 } 23.9]$, 可知，$\partial H_M(x^*) = M^T \partial H(Mx^*)$。因此，存在$z^* \in \partial H(Mx^*)$，使得

$$F'(x^*) + M^T z^* = F'(x^*) - Dz^* = 0. \tag{7}$$

由于 Mx^* 的第 e 个分量为 $x_j^* - x_i^*$，由 (5) 可得

$$z^*_{i,j} \in \begin{cases} \{-\lambda_{i,j}\}, & \text{if } x_j^* - x_i^* < 0,\\ [-\lambda_{i,j}, \mu_{i,j}], & \text{if } x_j^* - x_i^* = 0, \quad \forall\, (i,j) \in E.\\ \{\mu_{i,j}\}, & \text{if } x_j^* - x_i^* > 0, \end{cases}$$

由此，我们得到 KKT 条件 (6)。根据 (7) 以及 $\text{rank}(D) = |E|$ 这一事实，可知 z^* 是唯一的。至此，证明完毕。 □

接下来，我们研究一种用于求解问题 (1) 的朴素活动集方法。对于每条边 $(i, j) \in E$，我们可以为其关联一个符号 $\# \in \{<, =, >\}$，从而得到一个三元组 $(i, j, \#)$，表示关系$x_i \# x_j$。为保持一致性，在处理边 (i, j) 且 $\lambda_{i,\,j} = +\infty$ （或 $\mu_{i,\,j} = +\infty$）时，相应的符号 # 只能从 $\{<, =\}$ （或 $\{>, =\}$）中选取。我们用 $\mathcal{A}$ 表示所有这些三元组的集合，并将其称为与问题 (1) 相关联的活动集。接着，活动集 $\mathcal{A}$从 (1) 诱导出以下 $\mathcal{A}$约化问题：

$$\begin{aligned}
\min_{x\in\Re^{|V|}} \quad & \sum_{i\in V} f_i(x_i) + \sum_{(i,j)\in\mathcal{A}_>} \lambda_{i,j}(x_i - x_j) + \sum_{(i,j)\in\mathcal{A}_<} \mu_{i,j}(x_j - x_i),\\
\text{s.t.} \quad & x_i = x_j, \quad \forall\, (i,j) \in \mathcal{A}_=,
\end{aligned} \tag{8}$$

其中 $\mathcal{A}_{\#} := \{(i,j) \mid (i,j,\#) \in \mathcal{A}\}$。若 $\mathcal{A}_{=} = \emptyset$，则 (8) 化为一个无约束优化问题；由于其目标函数可分离、光滑且强凸，因此可以高效求解。对于 i、$j \in V$，当且仅当存在一条 $\mathcal{A}_{=}$中的无向路径连接 i 和j时，我们称它们是 $\mathcal{A}$连通的；该路径由将 $\mathcal{A}_{=}$ 中的所有边视为无向边得到。令 $P_{\mathcal{A}}$ 为 G中所有 $\mathcal{A}$连通分量的集合。不难看出， $P_{\mathcal{A}}$ 自然构成 G的一个划分。因此，我们将 $P_{\mathcal{A}}$ 称为由 $\mathcal{A}$ 诱导的划分。不失一般性，假设 $P_{\mathcal{A}} = \{B_k\}_{k=1}^{K}$，且每个 B_k 都是 G的子树；由此可见， $\mathcal{A}$约化问题 (8) 可解耦为 K 个独立子问题，如下所示：

$$\min_{x \in \Re^{|V_{B_k}|}} \left\{ \sum_{i \in V_{B_k}} \hat{f}_i(x_i) \mid x_i = x_j, \forall\, (i,j) \in E_{B_k} \right\}, \quad 1 \le k \le K, \tag{9}$$

其中，对每个 $i \in V_{B_k}$,

$$\hat{f}_i(x_i) := f_i(x_i) + \Big(\sum_{j:(i,j)\in\mathcal{A}_{>}} \lambda_{i,j} - \sum_{j:(i,j)\in\mathcal{A}_{<}} \mu_{i,j} \Big) x_i + \Big(\sum_{l:(l,i)\in\mathcal{A}_{<}} \mu_{l,i} - \sum_{l:(l,i)\in\mathcal{A}_{>}} \lambda_{l,i} \Big) x_i.$$

显然，问题 (9) 中的简单约束可以消去。由此得到的无约束优化问题具有单变量光滑强凸目标函数，因此可以高效求解。这样，我们便得到 $\mathcal{A}$-约化问题 (8) 的最优解。

遗憾的是，活动集 $\mathcal{A}$ 最多可能有 $3^{|E|}$ 种不同的选择。因此，逐一探索所有可能的不同活动集这一朴素方法，需要求解指数数量的 $\mathcal{A}$-约化问题。为降低这项高昂的计算成本，我们将在下一节介绍一种新颖的基于活动集的递归算法。

3 求解问题 (1) 的递归算法

本节介绍用于求解问题 (1) 的递归算法。首先我们指出，不失一般性，可以假设 (1) 中的有向树 G 是一棵以节点 1 为根节点的树形图。此外，我们可以将 G 分解为一系列子树 $\{G_m = (V_m, E_m)\}_{m=1}^{n}$，其中 $G_1 \subset G_2 \subset \cdots \subset G_n = G$和 $V_m = \{1, 2, \ldots, m\}$对 $1 \le m \le n$成立，并且边集 $E_{m+1} \setminus E_m$ 恰好包含一条边 $(i_m, m+1)$，其中 $i_m \in V_m$。更多细节见附录。

对于每个 $1 \le m \le n$，问题 (1) 限制在子树 G_m 上时，具有以下形式：

$$\min_{x \in \Re^{|V_m|}} \sum_{i \in V_m} f_i(x_i) + \sum_{(i,j)\in E_m} \lambda_{i,j}(x_i - x_j)_+ + \sum_{(i,j)\in E_m} \mu_{i,j}(x_j - x_i)_+. \tag{10}$$

由引理 2 不难看出，问题 (10) 的唯一原始-对偶最优对，记为 $(x^{(m)}, z^{(m)}) \in \Re^{|V_m|} \times \Re^{|E_m|}$，满足以下 KKT 系统：

$$\begin{aligned} &\sum_{k:(i,k)\in E_m} z_{i,k} - \sum_{k:(k,i)\in E_m} z_{k,i} = f_i'(x_i), \quad \forall\ i \in V_m, \\ &z_{i,j} \in \begin{cases} \{-\lambda_{i,j}\}, & \text{if } x_i > x_j, \\ [-\lambda_{i,j}, \mu_{i,j}], & \text{if } x_i = x_j, \\ \{\mu_{i,j}\}, & \text{if } x_i < x_j, \end{cases} \quad \forall\ (i,j) \in E_m. \end{aligned} \tag{11}$$

为方便起见，唯一最优对 $(x^{(m)}, z^{(m)})$ 也称为 G_m-最优对。通过仔细利用 KKT 条件 (11) 中的特殊结构，我们提出以递归方式求解问题 (1)。具体而言，对于 $m = 1$，.....，$n - 1$，我们将从 G_m-最优对 $(x^{(m)}, z^{(m)})$ 递归生成 G_{m+1}-最优对 $(x^{m+1}, z^{(m+1)})$。

我们在算法 1 中总结了上述递归方法的详细步骤。在该算法中，generate 子程序旨在根据 G_m-最优对生成 G_{m+1}-最优对。在下一小节中，我们将说明这一过程是通过一种新颖的主动集搜索方案实现的。因此，我们自然将算法 1 称为基于活动集的递归方法（ASRA）。

算法 1 ASRA：用于求解问题 (1) 的基于活动集的递归方法

1: 初始化： $x_1^{(1)} = (f_1^*)'(0) \in \Re$，并且 $z^{(1)} = \emptyset$ 2: 对于 $m = 1,\ldots,n-1$，执行 3: $(x^{(m+1)}, z^{(m+1)}) = \text{generate}(x^{(m)}, z^{(m)}, G_{m+1})$ 4: 结束循环 5: 返回：$(x^{(n)}, z^{(n)}) \in \Re^n \times \Re^{n-1}$

3.1 generate 子程序

为了从给定的 G_m-最优对高效地求得 G_{m+1}-最优对，我们将考察由子树 G_{m+1} 诱导的子问题所对应的 KKT 条件。具体形式如下：

$$\sum_{k:(i,k)\in E_m} z_{i,k} - \sum_{k:(k,i)\in E_m} z_{k,i} = f_i'(x_i), \quad \forall\, i \in V_m \backslash \{i_m\}, \tag{12}$$

$$z_{i,j} \in \begin{cases} \{-\lambda_{i,j}\}, & \text{if } x_i > x_j, \\ [-\lambda_{i,j}, \mu_{i,j}], & \text{if } x_i = x_j, \quad \forall\, (i,j) \in E_m, \\ \{\mu_{i,j}\}, & \text{if } x_i < x_j, \end{cases} \tag{13}$$

$$\sum_{k:(i_m,k)\in E_m} z_{i_m,k} - \sum_{k:(k,i_m)\in E_m} z_{k,i_m} + z_{i_m,m+1} = f_{i_m}'(x_{i_m}), \tag{14}$$

$$-z_{i_m,\ m+1} = f_{m+1}'(x_{m+1}), \tag{15}$$

$$z_{i_m,m+1} \in \begin{cases} \{-\lambda_{i_m,m+1}\}, & \text{if } x_{i_m} > x_{m+1}, \\ [-\lambda_{i_m,m+1}, \mu_{i,j}], & \text{if } x_{i_m} = x_{m+1}, \\ \{\mu_{i_m,m+1}\}, & \text{if } x_{i_m} < x_{m+1}. \end{cases} \tag{16}$$

可以看出，我们没有将 KKT 条件写成一个完整的方程组，而是单独列出了与对偶变量 $z_{i_m,\ m+1}$ 相关的条件，即 (14)、(15) 和 (16)；该对偶变量对应于新添加的边 $\{(i_m, m+1)\} = E_{m+1} \setminus E_m$。基于上述 KKT 条件，我们得到以下关于 $z_{i_m,\ m+1}$ 符号的命题。

命题1. 有 $z_{i_m}^{(m+1)}\, {}_{,m+1} f_{m+1}'(x_{i_m}^{(m)}) \le 0$，其中 $(x^{(m)}, z^{(m)})$ 和 $(x^{(m+1)}, z^{(m+1)})$ 分别是 G_m-最优对和 G_{m+1}-最优对。

证明. 注意，当 $f'_{m+1}(x_{i_m}^{(m)}) = 0$ 时，所需结果自然成立。对于其余部分，我们只证明 $f'_{m+1}(x_{i_m}^{(m)}) > 0$ 的情形，因为针对 $f'_{m+1}(x_{i_m}^{(m)}) < 0$ 情形的证明可由此处的论证轻易修改得到。

假设 $f'_{m+1}(x_{i_m}^{(m)}) > 0$，则我们将证明 $z_{i_m}^{(m+1)}, {}_{m+1} \leq 0$。反设 $z_{i_m}^{(m+1)}, {}_{m+1} > 0$。由 (15) 可得 $x_{m+1}^{(m+1)} = (f_{m+1}^*)'(-z_{i_m\, m+1}^{(m+1)}) < (f_{m+1}^*)'(0)$。此外，由 (16) 可知 $x_{m+1}^{(m+1)} \geq x_{i_m}^{(m+1)}$。因此，根据 $(f_{m+1}^*)'$ 的严格单调性，可得以下不等式:

$$x_{i_m}^{(m)} > (f_{m+1}^*)'(0) > x_{m+1}^{(m+1)} \geq x_{i_m}^{(m+1)}. \tag{17}$$

现在，由 (12)、(13) 和 (14) 可知，对于 $i \in V_m$，$\widetilde{x} \in \Re^{|V_m|}$ 与 $\widetilde{x}_i = x_i^{(m+1)}$ 构成以下优化问题的最优解:

$$\min_{x\in\Re^{|V_m|}} F_1(x) := \sum_{i\in V_m} f_i(x_i) + \sum_{(i,j)\in E_m} \{\lambda_{i,j}(x_i - x_j)_+ + \mu_{i,j}(x_j - x_i)_+\} - z_{i_m,m+1}^{(m+1)} x_{i_m}.$$

同时，由于 $(x^{(m)}, z^{(m)})$ 是 G_m-最优对，$x^{(m)}$ 是以下优化问题的最优解:

$$\min_{x\in\Re^{|V_m|}} F_0(x) := \sum_{i\in V_m} f_i(x_i) + \sum_{(i,j)\in E_m} \{\lambda_{i,j}(x_i - x_j)_+ + \mu_{i,j}(x_j - x_i)_+\}.$$

因此，有

$$0 \geq F_1(\widetilde{x}) - F_1(x^{(m)}) = F_0(\widetilde{x}) - F_0(x^{(m)}) + z_{i_m,m+1}^{(m+1)}(x_{i_m}^{(m)} - \widetilde{x}_{i_m}).$$

由于 $F_0(\widetilde{x}) - F_0(x^{(m)}) \geq 0$, $z_{i_m}^{(m+1)}, {}_{m+1} > 0$，且 $\widetilde{x}_{i_m} = x_{i_m}^{(m+1)}$，我们得到 $x_{i_m}^{(m)} - x_{i_m}^{(m+1)} \leq 0$, 这与 (17) 矛盾。因此，我们有 $z_{i_m}^{(m+1)}, {}_{m+1} \leq 0$ 且 $z_{i_m}^{(m+1)}, {}_{m+1} f'_{m+1}(x_{i_m}^{(m)}) \leq 0$，证明完毕。 □

根据命题1，我们可以根据 $f'_{m+1}(x_{i_m}^{(m)})$ 的值确定 $z_{i_m}^{(m+1)}, {}_{m+1}$ 的符号。此外，如果 $f'_{m+1}(x_{i_m}^{(m)}) = 0$，我们可以很容易地构造如下的 G_{m+1}-最优对:

$$x_i^{(m+1)} = \begin{cases} x_i^{(m)}, \forall\, i \in V_m, \\ x_{i_m}^{(m)}, \; i = m+1, \end{cases} \text{ and } z_{i,j}^{(m+1)} = \begin{cases} z_{i,j}^{(m)}, \forall\, (i,j) \in E_m, \\ 0, \; (i,j) = (i_m, m+1). \end{cases}$$

因此，在后续讨论中，我们关注 $f'_{m+1}(x_{i_m}^{(m)}) \neq 0$ in 的情形。为此，我们考虑以下以参数 $t \in \Re$ 为参数的参数优化问题:

$$\min_{x\in\Re^{|V_{m+1}|}} \sum_{i\in V_{m+1}} f_i(x_i) + \sum_{(i,j)\in E_m} \{\lambda_{i,j}(x_i - x_j)_+ + \mu_{i,j}(x_j - x_i)_+\} - t(x_{i_m} - x_{m+1}), \tag{18}$$

其 KKT 条件如下所示:

$$\begin{aligned}
&\sum_{k:(i,k)\in E_m} z_{i,k} - \sum_{k:(k,i)\in E_m} z_{k,i} + 1_{\{i=i_m\}} t = f_i'(x_i), \quad \forall\, i \in V_m, \\
&z_{i,j} \in \begin{cases} \{-\lambda_{i,j}\}, & \text{if } x_i > x_j, \\ [-\lambda_{i,j}, \mu_{i,j}], & \text{if } x_i = x_j, \\ \{\mu_{i,j}\}, & \text{if } x_i < x_j, \end{cases} \quad \forall\, (i,j) \in E_m, \\
&-t = f_{m+1}'(x_{m+1}).
\end{aligned} \tag{19}$$

由于每个 f_i 都是强凸的，问题 (18) 对每个 $t \in \Re$ 都有唯一的最优解，记为 $x^*(t)$。此外，根据 Fenchel-Rockafellar对偶定理 [18] 和每个 f_i 的可微性，我们知道问题 (18) 存在唯一的对偶最优解，记为 $z^*(t)$；它与 $x^*(t)$ 一起满足 KKT 条件 (19)。若对某个 $t^* \in \Re$，有

$$t^* \in \begin{cases} \{-\lambda_{i_m,m+1}\}, & \text{if } x^*_{i_m}(t^*) > x^*_{m+1}(t^*), \\ [-\lambda_{i_m,m+1}, \mu_{i_m,m+1}], & \text{if } x^*_{i_m}(t^*) = x^*_{m+1}(t^*), \\ \{\mu_{i_m,m+1}\}, & \text{if } x^*_{i_m}(t^*) < x^*_{m+1}(t^*). \end{cases} \tag{20}$$

然后，通过比较方程 (19) 和 (20) 以及方程 (12) 至 (16) 中的 KKT 条件，我们可以根据 $(x^*(t^*)\ z^*(t^*))$ 得到 G_{m+1}-最优对。实际上，可以通过以下方式构造 G_{m+1}-最优对 $(x^{(m+1)}, z^{(m+1)})$:

$$x^{(m+1)} = x^*(t^*), \text{ and } z^{(m+1)}_{i,j} = z^*_{i,j}(t^*) \text{ for } (i,j) \in E_m, \text{ and } z^{(m+1)}_{i_m,m+1} = t^*.$$

这一观察也表明，可以利用命题1确定 t^* 的符号。

为了找到所需的 t^*，我们从初始猜测 $t_0 = 0$ 开始。注意，当 $t_0 = 0$时，相应的原始-对偶最优对 $(x^*(t_0)\ z^*(t_0))$ 已知，其中 $x^*_i(t_0) = x_i^{(m)}$ 对于 $i \in V_m$，且 $x^*_{m+1}(t_0) = (f^*_{m+1})'(-t_0)$，以及 $z^*(t_0) = z^{(m)}$。然后，通过比较 $x^*_{m+1}(t_0)$ 和 $x^*_{i_m}(t_0)$，我们可以轻松检查 $t_0 = 0$ 是否满足 (20)。如果 $x^*_{m+1}(t_0) \neq x^*_{i_m}(t_0)$，我们可以利用命题1确定应当减小还是增大 t。不失一般性，假设 $f'_{m+1}(x^*_{i_m}(t_0)) = f'_{m+1}(x^{(m)}_{i_m}) > 0$。根据上述讨论和命题1，我们可知 $t^* < 0$。然后，我们采用活动集策略，迭代更新对 t^* 的猜测。

从初始猜测 $t_0 = 0$ 出发，我们将式 (18) 中与 E_m 对应的活动集记为

$$\mathcal{A}^0 = \{(i,j,\#) \mid (i,j) \in E_m,\ x^*_i(t_0) \,\#\, x^*_j(t_0)\}, \text{ where } \# \in \{<, =, >\}. \tag{21}$$

然后，我们将由 $\mathcal{A}^0_=$ 中的边所导出的等式约束加入问题 (18)，得到问题 (18) 的 $\mathcal{A}^0$-约化问题。关键观察是，该 $\mathcal{A}^0$-约化问题的原始-对偶最优解对可以写成参数 t 的半闭式函数，记为 $(x^0(t)\ z^0(t))$。接着，我们按如下方式构造问题 (18) 的对偶候选解 $\tilde{z}^0(t)$:

$$\tilde{z}^0_{i,j}(t) = \begin{cases} z^0_{i,j}(t), & \text{if } (i,j) \in \mathcal{A}^0_=, \\ z^*_{i,j}(t_0), & \text{otherwise}, \end{cases} \quad \forall\, (i,j) \in E_m.$$

我们将 证明如果 $(x^0(t)\ \tilde{z}^0(t))$ 满足 (19) 中的互补条件，即，

$$\tilde{z}^0_{i,j}(t) \in \begin{cases} \{-\lambda_{i,j}\}, & \text{if } x^0_i(t) > x^0_j(t), \\ [-\lambda_{i,j}, \mu_{i,j}], & \text{if } x^0_i(t) = x^0_j(t), \\ \{\mu_{i,j}\}, & \text{if } x^0_i(t) < x^0_j(t), \end{cases} \quad \forall\ (i,j) \in E_m,$$

那么，$(x^0(t)\ \tilde{z}^0(t))$ 就是问题 (18) 的原始-对偶最优解对。

基于这一观察，通过搜索满足上述互补条件的最小可能 t_1 ，构造新的 t^* ：$-\lambda_{i_m,}$ $_{m+1} \le t^* \le t_1 \le t_0 = 0$ and $(x^0(t_1)\ \tilde{z}^0(t_1))$。于是，我们有 $(x^*(t_1)\ z^*(t_1)) = (x^0(t_1)\ \tilde{z}^0(t_1))$，并且可以检查 t_1 是否满足系统 (20)。如果不满足，则构造新的活动集 $\mathcal{A}^1$ ，并继续上述过程，直到找到 t^* 。简而言之，我们的方法总结如下流程图：

$$(t_0, x^*(t_0), z^*(t_0), \mathcal{A}^0) \Rightarrow \cdots \Rightarrow (t_q, x^*(t_q), z^*(t_q), \mathcal{A}^q) \Rightarrow \cdots \Rightarrow (t^*, x^*(t^*), z^*(t^*), \mathcal{A}^*).$$

接下来，我们将讨论该过程的详细步骤，并证明 t^* 的搜索过程至多经过 $2m-1$ 步后终止。

在 t_q 且 $t^* < t_q \le t_0$时，我们假设 $(x^*(t_q),\ z^*(t_q))$ 和相应的活动集 $\mathcal{A}^q$ 已知。接着，我们构造如下以 $\mathcal{A}^q$ 为约化对象、参数为 $t \in \Re$ 的参数优化问题：

$$\begin{aligned} \min_{x \in \Re^{|V_{m+1}|}} \quad & \sum_{i \in V_{m+1}} f_i(x_i) + \sum_{(i,j) \in \mathcal{A}^q_>} \lambda_{i,j}(x_i - x_j) + \sum_{(i,j) \in \mathcal{A}^q_<} \mu_{i,j}(x_j - x_i) - t(x_{i_m} - x_{m+1}), \\ \text{s.t.} \quad & x_i = x_j, \quad \forall\ (i,j) \in \mathcal{A}^q_=, \end{aligned} \tag{22}$$

其唯一的原始-对偶最优对记为 $(x^q(t)\ z^q(t))$。若 $\mathcal{A}^q_= = \emptyset$，则令 $z^q(t) = \emptyset$。这里，我们要求 $\mathcal{A}^q$ 与 $(x^*(t_q)\ z^*(t_q))$ 满足以下相容性条件，这也作为归纳假设。

假设 2. 活动集 $\mathcal{A}^q$ 与原始-对偶对 $(x^*(t_q)\ z^*(t_q))$ 相容。也就是说， $x^*(t_q)$ 是问题 (22) 在 $t = t_q$ 处的最优解，即 $x^q(t_q) = x^*(t_q)$，且相应的对偶最优解 $z^q(t_q)$ 可通过 $z^q_{i\ ,\ j}(t_q) = z^*_{i\ ,\ j}(t_q)$ 对于 $(i,j) \in \mathcal{A}^q_=$ 构造得到。此外，有 $x^*_{i_m}(t_q) - x^*_{m+1}(t_q) > 0$。

我们强调，根据 $\mathcal{A}^0$ 的构造，不难发现，活动集 $\mathcal{A}^0$ 与原始-对偶解对 $(x^*(t_0)\ z^*(t_0))$ 是相容的，并且 $x^*_{i_m}(t_0) - x^*_{m+1}(t_0) > 0$。接下来，我们着重从 $(t_q, x^*(t_q)\ z^*(t_q)\ \mathcal{A}^q)$ 求得 $(t_{q+1}, x^*(t_{q+1})\ z^*(t_{q+1})\ \mathcal{A}^{q+1})$。

我们首先研究问题 (22) 对应的最优原始-对偶解对。具体而言，我们不针对每个 $t \ne t_q$ 求解问题 (22)，而是在下面的命题中推导出满足假设 2 时 $(x^q(t)\ z^q(t))$ 的半闭式公式。我们还表明，只要满足某些互补条件，就可以从 $(x^q(t)\ z^q(t))$ 得到问题 (18) 的最优原始-对偶解对。

命题 2. 设 $P_{\mathcal{A}^q}$ 是由 $\mathcal{A}^q$ 诱导的 G_m 的划分，且 $B^q \in P_{\mathcal{A}^q}$ 是满足 $i_m \in B^q$ 的子树。那么，在假设 2 下，对任意 $t \in \Re$，原始最优解 $x^q(t)$ 具有

以下形式:

$$\begin{cases} x_i^q(t) = x_i^*(t_q), \quad \forall\, i \in V_m \backslash V_{B^q}, \\ x_i^q(t) = \big((\sum\limits_{i \in V_{B^q}} f_i)^*\big)'\Big(t + \beta^q\Big), \quad \forall\ i \in V_{B^q}, \\ x_{m+1}^q(t) = (f_{m+1}^*)'(-t), \end{cases} \tag{23}$$

其中

$$\beta^q = \sum_{\substack{(i,k)\in E_m \\ i\in V_{B^q}, k\notin V_{B^q}}} z_{i,k}^*(t_q) - \sum_{\substack{(k,i)\in E_m \\ k\notin V_{B^q}, i\in V_{B^q}}} z_{k,i}^*(t_q).$$

选择 i_m 作为 B^q的祖先。于是，对于任意 $t \in \Re$， $z^q(t)$ 可表示为

$$\begin{cases} z_{i,j}^q(t) = z_{i,j}^*(t_q), \quad \forall (i,j) \in \mathcal{A}_=^q \backslash E_{B^q}, \\ z_{i,j}^q(t) = \begin{cases} \sum\limits_{l\in C_i} f_l'(x_l^q(t)) - \alpha_{i,j}^q, \ \ \textit{if } i \lhd j, \\ \sum\limits_{l\in C_j} -f_l'(x_l^q(t)) + \alpha_{i,j}^q, \ \ \textit{if } i \lhd j, \end{cases} \quad \forall (i,j) \in E_{B^q}, \end{cases} \tag{24}$$

其中，对于任意 $i \in V_{B^q}$， $C_i := \{j \in V_{B^q} \mid j \lhd i\} \cup \{i\}$，且

$$\alpha_{i,j}^q = \begin{cases} \sum\limits_{\substack{(l,k)\in E_m \\ l\in C_i, k\notin V_{B^q}}} z_{l,k}^*(t_q) - \sum\limits_{\substack{(k,l)\in E_m \\ l\in C_i, k\notin V_{B^q}}} z_{k,l}^*(t_q), \ \ \textit{if } i \lhd j, \\ \sum\limits_{\substack{(l,k)\in E_m \\ l\in C_j, k\notin V_{B^q}}} z_{l,k}^*(t_q) - \sum\limits_{\substack{(k,l)\in E_m \\ l\in C_j, k\notin V_{B^q}}} z_{k,l}^*(t_q), \ \ \textit{if } j \lhd i, \end{cases} \quad \forall (i,j) \in E_{B^q}.$$

令 $\Omega^q = \{(i,j) \in E_m \backslash E_{B^q} \mid$， i和 j中恰有一个属于 $V_{B^q}\}$。若

$$z_{i,j}^q(t) \in [-\lambda_{i,j}, \mu_{i,j}], \quad \forall\, (i,j) \in E_{B^q}, \tag{25}$$

$$z_{i,j}^*(t_q) \in \begin{cases} \{-\lambda_{i,j}\}, & \text{if } x_i^q(t) > x_j^q(t), \\ [-\lambda_{i,j}, \mu_{i,j}], & \text{if } x_i^q(t) = x_j^q(t), \\ \{\mu_{i,j}\}, & \text{if } x_i^q(t) < x_j^q(t), \end{cases} \quad \forall\, (i,j) \in \Omega^q, \tag{26}$$

则 $(x^q(t)\ \ z^q(t))$ 是 KKT 系统 (19) 的解，其中

$$\tilde{z}_{i,j}^q(t) = \begin{cases} z_{i,j}^q(t), \ \ \textit{if } (i,j) \in \mathcal{A}_=^q, \\ z_{i,j}^*(t_q), \ \ \textit{otherwise}, \end{cases} \quad \forall\, (i,j) \in E_m. \tag{27}$$

证明。在不失一般性的前提下，我们可以假设 $P_{\mathcal{A}^q} = \{B_k\}_{k=1}^K \cup B^q$ 其中 B_k、 $1 \le k \le K$ 和 B^q 是 G_m 的子树。然后，问题 (22) 可以分解为 $K+2$ 个独立子问题，分别对应子树 B_k、 B^q 和单元素集合 $\{m+1\}$。注意，参数 t 只

出现在对应子树 B^q 和单元素集合 $\{m+1\}$ 的子问题中。因此，根据假设 2，不难推得，对于任意 $t \in \Re$,

$$x_i^q(t) = x_i^*(t_q),\ i \in V_m \backslash V_{B^q}, \quad \text{and} \quad z_{i,j}^q(t) = z_{i,j}^*(t_q),\ (i,j) \in \mathcal{A}_=^q \setminus E_{B^q}.$$

与 $\{m+1\}$ 相关的子问题可通过 $x_{m+1}^q(t) = (f_{m+1}^*)'(-t)$轻松求解。因此，我们只需关注与子树 B^q 相关的子问题：

$$\min_{x \in \Re^{|V_{B^q}|}} \left\{ \sum_{i \in V_{B^q}} \hat{f}_i(x_i) - t x_{i_m} \mid x_i = x_j,\ \forall\, (i,j) \in E_{B^q} \right\}, \tag{28}$$

其中

$$\hat{f}_i(x_i) := f_i(x_i) + \sum_{\substack{k \notin V_{B^q} \\ (k,i) \in E_m}} z_{k,i}^*(t_q) x_i - \sum_{\substack{k \notin V_{B^q} \\ (i,k) \in E_m}} z_{i,k}^*(t_q) x_i, \quad \forall\, i \in V_{B^q}.$$

令 $\mathcal{L}$ 为问题 (28) 对应的拉格朗日函数

$$\mathcal{L}(x;z) = \sum_{i \in V_{B^q}} \hat{f}_i(x_i) - t x_{i_m} - \sum_{(i,j) \in E_{B^q}} z_{i,j}(x_i - x_j), \quad \forall\, (x,z) \in \Re^{|V_{B^q}|} \times \Re^{|E_{B^q}|}.$$

则问题 (28) 的最优原始-对偶解对满足以下 KKT 系统：

$$\begin{cases} x_i = x_j, \quad \forall\, (i,j) \in E_{B^q}, \\ f_i'(x_i) + \sum\limits_{\substack{k \notin V_{B^q} \\ (k,i) \in E_m}} z_{k,i}^*(t_q) + \sum\limits_{\substack{k \in V_{B^q} \\ (k,i) \in E_{B^q}}} z_{k,i} - \sum\limits_{\substack{k \notin V_{B^q} \\ (i,k) \in E_m}} z_{i,k}^*(t_q) - \sum\limits_{\substack{k \in V_{B^q} \\ (i,k) \in E_{B^q}}} z_{i,k} - 1_{\{i=i_m\}} t = 0, \quad \forall\, i \in V_{B^q}. \end{cases} \tag{29}$$

对所有 $i \in V_{B^q}$求和，可由上述系统推出

$$\sum_{i \in V_{B^q}} f_i'(x_i^q(t)) = - \sum_{\substack{(k,i) \in E_m \\ k \notin V_{B^q}, i \in V_{B^q}}} z_{k,i}^*(t_q) + \sum_{\substack{(i,k) \in E_m \\ i \in V_{B^q}, k \notin V_{B^q}}} z_{i,k}^*(t_q) + t,$$

即,

$$x_i^q(t) = ((\sum_{i \in V_{B^q}} f_i)^*)'(t + \sum_{\substack{(i,k) \in E_m \\ i \in V_{B^q}, k \notin V_{B^q}}} z_{i,k}^*(t_q) - \sum_{\substack{(k,i) \in E_m \\ k \notin V_{B^q}, i \in V_{B^q}}} z_{k,i}^*(t_q)), \quad \forall\ i \in V_{B^q}.$$

接下来，我们从上述 KKT 系统 (29) 中得到与 $z_{i,j}$ 对于 $(i,j) \in E_{B^q}$相对应的以下线性方程组：

$$\sum_{k:(i,k) \in E_{B^q}} z_{i,k} - \sum_{k:(k,i) \in E_{B^q}} z_{k,i} = f_i'(x_i^q(t)) + \sum_{\substack{k \notin V_{B^q} \\ (k,i) \in E_m}} z_{k,i}^*(t_q) - \sum_{\substack{k \notin V_{B^q} \\ (i,k) \in E_m}} z_{i,k}^*(t_q), \quad \forall\, i \in V_m \setminus \{i_m\}.$$

由于 i_m 是子树 B的祖先，根据引理 1，我们得到 $z_{i,j}^q(t)$，$(i,j) \in E_{B^q}$的更新公式。因此，我们证明了 (24)。

最后，不难看出，若假设条件 (25) 和 (26) 成立，则 $x^q(t)$ 和 $z^q(t)$ 满足 KKT 系统 (19) 中的互补条件。注意到 (27) 和 KKT 系统 (29)，(19) 中的其余方程自动成立。

□

采用半　　根据命题 2 中的 -闭式公式，我们计算出以下下界 $\Delta t_q \le 0$　　：

$$\Delta t_q := \min\{\Delta t \mid \text{(25) and (26) hold for all } t \in [t_q + \Delta t, t_q]\}.$$

计算分为两部分。首先，我们关注对于（i，j）$\in E_{B^q}$，$z^q_{i,\,j}$（t）的值。对于任意（i，j）$\in E_{B^q}$，由（24）可知，$z^q_{i,\,j}$（t_q）$\in [-\lambda_{i,j}, \mu_{i,j}]$和$z^q_{i,\,j}$（$t$）关于 t 在 $i \triangleleft j$时递增，在 $j \triangleleft i$ 时递减。我们将阈值 Δ（E_{B^q}）定义如下:

$$\Delta(E_{B^q}) := \begin{cases} \max\limits_{(i,j)\in E_{B^q}} \Delta t_{i,j}, & \text{if } E_{B^q} \neq \emptyset, \\ -\infty, & \text{otherwise.} \end{cases} \tag{30}$$

这里，每个$\Delta t_{i,\,j} \le 0$ 满足

$$z^q_{i,j}(t_q + \Delta t_{i,j}) = -\lambda_{i,j}, \quad \text{if } i \triangleleft j, \quad \text{and} \quad z^q_{i,j}(t_q + \Delta t_{i,j}) = \mu_{i,j}, \quad \text{if } j \triangleleft i. \tag{31}$$

接下来，考察 Ω^q 中与（26）对应的边。为此，我们将 Ω^q 分为两部分，即:

$$\Omega^q_+ = \{(i,j) \in \Omega^q \mid i \in V_{B^q},\, j \in V_m \backslash V_{B^q}\} \text{ and } \Omega^q_- = \{(i,j) \in \Omega^q \mid i \in V_m \backslash V_{B^q},\, j \in V_{B^q}\}, \tag{32}$$

并分别处理。由（23）可知，对于所有 $i \in V_{B^q}$，x^q_i（t）取值相同，且关于 t递增。因此，对于任意 $i \in V_{B^q}$，我们可将 x_{B^q}（t）$= x^q_i$（t）简记为 x_{B^q}（t）。然后，我们计算阈值 Δ（Ω^q）：$=$ max{Δ（Ω^q_+） Δ（Ω^q_-）}，其中

$$\Delta(\Omega^q_+) := \begin{cases} \Delta\bar{t} \text{ satisfying } x_{B^q}(t_q + \Delta\bar{t}) = \max\limits_{(i,j)\in\Omega^q_+ \cap \mathcal{A}^q_>} x^*_j(t_q), & \text{if } \Omega^q_+ \cap \mathcal{A}^q_> \neq \emptyset, \\ -\infty, \text{ otherwise,} \end{cases} \tag{33}$$

and

$$\Delta(\Omega^q_-) := \begin{cases} \Delta\bar{t} \text{ satisfying } x_{B^q}(t_q + \Delta\bar{t}) = \max\limits_{(i,j)\in\Omega^q_- \cap \mathcal{A}^q_<} x^*_i(t_q), & \text{if } \Omega^q_- \cap \mathcal{A}^q_< \neq \emptyset, \\ -\infty, \text{ otherwise.} \end{cases} \tag{34}$$

容易验证,

$$\Delta t_q = \max\{\Delta(E_{B^q}), \Delta(\Omega^q)\}. \tag{35}$$

因此，利用命题 2，对于任意 $t \in [t_q + \Delta t_q, t_q]$，我们可以得到问题 (18) 的最优解 $x^*(t)$ 的半闭式形式，以及其对应的对偶最优解 $z^*(t)$。

现在，我们可以讨论 t_{q+1} 的搜索了。注意，根据假设 2，我们有

$$x^q_{i_m}(t_q) - x^q_{m+1}(t_q) = x^*_{i_m}(t_q) - x^*_{m+1}(t_q) > 0.$$

利用命题 2 中的闭式公式，我们知道 $x^q_{i_m}(t) - x^q_{m+1}(t)$ 关于 t 严格递增~，因此可以通过求解以下单变量非线性方程得到唯一的 $\Delta t_q < 0$：

$$x^q_{i_m}(t_q + \Delta\widetilde{t_q}) - x^q_{m+1}(t_q + \Delta\widetilde{t_q}) = 0,$$

这实际上就是以下单变量强凸优化问题对应的最优性条件:

$$t_q + \Delta\widetilde{t}_q = \operatorname*{argmin}_t \left\{ \Big(\sum_{i\in V_{Bq}} f_i \Big)^*(t+\beta^q) + (f^*_{m+1})(-t) \right\}.$$

~因此可以保证 Δt_q 存在。接着，我们设

$$t_{q+1} = \max\{t_q + \Delta t_q, t_q + \Delta\widetilde{t}_q, -\lambda_{i_m,m+1}\}. \tag{36}$$

显然，始终有 $t_{q+1} \in [t_q + \Delta t_q, t_q]$ 且

$$\begin{aligned} x^*_{i_m}(t_{q+1}) - x^*_{m+1}(t_{q+1}) &= x^q_{i_m}(t_{q+1}) - x^q_{m+1}(t_{q+1}) \\ &\geq x^q_{i_m}(t_q + \Delta\widetilde{t}_q) - x^q_{m+1}(t_q + \Delta\widetilde{t}_q) = 0. \end{aligned} \tag{37}$$

接下来，我们通过以下引理揭示 t_{q+1} 与 t^* 之间的关系。

引理 3. 有$-\lambda_{i_m,\ m+1} \leq t^* \leq t_{q+1} \leq t_q \leq 0$。此外，当且仅当 $t_{q+1} = t^*$ ($x^*_{i_m}$) $t_{q+1} - x^*_{m+1}(t_{q+1}) = 0$ 或者$t_{q+1} = -\lambda_{i_m,\ m+1}$时，$t_{q+1} = t^*$ 成立。

证明. 若 $t^* > t_{q+1}$，由式 (36) 可得$t^* > t_{q+1} \geq -\lambda_{i_m,\ m+1}$。进而由式 (20) 可得

$$x^q_{i_m}(t^*) - x^q_{m+1}(t^*) = x^*_{i_m}(t^*) - x^*_{m+1}(t^*) = 0.$$

然而，由式 (37) 以及 $x^q_{i_m}(t) - x^q_{m+1}(t)$ 的严格单调性可知

$$x^q_{i_m}(t^*) - x^q_{m+1}(t^*) > x^q_{i_m}(t_{q+1}) - x^q_{m+1}(t_{q+1}) \geq 0.$$

由此得到矛盾。因此，$t^* \leq t_{q+1}$。

接下来，如果 $x^*_{i_m}(t_{q+1}) - x^*_{m+1}(t_{q+1}) = 0$ 或 $t_{q+1} = -\lambda_{i_m,\ m+1}$，则很容易验证，$t_{q+1}$,$x^*_{i_m}(t_{q+1})$ 和 $x^*_{m+1}(t_{q+1})$ 满足 (20)，即 $t^* = t_{q+1}$。反过来，如果 $t^* = t_{q+1}$，则有 $t_{q+1} \geq -\lambda_{i_m,\ m+1}$。如果 $t_{q+1} > -\lambda_{i_m,\ m+1}$，则可直接由 (20) 得出 $x^*_{i_m}(t^*) - x^*_{m+1}(t^*) = 0$。至此，我们完成了该引理的证明。□

~注释 1. 在整个搜索过程中，对于t^*，只需至多计算一次 Δt_q。事实上，令

$$\Delta_* := \begin{cases} x^q_{i_m}(t_q + \Delta t_q) - x^q_{m+1}(t_q + \Delta t_q), & \text{if } \Delta t_q > -\infty, \\ -\infty, & \text{otherwise.} \end{cases}$$

若 $\Delta_* \geq 0$，则由 $x^q_{i_m}(t) - x^q_{m+1}(t)$ 的严格单调性可知，必有 $\Delta\widetilde{t}_q \leq \Delta t_q$。在这种情况下，我们可以直接设定

$$t_{q+1} = \max\{t_q + \Delta t_q, -\lambda_{i_m,m+1}\},$$

~而无需计算 Δt_q。只有当 $\Delta_* < 0$ 时，我们才计算 Δt_q 并设定

$$t_{q+1} = \max\{t_q + \Delta\widetilde{t}_q, -\lambda_{i_m,m+1}\}.$$

~然后，根据引理 3，可得 $t_{q+1} = t^*$。因此，Δt_q 至多只需计算一次。

若 $t_{q+1} \neq t^*$，由式 (36)、(37) 和引理 3 可知 $t^* < t_{q+1}$，且

$$t_{q+1} = t_q + \Delta t_q, \quad \text{and} \quad x^*_{i_m}(t_{q+1}) - x^*_{m+1}(t_{q+1}) > 0. \tag{38}$$

接下来，我们给出 出 $\mathcal{A}^{q+1}$ 的构造细节。令 $\mathcal{M}(E_{B^q}) = \mathcal{M}(E^+_{B^q}) \cup \mathcal{M}(E^-_{B^q})$，其中

$$\begin{cases} \mathcal{M}(E^+_{B^q}) = \{(i,j) \in E_{B^q} \mid \Delta t_{i,j} = \Delta t_q, \text{ and } i \triangleleft j\}, \\ \mathcal{M}(E^-_{B^q}) = \{(i,j) \in E_{B^q} \mid \Delta t_{i,j} = \Delta t_q, \text{ and } j \triangleleft i\}, \end{cases} \tag{39}$$

以及 $\mathcal{M}(\Omega^q) = \mathcal{M}(\Omega^q_+) \cup \mathcal{M}(\Omega^q_-)$，其中

$$\begin{cases} \mathcal{M}(\Omega^q_+) = \{(i,j) \in \Omega^q_+ \cap \mathcal{A}(t_q)_> \mid x^q_i(t_q + \Delta t_q) = x^*_j(t_q)\}, \\ \mathcal{M}(\Omega^q_-) = \{(i,j) \in \Omega^q_- \cap \mathcal{A}(t_q)_< \mid x^q_j(t_q + \Delta t_q) = x^*_i(t_q)\}. \end{cases} \tag{40}$$

活动集 $\mathcal{A}^{q+1}$ 通过以下方式构造

$$\begin{cases} \mathcal{A}^{q+1}_= = \big(\mathcal{A}^q_= \cup \mathcal{M}(\Omega^q)\big) \backslash \mathcal{M}(E_{B^q}), \\ \mathcal{A}^{q+1}_> = \big(\mathcal{A}^q_> \cup \mathcal{M}(E^+_{B^q})\big) \backslash \mathcal{M}(\Omega^q_+), \\ \mathcal{A}^{q+1}_< = \big(\mathcal{A}^q_< \cup \mathcal{M}(E^-_{B^q})\big) \backslash \mathcal{M}(\Omega^q_-). \end{cases} \tag{41}$$

与 (27) 类似，我们可以由 $z^q(t_{q+1})$ 构造 $\widetilde{z}^q(t_{q+1})$，如下所示：

$$\widetilde{z}^q_{i,j}(t_{q+1}) = \begin{cases} z^q_{i,j}(t_{q+1}), & \text{if } (i,j) \in \mathcal{A}^q_=, \\ z^*_{i,j}(t_q), & \text{otherwise}, \end{cases} \qquad \forall\, (i,j) \in E_m.$$

于是，我们得到问题 (18) 的最优原始-对偶解对 $(x^*(t_{q+1})\ z^*(t_{q+1})) = (x^q(t_{q+1})\ \widetilde{z}^q(t_{q+1}))$，其中 $t = t_{q+1}$。

接下来，根据 (41) 中 $\mathcal{A}^{q+1}$ 的构造以及 (36) 中 t_{q+1} 的计算步骤，可以很容易地验证，新的活动集 $\mathcal{A}^{q+1}$ 与原始-对偶对 $(x^*(t_{q+1})\ z^*(t_{q+1}))$ 是兼容的。结合 (38)，我们可以对 $q \in \mathbb{N}$ 进行归纳，从而得到：对于所有 $q \in \mathbb{N}$，只要 $t_q \neq t^*$，就始终有 $\mathcal{A}^q$ 与 $(x^*(t_q)\ z^*(t_q))$ 兼容，并且

$$x^*_{i_m}(t_q) - x^*_{m+1}(t_q) > 0.$$

因此，我们可以迭代地重复上述搜索过程，即从 $(t_q, x^*(t_q)\ z^*(t_q)\ \mathcal{A}^q)$ 到 $(t_{q+1}, x^*(t_{q+1})\ z^*(t_{q+1})\ \mathcal{A}^{q+1})$，直到得到 t^*。搜索过程的细节总结于算法 2 中。我们将其称为 update$^-$ 子程序，因为在这种情况下 $t^* < 0$。对应于 $t^* > 0$ 情形的过程称为 update$^+$ 子程序，可由 update$^-$ 子程序轻松调整得到。update$^+$ 子程序的细节见附录。

在介绍 generate 子程序的详细内容之前，我们先通过以下引理对活动集 $\mathcal{A}^{q+1}$ 作出一些关键观察。

引理 4. 对于任意给定的 $q \in \mathbb{N}$，以下命题成立：

(a) 如果 $t_{q+1} \neq t^*$，则 $\mathcal{A}^{q+1}_= \neq \mathcal{A}^q_=$；

Al算法 2 $(t_{q+1}, x^*(t_{q+1})\ z^*(t_{q+1})\ \mathcal{A}^{q+1}, t^*) = 更新^{-}(t_q, x^*(t_q)\ z^*(t_q)\ \mathcal{A}^q, \lambda \qquad)$

Input: $(t_q, x^*(t_q), z^*(t_q), \mathcal{A}^q)$, $\lambda \geq 0$;
Compute $\Delta(E_{B^q}), \Delta(\Omega_+^q), \Delta(\Omega_-^q)$ via definitions (30), (33) and (34)
$\Delta(\Omega^q) = \max\{\Delta(\Omega_-^q), \Delta(\Omega_+^q)\}$
$\Delta t_q = \max\{\Delta(E_{B^q}), \Delta(\Omega^q)\}$
$\Delta^* = x_{i_m}^q(t_q + \Delta t_q) - x_{m+1}^q(t_q + \Delta t_q)$
if $\Delta^* \geq 0$ **then**
    $t_{q+1} = \max\{t_q + \Delta t_q, -\lambda\}$
else
    $\Delta\widetilde{t}^q = -t_q + \underset{t}{\operatorname{argmin}} \left\{(\sum_{i\in V_{B^q}} f_i)^*(t + \beta^q) + (f_{m+1}^*)(-t)\right\}$
    $t_{q+1} = \max\{t_q + \Delta\widetilde{t}^q, -\lambda\}$
end if
$(x^*(t_{q+1}), z^*(t_{q+1})) = (x^q(t_{q+1}), \widetilde{z}^q(t_{q+1}))$
if $t_{q+1} = -\lambda$ **or** $x_{i_m}^*(t_{q+1}) = x_{m+1}^*(t_{q+1})$ **then**
    $t^* = t_{q+1}$
    Let $\mathcal{A}^{q+1} = \{(i, j, \#) \mid (i, j) \in E_m,\ x_i^*(t_{q+1}) \,\#\, x_j^*(t_{q+1})\}$
else
    $t^* = \emptyset$
    Update $\mathcal{A}^{q+1}$ from $\mathcal{A}^q$ via (41)
end if
Output: $(t_{q+1}, x^*(t_{q+1}), z^*(t_{q+1}), \mathcal{A}^{q+1}, t^*)$

(b) 若 $(i, j) \in \mathcal{M}(E_{B^q})$，则对于任意满足 $\widehat{q} > q$ 和 $t_{\widehat{q}} \neq t^*$ 的 $\widehat{q} \in \mathbb{N}$，$(i, j) \notin \mathcal{A}_{=}^{\widehat{q}}$.

证明。我们先证明 (a)。若 $t_{q+1} \neq t^*$，由 (38) 可得 $t_{q+1} = t_q + \Delta t_q > t^*$。因此，两个集合 $\mathcal{M}(E_{B^q})$ 和 $\mathcal{M}(\Omega^q)$ 中至少有一个非空。由于 $\mathcal{A}_{=}^{q+1} = \big(\mathcal{A}^{q+1} \cup \mathcal{M}(\Omega^q)\big)\backslash\mathcal{M}(E_{B^q})$ 且 $\mathcal{M}(E_{B^q}) \cap \mathcal{M}(\Omega^q) = \emptyset$，所需结果由此得出。

接下来证明 (b)。我们先考虑 $i \triangleleft j$ 的情形。若 $(i, j) \in \mathcal{M}(E_{B^q})$ 且 $i \triangleleft j$，由 (31)、(39) 和 (41) 可知

$$z_{i,j}^q(t_q + \Delta t_q) = -\lambda_{i,j}, \text{ and } (i, j) \in \mathcal{M}(E_{B^q}^+) \subseteq \mathcal{A}_{>}^{q+1}.$$

由于 $(i, j) \in \mathcal{A}_{>}^{q+1}$，因此 i 和 j 中至少有一个不属于 B^{q+1}，即 $(i, j) \notin E_{B^{q+1}}$。由于 $i \triangleleft j$，我们有以下两种可能的情况:

(i) $j \in B^{q+1}, i \notin B^{q+1}$。在这种情况下，我们有 $(i, j) \in \Omega_-^{q+1}$。由于 $(i, j) \in \mathcal{A}_{>}^{q+1}$，由 (40) 可得 $(i, j) \notin \mathcal{M}(\Omega_+^{q+1})$。因此，(41) 意味着 $(i, j) \in \mathcal{A}_{>}^{q+2}$。

(ii) $j \notin B^{q+1}, i \notin B^{q+1}$。由 (32) 可知，$(i, j) \notin \Omega^{q+1}$。因此，由 (40) 和 (41) 可得 $(i, j) \in \mathcal{A}_{>}^{q+2}$。

因此，在这两种情况下，我们都有 $(i, j) \notin \Omega_+^{q+2}$ 和 $(i, j) \in \mathcal{A}_{>}^{q+2}$。根据归纳法，我们可以证明，对于所有 $\widehat{q} > q$，$\widehat{\quad}$ $\widehat{(i, j) \notin \Omega_+^q}$ 和 $(i, j) \in \mathcal{A}_{>}^q$。

$\widehat{\quad}$ $\widehat{\quad}$类似地，对于 $j \triangleleft i$ 的情形，我们可以得到 $(i, j) \notin \Omega_-^q$，且 $(i, j) \in \mathcal{A}_{<}^q$ 对所有 $\widehat{q} > q$ 均成立。因此，我们完成了证明。 □

有了这两个子程序 update$^-$ 和 update$^+$，我们现在可以详细介绍算法 3 中的 generate 子程序。显然，generate 子程序的复杂度主要取决于 while 循环的执行次数（即算法 3 中第 9–12 行和第 15–18 行）。

算法 3 generate 子程序：$(x^{(m+1)}, z^{(m+1)}) = \text{generate}(x^{(m)}, z^{(m)}, G_{m+1})$

输入：$x^{(m)} \in \Re^m, z^{(m)} \in \Re^{m-1}, G_{m+1} = (V_{m+1}, E_{m+1})$
令 $x_i^*(0) = x_i^{(m)}$ 对 $i \in V_m$ 和 $x^*_{m+1}(0) = (f^*_{m+1})'(0)$
令 $z_{i,j}^*(0) = z_i^{(m)}$，$_j$ 对于 $(i,j) \in E_m$ 和 $t^* = \emptyset$

如果 $f'_{m+1}(x^*_{i_m}(0)) = 0$ 那么
  $t^* = 0$
否则如果 $f'_{m+1}(x^*_{i_m}(0)) > 0$ 那么
  令 $t_0 = 0$、$q = 0$ 和 $\mathcal{A}^0$ 为根据式 (21) 从 $x^*(0)$ 构造的活动集
  当 $t^* = \emptyset$ 时执行
    $(t_{q+1}, x^*(t_{q+1}), z^*(t_{q+1}), \mathcal{A}^{q+1}, t^*) = \text{更新}^-(t_q, x^*(t_q), z^*(t_q), \mathcal{A}^q, \lambda_{i_m,\ m+1})$
    $q = q + 1$
  结束循环
否则
  令 $t_0 = 0$, $q = 0$ 和 $\mathcal{A}^0$ 为根据 (21) 从 $x^*(0)$ 构造的活动集
  当 $t^* = \emptyset$ 时
    $(t_{q+1}, x^*(t_{q+1}), z^*(t_{q+1}), \mathcal{A}^{q+1}, t^*) = \text{更新}^+(t_q, x^*(t_q), z^*(t_q), \mathcal{A}^q, \mu_{i_m,\ m+1})$
    $q = q + 1$
  结束 while 循环
结束 if 语句
令 $x^{(m+1)} = x^*(t^*)$、$z_{i,j}^{(m+1)} = z_{i,j}^*(t^*)$ 对于 $(i,j) \in E_m$，并令 $z_{i_m\ m+1}^{(m+1)} = t^*$
返回：$(x^{(m+1)}, z^{(m+1)}) \in \Re^{m+1} \times \Re^m$

引理 5. generate 子程序中的 while 循环最多经过 $2m-1$ 次迭代即可找到 t^*。

证明. 不失一般性，我们只考虑 $f'_{m+1}(x^*_{i_m}(0)) > 0$ 的情形，即 $t^* < 0$。假设 while 循环执行 $2m-2$ 次后，仍未找到 t^*。也就是说，对于所有 $i = 0$，算法生成了 $\{(t_i, x^*(t_i), z^*(t_i), \mathcal{A}^i)\}_{i=1}^{2m-2}$ 和 $t_i > t^*$，$\ldots, 2m-2$。由引理4(a)，我们知道

$$\mathcal{A}^q_= \neq \mathcal{A}^{q+1}_=, \quad \forall\, q = 0, \ldots, 2m-3. \tag{42}$$

接下来，根据引理 4(b)，我们注意到，若某条边 $(i,j) \in E_m$ 从 $\mathcal{A}^q_=$ 中因某个 q而被移除，则对于所有 $2m-2 \geq \widehat{q} \geq q \geq 0$，都有 $(i,j) \notin \mathcal{A}^{\widehat{q}}_=$。因此，对于每条边 $(i,j) \in E_m$，它至多只能被添加到 $\mathcal{A}^q_=$ 一次并从中移除一次。结合式 (42) 以及 $|E_m| = m-1$这一事实可知，在 t_{2m-2}处，E_m 中的每条边都已被添加到某个 $\mathcal{A}^q_=$并从中移除。因此，$\mathcal{A}^{2m-2}_= = \emptyset$，且集合 $\mathcal{A}^{2m-2}_>$和 $\mathcal{A}^{2m-2}_<$在后续迭代中保持不变，即 $E_{B^q} = \emptyset$、$\Omega^{2m-2}_+ \cap A^{2m-2}_> = \emptyset$ 和 $\Omega^{2m-2}_- \cap A^{2m-2}_< = \emptyset$。因此，根据式 (35) 中的定义，我们有 $\Delta t_{2m-2} = -\infty$。由式 (36) 和引理 3 可得 $t_{2m-1} = t^*$，证明完毕。 □

引理 5 保证可以通过 generate 子程序高效地找到 t^*。同时，还可得到 t^*以及 G_{m+1}-最优对 $(x^{(m+1)}, z^{(m+1)})$，即 generate 子程序的输出。因此，我们自然可以得出算法 1 的正确性。

Theo注 1. 算法 1 的输出 $x^{(n)}$ 是问题 (1) 的最优解。

本节末尾，我们简要分析算法 1 的最坏情况复杂度。这里，假设对于给定的强凸可微函数 f 和 $x \in \Re$，求出满足 $f'(t) = x$ 的 t 的计算复杂度为 $\mathcal{O}(1)$。那么，update$^-$ （以及 update$^+$）的计算复杂度为 $\mathcal{O}(m)$。根据引理 5，可知 generate 子程序的计算复杂度为 $\mathcal{O}(m^2)$。因此，算法 1 的计算复杂度为 $\mathcal{O}(n^3)$。

4 结论

本文聚焦于具有树诱导的广义序约束的凸保序回归问题 (1)。受 PAVA 成功经验的启发，我们精心设计了一种高效的基于活动集的递归方法 ASRA 来求解 (1)。在温和的假设下，我们证明 ASRA 具有多项式时间计算复杂度。

5 附录

5.1 关于 G的树形图假设

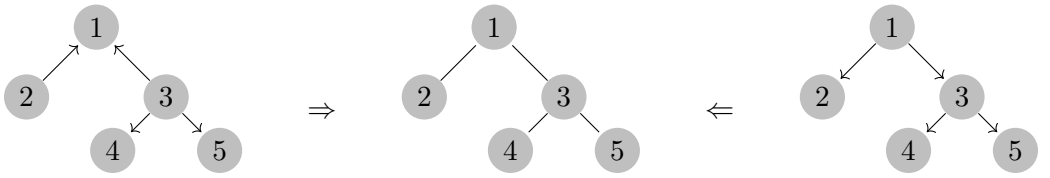
对于问题 (1) 的表述中给定的 $G = (V, E)$，令 $\widehat{G} = (V, \widehat{E})$ 为与 G 共享同一底层图的树形图。因此，对于任意 $(i, j) \in E$，要么有 $(i, j) \in \widehat{E}$，要么有 $(j, i) \in \widehat{E}$。于是，对于任意 $(i, j) \in E$，令

$$\widehat{\lambda}_{i,j} = \begin{cases} \lambda_{i,j}, & \text{if } (i,j) \in E, \\ \mu_{j,i}, & \text{if } (j,i) \in E, \end{cases} \quad \text{and} \quad \widehat{\mu}_{i,j} = \begin{cases} \mu_{i,j}, & \text{if } (i,j) \in E, \\ \lambda_{j,i}, & \text{if } (j,i) \in E. \end{cases}$$

容易验证，问题 (1) 等价于以下优化问题：

$$\min_{x \in \Re^V} \quad \sum_{i \in V} f_i(x_i) + \sum_{(i,j) \in \widehat{E}} \widehat{\lambda}_{i,j}(x_i - x_j)_+ + \sum_{(i,j) \in \widehat{E}} \widehat{\mu}_{i,j}(x_j - x_i)_+.$$

因此，我们可以假设 (1) 中的有向树 G 是一棵树形图。



(a) 有向树 G （b） G and 的底层图 $\widehat{G}$ （c）一棵树形图 $\widehat{G}$

图2: 一棵有向树 G 和一棵树形图 G 具有相同的底层图 h.

接下来，我们讨论 G 的分解。对于一棵树形图 $G=(V,E)$，令 $n=|V|$。不失一般性，我们假设节点 1 是 G 的根，并将 G 中的节点排列为：对于任意边 $(i,j)\in E$，总有 $i<j$。然后，我们定义 $G_n=G$，并令 $G_{m-1}=(V_{m-1},E_{m-1})$ 为从 $G_m=(V_m,E_m)$ 中删除节点 m 及其相关边后得到的子图，其中 $n\geq m\geq 2$。由于对于任意 $(i,j)\in E$，都有 $i<j$，因此我们知道节点 m 必定是 G_m 的叶节点；于是，根据 [23]，G_{m-1} 仍是一棵有向树，且对于 $m=2$、...、n，$G_{m-1}\subset G_m$。容易验证，对于 $1\leq m\leq n$，$V_m=\{1,2,...,m\}$，并且 $\{(i_m,m+1)\}=E_{m+1}\backslash E_m$，其中对于 $1\leq m\leq n-1$，$i_m\in V_m$。

5.2 Theupdate$^+$ 子程序

这里简要介绍 update$^+$ 子程序，它对应于 $t^*>0$ 的情形。假设我们已得到一个猜测 t_q，它满足 $t^*\ 0=t_0\leq t_q<t^*\leq\mu_{i_m,\ m+1}$。与此同时，相应的原始 - 对偶最优解对 $(x^*(t_q),\ z^*(t_q))$ 和活动集 $\mathcal{A}^q$ 已可用，使得 $\mathcal{A}^q$ 和 $(x^*(t_q)\ z^*(t_q))$ 相容，并且 $x^*_{i_m}(t_q)-x^*_{m+1}(t_q)<0$。那么，命题 2 中 $\mathcal{A}^q$-约化问题的半闭式公式 (23) 和 (24) 仍然成立。

这里我们需要搜索

$$\Delta t_q:=\max\{\Delta t\mid (25)\text{ and }(26)\text{ hold for all }t\in[t_q,t_q+\Delta t]\}.$$

首先，令

$$\Delta(E_{B^q})=\begin{cases}\min\limits_{(i,j)\in E_{B^q}}\Delta t_{i,j}, & \text{if } E_{B^q}\neq\emptyset,\\ +\infty, & \text{otherwise,}\end{cases}\tag{43}$$

其中，每个 $\Delta t_{i,\ j}\geq 0$ 都满足：

$$z^q_{i,j}(t_q+\Delta t_{i,j})=\mu_{i,j},\quad \text{if } i\lhd j,\quad \text{and}\quad z^q_{i,j}(t_q+\Delta t_{i,j})=-\lambda_{i,j},\quad \text{if } j\lhd i.$$

接下来，令 $\Delta(\Omega^q)=\min\{\Delta(\Omega^q_+)\ \Delta(\Omega^q_-)\}$，其中

$$\Delta(\Omega^q_+):=\begin{cases}\Delta\bar{t}\text{ satisfying }x_{B^q}(t_q+\Delta\bar{t})=\min\limits_{(i,j)\in\Omega^q_+\cap\mathcal{A}^q_<}x^*_j(t_q), & \text{if }\ \Omega^q_+\cap\mathcal{A}^q_<\neq\emptyset,\\ +\infty,\ \text{otherwise,}\end{cases}\tag{44}$$

and

$$\Delta(\Omega^q_-):=\begin{cases}\Delta\bar{t}\text{ satisfying }x_{B^q}(t_q+\Delta\bar{t})=\min\limits_{(i,j)\in\Omega^q_-\cap\mathcal{A}^q_>}x^*_i(t_q), & \text{if }\Omega^q_-\cap\mathcal{A}^q_>\neq\emptyset,\\ +\infty,\ \text{otherwise.}\end{cases}\tag{45}$$

然后，$\Delta t_q=\min\{\Delta(E_{B^q})\ \Delta(\Omega^q)\}$。通过求解 $x^q_{i_m}(t_q+\Delta\widetilde{t}_q)-x^q_{m+1}(t_q+\Delta\widetilde{t}_q)=0$ 计算 $\Delta\widetilde{t}_q\geq 0$，并设为

$$t_{q+1}=\min\{t_q+\Delta t_q,t_q+\Delta\widetilde{t}_q,\mu_{i_m,m+1}\}.$$

若 $t_{q+1}<t^*$，我们将按以下方式更新活动集 $\mathcal{A}^{q+1}$。令 $\mathcal{M}(E_{B^q})=\mathcal{M}(E^+_{B^q})\cup\mathcal{M}(E^-_{B^q})$，其中

$$\begin{cases}\mathcal{M}(E^+_{B^q})\ =\ \{(i,j)\in E_{B^q}\mid\Delta t_{i,j}=\Delta t_q,\ \text{and }i\lhd j\},\\ \mathcal{M}(E^-_{B^q})\ =\ \{(i,j)\in E_{B^q}\mid\Delta t_{i,j}=\Delta t_q,\ \text{and }j\lhd i\},\end{cases}$$

以及 $\mathcal{M}(\Omega^q) = \mathcal{M}(\Omega^q_+) \cup \mathcal{M}(\Omega^q_-)$，其中

$$\begin{cases} \mathcal{M}(\Omega^q_+) &= \{(i,j) \in \Omega^q_+ \cap \mathcal{A}(t_q)_< \mid x^q_i(t_q + \Delta t_q) = x^*_j(t_q)\}, \\ \mathcal{M}(\Omega^q_-) &= \{(i,j) \in \Omega^q_- \cap \mathcal{A}(t_q)_> \mid x^q_j(t_q + \Delta t_q) = x^*_i(t_q)\}. \end{cases}$$

那么，$\mathcal{A}^{q+1}$ 可通过以下方式获得

$$\begin{cases} \mathcal{A}^{q+1}_= = \big(\mathcal{A}^q_= \cup \mathcal{M}(\Omega^q)\big) \backslash \mathcal{M}(E_{B^q}), \\ \mathcal{A}^{q+1}_> = \big(\mathcal{A}^q_> \cup \mathcal{M}(E^-_{B^q})\big) \backslash \mathcal{M}(\Omega^q_-), \\ \mathcal{A}^{q+1}_< = \big(\mathcal{A}^q_< \cup \mathcal{M}(E^+_{B^q})\big) \backslash \mathcal{M}(\Omega^q_+). \end{cases} \tag{46}$$

我们在算法4中总结了更新$^+$ 子程序。

算法4 $(t_{q+1},\ x^*(t_{q+1})\ \ z^*(t_{q+1})\ \ \mathcal{A}^{q+1},\ t^*) =$ 更新$^+$ $(t_q,\ x^*(t_q)\ \ z^*(t_q)\ \ \mathcal{A}^q,\ \mu)$

Input: $(t_q, x^*(t_q), z^*(t_q), \mathcal{A}^q)$, $\mu \geq 0$;
Compute $\Delta(E_{B^q}), \Delta(\Omega^q_+), \Delta(\Omega^q_-)$ via definitions (43), (44) and (45)
$\Delta(\Omega^q) = \min\{\Delta(\Omega^q_-), \Delta(\Omega^q_+)\}$
$\Delta t_q = \min\{\Delta(E_{B^q}), \Delta(\Omega^q)\}$
$\Delta^* = x^q_{i_m}(t_q + \Delta t_q) - x^q_{m+1}(t_q + \Delta t_q)$
if $\Delta^* \leq 0$ **then**
    $t_{q+1} = \min\{t_q + \Delta t_q, \mu\}$
else
    $\Delta\widetilde{t}_q = -t_q + \underset{t}{\operatorname{argmin}} \left\{(\sum_{i \in V_{B^q}} f_i)^*(t + \beta^q) + (f^*_{m+1})(-t)\right\}$
    $t_{q+1} = \min\{t_q + \Delta\widetilde{t}_q, \mu\}$
end if
$(x^*(t_{q+1}), z^*(t_{q+1})) = (x^q(t_{q+1}), \widetilde{z}^q(t_{q+1}))$
if $t_{q+1} = \mu$ **or** $x^*_{i_m}(t_{q+1}) = x^*_{m+1}(t_{q+1})$ **then**
    $t^* = t_{q+1}$
    Let $\mathcal{A}^{q+1} = \{(i,j,\#) \mid (i,j) \in E_m,\ x^*_i(t_{q+1}) \,\#\, x^*_j(t_{q+1})\}$
else
    $t^* = \emptyset$
    Update $\mathcal{A}^{q+1}$ from $\mathcal{A}^q$ via (46)
end if
Output:$(t_{q+1}, x^*(t_{q+1}), z^*(t_{q+1}), \mathcal{A}^{q+1}, t^*)$

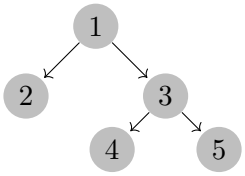
5.3 ASRA 的示例说明

我们给出一个应用 ASRA 求解问题（1）的示例。设 $G =$ （V， E）为图3a所示的有向树，其中 $V = \{1,2,3,4,5\}$ 和 $E =$ {（1, 2）、（1, 3）、（3, 4）、（3, 5）}。令

$$f_i(x_i) = \frac{1}{2}(x_i - y_i)^2 \text{ for } i = 1, ..., 4, \text{ where } y = (4,2,2,8) \in \Re^4, \text{ and } f_5(x_5) = x_5^2 + \frac{1}{4}x_5^4,$$

并将正则化参数设置如下:

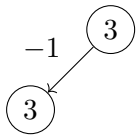
$$(\lambda_{1,2}, \mu_{1,2}) = (+\infty, 0),\ (\lambda_{1,3}, \mu_{1,3}) = (0, +\infty),\ (\lambda_{3,4}, \mu_{3,4}) = (0,4),\ \text{and } (\lambda_{3,5}, \mu_{3,5}) = (3,3).$$



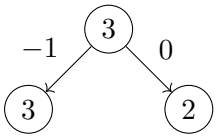
(a) 有向树 $G = (V, E)$



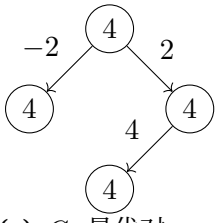
(b) G_1-最优对



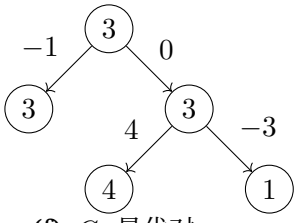
(c) G_2-最优对



(d) G_3-最优对



(e) G_4-最优对



(f) G_5-最优对

图3：应用 ASRA 求解给定 $G = (V, E)$ 的问题 (1) 的示例。第一个子图表示有向树 $G = (V, E)$，其余五个子图展示了针对 $m = 1$、2、3、4、5 的 G_m-最优对；$i \in V$ 的 x_i 值标注在圆圈内，而 $(i, j) \in E$ 的 $z_{i,j}$ 值标注在边上方。

ASRA 的详细步骤如下：

(i) 首先，我们以 $x_1^{(1)} = 4$ 初始化。(ii) 由于 $(f_2^*)'(x_1^{(1)}) > 0$，因此有 $t^* \leq 0$。我们从 $t_0 = 0$ 开始，并在 $t^* = t_1 = -1$ 处终止。于是，G_2-最优对 $(x^{(2)}, z^{(2)})$ 为 $x^{(2)} = (3, 3)$ 和 $z_1^{(2)},_2 = -1$。(iii) 由于 $(f_3^*)'(x_1^{(2)}) > 0$，我们有 $t^* \leq 0$。此处，我们有 $t^* = t_0 = -\lambda_1,_3 = 0$。相应的 G_3-最优对 $(x^{(3)}, z^{(3)})$ 为 $x^{(3)} = (3, 3, 2)$，以及 $z_1^{(3)},_2 = -1, z_1^{(3)},_3 = 0$。(iv) 由于 $(f_4^*)'(x_3^{(3)}) < 0$，因此有 $t^* \geq 0$。从 $t_0 = 0$ 开始，我们首先到达 $t_1 = 1$，并修改相应的活动集，即将 (2, 3, >) 替换为 (2, 3, =)，然后继续搜索 t^*。我们在 $t^* = t_2 = 4$ 处终止。因此，G_4-最优对 $(x^{(4)}, z^{(4)})$ 为$x^{(4)} = (4, 4, 4, 4)$，以及 $z_1^{(4)},_2 = -2, z_1^{(4)},_3 = 2, z_3^{(4)},_4 = 4$。(v) 由于 $(f_5^*)'(x_3^{(4)}) > 0$，我们有 $t^* \leq 0$。从 $t_0 = 0$ 开始，我们首先到达 $t_1 = 0$，并在相应的活动集中将 (3, 4, =) 替换为 (3, 4, <)。随后，我们在 $t^* = t_2 = -3$ 处结束搜索，G_5-最优对 $(x^{(5)}, z^{(5)})$ 为 $x^{(5)} = (3, 3, 3, 4, 1)$，以及 $z_1^{(5)}$，$_2 = -1, z_1^{(5)},_3 = 0, z_3^{(5)},_4 = 4, z_3^{(5)},_5 = -3$。

因此，问题 (1) 的最优解为 $x^* = (3,3,3,4,1)$。上述过程的示意图见图3。

参考文献

[1] M. Ayer, H. D. Brunk, G. M. Ewing, W. T. Reid, and E. Silverman, 基于不完整信息抽样的经验分布函数, 《数学统计年鉴》26 (1955), 第 641–647 页。

[2] A. Barbero and S. Sra, 用于多维全变分正则化的模块化近端优化, 《机器学习研究杂志》19 (2018), 第 1–82 页。[3] D. Bertsimas and J. N. Tsitsiklis,《线性优化导论》, Athena Scientific, 马萨诸塞州, 1997。[4] M. J. Best and N. Chakravarti, 保序回归的活动集算法：一个统一框架, 《数学规划》47 (1990), 第 425–439 页。[5] M. J. Best, N. Chakravarti, and V. A. Ubhaya, 在简单链约束下最小化可分离凸函数, 《SIAM 优化杂志》10 (2000), 第 658–672 页。[6] H. D. Brunk, 单调参数的最大似然估计, 《数学统计年鉴》26 (1955), 第 607–616 页。[7] N. Chakravarti, 保序中位数回归：一种线性规划方法, 《运筹学数学》14 (1989), 第 303–308 页。[8] N. Chakravarti, 可由有根树表示的序上的保序中位数回归, 《海军研究后勤学》39 (1992), 第 599–611 页。[9] X. Chang, Y.-L. Yu, Y. Yang, and E. P. Xing, 未剪辑视频中复杂事件分析的语义池化, 《IEEE 模式分析与机器智能汇刊》39 (2016), 第 1617–1732 页。[10] L. Condat, 一维全变分去噪的直接算法, 《IEEE 信号处理快报》20 (2013), 第 1054–1057 页。[11] N. Deo, 工程与计算机科学中的图论及其应用, Prentice Hall, 新泽西州, 1974。[12] M. Frisen, 单峰回归, 《统计学家》35 (1986), 第 479–485 页。[13] C. Lu and D. S. Hochbaum, 一维广义全变分问题的统一方法, 《数学规划》194 (2022), 第 415–442 页。[14] H. Höefling, 融合套索信号逼近器的路径算法, 《计算与图形统计学杂志》19 (2010), 第 984–1006 页。[15] V. Kolmogorov, T. Pock, and M. Rolinek, 树上的全变差, 《SIAM 成像科学杂志》9 (2016), 第 605–636 页。

[16] I. Matyasovszky, 气候时间序列红噪声谱的估计, 《匈牙利气象局季刊》117 (2013), 第 187–200 页。

[17] A. Restrepo and A. C. Bovik，局部单调回归，《IEEE 信号处理汇刊》，41（1993），第2796–2810页。[18] R. T. Rockafellar，《凸分析》，普林斯顿大学出版社，普林斯顿，新泽西州，1970。[19] Y. U. Ryu， R. Chandrasekaran， and V. Jacob，使用保序预测技术进行预后，《管理科学》，50（2004），第777–785页。
[20] M. J. Silvapulle and P. K. Sen，《约束统计推断：不等式、序与形状约束》，约翰·威立父子公司，2005。[21] Q. F. Stout，通过前缀保序回归进行单峰回归，《计算统计与数据分析》，53（2008），第289–297页。[22] R. Tibshirani， H. Höefling， and R. Tibshirani，近似保序回归，《计量学》，53（2011），第54–61页。[23] D. B. West，图论导论，第2版，上萨德尔河：Prentice Hall，2001。[24] C. Wu， J. Thai， S. Yadlowsky，
A. Pozdnoukhov， and A. Bayen，Cellpath：通过凸优化融合蜂窝网络与交通传感器数据以估计路线流量，《交通研究C辑：新兴技术》，59（2015），第111–128页。
[25] Y.-L. Yu and E. P. Xing，保序回归及相关问题的精确算法，《物理学杂志：会议系列》699，2016。[26] Z. Yu， X. Chen， and X. D. Li，广义近似保序回归的动态规划方法，《数学规划计算》，15（2023），第195–225页。