Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in DISCRETE MATHEMATICS

Electronic Notes in Discrete Mathematics 52 (2016) 109–116 www.elsevier.com/locate/endm

Separation Algorithm for Tree Partitioning Inequalities

Sunil Chopra[a], Kangbok Lee[b,1], Minseok Ryu[c] and Sangho Shim[d]

[a] *J.L. Kellogg Graduate School of Management, Northwestern University, Evanston, IL 60208, USA*

[b] *Department of Business and Economics, York College, The City University of New York, Jamaica, NY 11451, USA*

[c] *Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109, USA*

[d] *Department of Engineering, Robert Morris University, Moon Twp, PA 15108*

Abstract

We consider the tree partition problem to partition the node set of a tree into subsets where the induced subgraph by each subset is connected and the total weight of nodes in a subset cannot exceed the capacity of the subset. We identify exponentially many valid inequalities for an integer programming formulation of the problem and develop a linear time separation algorithm for the valid inequalities.

Keywords: tree partitioning, integer programming, valid inequality, separation algorithm

[1] Corresponding Author: `klee5@york.cuny.edu`

http://dx.doi.org/10.1016/j.endm.2016.03.015

1 Introduction

Given a tree $G = (V, E)$, the *tree partition problem* into at most k sub-trees is to partition the node set V into k subsets $N_1, ..., N_k$ at minimum total cost of inner-subset edges such that the induced graph by each non-empty subset is connected (subsets N_i may be empty.) We assume each subset N_i has its own capacity U_i. That is, the sum of the weights w_v of the nodes v in each subset N_i cannot exceed U_i. Assuming a uniform capacity $U_i = U$ for all i, Lukes [4] developed a pseudo-polynomial time dynamic programming algorithm. In our paper we assume that capacities U_i may be different from each other.

We formulate the tree partition problem as an integer programming problem. A partition $\Pi = (N_1, ..., N_k)$ is encoded into a binary incidence vector (x, y) where $x_{vt} = 1$ if node $v \in N_t$ for $v \in V$, $t \in \{1, \ldots, k\}$, and $y_e = 1$ if edge e is not cut by the partition for $e \in E$; *i.e.*, e is an inner-subset edge. We refer to y_e as an *edge variable*. The objective is to minimize the total cost of uncut (inner-subset) edges, $\sum_{e \in E} c_e y_e$ where c_e is the cost of edge e.

Without the connectivity restriction of each subset, the convex hull of the feasible binary incidence vectors is known in [2] to be spanned by the integer solutions which satisfy the following system of linear constraints:

(1) $(x, y) \geq 0,$

(2) $\displaystyle\sum_{t=1}^{k} x_{vt} = 1 \quad \text{for all nodes } v,$

(3) $x_{ut} + x_{wt} - y_{uw} \leq 1 \quad \text{for all edges } uw \text{ and all } t = 1, ..., k,$

(4) $$\left.\begin{array}{r} x_{ut} - x_{wt} + y_{uw} \leq 1 \\ -x_{ut} + x_{wt} + y_{uw} \leq 1 \end{array}\right\} \quad \text{for all edges } uw \text{ and all } t = 1, ..., k,$$

(5) $\displaystyle\sum_{u \in V} w_u x_{ut} \leq U_t \quad \text{for all } t = 1, ..., k.$

We refer to (2) as *partitioning equations*, (3) as *edge inequalities*, the pair of inequalities in (4) as *arc inequalities* and (5) as *capacity inequalities*, respectively.

Additional constraints are required for the induced subgraph by each subset N_i to be connected. The *path inequalities* are defined to be

(6) $x_{ut} + x_{vt} - x_{ht} \leq 1$ for all t and for all $u \neq v \in V$ and all $h \in \text{Path}(u, v)$,

where $\text{Path}(u, v)$ denotes the set of intermediate vertices in the path from u to v ($\text{Path}[u, v]$ denotes the set of vertices in the path including u and v.) Note that the number of the path inequalities is $O(kn^3)$ in an arbitrary tree and