Notes for Classic IP & CO Paper List

Lan Peng, PhD Student

Department of Industrial and Systems Engineering
University at Buffalo, SUNY
lanpeng@buffalo.edu

February 5, 2021

February 5, 2021

To My Beloved Motherland China

Contents

Chapter 1

Partitioning Procedures for Solving Mixed-Variables Programming Problems

J. F. Benders

***Numerische Mathematik*, 1962**

1.1 Introduction & Preliminaries

The mixed-variables programming problems considered is the following

$$\begin{aligned} (P) \quad \max \quad & \mathbf{c}^\top \mathbf{x} + f(\mathbf{y}) \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} + F(\mathbf{y}) \leq \mathbf{b} \\ & \mathbf{x} \in \mathbb{R}^p \\ & \mathbf{y} \in S \end{aligned}$$

in which

- S - is an arbitrary subset of $\mathbb{R}^q$
- $\mathbf{A}$ - is an $m \times p$ matrix
- $f(\mathbf{y})$ - is a scalar function defined on S
- $F(\mathbf{y})$ - is an m-component vector function defined on S
- $\mathbf{b}$ - RHS, $\mathbf{b} \in \mathbb{R}^m$
- $\mathbf{c}$ - Cost coefficient, $\mathbf{c} \in \mathbb{R}^p$

Example. MIP, LP

The basic idea is to partition a given problem into two sub problems, a programming problem ($F(\mathbf{y})$) and a LP ($\mathbf{Ax}$). This happens when the problem indicates a natural partition of the variables.
Denote $\mathbf{u}, \mathbf{v}, \mathbf{z}$ as vectors in $\mathbb{R}^m$, and u_0, x_0, and z_0 as scalars.
For $\mathbf{A}$ and $\mathbf{c}$ defined previously, we define

- the convex polyhedral cone C in $\mathbb{R}^{m+1}$ by

$$C = \{ \begin{pmatrix} u_0 \\ \mathbf{u} \end{pmatrix} | \mathbf{A}^\top \mathbf{u} - \mathbf{c} u_0 \geq \mathbf{0},\ \mathbf{u} \geq \mathbf{0},\ u_0 \geq 0 \} \tag{1.1}$$

- the convex polyhedral cone C_0 in $\mathbb{R}^m$ by

$$C_0 = \{ \mathbf{u} | \mathbf{A}^\top \mathbf{u} \geq \mathbf{0},\ \mathbf{u} \geq \mathbf{0} \} \tag{1.2}$$

- the convex polyhedral P in $\mathbb{R}^m$ by

$$P = \{ \mathbf{u} | \mathbf{A}^\top \mathbf{u} \geq \mathbf{c},\ \mathbf{u} \geq \mathbf{0} \} \tag{1.3}$$

1.2 A partitioning theorem

Rewrite (P) into an equivalent form as following

$$
\begin{aligned}
(P') \quad \max \quad & x_0 \\
\text{s.t.} \quad & x_0 - \mathbf{c}^\top \mathbf{x} - f(\mathbf{y}) \le 0 \\
& \mathbf{A}\mathbf{x} + F(\mathbf{y}) \le \mathbf{b} \\
& \mathbf{x} \ge \mathbf{0} \\
& \mathbf{y} \in S
\end{aligned}
$$

Clearly, $(x_0^*, \mathbf{x}^*, \mathbf{y}^*)$ is optimal for (P') iff $x_0^* = \mathbf{c}^\top \mathbf{x}^* + f(\mathbf{y}^*)$ and $(\mathbf{x}^*, \mathbf{y}^*)$ is optimal for (P).
For one of the subproblem of (P), the LP, we have

$$
\begin{aligned}
(subLP) \quad \max \quad & \mathbf{c}^\top \mathbf{x} \\
\text{s.t.} \quad & \mathbf{A}\mathbf{x} \le \mathbf{b}' \\
& \mathbf{x} \ge \mathbf{0} \\
& \mathbf{x} \in \mathbb{R}^p
\end{aligned}
$$

take the dual of $(subLP)$, we have

$$
\begin{aligned}
(dualLP) \quad \min \quad & \mathbf{u}^\top \mathbf{b}' \\
\text{s.t.} \quad & \mathbf{u}^\top \mathbf{A} \ge \mathbf{c} \\
& \mathbf{u} \ge \mathbf{0} \\
& \mathbf{u} \in \mathbb{R}^m
\end{aligned}
$$

Chapter 2

The Traveling-Salesman Problem and Minimum Spanning Tree

Michael Held and Richard M. Karp

***Operations Research*, 1969**

2.1 The Traveling-Salesman Problem and a Related Spanning-Tree Problem

Definition 2.1.1 (1-tree)**.** In graph $G = (V, E)$, where $V = \{1, 2, \cdots, n\}$, a 1-tree consists of a tree on the vertex set $\{2, 3, \cdots, n\}$, together with two distinct edges at vertex 1.

Thus, a 1-tree has a single cycle, this cycle contains vertex 1 and vertex 1 always has degree 2. A minimal weighted 1-tree can be found by constructing a minimum spanning tree on the vertex set $\{2, 3, \cdots, n\}$, and then adjoining two edges of lowest weight at vertex 1.
Also notice that every tour is a 1-tree, and a 1-tree is a tour iff each of its vertices has degree 2. If a minimum-weight 1-tree is a tour, it is the solution of the TSP.

Example. An example of 1-tree can be found in figure 2.1, solid arcs are minimum spanning tree of $\{2, 3, \cdots, n\}$ and two dashed arcs links the MST to vertex 1 with minimal cost.

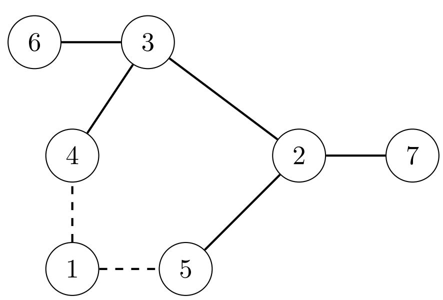

Figure 2.1: 1-tree

Lemma 2.1. *Let $\pi = (\pi_1, \pi_2, \cdots, \pi_n)$ be a real n-vector. If C^* is a minimum-weight tour with respect to the edge weights c_{ij}, then it is also a minimum-weight tour C' with respect to the edge weight $c_{ij} + \pi_i + \pi_j$.*

Proof. For tour C, the weight is $C = \sum_{(i,j)\in C} c_{ij}$. Therefore $C' - C^* = 2\sum_{i=1}^{n} \pi_i$, which is a constant. $\square$

Change the costs from c_{ij} to $c_{ij} + \pi_i + \pi_j$ only changes its minimum spanning 1-tree. Introduce a gap function $f(\pi)$, which is the cost of a minimum-weight tour minus the cost of a minimum-weighted 1-tree both with respect to the weights $c_{ij} + \pi_i + \pi_j$. Notice that if $f(\pi) = 0$ then we found optimal tour for TSP, thus, we consider the problem of finding $\min_\pi f(\pi)$, where

$$\begin{aligned} f(\pi) &= W + 2\sum_{i=1}^{n} \pi_i - \min_k (c_k + \sum_{i=1}^{n} \pi_i d_{ik}) \\ &= W - \min_k [c_k + \sum_{i=1}^{n} (d_{ik} - 2)] \end{aligned}$$

in which

- W - is the weight of a minimum tour with respect to the weights c_{ij}
- c_k - is the weight of the kth 1-tree with respect to the weight c_{ij}. Notice that the 1-trees of the graph are indexed by $1, 2, \ldots, q$ with k as a generic index.
- d_{ik} - is the degree of vertex i in the kth 1-tree.

The goal is to minimize $f(\pi)$ over π, which is equivalent to

$$\begin{aligned} \max \quad & w \\ \text{s.t.} \quad & w \leq c_k + \sum_{i=1}^{n} (d_{ik} - 2) \quad \forall k \end{aligned}$$

Dualizing, we obtain

$$\begin{aligned} \min \quad & \sum_k c_k y_k \\ \text{s.t.} \quad & \sum_k y_k = 1 \\ & y_k \geq 0 \quad \forall k \\ & \sum_k (2 - d_{ik}) y_k = 0 \quad i = 2, 3, \ldots, n-1 \end{aligned}$$

Notice that this LP model seeks a minimum-weight "convex combination of 1-trees" such that each vertex has, on average, degree two.

2.2 Relation to a Linear Program

The following is DFJ formulation for TSP

$$\begin{aligned} \min \quad & \sum_{1 \leq i < j \leq n} c_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{i<j} x_{ij} + \sum_{i>j} x_{ji} = 2, \quad i \in \{1, 2, \cdots, n\} \\ & \sum_{\substack{i \in S \\ j \notin S \\ i<j}} x_{ij} \leq |S| - 1, \quad S \subset \{2, 3, \cdots, n\} \\ & x_{ij} \in \{0, 1\} \end{aligned}$$

Equivalently, we can transform it into

$$
\begin{aligned}
\min \quad & \sum_{1\le i<j\le n} c_{ij}x_{ij} && (2.1)\\
\text{s.t.} \quad & \sum_{i<j} x_{ij} + \sum_{i>j} x_{ji} = 2, \quad i \in \{2,3,\cdots,n-1\} && (2.2)\\
& \sum_{j} x_{1j} = 2 && (2.3)\\
& \sum_{1\le i<j\le n} x_{ij} = n && (2.4)\\
& \sum_{\substack{i\in S\\ j\notin S\\ i<j}} x_{ij} \le |S|-1, \quad S \subset \{2,3,\cdots,n\} && (2.5)\\
& x_{ij} \le 1 && (2.6)\\
& x_{ij} \ge 0 && (2.7)\\
& x_{ij} \in \{0,1\} && (2.8)
\end{aligned}
$$

Denote constraints (2.2) by $\mathbf{A}\mathbf{x} = \mathbf{b}$, and the constraints (2.3), (2.4), (2.5) and (2.6) by $\mathbf{A}'\mathbf{x} = \mathbf{b}'$.

2.3 A Column-Generation Technique

2.4 An Ascent Method

2.5 A Branch-and-Bound Method

2.6 Minimal 1-Trees and Matroids

2.7 Other Applications