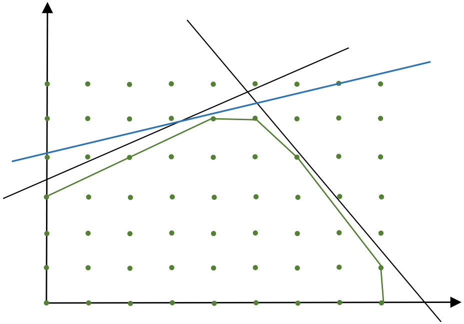
CE 205A
Transportation Logistics

Lecture 11

Cutting Plane Algorithms

Previously on Transportation Logistics

The idea behind cutting plane algorithms is to solve the LP relaxations and identify a hyperplane that separates the LP relaxation and integer feasible solutions of the problem.



This is done iteratively by solving a *separation problem* which determines new cuts to be added to the problem.

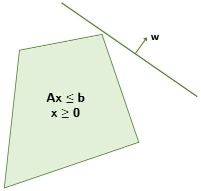
Previously on Transportation Logistics

Definition

An inequality $\mathbf{w}^\mathsf{T}\mathbf{x} \leq w_0$ is a valid inequality for $X \subseteq \mathbb{R}^n$ if $\mathbf{w}^\mathsf{T}\mathbf{x} \leq w_0 \; \forall \; \mathbf{x} \in X.$ A valid inequality is also denoted as $(\mathbf{w}, w_0)$.

Proposition

An inequality $\mathbf{w}^\mathsf{T}\mathbf{x} \leq w_0$ *is a valid inequality for* $X = \{\mathbf{x} : \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\} \Leftrightarrow \exists \; \mathbf{y} \geq \mathbf{0}$, *such that* $\mathbf{A}^\mathsf{T}\mathbf{y} \geq \mathbf{w}$ *and* $\mathbf{b}^\mathsf{T}\mathbf{y} \leq w_0$.


$\mathbf{w}$
$\mathbf{A}\mathbf{x} \leq \mathbf{b}$
$\mathbf{x} \geq \mathbf{0}$

Previously on Transportation Logistics

Consider the set $X = \{\mathbf{x} \in \mathbb{R}^n_+ : \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$, where $\mathbf{A} \in \mathbb{R}^{m \times n}_+$ and $\boldsymbol{\lambda} \geq \mathbf{0}$. Recall that the following inequality is valid for X.

$$\sum_{i=1}^{m} \lambda_i \mathbf{A}_{i.}\mathbf{x} \leq \sum_{i=1}^{m} \lambda_i b_i$$

The inequality can also be written as

$$\sum_{j=1}^{n} \boldsymbol{\lambda}^{\mathsf{T}} \mathbf{A}_{.j} x_j \leq \boldsymbol{\lambda}^{\mathsf{T}} \mathbf{b}$$

Since, $\mathbf{x} \geq 0$, rounding the coefficients still makes it a valid inequality for X

$$\sum_{j=1}^{n} \lfloor \boldsymbol{\lambda}^{\mathsf{T}} \mathbf{A}_{.j} \rfloor x_j \leq \boldsymbol{\lambda}^{\mathsf{T}} \mathbf{b}$$

Finally, the following inequality is valid for $X \cap \mathbb{Z}^n_+$ since the variables are integral.

$$\sum_{j=1}^{n} \lfloor \boldsymbol{\lambda}^{\mathsf{T}} \mathbf{A}_{.j} \rfloor x_j \leq \lfloor \boldsymbol{\lambda}^{\mathsf{T}} \mathbf{b} \rfloor$$

Previously on Transportation Logistics

Consider the Knapsack constraint $X = \{\mathbf{x} \in \{0,1\}^n : \sum_{j=1}^n a_j x_j \leq b\}$. Let $N = \{1, \ldots, n\}$. Assume that $b > 0$ and $a_j > 0$ for all j. Is this restrictive?

Definition (Cover)

A set $C \subseteq N$ is a cover/dependent set if $\sum_{j \in C} a_j > b$. A cover is minimal if $C \backslash \{j\}$ is not a cover or any $j \in C$.

Determine all covers of $2x_1 + 5x_2 + 3x_3 + x_4 \leq 6$.

- Which of these are minimal?
- What kind of valid inequalities are implied by covers?

Proposition

If $C \subseteq N$ is a cover for X, then $\sum_{j \in C} x_j \leq |C| - 1$ is valid for X.

Previously on Transportation Logistics

Before we begin, we need to partition the data into those corresponding to the free and pivot columns. We will call these *basic* and *non-basic* variables. The original LP is recast as

$$\begin{aligned} \min\ & \mathbf{c}^\mathsf{T}\mathbf{x} \\ \text{s.t. } & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned} \qquad \qquad \begin{aligned} \min\ & \mathbf{c}_B^\mathsf{T}\mathbf{x}_B + \mathbf{c}_N^\mathsf{T}\mathbf{x}_N \\ \text{s.t. } & \begin{bmatrix} \mathbf{B} & \mathbf{N} \end{bmatrix} \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{bmatrix} = \mathbf{b} \\ & \mathbf{x}_B, \mathbf{x}_N \geq \mathbf{0} \end{aligned}$$

We then set the non-basic variables to 0 and hence $\begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{bmatrix} = \begin{bmatrix} \mathbf{B}^{-1}\mathbf{b} \\ \mathbf{0} \end{bmatrix}$

Previously on Transportation Logistics

The series of steps in the Simplex algorithm involved finding inverses and performing matrix multiplications to obtain the reduced cost vector and the descent direction.

$$\begin{aligned} \min\ & z = \mathbf{c}^\mathsf{T}\mathbf{x} \\ \text{s.t. } & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

These steps can be compactly written in a table like format using row operations and Gauss-Jordan style elimination.

	$\mathbf{x}_B$	$\mathbf{x}_{NB}$	RHS
RC	$\mathbf{0}$	$\mathbf{c}_N^\mathsf{T} - \mathbf{c}_B^\mathsf{T}\mathbf{B}^{-1}\mathbf{N}$	$-\mathbf{c}_B^\mathsf{T}\mathbf{B}^{-1}\mathbf{b}$
$\mathbf{x}_B$	$\mathbf{I}$	$\mathbf{B}^{-1}\mathbf{N}$	$\mathbf{B}^{-1}\mathbf{b}$

The reduced costs follow from row operations that create $\mathbf{0}$ above $\mathbf{I}$. Each column in the Tableau is $\mathbf{B}^{-1}\mathbf{A}_{.j}$. Hence, the min-cost rule and the unbounded condition in the tableau method involves a sign reversal.

Lecture Outline

1 Generic Cutting Plane Algorithm
2 Gomory's Cutting Plane Algorithm

Lecture Outline

Generic Cutting Plane Algorithm

Generic Cutting Plane Algorithm

Introduction

Valid inequalities can potentially help improve LP relaxations of an optimization problem.

Hence, one could add them to a formulation and proceed with branch and bound. This method is also known as *cut and branch*.

However, there many valid inequalities to choose from. If a certain valid inequality is known to be facet-defining, it is more useful than the others.

But finding such inequalities is often non-trivial. Even if we did, adding them upfront may require adding an exponential number of constraints which is prohibitive.

Can we iteratively add 'useful' valid inequalities?

Generic Cutting Plane Algorithm

Introduction

Consider an IP formulation $X = \{\mathbf{x} \in \mathbb{Z}^n_+ : \mathbf{A}\mathbf{x} \le \mathbf{b}\}$. Let Conv($X$) be the convex hull of the feasible space and $P = \{\mathbf{x} \in \mathbb{R}^n_+ : \mathbf{A}\mathbf{x} \le \mathbf{b}\}$ be the LP relaxation space.

We can try to cut off the current solution to the LP relaxation using a specific valid inequality so that we move closer to Conv(X).

Algorithm Generic Cutting Plane Algorithm

```
$\mathbf{x}^* \leftarrow \arg\min_{\mathbf{x} \in P} \mathbf{c}^\mathsf{T}\mathbf{x}$
while $\mathbf{x}^* \notin \mathbb{Z}^n_+$ do
    Separation Problem:
    Determine a valid inequality $(\mathbf{w}, w_0)$ that satisfies $\mathbf{w}^\mathsf{T}\mathbf{x}^* > w_0$
    $P \leftarrow P \cap \{\mathbf{w}^\mathsf{T}\mathbf{x} \le w_0\}$
    $\mathbf{x}^* \leftarrow \arg\min_{\mathbf{x} \in P} \mathbf{c}^\mathsf{T}\mathbf{x}$
end while
```

Does $\mathbf{c}^\mathsf{T}\mathbf{x}$ always increase?

Generic Cutting Plane Algorithm

Separation Problem

Searching for valid inequalities that violate the current LP solution is usually restricted to a family of known valid inequalities $\mathcal{F}$.

In such cases, the while loop may not terminate since we may not reach the IP solution.

From a practical perspective, the algorithm can be terminated if all inequalities in $\mathcal{F}$ are exhausted or if the improvement in the objective is not significant.

The updated P can be used as a starting point for a branch and bound algorithm.

Generic Cutting Plane Algorithm

Example – Knapsack

Consider the knapsack constraint $X = \left\{ \mathbf{x} \in \{0,1\}^n : \sum_{j=1}^{n} a_j x_j \le b \right\}$. Let $N = \{1, \ldots, n\}$.

Suppose $\mathcal{F}$ is the family of cover inequalities. Note that $\sum_{j \in C} x_j \le |C| - 1$ is equivalent to

$$\sum_{j \in C} (1 - x_j) \ge 1$$

Suppose we solve the problem as an LP and get a factional solution $\mathbf{x}^*$. The idea in the cutting plane algorithm is to find a cover inequality that is not satisfied by $\mathbf{x}^*$ and add it to the LP relaxation?

Generic Cutting Plane Algorithm

Example – Knapsack

Can you find a cover $C \subseteq N$ for which $\sum_{j\in C} a_j > b$ and $\sum_{j\in C}(1-x_j^*) < 1$? Cast this as an optimization problem.

$$\begin{aligned} \zeta = \min \ & \sum_{j\in N}(1-x_j^*)z_j \\ \text{s.t. } & \sum_{j\in N} a_j z_j > b \\ & \mathbf{z} \in \{0,1\}^n \end{aligned}$$

If $\zeta < 1$, then indices for which zs are 1 give us a knapsack cover cut. Why solve a minimization problem?

Notice that we replaced one IP problem with another one. What's the advantage of this method? Do we get a minimal cover using this method? Can we solve this using approximations?

Generic Cutting Plane Algorithm

Example – Knapsack

Consider the knapsack constraint

$$11x_1 + 6x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19$$

Suppose a fractional solution $(0.5, 1, 0.5, 0, 0, 1, 0)$ is obtained in one of the iterations. Find a cover cut.

Lecture Outline

Gomory's Cutting Plane Algorithm

Gomory's Cutting Plane Algorithm
Introduction

The cutting plane algorithm discussed earlier is generic but the separation problem depends on the formulation.

Gomory proposed a cutting plane method that uses the solution from the final simplex tableau to generate new inequalities that cuts off the current LP solution.

This method is problem-agnostic and is theoretically guaranteed to converge to the IP solution. However, it is empirically known to suffer from poor convergence.

Gomory's Cutting Plane Algorithm

Method

Consider an IP problem (left) and its LP relaxation (right).

$$\begin{aligned} \min\ & \mathbf{c}^\mathsf{T}\mathbf{x} \\ \text{s.t. } & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \in \mathbb{Z}_+^n \end{aligned}$$

$$\begin{aligned} \min\ & \mathbf{c}_B^\mathsf{T}\mathbf{x}_B + \mathbf{c}_N^\mathsf{T}\mathbf{x}_N \\ \text{s.t. } & \begin{bmatrix} \mathbf{B} & \mathbf{N} \end{bmatrix} \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{bmatrix} = \mathbf{b} \\ & \mathbf{x}_B, \mathbf{x}_N \geq \mathbf{0} \end{aligned}$$

The constraints can also be written as

$$\mathbf{x}_B + \mathbf{B}^{-1}\mathbf{N}\mathbf{x}_N = \mathbf{B}^{-1}\mathbf{b}$$

If $\mathbf{B}^{-1}\mathbf{b}$ is integral when simplex is terminated, the solution obtained is optimal to the IP problem. Else, we can find a row i for which $(\mathbf{B}^{-1}\mathbf{b})_i$ is fractional.

Gomory's Cutting Plane Algorithm

Method

The corresponding row can be written as

$$(\mathbf{x}_B)_i + \sum_{j \in N} (\mathbf{B}^{-1}\mathbf{N})_{ij} x_j = (\mathbf{B}^{-1}\mathbf{b})_i$$

Since the above equation holds, the following inequality is also true.

$$(\mathbf{x}_B)_i + \sum_{j \in N} (\mathbf{B}^{-1}\mathbf{N})_{ij} x_j \leq (\mathbf{B}^{-1}\mathbf{b})_i$$

Applying the ideas from CG cuts, the following inequality is valid for the LP polytope.

$$(\mathbf{x}_B)_i + \sum_{j \in N} \lfloor (\mathbf{B}^{-1}\mathbf{N})_{ij} \rfloor x_j \leq \lfloor (\mathbf{B}^{-1}\mathbf{b})_i \rfloor$$

Gomory's Cutting Plane Algorithm

Writing a $\geq$ version of the previous equality and subtracting it from the rounded inequality,

$$\sum_{j \in N} \Big((\mathbf{B}^{-1}\mathbf{N})_{ij} - \lfloor (\mathbf{B}^{-1}\mathbf{N})_{ij} \rfloor \Big) x_j \geq (\mathbf{B}^{-1}\mathbf{b})_i - \lfloor (\mathbf{B}^{-1}\mathbf{b})_i \rfloor$$

We can write this valid inequality as $\mathbf{w}^\mathsf{T}\mathbf{x} \geq w_0$, where

$$\mathbf{w} = \begin{bmatrix} \mathbf{0}_B & (\mathbf{B}^{-1}\mathbf{N})_{i.} - \lfloor (\mathbf{B}^{-1}\mathbf{N})_{i.} \rfloor \end{bmatrix}$$

$$w_0 = (\mathbf{B}^{-1}\mathbf{b})_i - \lfloor (\mathbf{B}^{-1}\mathbf{b})_i \rfloor$$

Note that $w_0 \in (0, 1)$. Clearly, the optimal solution violates this valid inequality. (Why?)

Gomory's Cutting Plane Algorithm

Example

Solve the following IP using Gomory's cutting plane method.

$$\begin{aligned} \min \quad & -4x_1 + x_2 \\ \text{s.t.} \quad & 7x_1 - 2x_2 \leq 14 \\ & x_2 \leq 3 \\ & 2x_1 - 2x_2 \leq 3 \\ & \mathbf{x} \in \mathbb{Z}_+^2 \end{aligned}$$

The LP relaxation in standard form is

$$\begin{aligned} \min \quad & -4x_1 + x_2 \\ \text{s.t.} \quad & 7x_1 - 2x_2 + x_3 = 14 \\ & x_2 + x_4 = 3 \\ & 2x_1 - 2x_2 + x_5 = 3 \\ & \mathbf{x} \in \mathbb{R}_+^5 \end{aligned}$$

Gomory's Cutting Plane Algorithm

Example – Iteration 1

The LP relaxation solution is $\begin{bmatrix} 20/7 & 3 & 0 & 0 & 23/7 \end{bmatrix}$. The basic variables include x_1, x_2, and x_5.

$$\mathbf{B} = \begin{bmatrix} 7 & -2 & 0 \\ 0 & 1 & 0 \\ 2 & -2 & 1 \end{bmatrix} \ \mathbf{B}^{-1} = \begin{bmatrix} \frac{1}{7} & \frac{2}{7} & 0 \\ 0 & 1 & 0 \\ -\frac{2}{7} & \frac{10}{7} & 1 \end{bmatrix} \ \mathbf{N} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} 14 \\ 3 \\ 3 \end{bmatrix} \ \mathbf{B}^{-1}\mathbf{N} = \begin{bmatrix} \frac{1}{7} & \frac{2}{7} \\ 0 & 1 \\ -\frac{2}{7} & \frac{10}{7} \end{bmatrix} \ \mathbf{B}^{-1}\mathbf{b} = \begin{bmatrix} \frac{20}{7} \\ 3 \\ \frac{23}{7} \end{bmatrix}$$

Hence, we add the cut

$$\frac{1}{7}x_3 + \frac{2}{7}x_4 \geq \frac{20}{7} - \left\lfloor \frac{20}{7} \right\rfloor = \frac{6}{7}$$

Gomory's Cutting Plane Algorithm

Example – Iteration 2

Repeat by solving the following LP

$$\begin{aligned}
\min\ & -4x_1 + x_2 \\
\text{s.t.}\ & 7x_1 - 2x_2 + x_3 = 14 \\
& x_2 + x_4 = 3 \\
& 2x_1 - 2x_2 + x_5 = 3 \\
& \tfrac{1}{7}x_3 + \tfrac{2}{7}x_4 - x_6 = \tfrac{6}{7} \\
& \mathbf{x} \in \mathbb{R}^6_+
\end{aligned}$$

Gomory's Cutting Plane Algorithm

Example – Iteration 2

The new optimal LP solution is $\begin{bmatrix}2 & 1/2 & 1 & 5/2 & 0 & 0\end{bmatrix}$ with x_1, x_2, x_3, and x_4 in the basis.

$$\mathbf{B} = \begin{bmatrix}7 & -2 & 1 & 0\\ 0 & 1 & 0 & 1\\ 2 & -2 & 0 & 0\\ 0 & 0 & \frac{1}{7} & \frac{2}{7}\end{bmatrix} \quad \mathbf{B}^{-1} = \begin{bmatrix}\frac{1}{7} & \frac{2}{7} & 0 & -1\\ \frac{1}{7} & \frac{2}{7} & -\frac{1}{2} & -1\\ \frac{2}{7} & -\frac{10}{7} & -1 & 5\\ -\frac{1}{7} & \frac{5}{7} & \frac{1}{2} & 1\end{bmatrix} \quad \mathbf{N} = \begin{bmatrix}0 & 0\\ 0 & 0\\ 1 & 0\\ 0 & -1\end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix}14\\ 3\\ 3\\ \frac{6}{7}\end{bmatrix} \quad \mathbf{B}^{-1}\mathbf{N} = \begin{bmatrix}0 & 1\\ -\frac{1}{2} & 1\\ -1 & -5\\ \frac{1}{2} & -1\end{bmatrix} \quad \mathbf{B}^{-1}\mathbf{b} = \begin{bmatrix}2\\ \frac{1}{2}\\ 1\\ \frac{5}{2}\end{bmatrix}$$

Hence, we add the cut

$$\frac{1}{2}x_5 \geq \frac{1}{2}$$

Gomory's Cutting Plane Algorithm

Example – Iteration 3

Repeat by solving the following LP

$$\begin{array}{rl} \min & -4x_1 + x_2 \\ & 7x_1 - 2x_2 + x_3 = 14 \\ & x_2 + x_4 = 3 \\ \text{s.t.} & 2x_1 - 2x_2 + x_5 = 3 \\ & \frac{1}{7}x_3 + \frac{2}{7}x_4 - x_6 = \frac{6}{7} \\ & \frac{1}{2}x_5 - x_7 = \frac{1}{2} \\ & \mathbf{x} \in \mathbb{R}^7_+ \end{array}$$

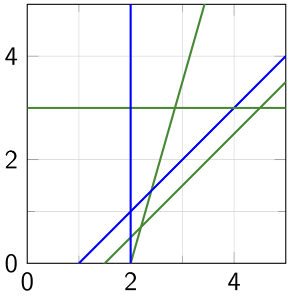
The optimal LP solution is $\begin{bmatrix} 2 & 1 & 2 & 2 & 1 \end{bmatrix}$ and integral. Hence, we terminate the algorithm.

Gomory's Cutting Plane Algorithm

Example

When implementing cutting plane methods, you do not need to create new variables but can add cuts to the original model using the original variables.

Write the cuts in terms of the original variables and plot the feasible region in each iteration.



Gomory's Cutting Plane Algorithm

CG Cuts

Gomory's cutting plane algorithm is equivalent to generating CG cuts with weights given by the ith row of $\mathbf{B}^{-1}$.

Specifically, we use the weights $(\mathbf{B}^{-1})_i - \lfloor (\mathbf{B}^{-1})_i \rfloor$.

Apply this to the earlier example and derive CG cuts. Remember to use the formulation in terms of the original variables in later iterations.

If you use the earlier formulations in which the cuts include slack variables, use the weights $(\mathbf{B}^{-1}\mathbf{A}_s)_i - \lfloor (\mathbf{B}^{-1}\mathbf{A}_s)_i \rfloor$, where $\mathbf{A}_s$ is the sub-matrix corresponding to the slack variables.

Gomory's Cutting Plane Algorithm

In the first iteration,

$$\mathbf{B} = \begin{bmatrix} 7 & -2 & 0 \\ 0 & 1 & 0 \\ 2 & -2 & 1 \end{bmatrix} \quad \mathbf{B}^{-1} = \begin{bmatrix} \frac{1}{7} & \frac{2}{7} & 0 \\ 0 & 1 & 0 \\ -\frac{2}{7} & \frac{10}{7} & 1 \end{bmatrix}$$

Using the row $\begin{bmatrix} \frac{1}{7} & \frac{2}{7} & 0 \end{bmatrix}$ of $\mathbf{B}^{-1}$, applying weights $\boldsymbol{\lambda} = (\frac{1}{7}, \frac{2}{7}, 0)$ to the original constraints

$$\begin{aligned} 7x_1 - 2x_2 &\leq 14 \\ x_2 &\leq 3 \\ 2x_1 - 2x_2 &\leq 3 \end{aligned}$$

we get $x_1 \leq \frac{20}{7}$. Rounding the LHS and RHS yields $x_1 \leq 2$.

Gomory's Cutting Plane Algorithm

CG Cuts

In the second iteration, the basis matrix for the original formulation is

$$\mathbf{B} = \begin{bmatrix} 7 & -2 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 2 & -2 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{B}^{-1} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{2} & 1 \\ 1 & 0 & -1 & -5 \\ 0 & 1 & \frac{1}{2} & -1 \end{bmatrix}$$

Using the row $\begin{bmatrix} 0 & 0 & -\frac{1}{2} & 1 \end{bmatrix}$ of $\mathbf{B}^{-1}$ and $\boldsymbol{\lambda} = (0, 0, \frac{1}{2}, 0)$

$$\begin{aligned} 7x_1 - 2x_2 &\leq 14 \\ x_2 &\leq 3 \\ 2x_1 - 2x_2 &\leq 3 \\ x_1 \qquad &\leq 2 \end{aligned}$$

gives the valid inequality $x_1 - x_2 \leq \frac{3}{2}$, which when rounded becomes $x_1 - x_2 \leq 1$.

Your Moment of Zen

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

Source: xkcd.com