CE 205A
Transportation Logistics

Lecture 16

Column Generation

Previously on Transportation Logistics

The term $\mathbf{c}^\mathsf{T} - \mathbf{c}_B^\mathsf{T}\mathbf{B}^{-1}\mathbf{A} \geq \mathbf{0}^\mathsf{T}$ is also called the *reduced cost vector*. Let us denote it using $\bar{\mathbf{c}}^\mathsf{T}$. What are the reduced costs of the basic variables?

$$\text{Reduced cost of variable } x_j = c_j - \mathbf{c}_B^\mathsf{T}\mathbf{B}^{-1}\mathbf{A}_{.j}$$

We care only about the reduced cost of the non-basic variables $\bar{\mathbf{c}}_N^\mathsf{T} = \mathbf{c}_N^\mathsf{T} - \mathbf{c}_B^\mathsf{T}\mathbf{B}^{-1}\mathbf{N}$

Theorem (Optimality Condition)

Suppose $\mathbf{x}^*$ is a basic feasible solution and $\bar{\mathbf{c}}_N \geq \mathbf{0}$, then $\mathbf{x}^*$ is optimal

This is an important result in linear programming. We'll soon see that it not only tells us when to stop but also shows us the direction in which we should move if we haven't reached optimality.

Previously on Transportation Logistics

1. Start with an initial basic feasible solution $\mathbf{x} = \begin{bmatrix} \mathbf{x}_B \\ \mathbf{x}_N \end{bmatrix} = \begin{bmatrix} \mathbf{B}^{-1}\mathbf{b} \\ \mathbf{0} \end{bmatrix}$
2. Compute the reduced cost vector $\bar{\mathbf{c}}_N^\top = \mathbf{c}_N^\top - \mathbf{c}_B^\top \mathbf{B}^{-1}\mathbf{N}$
3. If $\bar{\mathbf{c}}_N \geq 0$, then $\mathbf{x}$ is optimal and terminate, else go to Step 4
4. Pick $j^* : \bar{c}_{j^*} < 0$ and compute descent direction $\mathbf{d} = \begin{bmatrix} -\mathbf{B}^{-1}\mathbf{A}_{\cdot j^*} \\ \mathbf{e}_{j^*} \end{bmatrix}$
5. If $\mathbf{d}_B \geq 0$, then the LP is unbounded, else go to Step 6
6. Set $k^* \in \arg\min \left\{ -\frac{(x_B)_k}{(d_B)_k} : (d_B)_k < 0 \right\}$
7. Modify the basis by swapping $\mathbf{B}_{\cdot k^*}$ and $\mathbf{A}_{\cdot j^*}$, set $\mathbf{x}_B = \mathbf{B}^{-1}\mathbf{b}$, and go to Step 2.

Previously on Transportation Logistics

Make sure that the reduced costs of the basic variables are zeros. If not, perform row operations.

Identify an entering variable using the reduced costs. Note that we will no longer calculate reduced costs using cumbersome matrix operations. Everything that we need will be in the table.

	x_1 ↓	x_2	x_3	x_4	x_5	RHS
RC	-6	-4	0	0	0	0
x_3	1	1	1	0	0	6
x_4	2	1	0	1	0	9
x_5	2	3	0	0	1	16

Previously on Transportation Logistics

What happens when a new variable is added? Say a new decision variable x_7 is introduced in the example problem with cost coefficient -20 and $\mathbf{A}_{.7}^{\mathsf{T}} = \begin{bmatrix} 3 & 4 \end{bmatrix}$

This variable can be assumed to be non-basic and the current solution is basic feasible to the new problem. We hence need to check if the new variable can enter the basis.

Lecture Outline

Lecture Outline

Column Generation

Recall that the dual variables/marginal costs satisfy the following equation

$$\mathbf{y}^\mathsf{T} = \mathbf{c}_B^\mathsf{T}\mathbf{B}^{-1}$$

Let us take the primal problem in standard form.

Primal LP

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

Dual LP

$$\begin{aligned} \max\ & \mathbf{b}^\mathsf{T}\mathbf{y} \\ \text{s.t.}\ & \mathbf{A}^\mathsf{T}\mathbf{y} \leq \mathbf{c} \end{aligned}$$

What does the reduced cost optimality condition look like in terms of the dual variables? Having non-negative reduced costs is equivalent to achieving dual feasibility!

Column Generation

Introduction

In many real-world problems, the number of constraints or variables can be very large.

When a problem has several variables but a limited number of constraints, the size of the basis matrix is small and many of the variables are non-basic and are zero in the optimal solution. Yet, two major challenges remain.

- Identifying which columns belong to the optimal basis is non-trivial.
- Checking for optimality, that is the reduced costs of all the non-basic variables must be greater than or equal to zero is cumbersome.

These issues can be effectively addressed using the idea of column generation where

- The constraint matrix includes only a limited number of columns or variables to begin with.
- New variables are the corresponding column in the constraints are added recursively only if they have promising reduced cost.

Note that problems with a large number of constraints with few variables can be handled by converting it to its dual and using column generation.

Equivalently, one could use a similar row generation or constraint generation in which we begin with a limited number of constraints and add new ones recursively.

Column Generation

Master Problem

Consider an LP in the standard form

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

Suppose that the columns of $\mathbf{A}$ are prohibitively large in number such that it is impossible to store $\mathbf{A}$ in a computer's memory.

Instead of this LP, we first solve a *restricted master problem* (RMP) which is an LP with fewer columns. Note that this solution is basic feasible to the original problem (Why?).

$$\begin{aligned} \min \ & \sum_{j \in J} c_j x_j \\ \text{s.t. } & \sum_{j \in J} \mathbf{A}_{.j} x_j = \mathbf{b} \\ & x_j \geq 0 \, \forall j \in J \end{aligned}$$

Column Generation

Sub-problem

Since the restricted master has few variables and constraints, it can be efficiently solved using the simplex method.

The next step involves finding a column(s) to include in the set J and resolve the restricted master. To this end, a sub-problem of the following kind is optimized

$$\min_{j \in J^c} \bar{c}_j$$

where J^c is the complement of J, the columns excluded from the restricted master. In many cases, this optimization problem does not require enumerating the reduced costs of all the variables in J^c.

If the optimal solution to the sub-problem is negative then a new variable can enter the basis and is included in the restricted master. Else, the solution the restricted master is optimal to the original problem.

The reduced costs are evaluated using the duals of the master problem. We begin with an initial set of columns J that guarantees feasibility of the restricted master.

1. Solve the restricted master problem

$$\begin{aligned} \min \ & \sum_{j \in J} c_j x_j \\ \text{s.t.} \ & \sum_{j \in J} \mathbf{A}_{.j} x_j = \mathbf{b} \\ & x_j \geq 0 \, \forall j \in J \end{aligned}$$

   Suppose $\mathbf{y}$ represents the optimal dual solution.

2. Solve the sub-problem $z_{sub} = \min_{j \in J^c} (c_j - \mathbf{y}^\mathsf{T} \mathbf{A}_{.j})$. Let j^* be the optimal solution.

3. If $z_{sub} < 0$, $J \leftarrow J \cup \{j^*\}$ and go to Step 1, else terminate.

The ideas used in column generation can also be applied to problems with a few variables but a large number of constraints.

Consider the dual of the standard form as shown below.

$$\max \ \mathbf{b}^{\mathsf{T}}\mathbf{y}$$
$$\text{s.t. } \mathbf{A}^{\mathsf{T}}\mathbf{y} \leq \mathbf{c}$$

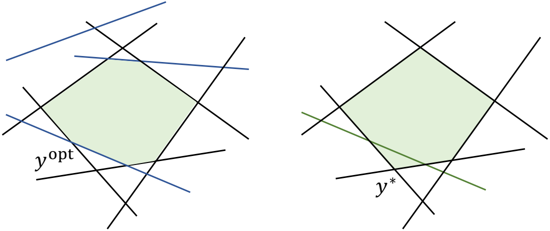
Instead of solving the complete problem, we solve a *relaxed* version of it with fewer constraints.

$$\max \ \mathbf{b}^{\mathsf{T}}\mathbf{y}$$
$$\text{s.t. } \mathbf{A}^{\mathsf{T}}_{.j}\mathbf{y} \leq c_j \, \forall \, j \in J$$

If the optimal solution to this problem solves all the left out constraints then then it is optimal to the original problem.

However, if some constraint is violated, we add it to the relaxed problem and resolve. How can we identify constraints that are violated easily?

To identify the violating constraint, we solve a *separation problem* in which $\mathbf{y}^*$ is separated from the original feasible region using a constraint which it violates just as done in cutting plane methods.



This constraint can be identified by solving $\min c_j - \mathbf{A}_{.j}^{\top}\mathbf{y}^* \, \forall \, j \in J^c$. What are the decision variables in this problem? If the optimal solution to this problem is ≥ 0, $\mathbf{y}^*$ is optimal to the original problem.

As before, row generation works when we can formulate this problem as another efficient optimization formulation.

Column Generation

Implementation Specifics

The column generation procedure can be implemented in many variants, each of which can be guaranteed to produce optimal solutions.

- Removing the leaving variable after each new column is found or including all the columns discovered so far and solving the restricted master.
- Deleting unused columns after a specific number of iterations.
- Adding multiple columns in each iterations which can be identified using an exact of heuristic method. Why does this work?

Lecture Outline

Examples

Suppose the large roll can be cut into smaller rolls in n feasible patterns and that a_{ij} is the number of rolls for i that can be produced from one roll of pattern j.

Let x_j be the number of rolls cut according to pattern j.

$$\begin{aligned} \min \ & \sum_{j=1}^{n} x_j \\ \text{s.t.} \ & \sum_{j=1}^{n} a_{ij} x_j = b_i \, \forall \, i = 1, \ldots, m \\ & x_j \geq 0 \, \forall \, j = 1, \ldots, n \end{aligned}$$

It doesn't hurt to solve this as an LP and round it to get an integer solution.

Note that a column of the jth feasible pattern contains m elements which must satisfy the following conditions

$$\sum_{i=1}^{m} a_{ij} w_i \leq W$$

An initial basic feasible solution to start the RMP is easy (Why?). Pick m patterns $j = 1, \ldots, m$ each of which cuts one roll of size w_i or $\lfloor W/w_i \rfloor$ rolls from a larger roll.

Also observe that the elements of $\mathbf{A}_{.j}$ indicate the a_{ij} values, i.e., the number of rolls of widths i that can be cut.

Examples

Cutting Stock Problem

Thus, minimizing the reduced cost $\bar{c}_j = 1 - \mathbf{y}^\mathsf{T}\mathbf{A}_{.j}$ is equivalent to

$$\max_{j \in J^c} \mathbf{y}^\mathsf{T}\mathbf{A}_{.j}$$

What are the decision variables and constraints for the above problem?

$$\begin{aligned} \max \ & \sum_{i=1}^{m} y_i a_i \\ \text{s.t.} \ & \sum_{i=1}^{m} w_i a_i \leq W \\ & a_i \in \mathbb{Z}_+^n \ \forall\, i = 1, \ldots, m \end{aligned}$$

Does this resemble any problem that we saw earlier? The above knapsack problem can be solved using branch and bound or dynamic programming. Since m is usually small, solving this is not very complicated.

Examples

Cutting Stock Problem

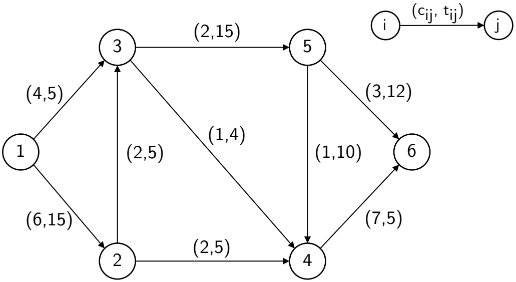
Solve the following cutting stock problem instance using column generation. Assume that $W = 100$.

w_i	b_i
45	97
36	610
31	395
14	211

Examples

Resource Constrained Shortest Paths

Consider a shortest path problem variant in which the goal is to find the cheapest path subject to a constraint that the total time is less than 27.



Formulate this as an optimization problem. Can we solve it as a linear program and get integer optimal solutions?

Suppose the time budget is B.

$$\min \sum_{(i,j)\in A} c_{ij}x_{ij}$$

$$\text{s.t.} \sum_{j:(i,j)\in A} x_{ij} - \sum_{h:(h,i)\in A} x_{hi} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{otherwise} \end{cases}$$

$$\sum_{(i,j)\in A} t_{ij}x_{ij} \leq B$$

$$x_{ij} \in \{0,1\} \, \forall \, (i,j) \in A$$

The extra constraint disrupts the total unimodularity property and hence we can no longer solve this as a linear program.

Can you reformulate this problem using path variables y_p? Suppose P is the set of all paths between the OD pair.

$$\begin{aligned} \min \; & \sum_{p \in P} c_p y_p \\ \text{s.t.} \; & \sum_{p \in P} t_p y_p \leq B \\ & \sum_{p \in P} y_p = 1 \\ & y_p \in \{0, 1\} \; \forall \, p \in P \end{aligned}$$

Just as the cutting stock problem, the size of the P can be exponentially large. Hence, we work with an LP version of the RMP where P is restricted to a subset P'. How do we generate new columns using a pricing subproblem?

Examples

Resource Constrained Shortest Paths

Define dual variables w_1 and w_2 for the first and second constraint of the LP relaxation of the restricted version of this master problem.

The first constraint can be written in standard form using a slack but that does not affect the pricing problem.

The reduced cost is therefore

$$\bar{c}_p = c_p - \begin{bmatrix} w_1 & w_2 \end{bmatrix} \begin{bmatrix} t_p \\ 1 \end{bmatrix} = c_p - w_1 t_p - w_2$$

To find a path which minimizes this expression, what are the decision variables? Can we write this as an optimization problem involving link flow variables?

Note that w_1 and w_2 are constants in the following pricing sub-problem.

Note that the pricing problem does not have the complicating constraint. Hence, standard labeling methods can be used. The arc weights can be negative and hence we should ensure that the path is elementary.

$$z_{sub} = \min \sum_{(i,j) \in A} c_{ij} x_{ij} - w_1 t_{ij} x_{ij} - w_2$$

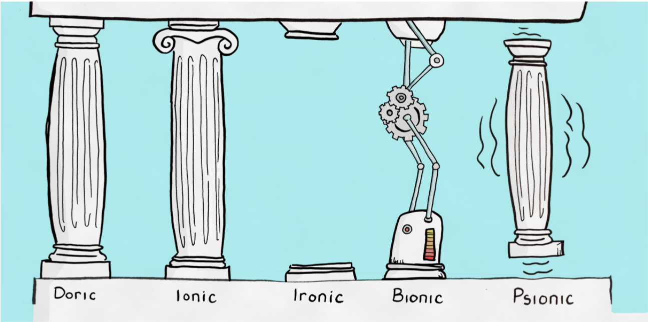
$$\text{s.t.} \quad \sum_{j:(i,j) \in A} x_{ij} - \sum_{h:(h,i) \in A} x_{hi} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{otherwise} \end{cases}$$

$$x_{ij} \in \{0, 1\} \, \forall \, (i,j) \in A$$

If $z_{sub} < 0$, then we add the new path formed by the links belonging to the path to the RMP and resolve it to get new dual variables.

Apply this method to find the optimal resource constrained path in the earlier instance.

Your Moment of Zen



Source: joshwoodjokes.com