Column Generation: Theory, Algorithms, and Applications

A Column Generation

A.1 Introduction

Column generation is a decomposition technique for solving large-scale linear programs with an exponential number of variables (columns). It is especially useful in problems like cutting stock, vehicle routing, and crew scheduling, where enumerating all variables is impractical.

A.2 Mathematical Formulation

Consider the master problem:

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

where A has too many columns to enumerate explicitly.

A.3 The Column Generation Algorithm

1. Start with a restricted master problem (RMP) containing a subset of columns.
2. Solve the RMP to obtain dual prices (multipliers) π.
3. Solve the pricing subproblem: find a new column (variable) with negative reduced cost (for minimization).
4. If such a column exists, add it to the RMP and repeat. If not, the current solution is optimal.

A.4 Reduced Cost and the Pricing Problem

The reduced cost of a column a_j with cost c_j is:

$$\bar{c}_j = c_j - \pi^T a_j$$

The pricing problem is to find a column with $\bar{c}_j < 0$ (for minimization). This is often a combinatorial optimization problem itself.

A.5 Numerical Example: Cutting Stock Problem

Suppose we need to cut rolls of length 10 into pieces of lengths 3 and 4 to meet demands of 4 and 3, respectively.

- **Step 1:** Start with simple patterns (e.g., three 3s, two 4s).
- **Step 2:** Solve the RMP to get dual prices π_1, π_2 for the two item types.
- **Step 3:** The pricing problem is a knapsack problem: maximize $3\pi_1 + 4\pi_2$ subject to $3a_1 + 4a_2 \leq 10$, $a_1, a_2 \geq 0$.
- **Step 4:** If a new pattern with negative reduced cost is found, add it to the RMP and repeat.
- **Step 5:** Stop when no improving pattern exists.

A.6 Classic Applications of Column Generation

Vehicle Routing Problem (VRP). In the VRP, the goal is to find the minimum-cost set of routes for a fleet of vehicles to serve a set of customers. The set covering formulation is:

$$\begin{aligned} \text{minimize} \quad & \sum_{p \in \mathcal{P}} c_p x_p \\ \text{subject to} \quad & \sum_{p: i \in p} x_p \geq 1 \quad \forall i \in \text{customers} \\ & x_p \in \{0, 1\} \quad \forall p \in \mathcal{P} \end{aligned}$$

where $\mathcal{P}$ is the set of all feasible routes (exponentially large). Column generation is used to generate only the necessary routes. The pricing problem is a shortest path problem with resource constraints (e.g., capacity, time windows), solved for each vehicle.

Bin Packing Problem. The bin packing problem can be formulated as a set covering problem:

$$\begin{aligned} \text{minimize} \quad & \sum_{p \in \mathcal{P}} x_p \\ \text{subject to} \quad & \sum_{p: i \in p} x_p \geq 1 \quad \forall i \in \text{items} \\ & x_p \in \{0, 1\} \quad \forall p \in \mathcal{P} \end{aligned}$$

where each p is a feasible packing of items into a bin. The pricing problem is a knapsack problem: find a packing (column) with negative reduced cost given the current dual prices.

A.7 Details on the Pricing Problem

In both VRP and bin packing, the pricing problem is a combinatorial optimization problem:

- **VRP:** Find a route (path) with minimum reduced cost, subject to vehicle and customer constraints. This is typically solved using dynamic programming or label-setting algorithms.
- **Bin Packing:** Find a subset of items (a bin) with minimum reduced cost, i.e., solve a knapsack problem with dual prices as item profits.

Reduced Cost in Set Covering. For a column (pattern/route) p, the reduced cost is:

$$\bar{c}_p = c_p - \sum_{i \in p} \pi_i$$

where π_i are the dual variables from the master problem. A new column is added if $\bar{c}_p < 0$ (for minimization).

Motivation and Context

As optimization problems grow in size and complexity, especially with an exponential number of variables, classical LP methods become impractical. Column generation is a decomposition technique that allows us to solve such problems efficiently by generating variables only as needed.

Motivating Example

Imagine a paper mill that must cut large rolls into smaller pieces to meet customer demands (the cutting stock problem). Enumerating all possible cutting patterns is infeasible, but column generation allows us to find the optimal solution by generating only the most useful patterns.

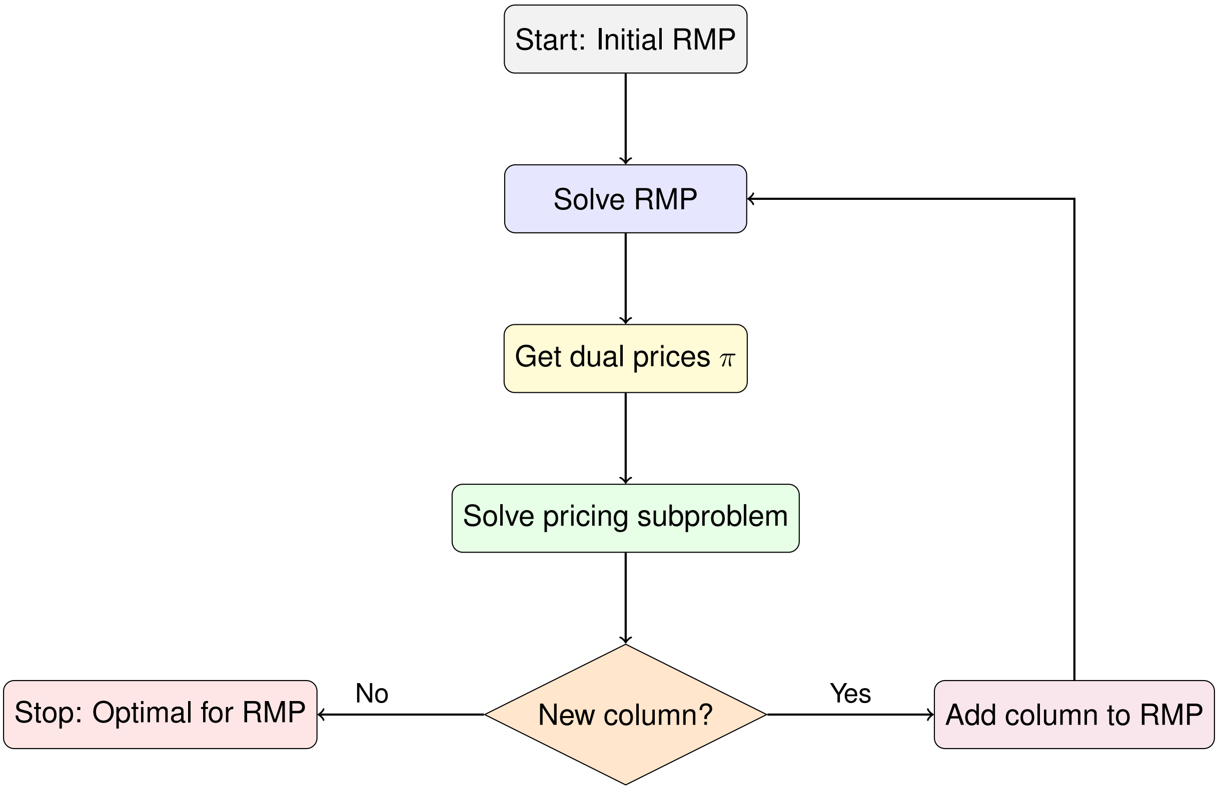
Summary and Transition

Column generation is a powerful tool for large-scale optimization, enabling the solution of problems with an exponential number of variables. The next topic, Branch-and-Price, integrates column generation with branch-and-bound to solve large-scale integer programs, where the pricing subproblem and branching must be carefully coordinated.

A.8 Practice Exercises

1. Formulate the pricing problem for the vehicle routing problem with time windows.
2. Explain why column generation is effective for the cutting stock problem.
3. Implement a simple column generation algorithm for a small instance of the cutting stock problem.

A.9 Algorithm Flow: Column Generation



A.10 Detailed Example: Knapsack Problem with Column Generation

Consider a knapsack problem with 10 objects of weights $[2, 3, 4, 5, 6, 7, 8, 9, 10, 11]$ and profits $[3, 4, 8, 8, 10, 13, 15, 17, 20, 22]$, and knapsack capacity $C = 20$.

Step 1: Start with a few simple patterns (e.g., each object alone, or greedy packings).

Step 2: Solve the restricted master problem (RMP) to get dual price π for the capacity constraint.

Step 3: The pricing subproblem is:

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{10} (p_i - \pi w_i) y_i \\ \text{subject to} \quad & \sum_{i=1}^{10} w_i y_i \le C \\ & y_i \in \{0, 1\} \end{aligned}$$

This is a knapsack problem with adjusted profits. If the optimal value is greater than zero, add the corresponding pattern (column) to the RMP.

Step 4: Repeat until no improving pattern exists (i.e., all reduced costs are non-negative).
Step 5: The final solution is optimal for the LP relaxation.

A.11 Pricing Subproblem: Set Covering Example

In set covering problems, the master problem is:

$$\begin{aligned} \text{minimize} \quad & \sum_{p \in \mathcal{P}} c_p x_p \\ \text{subject to} \quad & \sum_{p: i \in p} x_p \geq 1 \quad \forall i \in \mathcal{I} \\ & x_p \geq 0 \quad \forall p \in \mathcal{P} \end{aligned}$$

where $\mathcal{P}$ is an exponentially large set of subsets (columns).
The pricing subproblem is:

$$\text{find } p^* = \arg\min_{p \in \mathcal{P}} \left(c_p - \sum_{i \in p} \pi_i \right)$$

where π_i are the dual variables. This is often a combinatorial optimization problem (e.g., shortest path, knapsack).

Why Set Covering? Many real-world problems (e.g., vehicle routing, crew scheduling, bin packing) can be formulated as set covering problems, making column generation a natural and powerful approach. The set covering structure allows for efficient decomposition and pricing.