Reading for This Lecture

- “Selected Topics in Column Generation,” Lübbecke and Desrosiers
Column Generation

• The cutting plane method can be viewed as a technique for solving integer linear programs with a large number of constraints.

• In the context of integer programming, these large LPs arise as (partial) descriptions of the convex hull of feasible solutions to an integer program.

• *Column generation*, on the other hand, is a method for solving LPs with a large number of potential columns.

• Theoretically, this is nothing more than a cutting plane method applied to the dual linear program, but it is useful to consider the method separately.

• When column generation is combined with branch and bound, we obtain a method called *branch and price*.
Formulations Involving Many Columns

- Formulations involving many columns can arise in a number of different ways.
  - Applying a decomposition method, such as Dantzig-Wolfe, to an existing formulation results in a reformulation with many columns.
  - Extended formulations can arise through some other reformulation technique that lifts the problem to a higher-dimensional space (lot-sizing).
  - Formulations with many columns may be the “natural” formulation for some problems.

- Even when we start natively with a formulation that has an exponential number of columns, there is often an underlying “compact formulation”.

- Typically, we have a way of writing down the set of columns as the feasible set of a mathematical program of polynomial size.

- In such a case, we can often reformulate the problem in this lower-dimensional space.
Basic Idea of Solution Method

• We solve the LP to optimality using simplex with only a subset of the columns.

• We then ask whether any column that has been left out has positive reduced cost—if so, that column is added and we reoptimize.

• The problem of determining the column with most positive reduced cost is an optimization problem.

• This is called the column generation subproblem.
**Generic Column Generation Algorithm**

- We are interested in solving an LP with a large number of columns.
- For simplicity, we assume the LP is in standard form.
- Consider the *restricted problem* obtained by considering only the subset of the columns indexed by set \( I \).

\[
\begin{align*}
\max & \quad \sum_{i \in I} c_i x_i \\
\text{s.t.} & \quad \sum_{i \in I} A_i x_i = b \\
& \quad x \geq 0
\end{align*}
\]  

(RMP)

- Solve this LP and calculate the optimal dual solution \( u \).
- Now generate a new column \( A_j \) for which \( c_j - c_B B^{-1} A_j = c_j - u A_j > 0 \).
- This can be done by solving the *column generation subproblem*

\[
\max_{a \in C} c_a - u a,
\]

where \( C \) is the global set of columns.
**Overall Algorithm**

1. Generate initial columns and add them to the restricted set $I_0$.
2. $k \leftarrow 0$, $\bar{c}_{max} \leftarrow \infty$
3. **while** $\bar{c}_{max} > 0$ **do**
4. Solve the *restricted master problem*

$$\max \sum_{i \in I_k} c_i x_i$$

s.t. $\sum_{i \in I_k} A_i x_i = b$

$x \geq 0$  \hspace{1cm} (RMP)

5. Solve the *column-generation subproblem* to obtain

$$a^* \in \arg\max_{a \in C} c_a - u^k a,$$

and set $\bar{c}_{max} = c_{a^*} - u^k a^*$

6. **if** $\bar{c}_{max} > 0$ **then**
7. $I_{k+1} \leftarrow I_k \cup \{a^*\}$
8. **end if**
9. $k \leftarrow k + 1$
10. **end while**
**Example: The Cutting Stock Problem**

- The cutting stock problem was one of the first applications of column generation.
- We are selling rolls of paper in specified widths $w_i$, $i = 1, \ldots, m$.
- For each width $i$, we have a given demand $d_i$ that must be satisfied.
- There are large rolls from which the smaller rolls are cut with width $W$.
- We want to minimize the total number of larger rolls we need to use.
- An IP formulation of this problem is

\[
\begin{align*}
\min & \quad \sum_{i=1}^{n} \lambda_i \\
\text{s.t.} & \quad \sum_{i=1}^{n} \lambda_i a^i \geq d \\
& \quad \lambda_i \geq 0, i = 1, \ldots, n, \\
& \quad \lambda_i \quad \text{integer}, i = 1, \ldots, n
\end{align*}
\]

where the columns $a^i$ represent the *feasible patterns*. 
Pattern Generation for Cutting Stock

- The potential columns correspond to feasible patterns.
- A given column vector $a$ corresponds to a feasible pattern if and only if
  $$\sum_{i=1}^{m} w_i a_i \leq W$$
  and $a$ contains only non-negative integers.
- The objective function coefficient of every pattern (column) is 1.
- Finding the column with the smallest reduced cost is a knapsack problem:
  $$\max \sum_{i=1}^{m} u_i a_i$$
  s.t. $$\sum_{i=1}^{m} w_i a_i \leq W$$
  $$a_i \geq 0$$
  $$a_i \text{ integer}$$
Example: Set Partitioning Models

• Recall the set partitioning problem.
• In this problem, $A$ is a 0-1 matrix and we wish to find

$$\min\{cx \mid Ax = 1, x \in \mathbb{B}^n\}$$

• Examples of Set Partitioning Models
  – Airline Crew Scheduling
  – Winner Determination in Combinatorial Auctions
  – Vehicle Routing

• Note that in each case, the columns have to satisfy a particular structure that defines the column generation subproblem.

• One advantage of these formulations is that we can implicitly introduce constraints that are otherwise difficult to model.
Generating Initial Columns

- One aspect that is very different from a typical cutting plane method is that we do not necessarily have a feasible linear program to begin with.

- In fact, we may not even have a feasible relaxation.

- We need to generate some initial columns.

- How this is done has a big affect on the effectiveness of the overall algorithm.

- Options
  - Use knowledge of problem structure to generate solutions heuristically.
  - Use problem structure to generate a set of solutions guaranteed to be feasible.
  - Solve the subproblem with randomly perturbed objectives.
  - Do a few iterations of Lagrangian relaxation.
  - Use the generic phase 1 (discussed next).
A Generic Algorithm for Phase I

- The initial set of columns may not yield a feasible relaxation.
- The proof of infeasibility is a dual ray with negative cost that can serve as an objective in finding the next column.
- Essentially, we want to find a new column that “cuts” this dual ray.
- This process continues until the relaxation is feasible.
- Alternatively, we can add artificial variables to ensure feasibility.
- These options mirror the ones we have in standard linear programming.
Using Heuristic Methods to Solve the Subproblem

- Technically, all that is needed in each iteration is some column with positive reduced cost.

- It may not matter if the entering column is the one with the most positive reduced cost.

- We can thus use a quick and dirty heuristic method to generate columns initially.

- We can stop anytime after we find a column with positive reduced cost.

- Note, however, that we do not get a true bound in this case.
Stabilization

• One of the well-known difficulties with column generation is the slow convergence.

• This can sometimes be due to wild fluctuations in the dual solution.

• There are a number of techniques for dealing with this phenomena.
  – Artificially bounding the dual variables.
  – Penalizing changes in the dual solution.
  – Trust region method.
  – Weighted Dantzig-Wolfe.
  – Using an interior point method to solve the RMP.

• These methods can have a big impact in practice.