Chapter 6 Large scale optimization

6.1 Delayed column generation

Consider LP

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned} \qquad A \in \mathbb{R}^{m \times n}, \ \operatorname{rank}(A) = m, \quad m \ll n$$

and suppose that the number of columns of A is huge such that A cannot be generated or stored in memory.

Remember: Revised simplex method only requires m basic columns and the new column that will enter the basis.

Problem: How to find column j that should enter the basis in the next iteration (i.e., $\bar{c}_j < 0$)?

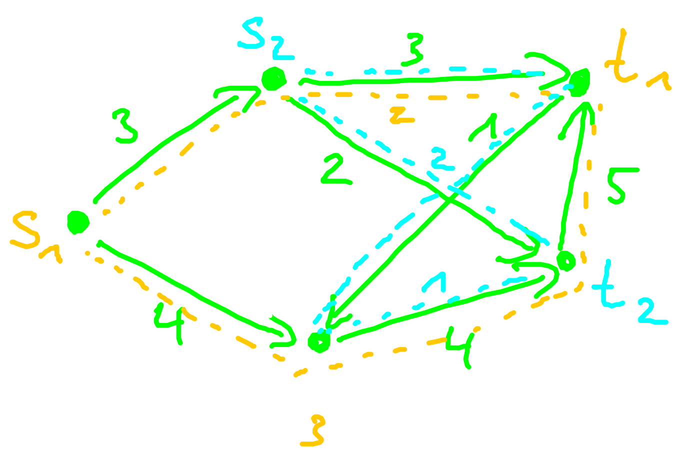
"Pricing Problem"

Solution: Sometimes one can find j with $\bar{c}_j = \min_i \bar{c}_i$ efficiently.

Example: Min Cost Multicommodity Flows

Given: Directed graph $D=(V,A)$ with arc capacities $u: A \to \mathbb{R}_{\geq 0}$ and costs $c: A \to \mathbb{R}_{\geq 0}$; k source-sink pairs $(s_i, t_i) \in V \times V$, $i = 1, \dots, k$ with demands $d_i \in \mathbb{R}_{\geq 0}$.

Task: Send d_i units of flow from s_i to t_i simultaneously for all $i=1,\dots,k$ without violating arc capacities and at minimum total cost.



$d_1 = 5$

$d_2 = 3$

LP formulation:

Let $\mathcal{P}_i = \{P \mid P \text{ is an } s_i\text{-}t_i\text{-path in } D\}$

$$\mathcal{P} = \bigcup_{i=1}^{k} \mathcal{P}_i \qquad c_P := \sum_{a \in P} c(a)$$

Introduce variables:

x_p denotes the amount of flow that is sent along path P. $\quad P \in \mathcal{P}$

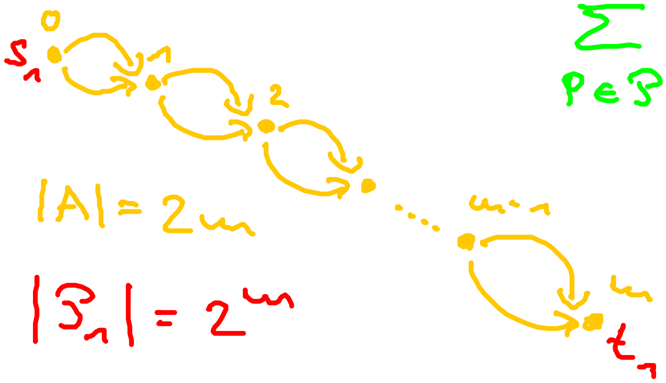
$$\min \sum_{P \in \mathcal{P}} c_P \cdot x_P$$

$$\text{s.t.} \quad \sum_{P \in \mathcal{P}: a \in P} x_P + s_a = u(a) \quad \forall a \in A \qquad y_a$$

$$\sum_{P \in \mathcal{P}_i} x_P = d_i \quad \forall i = 1, \dots, k \qquad z_i$$

$$x_P \geq 0 \quad \forall P \in \mathcal{P}$$

$$s_a \geq 0 \quad \forall a \in A$$



$|A| = 2m$

$|\mathcal{P}_1| = 2^m$

variables $= |\mathcal{P}| + |A|$ exponential in the input size (size of network).

Consider dual LP:

$$\max \sum_{a \in A} u(a) \cdot y_a + \sum_{i=1}^{k} d_i \cdot z_i$$

$$z_i + \sum_{a \in P} y_a \leq c_P \quad \forall P \in \mathcal{P}_i, i = 1, \dots, k$$

$$y_a \leq 0 \quad \forall a \in A$$

Notice that the reduced cost of a primal variable is negative if and only if the corresponding dual constraint is violated.

$$\bar{c}_{s_a} = -y_a \overset{?}{<} 0$$

This can be easily checked for the slack variables (only $|A|$ such variables)
How about path variables x_P, $P \in \mathcal{P}$?

$$z_i + \sum_{a \in P} y_a > c_P = \sum_{a \in P} c(a)$$

$$\iff z_i > \sum_{a \in P} \underbrace{(c(a) - y_a)}_{\geq 0}$$

Thus, the pricing problem for the x_P variables can be solved by computing a shortest s_i-t_i-path for all $i = 1, \ldots, k$ with respect to arc weights $c(a) - y_a \geq 0$.
If the length of a shortest s_i-t_i-path P_i is $< z_i$, we have found $\bar{c}_{P_i} < 0$.
Otherwise $\bar{c}_P \geq 0 \;\; \forall P \in \mathcal{P}_i$.

A variant involving retained columns

When solving an LP with delayed column generation, columns that exit the basis in some iteration, are usually discarded and might have to be rediscovered later again.
A variant of this method retains in

memory some or even all columns that have been in the basis previously.

6.3 Cutting plane methods

Delayed column generation viewed in terms of the dual LP.

$$\max p^T b$$
$$\text{s.t. } p^T \cdot A_i \leq c_i \quad i = 1, \dots, n$$

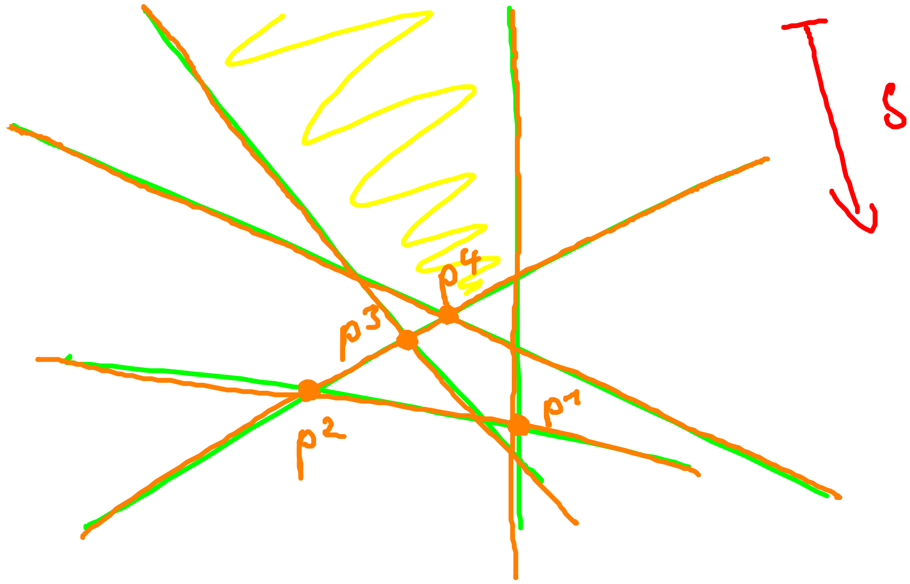
Instead of dealing with all n constraints, restrict to some subset $I \subseteq \{1, \dots, n\}$ and consider the relaxed problem:

$$\max p^T \cdot b$$
$$\text{s.t. } p^T \cdot A_i \leq c_i \ \forall i \in I$$

Let p^* be an optimal solution (basic feasible solution) to the relaxation.

(i) If p^* fullfills all constraints, then p^* is optimal for the original dual LP.

(ii) Otherwise, find a violated constraint and add it to I.



p4
p3
p2
p1
S