15 Large-Scale Linear Programming

15.1 Column Generation

Delayed Column Generation

Let $A \in \mathbb{R}^{m \times n}$ with $\operatorname{rank}(A) = m$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, and $m \ll n$.

$$\begin{aligned} \min \;\; & c^\top \cdot x \\ \text{s.t.} \;\; & A \cdot x = b \\ & x \geq 0 \end{aligned}$$

Suppose that the number of columns n is huge such that A cannot be generated and stored in your computer's memory.

Remember: Revised simplex method only requires m basic columns and the column which shall enter the basis.

Pricing problem: How to find column that should enter basis (i.e., $\bar{c}_j < 0$)?

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

Conclusion:

- Only work with few columns at a time (basic columns and some "promising" non-basic columns).
- Generate new relevant columns by solving pricing problem.

Example: Min-Cost Multi-Commodity Flows

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

Task: Send d_i units of flow from s_i to t_i for all i without violating arc capacities; minimize total cost.

Path-based LP formulation: Let $\mathcal{P}_i$ be the set of all s_i-t_i-dipaths in D, $\mathcal{P} := \bigcup_{i=1}^{k} \mathcal{P}_i$. Cost of path $P \in \mathcal{P}$ is $c_P := \sum_{a \in P} c(a)$.

$$
\begin{aligned}
\min \quad & \sum_{P \in \mathcal{P}} c_P \, x_P \\
\text{s.t.} \quad & \sum_{P \in \mathcal{P} : a \in P} x_P + s_a = u(a) && \text{for all } a \in A \\
& \sum_{P \in \mathcal{P}_i} x_P = d_i && \text{for all } i = 1, \dots, k \\
& x_P, s_a \geq 0 && \text{for all } P \in \mathcal{P}, a \in A
\end{aligned}
$$

Notice: The number of variables is exponential in the size of D.

Pricing Problem and Dual Separation Problem

Consider the dual LP:

$$\begin{aligned} \max \quad & \sum_{a \in A} u(a) \cdot y_a + \sum_{i=1}^{k} d_i \cdot z_i \\ \text{s.t.} \quad & z_i + \sum_{a \in P} y_a \le c_P && \text{for all } P \in \mathcal{P}_i,\ i = 1, \dots, k \\ & y_a \le 0 && \text{for all } a \in A \end{aligned}$$

Notice: The reduced cost of a primal variable is negative if and only if the corresponding dual constraint is violated (Observation 7.9)

$\longrightarrow$ dual separation problem

Easy for slack variable s_a: Check whether $y_a > 0$.

For path variable x_P, $P \in \mathcal{P}_i$: $\quad z_i + \sum_{a \in P} y_a > c_P = \sum_{a \in P} c(a)$

$$\iff \quad \sum_{a \in P} (c(a) - y_a) < z_i$$

Conclusion: Solve pricing problem by computing shortest s_i-t_i-paths w.r.t. arc weights $c(a) - y_a$, for $i = 1, \dots, k$.

15 Large-Scale Linear Programming

15.2 Cutting Planes

Cutting Plane Methods

Delayed column generation viewed in terms of the dual LP:

$$\max \quad p^\top \cdot b \qquad \text{s.t.} \quad p^\top \cdot A_i \leq c_i \qquad \text{for all } i = 1, \ldots, n$$

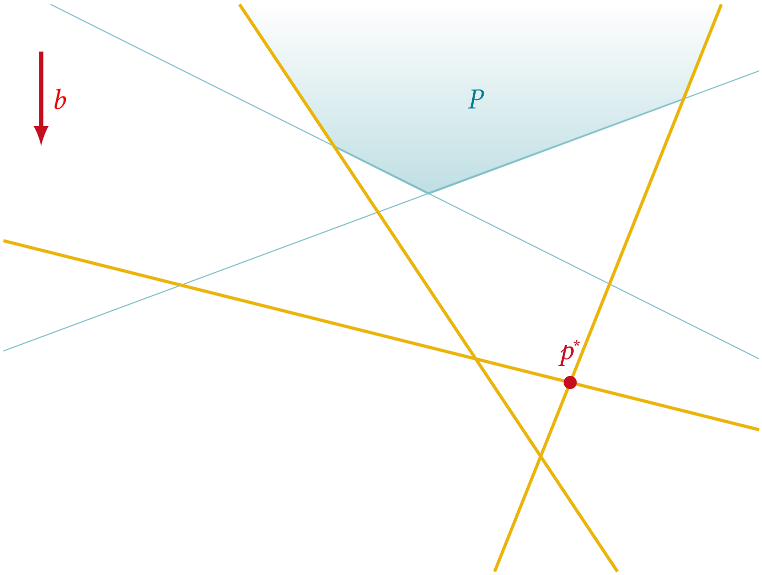
If n is huge, instead of dealing with all n constraints, restrict to subset $I \subset \{1, \ldots, n\}$ and consider relaxed problem

$$\max \quad p^\top \cdot b \qquad \text{s.t.} \quad p^\top \cdot A_i \leq c_i \qquad \text{for all } i \in I$$

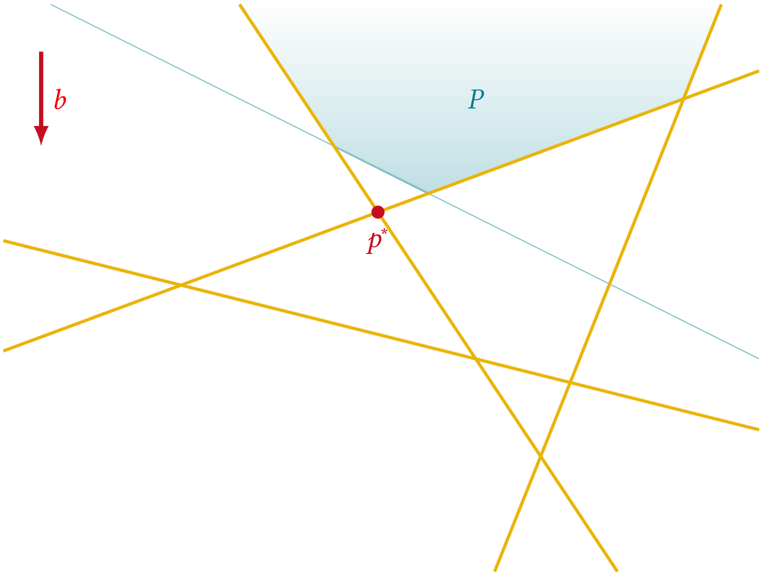
Let p^* be an optimal basic feasible solution:

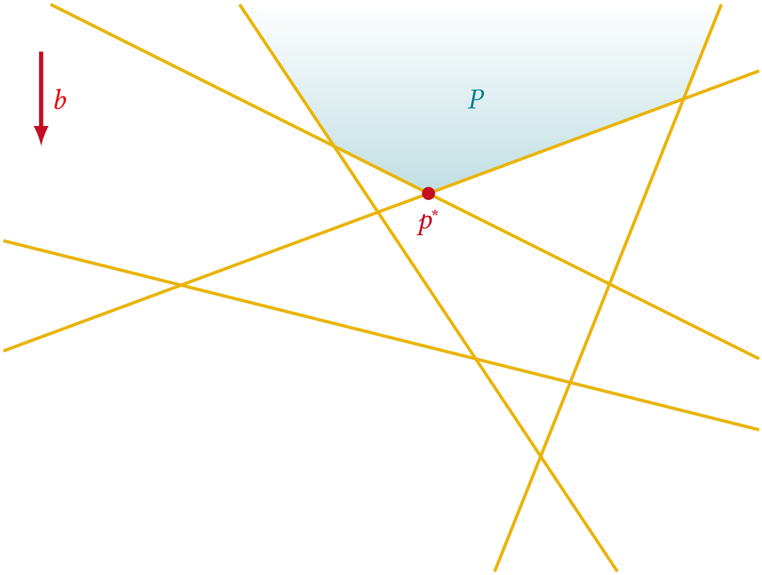
- If p^* is feasible for original LP, it is also optimal there.
- Otherwise, find a violated constraint and add it to relaxed problem.

Remark: Notice the similarity to the ellipsoid method where, in every iteration, the separation problem needs to be solved.



b
P
p*


b
P
p*


b
P
p*

Reminder: Solving the Subtour LP

For a given TSP instance, consider the subtour LP:

$$
\begin{aligned}
\min \quad & \sum_{e \in E} c_e \cdot x_e \\
\text{s.t.} \quad & \sum_{e \in \delta(v)} x_e = 2 && \text{for all nodes } v \in V, \\
& \sum_{e \in \delta(X)} x_e \geq 2 && \text{for all subsets } \emptyset \neq X \subsetneq V, \qquad (\star) \\
& 0 \leq x_e \leq 1 && \text{for all edges } e.
\end{aligned}
$$

Notice that there are $2^{n-1} - 1$ subtour elimination constraints $(\star)$.

The corresponding separation problem is a min-cut problem that can be solved efficiently by network flow methods.

Conclusion: Subtour LP is typically being solved by cutting plane methods.

15 Large-Scale Linear Programming

15.3 Dantzig-Wolfe-Decomposition

Dantzig-Wolfe Decomposition

Consider a linear program of the form

$$\begin{array}{rrcrcl} \min & c_1^\top \cdot x_1 & + & c_2^\top \cdot x_2 & & \\ \text{s.t.} & D_1 \cdot x_1 & + & D_2 \cdot x_2 & = & b_0 \\ & F_1 \cdot x_1 & & & = & b_1 \\ & & & F_2 \cdot x_2 & = & b_2 \\ & & & x_1,\ x_2 & \geq & 0 \end{array}$$

with $c_1 \in \mathbb{R}^{n_1}$, $c_2 \in \mathbb{R}^{n_2}$, $b_0 \in \mathbb{R}^{m_0}$, $b_1 \in \mathbb{R}^{m_1}$, $b_2 \in \mathbb{R}^{m_2}$.

Reformulation of the problem: For $i = 1, 2$, let $P_i := \{x_i \geq 0 \mid F_i \cdot x_i = b_i\}$.

$$\begin{array}{rrcrcl} \min & c_1^\top \cdot x_1 & + & c_2^\top \cdot x_2 & & \\ \text{s.t.} & D_1 \cdot x_1 & + & D_2 \cdot x_2 & = & b_0 \\ & x_1 \in P_1 & , & x_2 \in P_2 & & \end{array}$$

- Let x_i^j, $j \in \mathcal{J}_i$, be the extreme points of P_i.
- Let w_i^k, $k \in K_i$, be a complete set of extreme rays of P_i.

Reformulation

For $i = 1, 2$, $x_i \in \mathbb{R}^{n_i}$ it holds that $x_i \in P_i$ if and only if

$$x_i = \sum_{j \in \mathcal{J}_i} \lambda_i^j \cdot x_i^j + \sum_{k \in K_i} \theta_i^k \cdot w_i^k$$

for some $\lambda_i^j,\ \theta_i^k \geq 0$ and $\sum_{j \in \mathcal{J}_i} \lambda_i^j = 1$.

The reformulation thus leads to the following principal problem:

$$\begin{aligned}
\min \sum_{j \in \mathcal{J}_1} \lambda_1^j \,(c_1^\top x_1^j) + \sum_{k \in K_1} \theta_1^k \,(c_1^\top w_1^k) + \sum_{j \in \mathcal{J}_2} \lambda_2^j \,(c_2^\top x_2^j) + \sum_{k \in K_2} \theta_2^k \,(c_2^\top w_2^k) \\
\text{s.t.} \sum_{j \in \mathcal{J}_1} \lambda_1^j \begin{pmatrix} D_1 x_1^j \\ 1 \\ 0 \end{pmatrix} + \sum_{k \in K_1} \theta_1^k \begin{pmatrix} D_1 w_1^k \\ 0 \\ 0 \end{pmatrix} + \sum_{j \in \mathcal{J}_2} \lambda_2^j \begin{pmatrix} D_2 x_2^j \\ 0 \\ 1 \end{pmatrix} + \sum_{k \in K_2} \theta_2^k \begin{pmatrix} D_2 w_2^k \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} b_0 \\ 1 \\ 1 \end{pmatrix} \\
\lambda_1,\ \lambda_2,\ \theta_1,\ \theta_2 \geq 0
\end{aligned}$$

The principal problem has only $m_0 + 2$ constraints but a huge number of variables. $\longrightarrow$ Employ delayed column generation!

Example: Two-Commodity Flow Problem

Arc-based LP formulation of min-cost two-commodity flow problem:

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{2} \left(\sum_{a \in A} c(a) \cdot x_{i,a} \right) \\
\text{s.t.} \quad & \sum_{i=1}^{2} x_{i,a} \le u(a) && \text{for } a \in A \\
& \sum_{a \in \delta^-(v)} x_{i,a} - \sum_{a \in \delta^+(v)} x_{i,a} = \begin{cases} d_i & \text{if } v = t_i \\ -d_i & \text{if } v = s_i \\ 0 & \text{otherwise} \end{cases} && \text{for } i = 1, 2 \\
& x \ge 0
\end{aligned}
$$

- For $i = 1, 2$, let $P_i := \{x_i \mid x_i \text{ is } s_i\text{-}t_i\text{-flow of value } d_i\}$.
- Extreme points of polyhedron P_i: s_i-t_i-path flows of value d_i (denoted by x_i^P for s_i-t_i-path $P \in \mathcal{P}_i$)
- Extreme rays of polyhedron P_i: cycle flows; these can be ignored as they have positive cost.

Example (Cont.)

Principal problem:

$$\begin{aligned}
\min \quad & \sum_{P \in \mathcal{P}_1} \lambda_1^P \cdot (c_P\, d_1) + \sum_{P \in \mathcal{P}_2} \lambda_2^P \cdot (c_P\, d_2) \\
\text{s.t.} \quad & \sum_{P \in \mathcal{P}_1 : a \in P} \lambda_1^P \cdot d_1 + \sum_{P \in \mathcal{P}_2 : a \in P} \lambda_2^P \cdot d_2 \leq u(a) && \text{for } a \in A \\
& \sum_{P \in \mathcal{P}_i} \lambda_i^P = 1 && \text{for } i = 1, 2 \\
& \lambda \geq 0
\end{aligned}$$

- Setting $x_P := \lambda_i^P \cdot d_i$ for $P \in \mathcal{P}_i$ yields the path-based LP formulation!
- The ith subproblem (pricing problem for variables λ_i^P, $P \in \mathcal{P}_i$) is a shortest s_i-t_i-path problem.

Pricing Problem

Let B be a feasible basis to the principal problem and $p^\top := c_B^\top \cdot A_B^{-1}$ the associated dual solution: $p^\top = (q^\top, r_1, r_2)$ with $q \in \mathbb{R}^{m_0}$, $r_1,\ r_2 \in \mathbb{R}$.

Compute the reduced cost coefficient of a variable λ_1^j:

$$c_1^\top \cdot x_1^j - (q^\top, r_1, r_2) \cdot \begin{pmatrix} D_1 \cdot x_1^j \\ 1 \\ 0 \end{pmatrix} = (c_1^\top - q^\top \cdot D_1) \cdot x_1^j - r_1$$

Compute the reduced cost coefficient of a variable θ_1^k:

$$c_1{}^\top \cdot w_1^k - (q^\top, r_1, r_2) \cdot \begin{pmatrix} D_1 \cdot w_1^k \\ 0 \\ 0 \end{pmatrix} = (c_1^\top - q^\top \cdot D_1) \cdot w_1^k$$

In order to solve the pricing problem for variables λ_i^j and θ_i^k, we consider the following LP:

$$\min\ (c_i^\top - q^\top \cdot D_i) \cdot x_i \quad \text{s.t. } x_i \in P_i$$

This is called the ith subproblem.

Pricing Problem (Cont.)

Consider ith subproblem: $\min\ (c_i^\top - q^\top \cdot D_i) \cdot x_i \quad \text{s.t. } x_i \in P_i$

Case 1: ith subproblem is unbounded:

$\Longrightarrow$ simplex algorithm yields extreme ray w_i^k with $(c_i^\top - q^\top \cdot D_i) \cdot w_i^k < 0$

$\Longrightarrow$ reduced cost of θ_i^k is negative

$\longrightarrow$ generate column $\begin{pmatrix} D_i w_i^k \\ 0 \\ 0 \end{pmatrix}$ and let it enter the basis in pricipal problem.

Case 2: ith subproblem has finite optimal cost $< r_i$:

$\Longrightarrow$ simplex algorithm yields extreme point x_i^j with $(c_i^\top - q^\top \cdot D_i) \cdot x_i^j < r_i$

$\Longrightarrow$ reduced cost of λ_i^j is negative

$\longrightarrow$ generate column $\begin{pmatrix} D_i x_i^j \\ \vdots \end{pmatrix}$ and let it enter the basis in principal problem.

Case 3: ith subproblem has finite optimal cost $\geq r_i$:

$$\Longrightarrow \quad \begin{aligned} (c_i^\top - q^\top \cdot D_i) \cdot x_i^j &\geq r_i \quad \text{for all } j \in \mathcal{J}_i \text{ and} \\ (c_i^\top - q^\top \cdot D_i) \cdot w_i^k &\geq 0 \quad \text{for all } k \in K_i. \end{aligned}$$

$\Longrightarrow$ Variables λ_i^j and θ_i^k have reduced cost ≥ 0, for all $j \in \mathcal{J}_i$, $k \in K_i$.

Summary

- The given problem is transformed into an equivalent problem with few constraints but many variables.
- The pricing problem can be solved by solving smaller LPs over the polyhedra P_i.

Economic interpretation: Organization with two divisions and common objective $D_1 \cdot x_1 + D_2 \cdot x_2 = b_0$.

- Central planner assigns values q for each unit of contribution towards common objective.
- Division i wants to minimize $c_i^\top \cdot x_i$ s.t. its own constraint $x_i \in P_i$.
- Since x_i contributes $D_i \cdot x_i$ towards common objective, the overall objective for division i is $\min(c_i^\top - q^\top \cdot D_i) \cdot x_i$.
- The divisions propose solutions to the central planner who combines them with previous solutions and comes up with new values q.

Principal problem:

$$\begin{aligned}
\min \quad & \sum_{P\in\mathcal{P}_1} \lambda_1^P \cdot (c_P\, d_1) + \sum_{P\in\mathcal{P}_2} \lambda_2^P \cdot (c_P\, d_2) & \\
\text{s.t.} \quad & \sum_{P\in\mathcal{P}_1 : a\in P} \lambda_1^P \cdot d_1 + \sum_{P\in\mathcal{P}_2 : a\in P} \lambda_2^P \cdot d_2 \le u(a) & \text{for } a \in A \\
& \sum_{P\in\mathcal{P}_i} \lambda_i^P = 1 & \text{for } i = 1, 2 \\
& \lambda \ge 0 &
\end{aligned}$$

Let B be a feasible basis to the principal problem and $p^\top := c_B^\top \cdot A_B^{-1}$ the associated dual solution: $p^\top = (y^\top, z_1, z_2)$ with $y \in \mathbb{R}^A$, $z_1, z_2 \in \mathbb{R}$.

In order to solve the pricing problem for variables λ_i^P, we consider the LP:

$$\min\ (c^\top - y^\top) \cdot x_i \quad \text{s.t.}\ \ x_i \in P_i.$$

This is equivalent to finding a shortest s_i-t_i-path for arc weights $c(a) - y_a$.

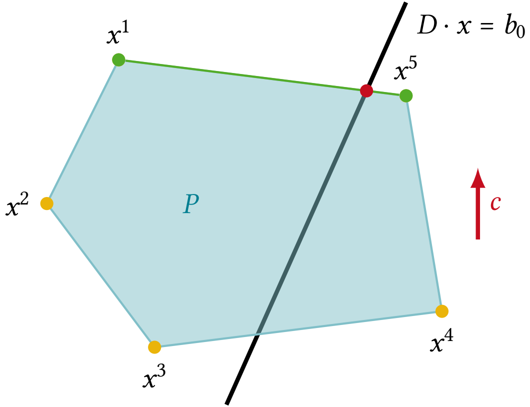
Generalization

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{t} c_i^\top \cdot x_i \\
\text{s.t.} \quad & \sum_{i=1}^{t} D_i \cdot x_i = b_0 \\
& F_i \cdot x_i = b_i \qquad \text{for } i = 1, \ldots, t \\
& x_1, \ldots, x_t \geq 0
\end{aligned}
$$

- Proceed as before $\longrightarrow$ t subproblems for pricing.
- Sometimes even useful for $t = 1$.

Example for $t = 1$

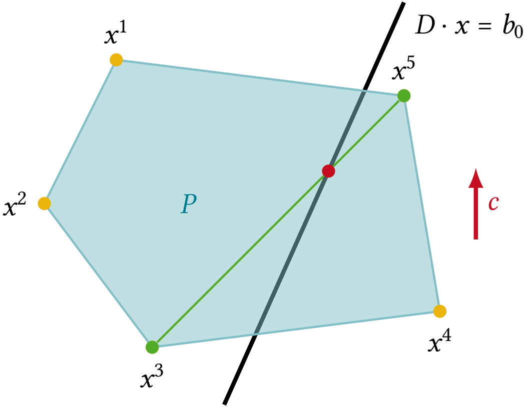
$$\min\left\{ \sum_{j=1}^{5} \lambda^j \,(c^\top x^j) \,\middle|\, \sum_{j=1}^{5} \lambda^j \,(D\, x^j) = b_0,\ \sum_{j=1}^{5} \lambda^j = 1,\ \lambda \geq 0 \right\}$$



Basis of the principal problem: $(1, 5)$

Example for $t = 1$

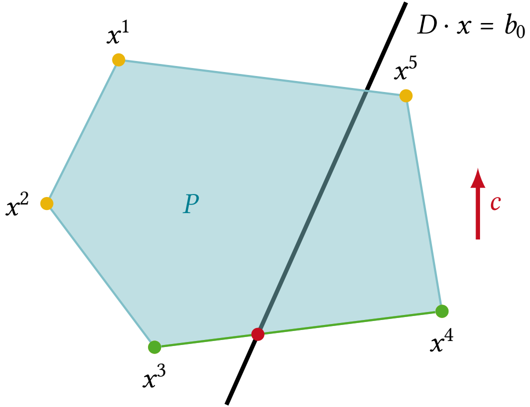
$$\min \left\{ \sum_{j=1}^{5} \lambda^j \,(c^\top x^j) \;\middle|\; \sum_{j=1}^{5} \lambda^j \,(D\, x^j) = b_0,\; \sum_{j=1}^{5} \lambda^j = 1,\; \lambda \geq 0 \right\}$$



Basis of the principal problem: $(3, 5)$

Example for $t = 1$

$$\min \left\{ \sum_{j=1}^{5} \lambda^j \, (c^\top x^j) \;\middle|\; \sum_{j=1}^{5} \lambda^j \, (D\, x^j) = b_0, \; \sum_{j=1}^{5} \lambda^j = 1, \; \lambda \geq 0 \right\}$$



Basis of the principal problem: $(3, 4)$

Dantzig-Wolfe Decomposition: Phase I

How to find an initial basic feasible solution?

- Use phase I of simplex method to find an extreme point x_i^1 of P_i, for $i = 1, \dots, t$.
- W.l.o.g. $\sum_{i=1}^{\top} D_i \cdot x_i^1 \le b_0$. Introduce slack variables $y \in \mathbb{R}^{m_0}$ and solve auxiliary principal problem:

$$\begin{aligned} \min \quad & \sum_{s=1}^{m_0} y_s \\ \text{s.t.} \quad & \sum_{i=1}^{\top} \left(\sum_{j \in \mathcal{J}_i} \lambda_i^j (D_i \cdot x_i^j) + \sum_{k \in K_i} \theta_i^k (D_i \cdot w_i^k) \right) + y = b_0 \\ & \sum_{j \in \mathcal{J}_i} \lambda_i^j = 1 \qquad \text{for } i = 1, \dots, t \\ & \lambda,\ \theta,\ y \ge 0 \end{aligned}$$

Introduction to

Linear and Combinatorial Optimization

15 Large-Scale Linear Programming

15.4 Benders Decomposition

Two-stage Optimization Problems

- Consider a linear program of the form

$$\begin{aligned} \min \quad & f^\top y + c^\top x \\ s.t. \quad & Fy \geq h \\ & Ay + Bx \geq b \\ & x \geq 0 \end{aligned}$$

- The above form occurs in many real-world problems with different stages of decision. Typically, y variables represent *here-and-now* decisions, while x variables represent *wait-and-see* decisions, whose domain depends on the value of the first stage variables.

- More generally, B can have a block-diagonal structure, so Benders Decomposition can be used to handle problems of the form:
- ⟶ Benders Decomposition can be viewed as Dantzig-Wolfe applied to the dual problem.

$$\begin{aligned} \min \quad & f^\top y + c_1^\top x_1 + \ldots + c_k^\top x_k \\ s.t. \quad & Fy \geq h \\ & A_1 y + B_1 x_1 \geq b_1 \\ & A_2 y + B_2 x_2 \geq b_2 \\ & \vdots \\ & A_k y + B_k x_k \geq b_k \\ & x_1, \ldots, x_k \geq 0 \end{aligned}$$

Recourse model

- It holds

$$\begin{array}{rllll} \min & f^\top y & + c^\top x & & = \quad \min \; f^\top y + Q(y) \\ s.t. & Fy & & \geq h & \qquad\qquad Fy \geq h \\ & Ay & + Bx & \geq b & \\ & x \geq 0 & & & \end{array}$$

  where $Q(y) := \min\{c^\top x : \; Bx \geq b - Ay, \; x \geq 0\}$ is the cost of the *recourse decisions.*
- Dualizing $Q(y)$ yields $Q(y) = \max\{u^\top(b - Ay) : \; B^\top u \leq c, \; u \geq 0\}$.
- The domain $\mathcal{U} := \{u : \; B^\top u \leq c, u \geq 0\}$ of the recourse problem does not depend on y. So if $\mathcal{U} \neq \emptyset$, the original problem rewrites

$$\begin{array}{rl} \min & f^\top y + z \\ & Fy \geq h \\ & u^\top(b - Ay) \leq z, \quad \forall u \in \mathcal{U} \end{array}$$

- This is a problem with infinitely many constraints, but we can reduce to finitely many by considering extreme points and extreme rays of $\mathcal{U}$.

Benders Decomposition

Let $\mathcal{U} := \{u : B^\top u \le c, u \ge 0\}$,

$R := \{\text{extreme rays of } \mathcal{U}\}$ and $V := \{\text{extreme points of } \mathcal{U}\}$

Theorem 15.1 If $\mathcal{U} \neq \emptyset$, we have

$$\begin{array}{llll} \min & f^\top y & + c^\top x & \\ s.t. & Fy & & \ge h \\ & Ay & + Bx & \ge b \\ & x \ge 0 & & \end{array} \quad = \quad \begin{array}{lll} \min & f^\top y + z & \\ & Fy \ge h & \\ & u^\top(b - Ay) \le 0, & \forall u \in R \\ & u^\top(b - Ay) \le z, & \forall u \in V \end{array}$$

Proof:

- If the optimal value is $< \infty$, then $\exists y^* : Q(y^*) < \infty$. For all z,

$$Q(y) \le z \iff u^\top(b - Ay) \le z, \forall u \in \mathcal{U} \iff \begin{cases} u^\top(b - Ay) \le 0, & \forall u \in R \\ u^\top(b - Ay) \le z, & \forall u \in V \end{cases}$$

- If the primal problem is infeasible, then either
  - $Y = \{y : Fy \ge h\} = \emptyset$
  - or the recourse problem is infeasible for all y. Since the dual recourse problem is feasible, this implies that the dual recourse problem is unbounded, for all y: $\forall y \in Y, \exists u \in R : u^\top(b - Ay) > 0$.

In both cases, the Benders decomposition is infeasible. □

Solving Benders Decomposition with Cutting Planes

- Initialize (possibly empty) subsets $\hat{R}$ and $\hat{V}$ of extreme rays/points of $\mathcal{U}$, and consider the restricted master problem

$$\begin{aligned} \min \quad & f^\top y + z \\ s.t. \quad & Fy \geq h \\ & u^\top (b - Ay) \leq 0, \forall u \in \hat{R} \\ & u^\top (b - Ay) \leq z, \forall u \in \hat{V} \end{aligned}$$

- In every iteration, given a previous solution $(\bar{y}, \bar{z})$ of the RMP, solve the separation problem

$$\max\{u^\top (b - A\bar{y}) \; : \; u \in \mathcal{U}\}.$$

- Since $\mathcal{U} \neq \emptyset$, only 2 cases can occur:
  - The subproblem is unbounded. Then, we find a ray $u \in R$ such that $u^\top (b - A\bar{y}) > 0$. $\longrightarrow$ Set $\hat{R} := \hat{R} \cup \{u\}$.
  - The subproblem has a finite optimal solution at some extreme point $u \in V$.

$\longrightarrow$ If $u^T (b - A\bar{y}) \leq \bar{z}$: STOP; solution is optimal.

$\longrightarrow$ Else, set $\hat{V} := \hat{V} \cup \{u\}$

Example: Two-stage stochastic transshipment

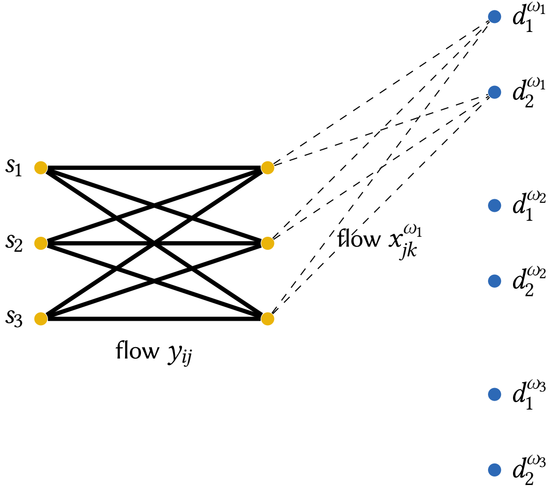
- An E-commerce company has some initial stock $s_i \geq 0$ of some commodity at location $i, \forall i \in I$.
- Between two selling periods, the company can resplenish the warehouses, by moving stocks from location $i \in I$ to location $j \in I$ (unit cost=f_{ij})
- There is a random demand d_k in region $k, \forall k \in K$. Serving one unit of this demand from location $j \in I$ costs c_{jk}.
- We have a set of historical samples $\Omega = \{d^1, ..., d^n\}$ for the vector of demands $d = (d_k)_{k \in K} \in \mathbb{R}^{|K|}_{\geq 0}$, and we estimate the probability distribution of the demands by

$$Pr[d = d^\omega] = \frac{1}{|\Omega|}, \qquad \forall \omega \in \Omega.$$

- The resplenishment flow $(y_{ij})_{i \in I, j \in I}$ is a here-and-now variable, which must be decided before observing the demand.
- The delivery flow $(x^\omega_{jk})_{j \in I, k \in K, \omega \in \Omega}$ can be decided after observing the demand, hence it depend on the observed scenario: we serve x^ω_{jk} units of the demand of region k from warehouse j in the scenario ω.
- We assume there is enough stock: $\sum_{i \in I} s_i \geq \sum_{k \in K} d^\omega_k, \forall \omega \in \Omega$

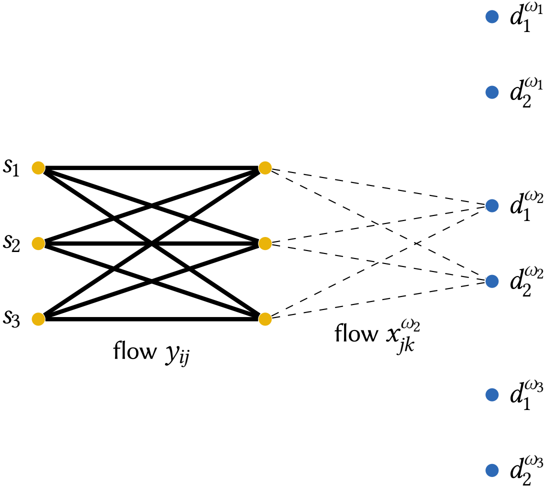
Example: Two-stage stochastic transshipment

Example with $|I| = 3$ warehouses, $|K| = 2$ demand regions and $|\Omega| = 3$ scenarios.



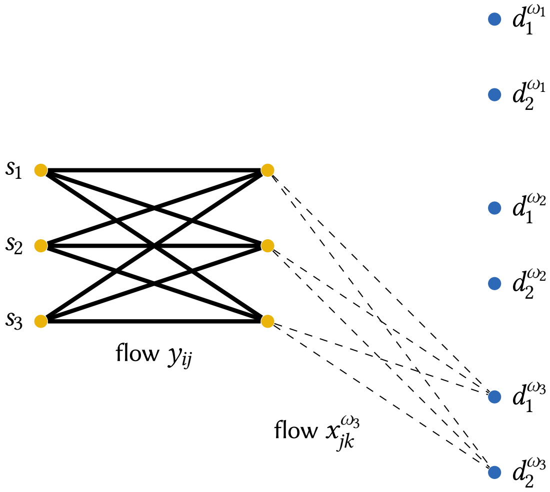
Example: Two-stage stochastic transshipment

Example with $|I| = 3$ warehouses, $|K| = 2$ demand regions and $|\Omega| = 3$ scenarios.



Example: Two-stage stochastic transshipment

Example with $|I| = 3$ warehouses, $|K| = 2$ demand regions and $|\Omega| = 3$ scenarios.



Two-stage stochastic transshipment: LP formulation

The problem of minimizing the total expected cost (over both decision stages) can be represented as a large LP:

$$\begin{aligned}
\min \quad & \sum_{i\in I, j\in I} f_{ij} y_{ij} + \sum_{\omega\in\Omega} \frac{1}{|\Omega|} \sum_{j\in I, k\in K} c_{jk} x_{jk}^{\omega} \\
s.t. \quad & \sum_{j\in I} y_{ij} = s_i, \quad \forall i \in I \\
& \sum_{k\in K} x_{jk}^{\omega} \le \sum_{i\in I} y_{ij}, \quad \forall j \in I, \forall \omega \in \Omega \\
& \sum_{j\in I} x_{jk}^{\omega} = d_k^{\omega}, \quad \forall k \in K, \forall \omega \in \Omega \\
& x, y \ge 0
\end{aligned}$$

- Denote by $s_j'(y) = \sum_{i\in I} y_{ij}$ the stock of j after the resplenishment
- The recourse problem for a fixed scenario $\omega \in \Omega$ is a simple transshipment problem

$$\begin{aligned}
Q_\omega(y) &= \min\{\sum_{jk} c_{jk} x_{jk} \; : \quad \sum_{k} x_{jk} \le s_j'(y), \quad \sum_{j} x_{jk} = d_k^{\omega}, \quad x \ge 0\} \\
&= \max\{-\sum_{j} u_j s_j'(y) + \sum_{k} v_k d_k^{\omega}, \quad v_k - u_j \le c_{jk}, \quad u \ge 0\}
\end{aligned}$$

Two-stage stochastic transshipment: Benders decomp…

- Restricted Master Problem (no extreme rays, recourse problem always feasible)

$$
\begin{aligned}
\min \quad & \sum_{i\in I, j\in I} f_{ij} y_{ij} + \sum_{\omega\in\Omega} \frac{1}{|\Omega|} z^\omega \\
s.t. \quad & \sum_{j\in I} y_{ij} = s_i, \quad \forall i \in I \\
& s'_j = \sum_{i\in I} y_{ij}, \quad \forall j \in I \\
& -\sum_{j\in I} u_j s'_j + \sum_k v_k d_k^\omega \le z^\omega, \quad \forall (u,v) \in \hat{V}^\omega \\
& y \ge 0
\end{aligned}
$$

- Recourse problem for scenario ω, given a first-stage solution y: solve the dual transhipment problem

$$
\max\{-\sum_j u_j s'_j(y) + \sum_k v_k d_k^\omega, \quad v_k - u_j \le c_{jk}, \forall j \in I, k \in K, \quad u \ge 0\}
$$

- If optimal recourse has value $\le z^\omega$ for all $\omega \in \Omega$: STOP.
- Otherwise, add an extreme point in $\hat{V}^\omega$ for each scenario $> z^\omega$ and re-solve RMP.

Benders decomposition and Integer variables

- In fact, one can also consider problems with integer first-stage variables
- This does not change anything in the derivation of the Benders decomposition, but we need a MIP solver to solve the RMP
- For example, we could modify the previous example to compute the number of trucks required for the first-stage transshipment.
  - The cost for a truck from i to j is f_{ij}^t.
  - Each truck has a capacity of C
  - This is equivalent to assuming that the cost for transporting y_{ij} units from i to j is equal to $\lceil \frac{y_{ij}}{C} \rceil \cdot f_{ij}^t$.
- Denote by n_{ij} the number of trucks driving from i to j. The RMP is changed as follows:
  - The objective function becomes: $\sum_{i \in I, j \in I} f_{ij}^t n_{ij} + \sum_{\omega \in \Omega} \frac{1}{|\Omega|} z^\omega$
  - We need to add the following constraints:

$$
\begin{aligned}
y_{ij} &\leq C \cdot n_{ij}, && \forall i, j \in I \\
n_{ij} &\in \mathbb{Z}_{\geq 0}, && \forall i, j \in I.
\end{aligned}
$$