Lecture 9: Dantzig-Wolfe Decomposition

(3 units)

Outline

- Dantzig-Wolfe decomposition
- Column generation algorithm
- Relation to Lagrangian dual
- Branch-and-price method
- Generated assignment problem and multi-commodity flow problem
- References

Dantzig-Wolfe decomposition

- Consider the following integer programming problem:

$$
\begin{array}{llllllll}
(P) & \min & (c^1)^T x^1 & +(c^2)^T x^2 & + \cdots & +(c^k)^T x^k & \\
& \text{s.t.} & A^1 x^1 & +A^2 x^2 & + \cdots & +A^K x^K & = b \\
& & D^1 x^1 & & & & \leq d_1 \\
& & & \cdots & & & \leq \cdot \\
& & & & \cdots & & \leq \cdot \\
& & & & & D^K x^K & \leq d_K \\
& & x^1 \in \mathbb{Z}_+^{n_1}, & \ldots, & & x^K \in \mathbb{Z}_+^{n_k} &
\end{array}
$$

- The constraint matrix has a block angular structure. Let

$$X^k = \{x^k \in \mathbb{Z}^{n_k} \mid D^k x^k \leq d_k\}.$$

- The sets are independent for $k = 1, \ldots, K$, only the joint constraint $\sum_{k=1}^{K} A^k x^k = b$ link together the different sets of variables.
- Dualizing the joint constraint, we have the following Lagrangian dual of (P):

$$(D) \quad \max_u L(u),$$

where

$$\begin{aligned} L(u) &= \min\{\sum_{k=1}^{K}((c^k)^T - u^T A^k)x^k + b^T u \mid x^k \in X^k, \ \forall k\} \\ &= \sum_{k=1}^{K} L_k(u) + b^T u. \end{aligned}$$

where $L_k(u) = \min\{((c^k)^T - u^T A^k)x^k \mid x^k \in X^k\}$.

- We have discussed Lagrangian relaxation and dual search in the previous lectures. Another way of exploiting the block-angular structure is Dantzig-Wolfe decomposition, which was invented by Dantzig and Wolfe in1961. The method is very closely connected to column generation and they are often used interchangeably.
- The set X^k in (P) can be either continuous (polyhedron) or discrete (integer set).
- Minkowski-Weyl's Theorem: Given the convex set $X = \{x \in \mathbb{R}^n \mid Ax \leq b\}$. X can be represented by the extreme points and extreme rays of X:

$$X = \{x = \sum_i \lambda_i x^i + \sum_j \mu_i y^j \mid \sum_i \lambda_i = 1, \ \lambda_i \geq 0, \ \mu_j \geq 0\}.$$

Changing representation

- When X^k is a bounded polyhedron, we can express X^k as

$$X^k = \{x^k = \sum_{t=1}^{T_k} \lambda_{kt} x^{kt} \mid \sum_{t=1}^{T_k} \lambda_{kt} = 1,\ \lambda_{kt} \geq 0\},$$

  where x^{kt}, $t = 1, \ldots, T_k$, are extreme points of X^k.
- When X^k is a finite integer set, we can express X^k as

$$X^k = \{x^k = \sum_{t=1}^{T_k} \lambda_{kt} x^{kt} \mid \sum_{t=1}^{T_k} \lambda_{kt} = 1,\ \lambda_{kt} \in \{0, 1\}\},$$

  where x^{kt}, $t = 1, \ldots, T_k$, list all the points of X^k.

IP Master Problem

- Now, we substitute the expression of x^k in (P), leading to the following IP Master Problem:

$$
\begin{aligned}
(IPM) \qquad \min \; & \sum_{k=1}^{K} \sum_{t=1}^{T_k} ((c^k)^T x^{kt}) \lambda_{kt} \\
\text{s.t.} \; & \sum_{k=1}^{K} \sum_{t=1}^{T_k} (A^k x^{kt}) \lambda_{kt} = b \\
& \sum_{t=1}^{T_k} \lambda_{kt} = 1, \; k = 1, \ldots, K, \\
& \lambda_{kt} \in \{0, 1\}, \; \forall k, t.
\end{aligned}
$$

- How to solve this equivalent problem? Note that the number of variables or columns in the constraint matrix is $\sum_{k=1}^{K} T_k$, which is usually exponentially large.

LP master problem

- We consider the LP relaxation of (IPM):

$$
\begin{aligned}
(LPM) \qquad \min\ & \sum_{k=1}^{K} \sum_{t=1}^{T_k} ((c^k)^T x^{kt}) \lambda_{kt} \\
\text{s.t.}\ & \sum_{k=1}^{K} \sum_{t=1}^{T_k} (A^k x^{kt}) \lambda_{kt} = b \\
& \sum_{t=1}^{T_k} \lambda_{kt} = 1,\ k = 1, \ldots, K, \\
& \lambda_{kt} \geq 0,\ \forall k, t.
\end{aligned}
$$

- How can we find all the points (or extreme points in the continuous case) of X^k? $\Rightarrow$ We can use the idea of column generation.

Dual of the LP master problem

- The dual problem of (LPM) is

$$
(DLPM) \qquad \begin{aligned} & \max \sum_{i=1}^{m} b_i \pi_i + \sum_{k=1}^{K} \mu_k \\ & \text{s.t. } \pi A^k x^{kt} + \mu_k \leq (c^k)^T x^{kt}, \\ & \qquad t = 1, \ldots, T_k,\ k = 1, \ldots, K, \end{aligned}
$$

  where $(\pi, \mu_k) \in \mathbb{R}^m \times \mathbb{R}^K$.
- Notice that a column in (LPM) corresponds a row in $(DLPM)$.

Column generation algorithm

- Initialization. Suppose that a subset of columns $\mathcal{P} = \cup_{k=1}^{K} \mathcal{P}_k$ (at least one for each k) is available. Consider the Restricted LP Master Problem:

$$
\begin{aligned}
(RLPM) \qquad & \min \sum_{k=1}^{K} \sum_{t \in \mathcal{P}_k} ((c^k)^T x^{kt}) \lambda_{kt} \\
& \text{s.t.} \ \sum_{k=1}^{K} \sum_{t \in \mathcal{P}_k} (A^k x^{kt}) \lambda_{kt} = b \\
& \sum_{t \in \mathcal{P}_k} \lambda_{kt} = 1, \ k = 1, \ldots, K, \\
& \lambda_{kt} \geq 0, \ t \in \mathcal{P}_k, \ k = 1, \ldots, K.
\end{aligned}
$$

- Let λ^* and $(\pi, \mu) \in \mathbb{R}^m \times \mathbb{R}^K$ be the optimal primal and dual solutions to $(RLPM)$, respectively.

- Primal feasibility. Any feasible solution of (LPM) can be expanded to a feasible solution of $(RLPM)$ (setting $\lambda_{kt} = 0$ for those columns not in $\mathcal{P}_k$. So

$$v(RLPM) = \sum_{i=1}^{m} \pi b_i + \sum_{k=1}^{K} \mu_k \geq v(LPM).$$

- Optimality check for (LPM). We need to check whether (π, μ) is dual feasible for (LPM). For any $x \in X^k$, the corresponding column is

$$\begin{pmatrix} (c^k)^T x \\ A^k x \\ e_k \end{pmatrix}.$$

The reduced cost for this column is $(c^k)^T x - \pi^T A^k x - \mu_k$. Solve the following subproblem:

$$(SP) \quad \zeta_k = \min\{((c^k)^T - \pi A^k)x - \mu_k \mid x \in X^k\}.$$

- Stopping criterion. If $\zeta_k \geq 0$, then the solution (π, μ) is dual feasible to (LPM) and so

$$\sum_{k=1}^{K} \sum_{t \in \mathcal{P}_k} ((c^k)^T x^{kt}) \lambda_{kt}^* = \sum_{i=1}^{m} \pi b_i + \sum_{k=1}^{K} \mu_k \leq v(LPM).$$

So the expanded solution of λ^* is optimal to (LPM).

- Generating a new column. If $\zeta_k < 0$ for some k, the column corresponding to the optimal solution $\tilde{x}$ of the subproblem (SP) has negative reduced cost. Introducing the column

$$\begin{pmatrix} (c^k)^T \tilde{x} \\ A^k \tilde{x} \\ e_k \end{pmatrix}$$

into $(RLPM)$ will lead to a new $(RLPM)$, which can be easily re-optimized using Primal Simplex Method.

- Dual (lower) bound. From the subproblem, we have

$$\zeta_k \leq ((c^k)^T - \pi A^k)x - \mu_k, \ \ \forall x \in X^k,$$

which implies

$$\pi A^k x + (\mu_k + \zeta_k) \leq (c^k)^T x, \ \ \forall x \in X^k.$$

Therefore, setting $\zeta = (\zeta_1, \ldots, \zeta_K)$, we have that $(\pi, \mu + \zeta)$ is dual feasible to (LPM). We thus have

$$\pi^T b + \sum_{k=1}^{K} (\mu_k + \zeta_k) \leq v(LPM).$$

Relation to Lagrangian dual

- Theorem:

$$v(LPM) = \min\{\sum_{k=1}^{K}(c^k)^T x^k \mid \sum_{k=1}^{K} A^k x^k = b,\ x^k \in conv(X^k)\}.$$

- Proof: Note that (LPM) is obtained by substituting

$$x^k = \sum_{t=1}^{T_k} \lambda_{kt} x^{kt},\ \ \sum_{t=1}^{T_k} \lambda_{kt} = 1,\ \lambda_{kt} \geq 0,$$

  where x^{kt}, $t = 1, \ldots, T_k$, are all the integer points in X^k. This is equivalent to $x^k \in conv(X^k)$.

- Theorem: $v(LPM) = v(D)$.
- Therefore, the LP relaxation of the IP master problem is actually equivalent to the Lagrangian dual.

- Consider the general IP problem:

$$\min\{c^T x \mid Ax \leq b,\ Dx \leq d,\ x \in \mathbb{Z}^n\}.$$

- Strength of the D-W decomposition and Lagrangian dual:

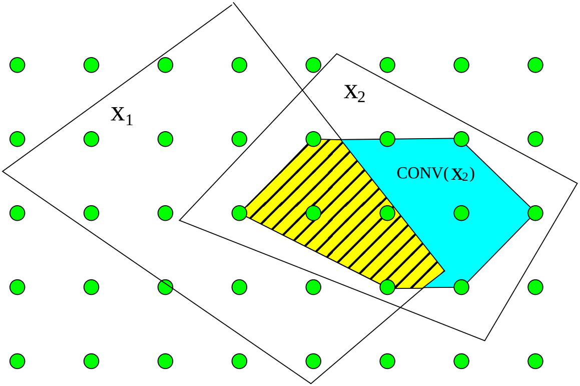

Figure: $X_1 = \{x \in \mathbb{Z}^n \mid Ax \leq b\}$, $X_2 = \{x \in \mathbb{Z}^n \mid Dx \leq d\}$.

Branch-and-Price Method

- Consider the general IP problem:

$$
\begin{aligned}
(IP) \qquad & \min c^T x \\
& \text{s.t. } Ax \le b \quad (\text{“complicating''constraints}) \\
& \qquad Dx \le d, \quad (\text{“easy''constraints}) \\
& \qquad x \in \mathbb{Z}^n.
\end{aligned}
$$

- We assume that optimization over the set $X_2 = \{x \in \mathbb{Z}^n \mid Dx \le d\}$ is “easy”. This will be used in solving the column generation subproblem.
- Let $x^1, \ldots, x^T$ be all the integer points of X_2. Then

$$X_2 = \{x = \sum_{i=1}^{T} \lambda_i x^i \mid \sum_{i=1}^{T} \lambda_i = 1,\ \lambda_i \in \{0,1\}\}.$$

- Use this representation of X_2, we can rewrite (IP) as

$$
\begin{aligned}
(IPM) \qquad & \min c^T(\sum_{i=1}^{T} \lambda_i x^i) \\
& \text{s.t. } A(\sum_{i=1}^{T} \lambda_i x^i) \le b, \\
& \sum_{i=1}^{T} \lambda_i = 1, \\
& \lambda_i \in \{0,1\},\ i = 1, \ldots, T.
\end{aligned}
$$

- This is a reformulation of (IP).

- The LP relaxation of (IPM) is at least as strong as the direct LP relaxation of (IP). (Because the LP relaxation of (IPM) is equivalent to the Lagrangian relaxation of (IP) by dualizing the "complicating" constraint $Ax \leq b$.
- We solve the LP relaxation of (IPM) using column generation.
- Adding one column to the LP master problem corresponds to adding one cutting plane to the dual problem of (LPM).
- The column generation subproblem is an optimization problem over X_2, which can be solved efficiently in many applications.
- We can embed this bounding scheme into a branch and price framework. "Price" here is referred to find the column with negative reduced cost by solving the subproblem.
- How to branch?

Branching with Dantzig-Wolfe Decomposition

- Unfortunately, branching on the variables (λ_i) of the reformulation doesn't work well because it's generally difficult to keep a variable from being generated again after it's been fixed to zero.
- Branching must be done in a way that does not destroy the structure of the column generation master problem and subproblem.
- We can do this by branching on the original variables, i.e., before the reformulation.
- In a 0-1 problem, branching on the jth original variable is equivalent to fixing the value of some element of the columns to be generated. This can usually be incorporated into the column generation subproblem.

Generalized Assignment Problem

- The problem is to assign m tasks to n machines subject to capacity constraints.
- An IP formulation of tho problem is

$$
\begin{aligned}
(GAP) \qquad & \max \sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij} z_{ij} \\
& \text{s.t.} \ \sum_{j=1}^{n} z_{ij} = 1, \quad i = 1, \dots, m, \\
& \sum_{i=1}^{m} w_{ij} z_{ij} \le d_j, \quad j = 1, \dots, n, \\
& z_{ij} \in \{0, 1\}, \ i = 1, \dots, m, j = 1, \dots, n.
\end{aligned}
$$

- Let's the columns in (GAP) represent feasible assignments of tasks to the machine.

$$
X_2 = \{z \in \{0, 1\}^{n \times m} \mid \sum_{i=1}^{m} w_{ij} z_{ij} \le d_j, \quad j = 1, \dots, n\}.
$$

- The Dantzig-Wolfe decomposition master problem is:

$$\begin{aligned}
\max \sum_{j=1}^{n} \sum_{i=1}^{m} p_{ij} & \left(\sum_{k=1}^{T_j} \lambda_j^j y_{ik}^j \right) \\
\text{s.t.} \ \sum_{j=1}^{n} \sum_{k=1}^{T_j} \lambda_j^j y_{ik}^j & = 1, \quad i = 1, \ldots, m, \\
\sum_{i=1}^{T_j} \lambda_k^j & = 1, \quad j = 1, \ldots, n, \\
\lambda_k^j & \in \{0, 1\}, \ j = 1, \ldots, n, \ k = 1, \ldots, T_j.
\end{aligned}$$

  This is a set-covering problem.

- The column generation subproblems are n knapsack problem with a single constraint $\sum_{i=1}^{m} w_{ij} z_{ij} \leq d_j$.

Multi-commodity flow problem

- Let $D = (V, A)$ be a directed graph with nonnegative capacities u_a for all $a \in A$.
- Given k commodities, i.e., node pairs $(s_1, t_1), \ldots, (s_k, t_k)$. For each $i = 1, \ldots, k$, we want to find an $s_i - t_i$-flow, (x_a^i), such that the sum of these flows on each arc does not exceed a given capacity. This flow is called a feasible multicommodity flow and can be expressed by the following constraints:

$$\sum_{a \in \delta^+(v)} x_a^i - \sum_{a \in \delta^-(v)} x_a^i = 0, \quad \forall v \in V \setminus \{s_i, t_i\},\ i = 1, \ldots, k,$$

$$\sum_{i=1}^{k} x_a^i \leq u_a, \quad \forall a \in A,$$

where $\delta^+(v) = \{(v, w) \mid (v, w) \in A\}$ and $\delta^-(v) = \{(u, v) \mid (u, v) \in A\}$.

Edge formulation

- The multicommodity flow problem (MCF) is to maximize the sum of the flow values of each commodes:

$$
\begin{aligned}
\max \sum_{i=1}^{k} & \left(\sum_{a\in\delta^+(s_i)} x_a^i - \sum_{a\in\delta^-(s_i)} x_a^i \right) \\
\text{s.t.} \quad & \sum_{a\in\delta^+(v)} x_a^i - \sum_{a\in\delta^-(v)} x_a^i = 0, \quad \forall v \in V \setminus \{s_i, t_i\},\ i = 1, \dots, k \\
& \sum_{i=1}^{k} x_a^i \le u_a, \quad \forall a \in A.
\end{aligned}
$$

- This formulation is called edge formulation of MCF. It is a linear program and can be solved polynomially.
- However, if we require the flow to be integer, i.e., $x_a^i \in \mathbb{Z}_+$, then the MCF is NP-hard.

Path formulation

- We now give a path formulation for MCF, which can be viewed as the reformulation of the edge formulation using Dantzig-Wolfe decomposition.
- Let $\mathcal{P}_{st}$ be the set of all $s - t$ paths. Let

$$\mathcal{P} = \mathcal{P}_{s_1,t_1} \cup \cdots \cup \mathcal{P}_{s_k,t_k}.$$

  Let $\mathcal{P}_a$ denote the set of paths that use arc a, i.e., $\mathcal{P}_a = \{p \in \mathcal{P} \mid a \in p\}$.
- For each $p \in \mathcal{P}_{s_i,t_i}$, we define a flow variable f_p. The multi-commodity flow problem can be expressed as

$$\begin{aligned} (MCF) \qquad & \max \sum_{p \in \mathcal{P}} f_p \\ & \text{s.t.} \sum_{p \in \mathcal{P}_a} f_p \le u_a, \quad \forall a \in A, \\ & f_a \ge 0, \quad \forall a \in A. \end{aligned}$$

Column generation for MCF

- Let $\mathcal{P}' \subset \mathcal{P}$ be a subset of all paths. Consider the restricted LP master problem:

$$(MCF') \qquad \max \sum_{p \in \mathcal{P}'} f_p$$
$$\text{s.t.} \sum_{p \in \mathcal{P}'_a} f_p \leq u_a, \quad \forall a \in A,$$
$$f_a \geq 0, \quad \forall a \in A.$$

- Note that a feasible solution $(f'_p)_{p \in \mathcal{P}'}$ for (MCF') can be expanded to a feasible solution $(f_p)_{p \in \mathcal{P}}$ to (MCF) (setting $f_p = 0$ for $p \in \mathcal{P} \setminus \mathcal{P}'$. Thus,

$$v(MCF') \leq v(MCF).$$

- The dual of (MCF') is

$$
\begin{aligned}
(DMCF') \qquad & \min \sum_{a \in A} u_a \mu_a \\
& \text{s.t.} \sum_{a \in p} \mu_a \geq 1, \quad \forall p \in \mathcal{P}', \\
& \qquad \mu_a \geq 0, \quad \forall a \in A.
\end{aligned}
$$

- Reduced cost≤ 0 $\Leftrightarrow$ Dual feasible. So, if

$$
\sum_{a \in p} \mu_a \geq 1, \quad \forall p \in \mathcal{P},
$$

then the current solution is optimal to (MCF). Otherwise, we need to find a new column to improve (MCF').

- Checking whether the above inequality is satisfied for all paths is called pricing subproblem. This can be done efficiently by computing the shortest path w.r.t. the weight vector μ_a $(a \in A)$ for each commodity (s_i, t_i) $(i = 1, \dots, k)$. If the shortest paths are all at least 1, then we mush have

$$\sum_{a \in p} \mu_a \geq 1, \quad \forall p \in \mathcal{P},$$

  Otherwise, we find a new path p that has the positive reduced cost and we can add this path (column) to (MCF').
- Finding a shortest (s_i, t_i)-path in graph (V, A) with nonnegative weights $\mu_a \geq 0$ $(a \in A)$ is polynomial (e.g., by Dijkstra's algorithm).

- Column generation algorithm for MCF

  Repeat

  1. Solve (MCF') for $\mathcal{P}'$
  2. Let μ be the optimal multiplier vector of (MCF')
  3. **for** $i = 1, \ldots, k$, **do**
     - Find a shortest path $s_i - t_i$-path w.r.t. weight vector (μ_a)
     - If weight of the shortest path p is less than 1, add p to (MCF')
  4. **end for**

  Until no path has been added

- How to get an integer solution?

Dantzig-Wolfe decomposition for MCF

- We now show that the path formulation can be obtained from Dantzig-Wolfe decomposition. We first write the edge formulation in a simple form:

$$\begin{aligned}
&\max c^T x \\
&\text{s.t. } Nx = 0, \qquad \text{(flow conservation)} \\
&\qquad Ux \leq u, \qquad \text{(capacity constraints)} \\
&\qquad x \geq 0.
\end{aligned}$$

- Let

$$P = \{x \geq 0 \mid Nx = 0\}.$$

  This polyhedron is a cone and has a single vertex 0 and a finite number of rays $r^1, \ldots, r^s$. By the Minkowski-Weyl theorem, we have

$$P = cone\{r^1, \ldots, r^s\} = \{\sum_{i=1}^{s} \lambda_i r^i \mid \lambda_i \geq 0\}.$$

- Using the above expression of P, we can rewrite the edge formulation as

$$\begin{aligned} \max \sum_{i=1}^{s} \lambda_i (c^T r^i) \\ \text{s.t.} \ \sum_{i=1}^{s} \lambda_i (N r^i) \leq u, \\ \lambda_i \geq 0, \ i = 1, \ldots, s. \end{aligned}$$

- Analyzing the structure of c and U, the above problem can be reduced to

$$\begin{aligned} \max \sum_{i=1}^{s} y_i \\ \text{s.t.} \ \sum_{i:a \in p_i} y_i \leq u_a, \quad a \in A \\ y_i \geq 0, \ i = 1, \ldots, s, \end{aligned}$$

which is equivalent to the path formulation (refer to Numhauser and Wolsey (1988) for more details)

References

1. F. Vanderbeck, M. W. P. Savelsbergh, A generic view of Dantzig-Wolfe decomposition in mixed-integer programming, Vol. 34(3), 296-306, Operations Research Letters, 2006.
2. F. Vanderbeck, L. A. Wolsey, Reformulation and decomposition of integer programs, in 50 Years of Integer Programming 1958-2008, Part 2, pp. 431-502, 2010.
3. Cynthia Barnhart, Ellis L. Johnson, George L. Nemhauser, Martin W. P. Savelsbergh and Pamela H. Vance, Branch-and-Price: Column Generation for Solving Huge Integer Programs, Operations Research, Vol. 46(3), 316-329, 1998.
4. Marc Pfetsch, Multi-commodity flow and column generation, Lecture Notes, ZIB, 2006.