Mathematical Programming Lecture 21 OR 630 Fall 2005 November 08, 2005
Notes by Ilya Alexandrovich Sheynzon

Dantzig-Wolfe Decomposition (continued)

$$
(P) \quad \begin{array}{rcccccl}
\min & c_1^T x_1 & + & \dots & + & c_k^T x_k & \\
 & A_{01} x_1 & + & \dots & + & A_{0k} x_k & = b_0 \\
 & A_{11} x_1 & & & & & = b_1 \\
 & & & \dots & & & \\
 & & & & & A_{kk} x_k & = b_k \\
\end{array}
$$

$$x_1 \geq 0, \dots, x_k \geq 0.$$

$$
(MP) \quad \begin{array}{llcllll}
\min & \sum_{j=1}^k (\sum_h (c_j^T v_{jh}) \lambda_{jh} & + & \sum_i (c_j^T d_{ji}) \mu_{ji}) & & & \\
 & \sum_{j=1}^k (\sum_h (A_{0j} v_{jh}) \lambda_{jh} & + & \sum_i (A_{0j} d_{ji})) & = & b_0 & \\
 & \sum_h \lambda_{jh} & & & = & 1, & j = 1, 2, \dots, k \\
 & \lambda_{jh}, \mu_{ji} \geq 0, & & & & & \text{all } j, h, i.
\end{array}
$$

Conclusion: We can solve (MP) by the revised simplex method by solving at most k subproblems at each iteration to prove optimality or generate a new column for (MP) with negative reduced cost.

Each iteration requires

1)Pass $\overline{y}$ and $\overline{z}$ down to subproblems, and solve 1 to k LP subproblems (SP_j) to prove optimality or generate column.

2) Form a column $\begin{pmatrix} A_{0j} v_{jh} \\ e_j \end{pmatrix}$ or $\begin{pmatrix} A_{0j} d_{ji} \\ 0 \end{pmatrix}$ to enter into the basis of (MP) and perform a pivot.

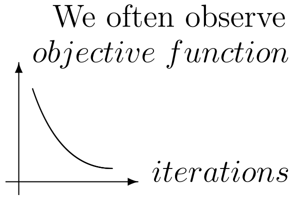
Pros and cons:

Pros: We're dealing with problem (MP) with fewer rows than (P) and with small subproblems. So we "should" save on arithmetic operations per iteration, and on storage.

Cons: We may need lots of iterations in the (SP_j)'s. We may need lots of iterations in (MP).

Typically, folklore claims that the number of iterations for simplex method to solve a problem with m equations in n unknowns is about $2m$ to $3m$ (For phase I and phase II). But this only holds for "reasonably" square problems, say $n \leq 10m$ (which is not true for (MP)).

We often observe long tails in convergence:



It may be worth terminating before the optimal solution is found, if we're within a guaranteed amount of optimal value.

Computational complexity of the simplex method and of linear programming

We start with a more informal analysis of the simplex method then give a more formal treatment of LP in general (hence, get new algorithms). Each iteration of the simplex method requires $O(mn)$ arithmetic operations $(+, -, *, \div$, comparison), so the question is: is the number of iterations required *polynomial* in m and n (polynomial-good, exponential-bad) (Cobham, Edmonds('60s); von Neumann('53))?

So far, our bounds on the number of iterations are by the number of basic feasible solutions, and this can be exponential. E.g., $\{y \in \mathcal{R}^m : 0 \leq y \leq e\}$ has 2^m vertices, but only $2m = n$ inequalities in m variables. Could the/a simplex method visit all of them?

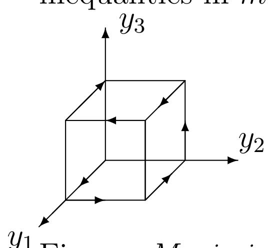


Figure: *Maximize* y_3*. Objective function non-decreasing path of* $2^3 = 8$ *vertices,* $m = 3$

Consider

$$
(D) \qquad \begin{array}{ll} \max & y_m \\ & \varepsilon \leq y_1 \leq 1 - \varepsilon \\ & \varepsilon y_{i-1} \leq y_i \leq 1 - \varepsilon y_{i-1}, \quad i = 2, .., m, \end{array}
$$

where $0 < \varepsilon < 1/2$. **Theorem**
(a) The feasible region of (D) has 2^m vertices.
(b) These vertices can be ordered $v^1, v^2, .., v^{2^m}$ so that $v_m^{k-1} < v_m^k, k = 2, .., 2^m$, and $[v^{k-1}, v^k]$ is an *edge* of this polytope for each j.
$[v, w] = \{(1-\lambda)v + \lambda w : 0 \leq \lambda \leq 1\}$ is an *edge* of polyhedron $Q \subseteq \mathcal{R}^m$ if $v \neq w$ and there is some $b \in \mathcal{R}^m$ with $\arg\max\left\{b^T y : y \in Q\right\} = [v, w] \Leftrightarrow v$ and w are vertices of Q and share $m-1$ linearly independent tight inequalities.
Proof:
(a) First, we construct 2^m vertices: for each vertex u of the unit cube $[0,1]^n$, let

$$
v_i = \begin{cases} \varepsilon v_{i-1}, & \text{if} \quad u_i = 0 \\ 1 - \varepsilon v_{i-1}, & \text{if} \quad u_i = 1, \text{ for } i = 1, 2, .., m \quad (v_0 \equiv 1). \end{cases}
$$

Then either $v_i \in (0, \varepsilon]$ or in $[1-\varepsilon, 1)$ (by induction), so v is feasible ($\varepsilon v_{i-1} < 1 - \varepsilon v_{i-1}$ since $\varepsilon < 1/2$) and all such v's are distinct (again since $\varepsilon < 1/2$), and all are vertices (satisfy m linearly independent constraints with equality, whose coefficients are columns of the matrix below):

$$\begin{pmatrix} 1 & \pm\varepsilon & 0 & ... & ... & 0 \\ 0 & 1 & \pm\varepsilon & ... & ... & 0 \\ 0 & 0 & 1 & ... & ... & 0 \\ ... & ... & ... & ... & ... & 0 \\ 0 & 0 & 0 & ... & 1 & \pm\varepsilon \\ 0 & 0 & 0 & ... & 0 & 1 \end{pmatrix}$$

These are in fact *all* the vertices. No vertex v can satisfy $\varepsilon v_{i-1} = v_i = 1 - \varepsilon v_{i-1}$, for then $v_{i-1} = \frac{1}{2\varepsilon} > 1$. So we can only choose one of each pair of inequalities $y_i \geq \varepsilon y_{i-1}, y_i \leq 1 - \varepsilon y_{i-1}$ to be tight. So each vertex chooses exactly one of each pair, so is a v constructed as above. $\square$

(b) By induction on m.

Base case: $m = 1$. Vertices ε and $1 - \varepsilon$.

Assume true for less than m, and consider case m. Look at the sequence of vertices of the feasible region of (D) for dimension $m-1$, say $u^1, u^2, ..., u^{2^{m-1}}$. Let $v^j := (u^j; \varepsilon u^j_{m-1})$ for $1 \leq j \leq 2^{m-1}$ and let $v^{2^m - j+1} = (u^j; 1 - \varepsilon u^j_{m-1})$ for $1 \leq j \leq 2^{m-1}$.

These are all vertices of the feasible region of (D) by the construction in (a).

For $2 \leq j \leq 2^{m-1}$, we have $v^j_m = \varepsilon u^j_{m-1} > \varepsilon u^{j-1}_{m-1} = v^{j-1}_m$ and $[v^{j-1}_m; v^j_m]$ is an edge of the polytope since the points on the edge satisfy enough equalities, since u^{j-1} and u^j both satisfy $m-2$ of the first $2(m-1)$ inequalities tightly (and so do v^{j-1} and v^j), but they also satisfy $y_m = \varepsilon y_{m-1}$, i.e., a total of $m-1$ linearly independent inequalities.

Similarly, $v^{2^m - j+2} = v^{2^m-(j-1)+1} = 1 - \varepsilon u^{j-1}_{m-1} > 1 - \varepsilon u^j_{m-1} = v^{2^m - j+1}_m$ for $2 \leq j \leq 2^{m-1}$, and again, $[v^{2^m - j+1}, v^{2^m - j+2}]$ is an edge of the polytope.

Finally, $v^{2^{m-1}} = (u^{2^{m-1}}; \varepsilon u^{2^{m-1}}_{m-1})$ and $v^{2^{m-1}+1} = v^{2^m - 2^{m-1}+1} = (u^{2^{m-1}}; 1 - \varepsilon u^{2^{m-1}}_{m-1})$.

$\varepsilon u^{2^{m-1}}_{m-1} < 1 - \varepsilon u^{2^{m-1}}_{m-1}$ $(\varepsilon < 1/2)$ and both vertices satisfy the same $m-1$ linearly independent inequalities tightly (the same as $u^{2^{m-1}}$). $\square$

If we add slack variables in the natural order:

$$y_1 - s_1 = \varepsilon, y_1 + s_2 = 1 - \varepsilon, y_2 - \varepsilon y_1 - s_3 = 0, ...,$$

and use Bland's rule, we generate exactly this sequence of vertices.

Indeed, we can construct a problem that takes $2^m - 1$ steps for Dantzig's most negative reduced cost rule:

$$\begin{array}{lll} \max & 2^{m-1}y_1 + ... + 2y_{m-1} + y_m & \\ & 1y_1 & \leq 5 \\ & 4y_1 + y_2 & \leq 5^2 \\ & ... & ... \\ & 2^m y_1 + 2^{m-1}y_2 + ... + 4y_{m-1} + y_m & \leq 5^m \\ & y \geq 0. & \end{array}$$