Computational Intractability 2010/5/6, revised 2011/5/26

Lecture 4

Professor: David Avis *Scribe:Bingbing Zhuang, DA*

Recall the standard form:

$$
\begin{aligned}
& max \sum_{j=1}^{n} c_j x_j \\
& \sum_{j=1}^{n} a_{ij} x_j \leq b_i, \quad i = 1, 2, \ldots, m \\
& x_j \geq 0 \quad j = 1, 2, \ldots, n
\end{aligned}
$$

$x_1, x_2, \ldots, x_n$ are *decision variables.* $x_{n+1}, x_{n+2}, \ldots, x_{n+m}$ are *slack variables.*
Assuming $b_i > 0$ for all i in (1) the origin is feasible.
In the dictionary,

$$
\begin{aligned}
x_{n+i} &= b_i - \sum_{j=1}^{n} a_{ij} x_j, i = 1, 2, \ldots, m \\
z &= \sum_{j=1}^{n} c_j x_j
\end{aligned}
$$

The LHS variables are *basic variables* with index set $B = \{n+1, n+2, \ldots, n+m\}$. The RHS variables are *non-basic variables* with index set $N = \{1, 2, \ldots, n\}$. A *pivot* switches a LHS variable with a RHS variable and updates the index sets B and N.

General dictionary ($|B| = m, |N| = n$)

$$
\begin{aligned}
x_i &= \overline{b}_i - \sum_{j \in N} \overline{a}_{ij} x_j, \quad i \in B \\
z &= \overline{z} + \sum_{j \in N} \overline{c}_j x_j
\end{aligned}
$$

Basic solution: set $x_j = 0, j \in N, x_i = \overline{b}_i,\ i \in B,\ z = \overline{z}$.
Basic feasible solution: All $x_i \geq 0$, ie. $\overline{b}_i \geq 0$ for $i \in B$.

1 Simplex Method

(Assuming that all $b_i > 0$)

1. Find $\overline{c}_j > 0$, $j \in N$. If none, terminate with optimum solution.
2. Choose i to minimize ratio $\overline{b}_i / \overline{a}_{ij}$ for $\overline{a}_{ij} > 0$. If none, terminate with unbounded solution.

3. Solve for basic variables $B = B\backslash\{i\} \cup \{j\}$ and get a new dictionary.

Outcomes for a general LP:

1. Optimum solution.
2. Unbounded.
3. No feasible solution. (Can happen only if some starting $b_i < 0$.)

1.1 Problem transformation to solve a problem with infeasible origin

The basic idea is to create a new LP' with feasible origin and compute an optimal solution. Depending on the solution the original problem is either feasible, so we contine and optimize it, or it is infeasible, so we terminate.

Steps:

1. Given a general LP with unfeasible origin, $b_i < 0$ for some i, transform it to LP' with feasible origin which is bounded.
2. Solve LP' to optimality.
3. Depending on the solution, either LP is infeasible or we get a feasible dictionary for LP with basis B.
4. Solve LP to get either an optimum or unbounded solution.

Steps 1 and 2 are often called *Phase 1* and steps 3 and 4 *Phase 2*.
Example:

$$
\begin{aligned}
maxz &= 5x_1 + 6x_2 + 9x_3 \\
& x_1 - 2x_2 - 3x_3 \le -5 \\
& -x_1 + x_2 + 2x_3 \le -3 \\
& x_1, x_2, x_3 \ge 0
\end{aligned}
$$

To get LP', add a new artificial variable $x_0 \ge 0$.

$$
\begin{aligned}
min\ w = x_0 \Leftrightarrow max\ \ - w = -x_0 \\
x_1 - 2x_2 - 3x_3 - x_0 \le -5 \\
-x_1 + x_2 + 2x_3 - x_0 \le -3
\end{aligned} \tag{1}
$$

$x_1 = x_2 = x_3 = 0, x_0 = 5$ is a feasible solution for LP'.
LP' is bounded since $w = x_0 \ge 0$.
We now get the optimum solution w^*. There are two cases:

1. $w^* = 0$: In this case the $x_1, x_2, \ldots, x_n$ basic solution to LP' is feasible for LP. Add back original objective z and continoue.

2. $w^* > 0$: In this case the original problem is infeasible. (Proof: Suppose LP is feasible. Take the feasible solution and set $x_0 = 0$. This solves LP' with $w = 0$, contradiction.)

The initial Dictionary for LP' is

$$\begin{aligned} x_4 &= -5 - x_1 + 2x_2 + 3x_3 + x_0 \\ x_5 &= -3 + x_1 - x_2 - 2x_3 + x_0 \\ -w &= -x_0 \end{aligned}$$

We now begin Phase 1. Pivot x_0 into B and pivot out most negative x_4.

$$\begin{aligned} x_0 &= 5 + x_1 - 2x_2 - 3x_3 + x_4 \\ x_5 &= 2 + 2x_1 - 3x_2 - 5x_3 + x_4 \\ -w &= -5 - x_1 + 2x_2 + 3x_3 - x_4 \end{aligned}$$

Now we use the standard simplex method to obtain the optimum dictionary:

$$\begin{aligned} x_1 &= 11 + x_3 + x_4 + 2x_5 - 3x_0 \\ x_2 &= 8 - x_3 + x_4 + x_5 - 2x_0 \\ -w &= 0 - x_0 \end{aligned}$$

Exercise. Do these pivots yourself and check the above.

This is the end of Phase 1. Since $w = 0$ we have a feasible solution to the original problem: $x_1 = 11, x_2 = 8, x_3 = 0$. This is the end of Phase 1. We now delete the artificial variable x_0 and reintroduce the original objective function z. We need to change variables in order to express z in terms of the non-basic variables x_3, x_4, x_5 by substituting for x_1 and x_2 from the previous dictionary.

$$z = 5x_1 + 6x_2 + 9x_3 = 103 + 8x_3 + 11x_4 + 16x_5$$

We can now begin Phase 2 with the dictionary

$$\begin{aligned} x_1 &= 11 + x_3 + x_4 + 2x_5 \\ x_2 &= 8 - x_3 + x_4 + x_5 \\ z &= 103 + 8x_3 + 11x_4 + 16x_5 \end{aligned} \tag{2}$$

Now if we choose x_5 as the entering variable we see that the problem is unbounded: for any $t \geq 0$ the solution $x_1 = 11 + 2t, x_2 = 8 + t, x_3 = 0$ is feasible with $z = 103 + 16t$. This is the *certificate of unboundedness.*

In conclusion we found a feasible solution to LP, which had infeasible origin, by finding an optimum solution to LP' which had a feasible origin. In fact, finding a feasible solution to an LP is, in general, as hard computationally as finding an optimum solution. For NP-hard problems such as the travelling salesman problem on a complete graph, the problems are quite different: any permutation gives a feasible solution, but it is NP-hard to get an optimum solution.

1.2 Cycling and Degneracy

Does the simplex method really terminate? Actually it can loop indefinitely, called cycling. Recall the basic idea of local improvement algorithms:

1. Given a starting feasible solution, improve it by increasing z but keep it staying feasible.

2. Each iteration gives us a dictionary with basis B.

$$|B| = m, |N| = n, B \cup N = \{1, 2, \ldots, n+m\}$$

   The number of different bases or dictionaries is $\binom{m+n}{n}$, which is finite. But can a dictionary be repeated later? The answer is yes if z does not increase. This happens if the ratio test for entering variable has minimum ratio zero, and such problems are called *degenerate*. The following example shows the problems that a degenerate problem may cause.

   Komei Fukuda's example
   Starting dictionary D_0 has basis $B = \{4, 5, 6\}, N = \{1, 2, 3\}$.

$$\begin{aligned} x_4 &= -2x_1 + x_2 - x_3 \\ x_5 &= -3x_1 - x_2 - x_3 \\ x_6 &= 5x_1 - 3x_2 + 2x_3 \\ z &= x_1 - 2x_2 + x_3 \end{aligned}$$

We select x_1 as entering and x_5 as leaving variable, and obtain D_1 with $B = \{4, 1, 6\}, N = \{5, 2, 3\}$.

$$\begin{aligned} x_4 &= \frac{2x_5}{3} + \frac{5x_2}{3} - \frac{x_3}{3} \\ x_1 &= -\frac{x_5}{3} - \frac{x_2}{3} - \frac{x_3}{3} \\ x_6 &= -\frac{5x_5}{3} - \frac{14x_2}{3} + \frac{x_3}{3} \\ z &= -\frac{x_5}{3} - \frac{7x_2}{3} + \frac{2x_3}{3} \end{aligned}$$

We select x_3 as entering and x_4 as leaving variable, and obtain D_2 with $B = \{3, 1, 6\}, N = \{5, 2, 4\}$.

$$\begin{aligned} x_3 &= -2x_5 + x_2 - x_4 \\ x_1 &= -3x_5 - x_2 - x_4 \\ x_6 &= 5x_5 - 3x_2 + 2x_4 \\ z &= x_5 - 2x_2 + x_4 \end{aligned}$$

Now D_2 has exactly the same coefficients as D_0 so we can pivot in the same location: column 1 and row 2. We will then get the identical dictionary coefficients as D_1, so we can pivot in the same place as for D_1. Since the number of dictionaries is finite, we have proved that a basis must repeat, and we have an infinite cycle. In fact it does not take long for this to happen. We select column 1 and row 2, ie. x_5 to enter and x_1 to leave. We know the new dictionary D_3 will have exactly D_1's coefficients:

$$\begin{aligned} x_3 &= \frac{2x_1}{3} + \frac{5x_2}{3} - \frac{x_4}{3} \\ x_5 &= -\frac{x_1}{3} - \frac{x_2}{3} - \frac{x_4}{3} \\ x_6 &= -\frac{5x_1}{3} - \frac{14x_2}{3} + \frac{x_4}{3} \\ z &= -\frac{x_1}{3} - \frac{7x_2}{3} + \frac{2x_4}{3} \end{aligned}$$

Now pivoting on column 3 and row 1 brings us back to $B = \{4, 5, 6\}$.

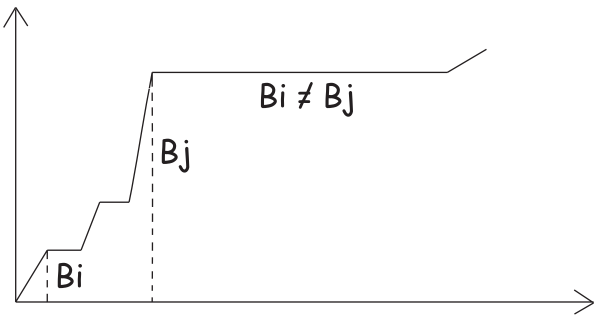

Figure 1: Stalling

The problem this example uncovers is the following:

1. We did a ratio test and get a minimum value of zero, so z did not increase (stalling, see Figure 1).

2. This means that many different dictionaries correspond to the same basic feasible solution.

Geometry is useful to understand what is happening. Consider the polytope (bounded polyhedron) which is a pyramid P with six faces. See Figure 2.

Each vertex in P can be represented by giving three faces that contain it. If each vertex is contained on exactly three faces the polytope is called *simple*, otherwise it is called *non-simple*. If a vertex lies on a face, it means that in the LP the corresponding slack variable for the constraint defining the face is zero. In other words, the non-basic variables identify three faces containing the given vertex. Simple polytopes correspond to non-degenerate LPs, and non-simple polytopes to degenerate LPs. So in the example, the origin is the intersection $F_1 \cap F_6 \cap F_5$. Note however that vertex v is on the insection of $F_1 \cap F_2 \cap F_3 \cap F_4 \cap F_5$. Therefore it does not have a unique representaion as the intersection of three faces: it can be represented by any three of the five, say $\{F_1, F_2, F_5\}$.

Now consider an objective function maximizing $z = x_2$. A pivot of the simplex method involves removing one face containing the current vertex (ie. current basic feasible solution) and replacing it with another face, to improve z if possible. A possible pivot from the

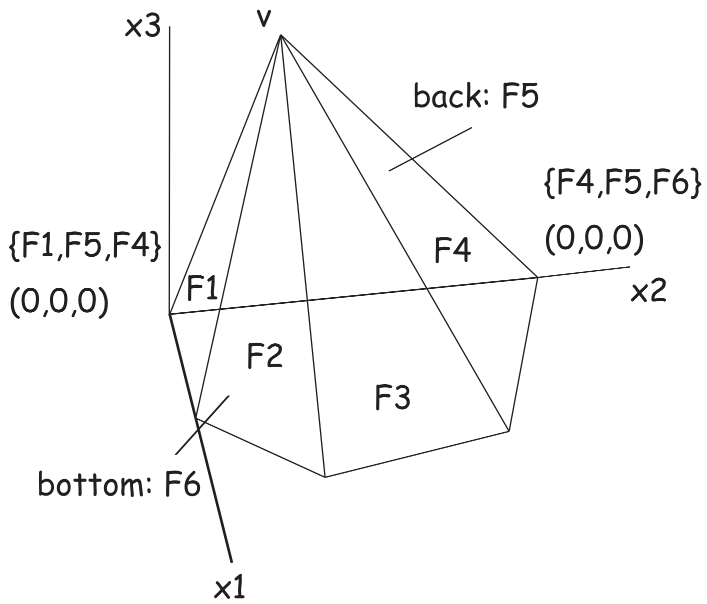

Figure 2: A non-simple polytope

origin replaces F_6 with F_2, increasing z. We get $\{F_1, F_2, F_5\}$. Now we may stall, ie. remain on vertex v, making pivots to $\{F_1, F_3, F_5\}$ and then to $\{F_1, F_4, F_5\}$. Now if we pivot to $\{F_1, F_2, F_5\}$ we are in a cycle. But if we pivot to $\{F_6, F_4, F_5\}$ we break the cycle and move to the optimum vertex at the right of the figure (which should be labelled something like (0,2,0)).

1.3 Handling Degenerate LPs

There are several ways to great versions of the simplex method that do not cycle. A simple, "engineering" method is to make all LPs non-degenerate. This can be done for the example in Figure 2 by perturbing a little bit each face so that no four faces intersect at a vertex. See Figure 3. In general a perturbation can be applied to a problem with n variables so that no $n + 1$ facets intersect.

A second method involves choosing a pivot selection rule that does not cycle. The simplest is the smallest subscript rule due to R. Bland:

1. For the entering variable, choose the smallest index j for which $\bar{c}_j$ is positive.

2. If the ratio test gives a tie for leaving variable, again choose the one with smallest subscript.

Exercise: Apply this rule to Fukuda's example and see that it does not cycle.

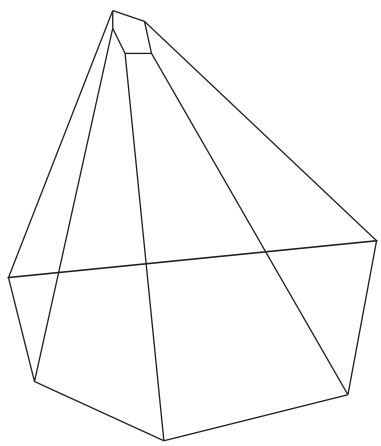

Figure 3: Perturbed pyramid is simple