Polyhedral Computation	2010/10/15

Lecture 3

Professor: David Avis *Scribe:Yosuke Suzuki*

We consider the problem of getting an initial feasible dictionary for the graph search method for vertex enumeration discussed in the previous lecture. Recall the setup:
Input: $m \times n$ matrix A, m-vector b
Output: List of vertices of the polytope $P = \{x \in R^n, b + Ax \geq 0\}$

By definition $v \in P$ is a vertex if it is the unique solution of an $n \times n$ subsystem of inequalities from P solved as equations. To be non-trivial, this problem implies $m \geq n$ and that the column rank of A must be n or there is no $n \times n$ subsystem with unique solution. However this is not sufficient, and there may in fact be no feasible dictionaries. We discuss how to find a starting feasible dictionary if there is one or give a proof if there is none. There are three steps.

Step 1: Introduce slacks obtaining the dictionary D_0:

$$x_{n+i} = b_i + \sum_{j=1}^{n} a_{ij} x_j$$
$$i = 1, 2, ...m$$

(We require $x_{n+i} \geq 0$ for a feasible solution.)

Step 2: Solve for the decision variables $x_1...x_n$ on the LHS by letting some subset of n slacks move to the RHS, getting D, with basis $B_1 = \{\text{indices of LHS variables}\}$ and cobasis $N_1 = \{\text{indices of RHS variables}\}$
(Note $\{1, 2, ...n\} \subseteq B$)

Step 3: If the dictionary obtained is infeasible, we use these steps to try to find a feasible solution:

Step 3-1: Choose the smallest index i of a slack variable x_i for which $x_i = b_i < 0$. If none, we have a feasible dictionary.

Step 3-2: Choose smallest index j s.t. the coefficient of x_j is positive in this row. If none, P is empty and has no vertices. Pivot and return to Step 3-1.

The following is an example with $n = 2$,$m = 4$ which requires Step 3. It is illustrated in Figure 1.

Suppose that after Step 2, $n = 2$,$m = 4$, we have $B_1 = \{1, 2, 5, 6\}, N_1 = \{3, 4\}$ and dictionary D_1.

$$\begin{aligned}
x_1 &= 1 + x_3 + x_4 \\
x_2 &= -1 + x_4 \\
x_5 &= -1 + x_3 - x_4 \\
x_6 &= -4 + 2x_3 + x_4
\end{aligned}$$

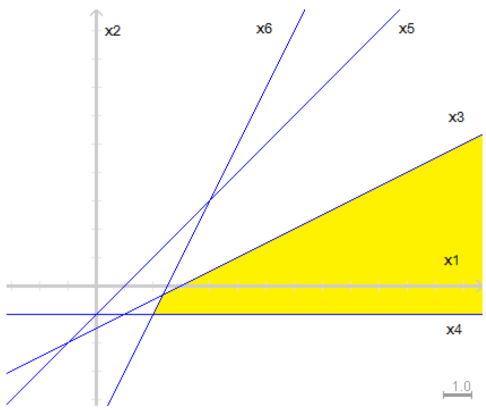

Figure 1: First example

We get a basic solution by setting $x_3 = x_4 = 0$, with $x_1 = 1, x_2 = -1, x_5 = -1, x_6 = -4$. Both x_5 and x_6 are infeasible slacks.

In this example, Step 3-1 chooses infeasible slack x_5, and Step 3-2 chooses x_3. Letting x_3 replace x_5 gives basis $B_2 = \{1, 2, 3, 6\}$, $N_2 = \{4, 5\}$ and gives the dictionary D_2:

$$\begin{aligned} x_1 &= 2 + 2x_4 + x_5 \\ x_2 &= -1 + x_4 \\ x_3 &= 1 + x_4 + x_5 \\ x_6 &= -2 + 3x_4 + 2x_5 \end{aligned}$$

This is infeasible as $x_6 = -2$ in the basic solution. Applying Step 3-1 we select x_6. In Step 3-2 the variable x_4 is chosen since x_5 has larger index. This gives dictionary D_3 with $B_3 = \{1, 2, 3, 3\}$ and $N = \{5, 6\}$. We give only the b-vector since it is a feasible dictionary.

$$\begin{aligned} x_1 &= \frac{10}{3} + \ldots \\ x_2 &= -\frac{1}{3} + \ldots \\ x_3 &= \frac{5}{3} + \ldots \\ x_4 &= \frac{2}{3} + \ldots \end{aligned}$$

This is feasible as slack variables are all non-negative. The corresponding vertex of P is $x_1 == \frac{10}{3}, x_2 = -\frac{1}{3}$.

Given this feasible dictionary we can use the graph search method described in the last lecture.

Let us consider the case where we terminate in Step 3-2. Here is an example:

$$\begin{aligned} x_1 &= 2 + 2x_4 + x_5 \\ x_2 &= -1 + x_4 \\ x_3 &= 1 + x_4 + x_5 \\ x_6 &= -2 - 3x_4 - 2x_5 \end{aligned}$$

Consider the last row of this dictionary. Whatever nonnegative value x_4 and x_5 take, x_6 must be negative. Therefore there can be no feasible solution and P is empty.

We have seen that if we terminate in Step 3-1 then we have a feasible dictionary with basic solution corresponding to a vertex of P. If we terminate in Step 3-2 we have a proof of infeasibility, since the infeasible equation in the dictionary is obtained by standard pivot operations on the original dictionary. It remains to prove that we always terminate in a finite number of steps.

1 Proof of termination

Suppose there exist A, b for which Step 3 loops forever. This means some basis repeats itself, a situation we call a cycle. Choose the smallest value of $m + n$ for which we have a cycle. Note that x_{n+m} must enter and leave the basis in this cycle, otherwise x_{n+m} could be deleted and we still have a cycle with smaller value of $m + n$.

- x_{n+m} enters the basis
  In this case the coefficients in the pivot row, say,

  $$x_i = b_i + \sum_{j \in N} a_{i,j}$$

  must have the sign pattern

  $$b_i < 0, \quad a_{i,n+m} > 0, \quad a_{i,j} < 0 \;\; j \in N \setminus \{n+m\}.$$

  If follows from this that whenever $x_j \geq 0, n+1 \leq j \leq n+m-1$ we must have $x_{n+m} > 0$. (In Step 3 N never contains the original decision variables $x_1, ..., x_n$.)

- x_{n+m} leaves the basis
  Consider the basic solution for this dictionary. Since x_{n+m} is chosen to leave the basis in Step 3-1 we must have that each basic slack variable is nonnegative except for $x_{n+m} < 0$. Slack co-basic variables have value zero.

Now we need only observe that the basic solution obtained in the second item above violates the condition given in the first item. This contradiction completes the proof.

2 The Adjacency Oracle

Given a feasible dictionary, want an adjacency oracle to get all adjacent feasible dictionaries obtainable by a single pivot. For each pivot, in principle we can choose any $i \in B, j \in$

$N(i \geq n+1)$. There are $n(m-n)$ such choices. However many may be rejected due to infeasibility. We define the adjacency oracle $Adj(N,i,j)$ as follows:

$$Adj(N,i,j) = \begin{cases} \emptyset & N \cup \{i\}/\{j\} \text{ gives an infeasible dictionary} \\ \emptyset & x_j \text{ has zero coefficient in row } x_i \\ N \cup \{i\}/\{j\} & \text{if this gives a feasible dictionary} \end{cases}$$

Consider the example, which is illustrated in Figure 2 :

$$\begin{aligned} 1 + x_1 &= x_3 \\ 2 - x_1 &= x_4 \\ 1 + x_2 &= x_5 \\ 2 - x_2 &= x_6 \\ 3 + x_1 + x_2 &= x_7 \end{aligned}$$

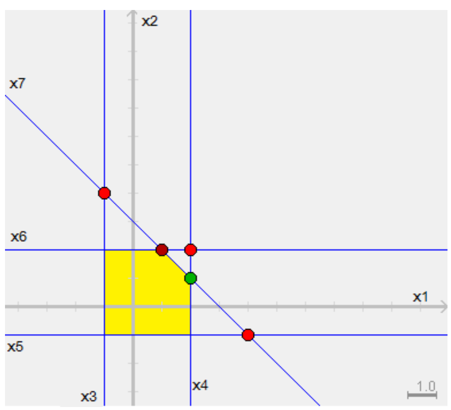

Figure 2: Second example

Consider the dictionary with $N = \{4, 7\}$.

$$\begin{aligned} x_1 &= 2 - x_4 \\ x_2 &= 1 + x_4 - x_7 \\ x_3 &= 3 - x_4 \\ x_5 &= 2 + x_4 - x_7 \\ x_6 &= 1 - x_4 + x_7 \end{aligned}$$

$$
\begin{aligned}
Adj(3,4) &= \emptyset \text{ (infeasible point (-1,4))} \\
Adj(3,7) &= \emptyset \text{ (no solution)} \\
Adj(5,4) &= \emptyset \text{ (infeasible point (4,-1))} \\
Adj(5,7) &= N = \{4,5\} \text{ (feasible vertex (2,-1))} \\
Adj(6,4) &= N = \{6,7\} \text{ (feasible vertex (1,2))} \\
Adj(6,7) &= \emptyset \text{ (infeasible point (2,2))}
\end{aligned}
$$

A simple (ie. non-degenerate) polyhedron is one for which each vertex lies on exactly n facets. In this case there are only n feasible adjacent dictionaries, one for each giving one new vertex.

3 More on the DFS algorithm

3.1 More on the DFS Algorithm

Data structure needed are stack S and list of vertices found L.

$DFS(V)$:

- Add v to L and S.
- Find an adjacent vertex to $top(S)$ not in L and add to L and S (bookkeeping).
- If none, remove $top(S)$.

If the graph is huge, book keeping dominates. Output size upperbound is $m^{\lfloor \frac{n}{2} \rfloor}$ vertices. There can be as many as $\binom{m}{n}$ feasible dictionaries for highly degenerate problems. In the next lecture we explain reverse search, which is a method of eliminating these two data structures.