Optimization and Approximation	ENS Lyon
Lecture 3	
Lecturer: Alantha Newman	*September 27, 2016*

1 A Closer Look at Basic Feasible Solutions

Recall the definition of a basic feasible solution:

Definition 1. *Let P be a polyhedron defined by linear equality and inequality constraints, and consider $\mathbf{x}^* \in \mathbb{R}^n$.*

1. *The vector $\mathbf{x}^*$ is a a* basic solution *if:*
   (a) *All equality constraints are tight;*
   (b) *Among the constraints that are tight at $\mathbf{x}^*$, n of them are linearly independent.*
2. *If $\mathbf{x}^* \in P$ is a basic solution, we say that it is a* basic feasible solution.

What exactly do we mean here by *linearly independent*? Given an $m \times n$ matrix $\mathbf{A}$, let $\mathbf{a}_i$ denote the i^{th} row of $\mathbf{A}$. Then we say that the constraints $\{\mathbf{a}_i\}$ for $i \in B$ are linearly independent if the equations corresponding to the tight constraints are linearly independent.

Definition 2. *A basic feasible solution is* non-degenerate *if there are* exactly *n tight constraints.*

Definition 3. *A basic feasible solution is* degenerate *if there are* more than *n tight constraints.*

We say that a linear programming problem is degenerate if it contains degenerate vertices or basic feasible solutions. It is NP-complete to determine if a given linear program is degenerate [CKM82].

Theorem 4. *Let $P = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq 0\}$. Let $\mathbf{x}^* \in P$ be a basic solution. The following statements are equivalent:*

1. *$\mathbf{x}^*$ is tight for n linearly independent constraints (all m constraints in $\mathbf{A}\mathbf{x} = \mathbf{b}$ and at least $n - m$ tight non-negativity constraints);*
2. *There exist m indices $B = \{B(1), B(2), \ldots, B(m)\}$ such that the columns $\mathbf{A}_{B(1)}, \mathbf{A}_{B(2)}, \ldots, \mathbf{A}_{B(m)}$ are linearly independent.*

1.1 Procedure for Constructing a Basic Solution

To construct a basic solution $\mathbf{x}^*$, we can do the following:

1. Choose m linearly independent columns $\mathbf{A}_{B(1)}, \mathbf{A}_{B(2)}, \ldots, \mathbf{A}_{B(3)}$.
2. Let $\mathbf{x}^*_i = 0$ for all $i \notin B = \{B(1), B(2), \ldots, B(m)\}$.
3. Solve the system of m equations $\mathbf{A}\mathbf{x} = \mathbf{b}$ for the unknowns $\mathbf{x}^*_{B(1)}, \mathbf{x}^*_{B(2)}, \ldots, \mathbf{x}^*_{B(m)}$.

If a basic solution constructed according to this procedure is non-negative, then it is a basic feasible solution. Conversely, every basic (feasible) solution can be constructed using this procedure. If $\mathbf{x}^*$ is a basic solution, then the variables $\{\mathbf{x}^*_{B(1)}, \mathbf{x}^*_{B(2)}, \ldots, \mathbf{x}^*_{B(m)}\}$ are called *basic variables*. The remaining variables are called *non-basic variables*. If $\mathbf{x}^*$ is non-degenerate, then all basic variables are non-zero. In a basic solution, at least n constraints are tight. Therefore, a basis corresponds to a unique solution. However, several bases can lead to the same solution (degeneracy).

1.2 Adjacent Basic Feasible Solutions

As seen in the last lecture, each iteration of the simplex algorithm corresponds to a dictionary, which corresponds to a basis and therefore a unique basic (feasible) solution. In each dictionary, the n tight constraints correspond to n hyperplanes. The solution associated with this dictionary is the point that is the intersection of these n hyperplanes. A legal pivot always moves from one basic feasible solution to an *adjacent* basic feasible solution.

Definition 5. *Let* $P = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{Ax} = \mathbf{b},\ \mathbf{x} \geq 0\}$. *Suppose* $\mathbf{x}_1, \mathbf{x}_2 \in P$ *are basic feasible solutions. Then* $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *are* adjacent *if they are each tight for some set of* $n-1$ *linearly independent constraints.*

Two adjacent basic feasible solutions share $n-1$ indices in their bases.

2 Finding an Initial Basic Feasible Solution

Last time, when we introduced the simplex algorithm, we used the following procedure to find an initial basic feasible solution. Given a linear program in canonical form:

$$\begin{aligned} \max\ & \mathbf{c}'\mathbf{x} \\ \mathbf{Ax}\ & \leq \mathbf{b} \\ \mathbf{x}\ & \geq 0, \end{aligned} \tag{Q}$$

we translated it to standard form and set all original variables to zero and set the j^{th} slack variable to b_j. Specifically, we did the following:

$$\begin{aligned} x_{n+j} &= b_j - \sum_{i=1}^{n} a_{ji} x_i, \quad j : 1 \leq j \leq m, \\ x_i &= 0, \quad i : 1 \leq i \leq n. \end{aligned}$$

If the b_j values are all non-negative, then this initial solution is both basic and feasible. But what is some of the b_j's are negative? Then we must find an initial basic feasible solution. The procedure to do this is usually called *Phase I* of the simplex algorithm. Given an intial feasible solution, the process of finding an optimal solution is called *Phase II.*

2.1 An Auxiliary Program

Let us consider the following example:

$$\begin{aligned} \max\ & 2x_1 + x_2 \\ \text{subject to:}\ & -2x_1 + x_2 \leq -2 \\ & x_1 - 2x_2 \leq -2 \\ & x_1 + x_2 \leq 7 \\ & x_1, x_2 \geq 0. \end{aligned} \tag{Q_1}$$

If we follow the standard initialization procedure, we obtain the following basic solution:

$$\{x_1, x_2, x_3, x_4, x_5\} = \{0, 0, -2, -2, 7\}.$$

This solution is infeasible. Moreover, it could even be the case that a given linear program is infeasible. Then it would not be possible to find an initial basic feasible solution. How do we determine whether or not this is the case? We will construct an *auxiliary* linear program whose optimal objective value is zero if and only if the original linear program. In the case of (Q_1), we obtain the following linear program (Q_1').

$$\begin{aligned} & \min \;\; x_0 \; (= \max -x_0) \\ \text{subject to:} \;\; & -2x_1 + x_2 \le -2 + x_0 \\ & x_1 - 2x_2 \le -2 + x_0 \\ & x_1 + x_2 \le 7 + x_0 \\ & x_0, x_1, x_2 \ge 0. \end{aligned} \tag{Q_1'}$$

Now (Q_1') is clearly an unbounded non-empty polyhedron. To see this, just choose x_0 to be sufficiently large. Moreover, the optimal value of (Q_1') is zero if and only if the domain of (Q_1) is empty. Let us now try to solve (Q_1'). Here is the initial dictionary:

$$\begin{aligned} x_3 &= -2 + 2x_1 - x_2 + x_0 \\ x_4 &= -2 - x_1 + 2x_2 + x_0 \\ x_5 &= +7 - x_1 - x_2 + x_0 \\ z &= \qquad\qquad\qquad - x_0. \end{aligned} \tag{D_0'}$$

This dictionary is *infeasible*. What can we do? Take one of the most negative (i.e. most infeasible) variables (in this case, x_3 or x_4) and swap with x_0 (x_0 enters basis, x_3 leaves). (This is an “illegal” pivot since it decreases the objective value. However, the resulting solution remains basic.) *This causes all variables to be non-negative, because it increases the values of all variables.*

$$\begin{aligned} x_0 &= +2 - 2x_1 + x_2 + x_3 \\ x_4 &= +0 - 3x_1 + 3x_2 + x_3 \\ x_5 &= +9 - 3x_1 \qquad + x_3 \\ z &= -2 + 2x_1 - x_2 - x_3. \end{aligned} \tag{D_1'}$$

Now since we have a feasible solution, we proceed with simplex as normal. Let x_1 enter and x_4 leave.

$$\begin{aligned} x_0 &= +2 + \frac{2x_4}{3} - x_2 + \frac{x_3}{3} \\ x_1 &= +0 - \frac{x_4}{3} + x_2 + \frac{x_3}{3} \\ x_5 &= +9 + x_4 - 3x_2 \\ z &= -2 - \frac{2x_4}{3} + x_2 - \frac{x_3}{3}. \end{aligned} \tag{D_2'}$$

Let x_2 enter and x_0 leave.

$$\begin{aligned} x_2 &= +2 + \frac{2x_4}{3} - x_0 + \frac{x_3}{3} \\ x_1 &= +2 + \frac{x_4}{3} - x_0 + \frac{2x_3}{3} \\ x_5 &= +3 - x_4 + 3x_0 - x_3 \\ z &= \qquad\qquad - x_0. \end{aligned} \tag{D_3'}$$

Now we cannot maximize anymore and we have a feasible solution for the original problem: $\{x_1, x_2\} = (2, 2)$.

	x_0	x_1	x_2	x_3	x_4	x_5	z
(D_0')	0	0	0	-2	-2	7	0
(D_1')	2	0	0	0	0	9	-2
(D_2')	2	0	0	0	0	9	-2
(D_3')	0	2	2	0	0	3	0

Table 1: The solutions corresponding to each dictionary when the simplex algorithm is run on (Q_1').

Now we can return to the original linear program (Q_1). Using the initial feasible solution found via the auxiliary program, the initial dictionary is:

$$\begin{aligned} x_2 &= 2 + \frac{2x_4}{3} + \frac{x_3}{3} \\ x_1 &= 2 + \frac{x_4}{3} + \frac{2x_3}{3} \\ x_5 &= 3 - x_4 - x_3 \\ z &= 6 + \frac{4x_4}{3} + \frac{5x_3}{3}. \end{aligned} \tag{D_0}$$

$$\begin{aligned} x_2 &= 3 + \frac{x_4}{3} - \frac{x_5}{3} \\ x_1 &= 4 - \frac{x_4}{3} - \frac{2x_5}{3} \\ x_3 &= 3 - x_4 - x_5 \\ z &= 11 - \frac{x_4}{3} - \frac{5x_5}{3}. \end{aligned} \tag{D_1}$$

So the optimal solution for (Q_1) is 11 at the point $(4, 3)$.

3 Overview of the Simplex Algorithm

We are given the following linear program (Q) in *canonical form.*

$$\begin{array}{rrcll} & \max & \sum_{i=1}^{n} c_i x_i & & \\ \text{subject to:} & \sum_{i=1}^{n} a_{ji} x_i & \leq & b_j, & j : 1 \leq j \leq m, \\ & x_i & \geq & 0, & i : 1 \leq i \leq n. \end{array}$$

3.1 Initialization: Phase I

Initial dictionary:

$$\begin{aligned} x_{n+j} &= b_j - \sum_{i=1}^{n} a_{ji} x_i, \quad j : 1 \leq j \leq m, \\ z &= \sum_{i=1}^{n} c_i x_i. \end{aligned}$$

1. If solution is *feasible* (i.e. all basic variables are non-negative), go to Phase II.
2. Otherwise use auxiliary program described in Section 2 to find an initial basic feasible solution.
   (a) Create auxiliary program (Q') and dictionary (D'_0).
   (b) Use "illegal" pivot to find basic feasible solution for (Q').
   (c) Run simplex until final dictionary (D'_F).
   (d) If value at (D'_F) is < 0, declare (Q) *infeasible.*
   (e) Else form initial dictionary (D_0) for (Q) and go to Phase II.

3.2 Iteration: Phase II

Assume we have a dictionary D with basis B that corresponds to a basic feasible solution, in which $x_i = 0$ for $i \notin B$:

$$\begin{aligned} x_j &= \bar{b}_j - \sum_{i \notin B} \bar{a}_{ji} x_i, \quad \forall j \in B, \\ z &= \gamma + \sum_{i \notin B} \bar{c}_i x_i. \end{aligned}$$

1. Choice of entering variable:
   (a) If $\bar{c}_i \leq 0$ for all $i \notin B$, there is no choice of entering variable. Hence, we have reached the maximum value for z. *The simplex algorithm terminates on an optimal solution with value* γ.
   (b) If there exists $\bar{c}_k > 0$, choose x_k as entering variable.
2. Choice of leaving variable:
   (a) If all $\bar{a}_{jk}$'s are ≤ 0. *Then* (Q) *is unbounded.*
   (b) Otherwise, among all $\bar{a}_{jk}$'s > 0, choose one such that $\frac{\bar{b}_j}{\bar{a}_{jk}}$ is minimum. This is the leaving variable.

References

[BT97] Dimitris Bertsimas and John N. Tsitsiklis. *Introduction to linear optimization.* Athena Scientific Belmont, MA, 1997.

[CKM82] R. Chandrasekaran, Santosh N. Kabadi, and Katta G. Murthy. Some NP-complete problems in linear programming. *Operations Research Letters*, 1(3):101–104, 1982.

[Van01] Robert J. Vanderbei. *Linear programming: Foundations and Extensions.* Kluwer Academic Publishers, Boston, MA, 2001.

These lecture notes are partly based on the following sources: lecture notes by Stéphan Thomassé from previous versions of the same course and two textbooks [BT97] and [Van01].