Week 2

1 The Simplex method

The Simplex method was developed by Dantzig in the late 40-ties.

1.1 The standard form

The simplex method is a general description algorithm that solves any LP-problem instance. To do so it first brings it into standard form

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b, \\ & x \geq \underline{0}, \end{array} \tag{1}$$

with $x, c \in \mathbb{R}^n$, A an $m \times n$ matrix and $b \in \mathbb{R}^m$. We assume that $m \leq n$ and that $\text{rank}(A) = m$. The latter is a rather severe restriction, but justified by Theorem 2.5 of the book, the proof of which is easy and which I leave to yourself to read.

So far we have derived all sorts of insights about the geometry of LP-problems and especially about their feasible sets in the general form $P = \{x \in \mathbb{R}^n | Ax \geq b\}$. All these insights are valid for the standard form, being a special case. Almost all these insights followed from the row-structure of the matrix A. For the standard form we can derive some more insights, based on A's column structure, and which are fundamental for the simplex method.

A basic feasible solution (bfs) in LP (1) is defined by a set of n linearly independent active constraints, m of which are directly imposed on us by the m equalities $Ax = b$. We have assumed that the rows of A are linearly independent. The other $n - m$ active constraints must come from the non-negativity constraints. Thus, in any bfs $n - m$ variables must have value 0. We call them the non-basic variables of the bfs. The other m variables of the bfs are the basic variables. Let us denote them by $x_B = (x_{B(1)}, \ldots, x_{B(m)})$. Their values come from the subsystem $Bx_B = b$ where B is the so-called basis matrix constituted of the columns $A_{B(1)}, \ldots, A_{B(m)}$ of A. For x to be a basic solution it is required that these columns are linearly independent, and this is also sufficient.

Theorem 1.1 (2.4 in [B& T]) *Let $P = \{x \in \mathbb{R}^n | Ax = b, x \geq \underline{0}\}$ with A having linearly independent rows. $x \in \mathbb{R}^n$ is a basic solution $\Leftrightarrow$ $Ax = b$ and there exist indices $B(1), \ldots, B(m)$ such that*
a) Columns $A_{B(1)}, \ldots, A_{B(m)}$ are linearly independent;
b) If $i \neq B(1), \ldots, B(m)$, then $x_i = 0$.

Proof. $\Leftarrow$ follows immediately from the fact that under the conditions a) and b) $Bx_B = b$ has a unique solution.

⇒: Let x be a basic solution, hence the unique solution of the system of n active constraints: $Ax = b$ and (after renumbering if necessary) $x_i = 0 \ \forall i \neq B(1), \ldots, B(k)$, for some $k \leq m$. Notice that more than $n - m$ of the x_i's can be 0 (degeneracy). Anyway, this satisfies b). Hence, x being a basic solution implies (through Theorem 2.2 in [B&T]) $\sum_{i=1}^{k} A_{B(i)} x_{B(i)} = b$ must have a unique solution and hence $A_{B(1)}, \ldots, A_{B(k)}$ are linearly independent. Since $\text{rank}(A) = m$, there are $m-k$ extra columns $A_{B(k+1)}, \ldots, A_{B(m)}$ in A that added to $A_{B(1)}, \ldots, A_{B(k)}$ keeps the set of columns linearly independent, satisfying a). □

Notice that the theorem merely speaks about *basic solutions* i.o. basic *feasible* solutions. For the latter, just feasibility is the condition to be added.

Given the definition of adjacency of two extreme points or bfs's, it is clear that in terms of the theorem, two bfs's are adjacent if
- their sets of indices $B(1), \ldots, B(m)$ differ by exactly one.
- they differ in exactly one variable being non-basic, hence also one variable being basic.

This makes that in the degenerate case we may call two bfs's adjacent if they correspond to the same point. A degenerate bfs is a bfs in which more than n constraints are active (met with equality). In the standard LP this means that more than $n - m$ variables have value 0, i.e., some basic variables have also value 0.

1.2 Simplex in the non-degenerate case

The Simplex algorithm goes from one bfs to an adjacent bfs. Suppose for the time being that the feasible set $P \subset \mathbb{R}^n$ is non-empty and *non-degenerate*. Thus, we assume here that all bfs's have exactly n active constraints, i.e., all basic variables have strictly positive values.

Suppose that we have arrived at some bfs x with basic columns $A_{B(1)}, \ldots, A_{B(m)}$, making up basis matrix B. We split x in its non-basic part $x_{NB} = 0$ and its basic part $x_B = B^{-1}b$ (B has rank m, hence has an inverse).

As we observed before, two bfs's are adjacent if they differ in only one non-basic variable. Suppose Simplex moves from x to the solution $\bar{x}$, which has basic variable x_j that is non-basic in x. Let $\bar{x} = x + \theta d$, for some (non-negative or, in the non-degenerate case, positive) θ to be determined, with $d_j = 1$, since we wish to make x_j positive, $d_i = 0$ for all $i \neq j$ and $i \neq B(1), \ldots, B(m)$, i.e. all i corresponding to non-basic variables, except for non-basic variable j, since we wish the other non-basic variables to remain non-basic. We call d_B the part of d corresponding to x_B. Then $Ax = b$ and $A\bar{x} = b$ implies

$$0 = Ad = Bd_B + A_j \ \Rightarrow \ d_B = -B^{-1}A_j$$

We need to take care that $\bar{x} \geq 0$, in particular that $\bar{x}_B \geq 0$:

$$\bar{x}_B = x_B + \theta d_B \geq 0,$$

which implies that for all $i = 1, \ldots, m$, such that $d_{B(i)} < 0$

$$\theta \leq -x_{B(i)}/d_{B(i)}. \tag{2}$$

Clearly, no bound is imposed on θ if $d_{B(i)} \geq 0$. Determine

$$\theta = \min_{\{i=1,\ldots,m | d_{B(i)}<0\}} -x_{B(i)}/d_{B(i)},$$

which is called the minimum ratio test, and let

$$\ell = \operatorname*{argmin}_{\{i=1,\ldots,m | d_{B(i)}<0\}} -x_{B(i)}/d_{B(i)}.$$

Please take care of the above confusing notation: ℓ is the number in the $B(i)$-sequence. It is not the number of the column of A. Thus, what we find here is $A_{B(\ell)}$ and not A_ℓ.

Then $x_{B(\ell)}$ becomes 0 and moves out of the basis; $\bar{x}_{B(\ell)} = 0$. It is replaced by x_j, the new basic variable: i.e., $A_{B(\ell)}$ in the basis matrix of x is replaced by A_j to obtain the basis matrix of $\bar{x}$. All other columns corresponding to x_B remain the same for $\bar{x}$:

$$\bar{B}(i) = B(i) \ \forall i \neq \ell, \text{ and } \ \bar{B}(\ell) = j.$$

This is in Simplex terms called a *pivot step*. We show that this new matrix is in fact a basis matrix.

Theorem 1.2 (3.2 in [B& T]) $A_{\bar{B}(1)}, \ldots, A_{\bar{B}(\ell-1)}, A_{\bar{B}(\ell)}, A_{\bar{B}(\ell+1)}, \ldots, A_{\bar{B}(m)}$ *are linearly independent.*

Proof. Suppose that they would be dependent, i.e.,

$$\sum_{i=1}^{m} \lambda_i A_{\bar{B}(i)} = 0$$

has a non-trivial solution implying that this also true for

$$B^{-1} \sum_{i=1}^{m} \lambda_i A_{\bar{B}(i)} = \sum_{i=1}^{m} \lambda_i B^{-1} A_{\bar{B}(i)} = 0.$$

We show that this cannot be true. Let e_i denote the i-th unit vector. Remember that $A_{\bar{B}(i)} = A_{B(i)}$ for all $i \neq \ell$. Hence,

$$B^{-1} A_{\bar{B}(i)} = B^{-1} A_{B(i)} = e_i \ \forall i \neq \ell,$$

so all of them have a 0 at the ℓ-th position.

$$B^{-1} A_{\bar{B}(\ell)} = B^{-1} A_j = -d_B$$

and, by definition of the minimum ratio test (2), $d_{B(\ell)} < 0$. Hence, $B^{-1} A_{\bar{B}(\ell)}$ is linearly independent of $B^{-1} A_{\bar{B}(i)}$, $i = 1, \ldots, m$. $\square$

As a result, indeed, $\bar{x} = x + \theta d$ is again a bfs. This is a simplex step from one non-degenerate bfs to an adjacent non-degenerate bfs.

The Simplex method makes such a step only when improving the objective; i.e., if in the direction d the objective sees improvement; i.e. $c^T d \leq 0$ (remember that we are minimizing by default), written out:

$$c_j d_j + c_B^T d_B = c_j + c_B^T d_B = c_j - c_B^T B^{-1} A_j \leq 0.$$

For those who remember from their basic LP-course: $\hat{c}_j = c_j - c_B^T B^{-1} A_j$ was called the *reduced costs* of variable j. We verify that the reduced costs of the basic variables are 0:

$$\hat{c}_{B(i)} = c_{B(i)} - c_B^T B^{-1} A_{B(i)} = c_{B(i)} - c_B^T e_{B(i)} = c_{B(i)} - c_{B(i)} = 0.$$

Simplex stops as soon as none of the reduced costs is negative and we verify this is correct.

Theorem 1.3 (3.1 in [B& T]) *Given bfs x with reduced cost vector $\hat{c}$.*
a) If $\hat{c} \geq 0$ then x is optimal;
b) If x is optimal and non-degenerate then $\hat{c} \geq 0$.

PROOF. a) Take any feasible y, $y \in P$. Define $d = y - x$ and show that $c^T d \geq 0$ given that $\hat{c} \geq 0$. Let B be the basis matrix corresponding to x.
We have $Ax = b$ and $Ay = b \Rightarrow Ad = Bd_B + \sum_{i \in NB} A_i d_i = 0 \Rightarrow d_B = -\sum_{i \in NB} B^{-1} A_i d_i \Rightarrow$

$$c^T d = c_B^T d_B + \sum_{i \in NB} c_i d_i = \sum_{i \in NB} (c_i - c_B^T B^{-1} A_i) d_i = \sum_{i \in NB} \hat{c}_i d_i.$$

The proof follows since for all $i \in NB$, $y_i \geq 0$ and $x_i = 0 \Rightarrow d_i \geq 0$.

b) Obvious by contradiction, read for yourself. □

There is another way in which the Simplex algorithm stops. It can happen that for some non-basic variable j the reduced cost $\hat{c}_j < 0$ and that in the direction d, with $d_j = 1$ and $d_B = -B^{-1} A_j$ and $d_i = 0$, for all $i \neq j, B(1), \ldots, B(m)$, there is no bound on θ in (2) because for all $i = 1, \ldots, m$, $d_{B(i)} \geq 0$. This means that we have an improvement direction in which we can decrease the objective value without limit and therefore we have an unbounded solution value $-\infty$.

Theorem 1.4 (3.3 in [B& T]) *Given an LP with a non-empty non-degenerate feasible polyhedron, the Simplex method terminates in a finite number of iterations arriving either at an optimal bfs or finding out that the solution value is unbounded.*

Recapitulating an iteration:

1. Given bfs x and basis matrix $B = \left(A_{B(1)}, \ldots, A_{B(m)}\right)$;

2. Compute reduced costs $\hat{c}_j = c_j - c_B^T B^{-1} A_j$; if all $\hat{c}_j \geq 0$ stop and conclude that x is an optimal solution; otherwise select j such that $\hat{c}_j < 0$;

3. Compute $u = B^{-1} A_j$; if $u \leq 0$ then output value $-\infty$; otherwise determine

$$\theta = \min_{\{i=1,\ldots,m|u_i>0\}} x_{B(i)}/u_i$$

and let

$$\ell = \operatorname*{argmin}_{\{i=1,\ldots,m|u_i>0\}} x_{B(i)}/u_i.$$

4. In the basis matrix replace $A_{B(\ell)}$ by A_j, and create new bfs y by setting $y_{B(i)} = x_{B(i)} - \theta u_i$, $i = 1, \ldots, m$, setting $y_j = \theta$ and $y_i = 0$ for all other i.

What is the running time of this iteration? Computing B^{-1} in the most straightforward way takes $O(m^3)$ elementary computer operations. Pre-multiplying this with c_B^T takes $O(m^2)$ operations. Computing the reduced costs of the $n - m$ non-basic variables takes $O((n - m)m)$. The rest takes only $O(m)$ time. Thus overall time is $O(m^3 + nm)$.

However, we do not need to recompute B^{-1} every time from scratch, since from one to the next iteration the basis matrices differ in only one column. Remember from your linear algebra course how you computed the inverse of a matrix by Gaussian elementary row operations both on the matrix and the identity matrix, resulting in the inverse on the place where you started with the identity matrix. Then it will be easy to see that if we have inverted already almost the whole matrix except for one column then it takes no more than m elementary row operations to compute the inverse, and each row operation costs only $O(m)$ time. Thus in fact each iteration of the simplex method can be implemented to take only $O(m^2 + nm)$ time.

The *revised simplex* method is a version of the simplex method that does not compute all reduced costs, but computes them one by one until it encounters the first negative one, and then pivots on that variable as entering in the basis. Most of you have seen the usual simplex tableau. In the revised simplex one the current solution and only B^{-1} (under slack variables or artificial slack variables) and $c_B^T B^{-1}$ (reduced costs of these slacks) are stored. Clearly in the worst case revised simplex will have to do the same amount of work as normal simplex, but it may be lucky and find already the first reduced cost to be negative in which case it saves $O(nm)$ time.

Read but don't learn the part in Section 3.4 about Practical Performance Enhancements.

Section 3.6 on a column geometry interpretation of the simplex method is skipped but I recommend you to read it. It may help in understanding Farkas Lemma in the next chapter.

1.3 Degeneracy and anticycling

This year I skip Degeneracy, but retain the notes for the interested students.
Degeneracy of a point x is the phenomenon that there are more than n constraints of P active in x. In the context of the standard formulation this implies that
- there are more choices of $B(1), \ldots, B(m)$ that satisfy Theorem 1.1 for x, or equivalently,
- there are more than $n-m$ variables that have value 0 in that point, or equivalently,
- there is a basic variable in any of the choices $B(1), \ldots, B(m)$ that has value 0, or equivalently,
- x is adjacent to itself.

Degeneracy is a nuisance in LP in several ways. E.g. in case of degeneracy Theorem 1.3b does not hold. Thus, in case of degeneracy we do not have a good optimality check.

Degeneracy typically occurs in a simplex iteration when in the minimum ratio test more than one basic variable determine θ, hence two or more variables go to 0, whereas only one of them will be leaving the basis. This is in itself not a problem, but making simplex iterations from a degenerate solution may give rise to cycling, meaning that after a certain number of iterations without improvement in objective value the method may turn back to the point where it started, see e.g. Example 3.6 in the book. The third nuisance.

To avoid cycling most of us know Bland's Lexicographic Anticycling rule, but I assume few have seen the proof that it works. For describing the rule one should think of the rows of the matrix $B^{-1}[A \mid b]$ as $B^{-1}[b \mid A]$; i.e. every row starts with the value of the basic variable corresponding to that row. This column is then called the 0-th column in the book.

Definition 1.1 Lexicographic pivoting rule. Select as entering variable any x_j with $\hat{c}_j < 0$ and let $u = B^{-1}A_j$ (the j-th column in the simplex tableau). For each i with $u_i > 0$ divide the i-th row by u_i and select the lexicographically smallest row. If this is row ℓ then $x_{B(\ell)}$ leaves the basis.

This rule boils down to first performing the minimum ratio test. Let

$$L =: \{h \mid x_{B(h)}/u_h = \theta = \min_{\{i=1,\ldots,m \mid u_i > 0\}} x_{B(i)}/u_i\}.$$

If $|L| = 1$, i.e., if the test gives a unique solution then that is the row with smallest value in its 0-th position, after division with its u_i-value. If $|L| > 1$, then all the rows h, $h \in L$, start in the 0-th position with the same value $\theta = x_{B(h)}/u_h$ and that value is (by the minimum ratio test) smaller than the values $x_{B(i)}/u_i$ for any $i \notin L$. From amongst these rows $h \in L$ the lexicographically smallest is then chosen.

Theorem 1.5 (3.4 in [B& T]) *Suppose simplex starts with an arrangement of the columns (variables) such that all rows are lexicographically positive. Then each iteration of the simplex algorithm applying the lexicographic pivoting rule*
a) maintains lexicographically positive rows;
b) the objective row of reduced costs is strictly lexicographically increasing;
As a result the simplex algorithm terminates after a finite number of iterations.

PROOF. a) Let x_j enter the basis and $x_{B(\ell)}$ leave the basis. Thus, $u_\ell > 0$ and because row ℓ was selected according to the lexicographic pivoting rule, we have

$$\frac{\text{row } \ell}{u_\ell} \prec^L \frac{\text{row } i}{u_i}, \ \forall\, i \neq \ell \text{ with } u_i > 0. \tag{3}$$

The pivoting operation row ℓ is divided by u_ℓ, hence remains lexicographically positive.

If $u_i < 0$ then in the pivoting operation a positive number times row ℓ is added to row i and hence row i remains lexicographically positive. If $u_i > 0$ then u_i/u_ℓ times row ℓ is distracted from row i. Because of (3) the new row i is still lexicographically positive.

b) $\hat{c}_j < 0$. To make it 0 we add a positive number times the lexicographically positive row ℓ to the objective row, making it lexicographically increasing.

c) Follows directly from b) $\Rightarrow$ no basis is repeated. □

Read in the book how it can be made sure to obtain an initially lexicographical ordering of the rows.

1.4 Running time of the simplex method

We have seen that each iteration of the simplex method can be implemented to run in time polynomial in the number of variables n and the number of functional restrictions m. Thus to make simplex an efficient method only requires to prove that some pivoting rule exists under which only a polynomial number of iterations is necessary. We have reached an open problem in our field:

Open Problem 1. Does there exist a pivot rule that makes the simplex method stop with the correct answer after at most $f(n, m)$ iterations, with f a polynomial function of n and m.

It is not too difficult to think of an LP with a number of vertices that is exponential in n and both a starting vertex and a pivot rule such that all vertices are visited before reaching the optimal one. See the slightly perturbed unit n-cube in the book. Maybe it will be even necessary for a polynomial time simplex method to construct a pivot rule that sometimes accepts a step in a direction that does not improve the objective, maybe some simulated annealing type of pivot rule. This is really a challenging open problem and so much at the core of our field.

A related, weaker, question is given any starting point in an LP is there always a pivot rule (possibly accepting objective worsening) such that after a polynomial number of iterations an optimal solution is reached. This question is translated into polyhedral terms: given a polyhedron with its vertices (0-dimensional faces) and edges (1-dimensional faces) seen as a graph, does there exist a path from any vertex to any other vertex of length bounded by a polynomial function of n and m, where *length* is combinatorial length; the number of edges on the path? That is, is the *combinatorial diameter* of any polyhedron bounded by a polynomial function of n and m? This is a beautiful open problem, stated differently:

Open Problem 2. Does every polytope in dimension n described by h halfspaces (linear inequalities) have combinatorial diameter at most $f(h, n)$, with f some polynomial function.

Until 2010 we had the *Hirsch Conjecture* stating that $f(h, n) = h - n$. Hirsch stated his conjecture in 1956, originally as a conjecture on any polyhedron. But already in 1967 Klee and Walkup in 1967 disproved it by giving an unbounded polyhedra with diameter $h - n + \lfloor n/5 \rfloor$. After that Hirsch's conjecture remained alive under the assumption of polytopes, bounded polyhedra. But 43 years later, 54 years after the conjecture, in the beginning of summer 2010 Francisco Santos from Santander University disproved also this version of the conjecture (http://arxiv.org/abs/1006.2814). The counterexample to Hirsch' Conjecture has dimension 43 defined by 86 halfspaces. However, his proof still leaves Open Problem 2, even with the possibility that f is a linear function in h.

For some classes of polytopes the Hirsch Conjecture is known to be true. E.g., all $0 - 1$-polytopes, and all polytopes of dimension at most 6.

The best bound that has been established so far is $h^{\log_2 n+2}$ by Kalai and Kleitman in 1992 (http://www.ams.org/journals/bull/1992-26-02/S0273-0979-1992-00285-9/S0273-0979-1992-00285-9.pdf), which is less than exponential but more than polynomial. The proof is half a page, extremely elegant, using the probabilistic method. I recommend it to those of you who would like to know more about these research questions.

In spite of these open questions, the simplex method appears to work very efficiently in practice; on average $O(n)$ iterations seem to suffice. This has been supported by so-called probabilistic analysis, though it is very hard to come up with a representative probabilistic model for random feasible LP-instances. Or even more convincing is so-called smoothed analysis of the simplex method, in which it is shown that bad instances are very non-dense in the set of all possible instances, since tiny random perturbations of the coefficients gives a polynomial number of iterations in expectation (http://www.cs.yale.edu/homes/spielman/simplex/index.html). A very elegant (friendly) version of smoothed analysis of a form of the simplex method has recently been developed by Daniel Dadush and Sophie Huiberts from CWI https://arxiv.org/abs/1711.05667, as the Master thesis work of Sophie https://dspace.library.uu.nl/handle/1874/359199

Material of Week 2 from [B& T]

Chapter 3: 3.1–3.3, 3.7. We skip Sections 3.5 and 3.6. and this year also Section 3.4

Exercises of Week 2

3.7, 3.18. Those who wish may try 3.27.

Next week

Chapter 4