LP. Lecture 4

Chapter 4: efficiency of the simplex algorithm

- how to measure efficiency?
- worst case analysis, the Klee-Minty cube
- mean analysis and in practise

Status

How far are we now in our LP story?

- simplex method (Phase I and II)
- problems: degeneracy and cycling
- solution: anticycling rules, perturbations
- the fundamental theorem of LP

Next question: how good is the simplex algorithm ?

But: first a bit on equivalent optimization problems!

Equivalent optimization problems

Often it is useful to rewrite an optimization problem into a more convenient form. Then it is important to end up with an "equivalent optimization problem". Let us clarify what this means.

Let P and Q denote two optimization problems, with variable vectors $x \in \mathbb{R}^n$ in P and $y \in \mathbb{R}^k$ in Q. Here n and k may be different.

We say that P and Q are equivalent if

1. P is feasible if and only if Q is feasible.
2. P is unbounded if and only if Q is unbounded.
3. P has an optimal solution if and only if Q has an optimal solution. Furthermore, there is a function $f : \mathbb{R}^n \to \mathbb{R}^k$ such that x is optimal in P if and only if $f(x)$ is optimal in Q.

Note that one can prove that there is a symmetry here: it does not matter if the order of P and Q is interchanged in this definition.

As an example, and an exercise, consider the problems

$$\begin{array}{ll} \mathrm{P:} & \max\{c^T x : Ax \le b\} \\ \mathrm{Q:} & \max\{c^T(x' - x'') : A(x' - x'') \le b,\ x', x'' \ge O\} \end{array}$$

Show that P and Q are equivalent!

LP problems may, for instance, have some variables that are nonnegative (or nonpositive), some variables that are free; there may be inequalities ($\le$ og $\ge$), and equations. Learn how to rewrite such problems into equivalent LP's.

Efficiency

Two types of efficiency measures exist for algorithms:

- worst case: the maximum number of arithmetic operations for problem (instances) of a given size.
- mean (average): the mean og expected number of arithmetic operations for problem (instances) of a given size.

LP: the computational time increases when the number of variables or constraints is increased. So computational time is some function of "problem size".

How can we measure problem size?

- *simple approach:* counting the numbers in input, i.e., $m \times (n+1)$ in LP. Weakness: in "real problems" (where m and n are large) many entries in the coef. matrix are zero. This may be exploited in the simplex algorithm to speed up the computations.

- *more accurate:* the number of nonzeros. Weakness: arithmetic operations with large numbers take longer time than they do for smaller numbers. (I would rather compute $14 \cdot 7$ than $832573928 \cdot 3722984$, right!)
- *even more accurate:* the number of bits needed to store all the numbers on the computer.

How can we measure the total work (complexity) of an algorithm?

- computational time (CPU)
- the number of iterations
- the number of elementary arithmetic operations.

Our choice here: we simply use m and n as measures of problem size and the number of pivots as complexity measure.

Want to consider worst case analysis of the simplex algorithm. The answer will be, unfortunately, that the method in theory is not good. In practice, however, the situation is the opposite: the method works excellently!

The "explanation": the "hard LP problems" do not show up in real world problems.

In 1972 V.Klee and G.J.Minty found a class of LP problems that "kills" the simplex algorithm with the usual pivot rule (largest coefficient rule).

The LP problem has n variables and it turns out that the number of pivots required is $2^n - 1$. Thus, the number of pivots grows exponentially fast in n (the number of variables). Even for moderate values of n this number of iterations is hoplessly big! To indicate this, consider the following table given computational time for (assuming one million pivots per second, which is very optimistic!):

n :	10	20	50	100
n^2 :	100	400	2500	10000
2^n :	1024	1048576	1.125899e+15	1.267650e+30
tid	0.001 sec.	1.0 sec.	35 years	4.0e+16 years

We shall investigate this LP problem. The idea is to deform the cube $[0,1]^n$ in $\mathbb{R}^n$ in such a way that the simplex algorithm is forced to go through all the 2^n vertices! This gives $2^n - 1$ pivots.

Here is the Klee-Minty LP problem:

$$
\begin{array}{llllll}
\max & \sum_{j=1}^{n} 10^{n-j} x_j & & & & \\
\text{s.t.} & & & & & \\
& 2\sum_{j=1}^{i-1} 10^{i-j} x_j & + & x_i & \leq & 100^{i-1} \quad \text{for } i = 1, \ldots, n \\
& & & x_j & \geq & 0 \quad \text{for } j = 1, \ldots, n.
\end{array}
$$

Interpretation of the constraints:

- $i = 1$: $x_1 \leq 1$
- $i = 2$: $20x_1 + x_2 \leq 100$, so $x_2 \leq 100 - 20x_1 \approx 100$ since $0 \leq x_1 \leq 1$.
- $i = 3$: $200x_1 + 20x_2 + x_3 \leq 10000$, so $x_3 \leq 10000 - 200x_1 - 20x_2 \approx 10000$ since $0 \leq x_1 \leq 1$ and $0 \leq x_2 \leq 100$.
- ...

So roughly we have the constraints

$$\begin{array}{ccccc} 0 & \leq & x_1 & \leq & 1 \\ 0 & \leq & x_2 & \leq & 100 \\ & & \vdots & & \\ 0 & \leq & x_n & \leq & 100^{n-1} \end{array}$$

so that the set P of feasible solutions is roughly an n-dimensional rectangle (cube). P is therefore called the *Klee-Minty cube*.

To simplify the analysis we modify the problem a bit.

- choose numbers b_i such that $1 = b_1 \ll b_2 \ll \cdots \ll b_n$. For instance, choose b_1 so much smaller than b_2 that even if we multiply by certain numbers in the course of the simplex algorithm, the new value of b_1 will still be far less than the new value of b_2. Think about the b_i's ar independent variables; one can find appropriate values for these later.

- in the Klee-Minty problem we first replace the right side 100^{i-1} with b_i. Note that the old right sides are increased by a factor of 100 for each new row, and we have kept this intact by the choice of the b_i-s.
- we then replace the right side b_i by

$$\sum_{j=1}^{i-1} 10^{i-j} b_j + b_i.$$

  This is a "minor alteration" because the first term is a lot smaller than b_i (because $b_1, \ldots, b_{i-1}$ are suitably small in comparison to b_i.)
- finally we change the objective function in the LP problem by subtracting

$$(1/2) \sum_{j=1}^{i-1} 10^{n-j} b_j.$$

The result is a modified Klee-Minty problem:

$$\begin{array}{ll}\max & \sum_{j=1}^{n} 10^{n-j} x_j - (1/2) \sum_{j=1}^{i-1} 10^{n-j} b_j \\ \text{s.t.} & \\ & \begin{array}{rcll} 2\sum_{j=1}^{i-1} 10^{i-j} x_j \quad + \quad x_i & \leq & \sum_{j=1}^{i-1} 10^{i-j} b_j + b_i & \text{for } i \leq n \\ x_j & \geq & 0 & \text{for } j \leq n. \end{array}\end{array}$$

Note that the right-hand sides are positive, so we don't need Phase I of the simplex method.

Result: the simplex algorithm with the "largest coefficient pivot rule" uses $2^n - 1$ pivots to solve the modified Klee-Minty problem.

We will let the proof of this the be an exercise (Ex. 4.5 and 4.6 i Vanderbei).

We now solve the modified Klee-Minty problem for $n = 3$. Hopefully we will see a pattern which is valid for all n.

Dictionary 0:

$$
\begin{array}{rcrcrcrcrcrcr}
\eta & = & -\frac{100}{2}b_1 & - & \frac{10}{2}b_2 & - & \frac{1}{2}b_3 & + & 100x_1 & + & 10x_2 & + & x_3 \\
\hline
w_1 & = & b_1 & & & & & - & x_1 & & & & \\
w_2 & = & 10b_1 & + & b_2 & & & - & 20x_1 & - & x_2 & & \\
w_3 & = & 100b_1 & + & 10b_2 & + & b_3 & - & 200x_1 & - & 20x_2 & - & x_3
\end{array}
$$

Here x_1 will go into basis by the largest coefficient rule. Since b_1 is a lot *smaller* than b_2 and b_3, w_1 will leave basis. Complete the pivot (check the calculations based on $x_1 = b_1 - w_1$).

Dictionary 1:

$$
\begin{array}{rcrcrcrcrcrcr}
\eta & = & \frac{100}{2}b_1 & - & \frac{10}{2}b_2 & - & \frac{1}{2}b_3 & - & 100w_1 & + & 10x_2 & + & x_3 \\
\hline
x_1 & = & b_1 & & & & & - & w_1 & & & & \\
w_2 & = & -10b_1 & + & b_2 & & & + & 20w_1 & - & x_2 & & \\
w_3 & = & -100b_1 & + & 10b_2 & + & b_3 & + & 200w_1 & - & 20x_2 & - & x_3
\end{array}
$$

Fantastic! The new list of basis contains the same numbers apart from some alterations of sign! The alterations are:

- x_1 and w_1 switched roles
- only alterations in the two columns for x_1 and b_1
- and these two columns are multiplied by -1, apart from in the pivot equation, where there is no alteration of sign

Now x_2 goes in to basis and w_2 out.
Dictionary 2:

$$
\begin{array}{rcrcrcrcrcrcr}
\eta & = & -\frac{100}{2}b_1 & + & \frac{10}{2}b_2 & - & \frac{1}{2}b_3 & + & 100w_1 & - & 10w_2 & + & x_3 \\
\hline
x_1 & = & b_1 & & & & & - & w_1 & & & & \\
x_2 & = & -10b_1 & + & b_2 & & & + & 20w_1 & - & w_2 & & \\
w_3 & = & 100b_1 & - & 10b_2 & + & b_3 & - & 200w_1 & + & 20w_2 & - & x_3
\end{array}
$$

Dictionary 3:

$$
\begin{array}{rcrcrcrcrcrcr}
\eta & = & \frac{100}{2} b_1 & + & \frac{10}{2} b_2 & - & \frac{1}{2} b_3 & - & 100 x_1 & - & 10 w_2 & + & x_3 \\
\hline
w_1 & = & b_1 & & & & & - & x_1 & & & & \\
x_2 & = & 10 b_1 & + & b_2 & & & - & 20 x_1 & - & w_2 & & \\
w_3 & = & -100 b_1 & - & 10 b_2 & + & b_3 & + & 200 x_1 & + & 20 w_2 & - & x_3
\end{array}
$$

Dictionary 4:

$$
\begin{array}{rcrcrcrcrcrcr}
\eta & = & -\frac{100}{2} b_1 & - & \frac{10}{2} b_2 & + & \frac{1}{2} b_3 & + & 100 x_1 & + & 10 w_2 & - & w_3 \\
\hline
w_1 & = & b_1 & & & & & - & x_1 & & & & \\
x_2 & = & 10 b_1 & + & b_2 & & & - & 20 x_1 & - & w_2 & & \\
x_3 & = & -100 b_1 & - & 10 b_2 & + & b_3 & + & 200 x_1 & + & 20 w_2 & - & w_3
\end{array}
$$

Dictionary 5:

$$
\begin{array}{rcrcrcrcrcrcr}
\eta & = & \frac{100}{2}b_1 & - & \frac{10}{2}b_2 & + & \frac{1}{2}b_3 & - & 100w_1 & + & 10w_2 & - & w_3 \\
\hline
x_1 & = & b_1 & & & & & - & w_1 & & & & \\
x_2 & = & -10b_1 & + & b_2 & & & + & 20w_1 & - & w_2 & & \\
x_3 & = & 100b_1 & - & 10b_2 & + & b_3 & - & 200w_1 & + & 20w_2 & - & w_3
\end{array}
$$

Dictionary 6:

$$
\begin{array}{rcrcrcrcrcrcr}
\eta & = & -\frac{100}{2}b_1 & + & \frac{10}{2}b_2 & + & \frac{1}{2}b_3 & + & 100w_1 & - & 10x_2 & - & w_3 \\
\hline
x_1 & = & b_1 & & & & & - & w_1 & & & & \\
w_2 & = & -10b_1 & + & b_2 & & & + & 20w_1 & - & x_2 & & \\
x_3 & = & -100b_1 & + & 10b_2 & + & b_3 & + & 200w_1 & - & 20x_2 & - & w_3
\end{array}
$$

Dictionary 7:

$$
\begin{array}{rcrcrcrcrcrcr}
\eta & = & \frac{100}{2} b_1 & + & \frac{10}{2} b_2 & + & \frac{1}{2} b_3 & - & 100 x_1 & - & 10 x_2 & - & w_3 \\
\hline
w_1 & = & b_1 & & & & & - & x_1 & & & & \\
w_2 & = & 10 b_1 & + & b_2 & & & - & 20 x_1 & - & x_2 & & \\
x_3 & = & 100 b_1 & + & 10 b_2 & + & b_3 & - & 200 x_1 & - & 20 x_2 & - & w_3
\end{array}
$$

Optimal! Therefore: $7 = 2^3 - 1$ pivots.

Observations:

- in each pivot x_j and w_j switch roles for a certain j
- apart from alterations of signs all the numbers are preserved during the pivots
- but how does the sign alter? Ex. 4.5 and 4.6!
- the problem could be solved with only one pivot, if we instead of x_1 had taken x_3 in to basis! But the choice was controlled by the rule of pivot.

Comments on efficiency and LP

- Are there other pivot rules for the simplex algorithm that always avoid 2^n pivots ??? We would prefer if the number of pivots do not grow faster than f.ex. n^2 or n^3 (or a polynomial in $n =$ number of variables)!
- The answer is unknown! But for all suggested pivot rules someone has found 2^n "contradictions".
- K.-H. Borgwardt has given a statistical analysis which shows that the *expected* number of pivots (when the LP problems are "drawn randomly") grows as $n^3 m^{1/(n-1)}$
- In *practice* one assumes that the number of pivots typically lies between m and $2m$. This is very good! Note that the number of variables n matter less for the number of pivots (even though the calculation time increases by n).

Comments on efficiency and LP

- In 1979 the ellipsoid method was developed by L.Khachian: first polynomial time algorithm for LP. This was a theoretical breakthrough. This algorithm is theoretically "good", but hopeless in practise!
- In 1984 N. Karamarkar published a new LP algorithm based on a different set of ideas: unlike the simplex algorithm, a series of points, which are not basic solutions, are made; these points lie in the interior of the feasible set.
- During the last years a very active field of research has been interior point methods for LP. Part 3 of Vanderbei's book treats these methods.