Practical Work 1 : Klee-Minty complexity

Max Perraudin, Marie Dupont
IPSA - Aero 4 2019-2020
Course: Ma421 - Operational Research

***Abstract*— The goal of this project is to evaluate the complexity of solving the Klee-Minty Linear Programming problems using the simplex method. We use Python and the package Scipy which includes a solver in order to solve the problem of various order between 2 and 17. We found that the simplex method is exponential in time and number of iterations and we compared our results to the scientific literature. However, we are limited by the power of our personal computers. Furthermore, we do not know how exactly Scipy applies the simplex method.**

I. Introduction to the Simplex method

The simplex algorithm is a popular method for linear programming. It operates on linear programs in the canonical form :

$maximize \quad c^T x$

subject to $Ax \leq b$ and $x \geq 0$

with $x = (x_1, ..., x_n)$ the variables of the problem, $c = (c_1, ..., c_n)$ the coefficients of the objective function, A a $p*n$ matrix and $b = (b_1, ..., b_n)$ non-negative constants

II. The Klee-Minty linear programming problems

The problem is defined as follows :

$$max \sum_{j=1}^{n} 10^{n-j} x_j$$

$$2\sum_{j=1}^{i-1}(10^{n-j}x_j) + x_i \leq 100^{i-1} \quad i = 1, 2, ..., n$$

$$x_j \geq 0 \quad j = 1, 2, ..., n$$

This can be expressed under the following form :

$$\begin{cases} max 100x_1 + 10x_2 + x_3 \\ x_1 \leq 1 \\ 20x_1 + x_2 \leq 10^2 \\ 200x_1 + 20x_2 + x_3 \leq 10^4 \\ x_1 \geq 0 \quad x_2 \geq 0 \quad x_3 \geq 0 \end{cases}$$

The KleeMinty cube has been used to analyze the performance of many algorithms. It is an example that shows the worst-case computational complexity of many algorithms of linear optimization The time complexity of an algorithm counts the number of arithmetic operations sufficient for the algorithm to solve the problem.

III. Measure of the complexity : Methodology

In order to measure the complexity of the Simplex when solving the Klee-Minty Linear Programming problems, we can measure the time taken by the method to solve the problem, or we can measure the number of iterations needed to reach the solution. In our case we will study both the time and the number of iterations. To measure the time, we use the Python package Time. The syntax is as follows :

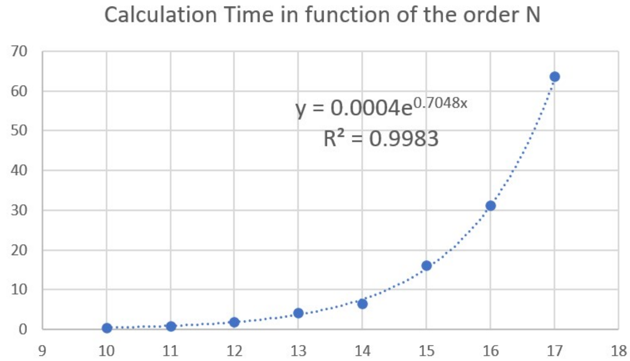
```
start_time = time()
res = linprog(c, A, b, method=
    'revised simplex')
stop_time = time()
calc_time = stop_time - start_time
```

Now to measure the number of iterations, we simply use the command `res.nit` with `res` being the returned datatype from `linprog`. Then we need to determine the order of the resulting trend-line associated with the time and iteration data. There are two possible options : Exponential in the shape of $y = ae^{bx}$, or Polynomial defined by $y = ax^b$.

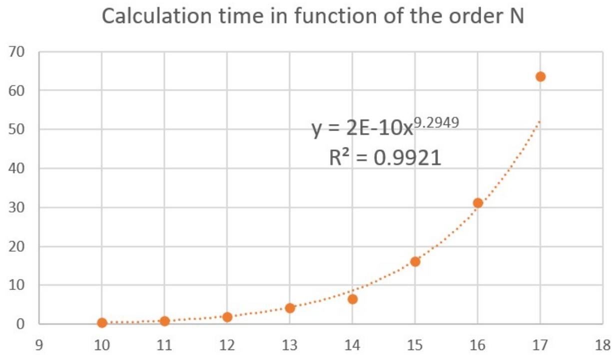
IV. Results

After running our program with the order N varying from 10 to 17, we extracted the number of iterations and the time into an Excel table and created two graphs.

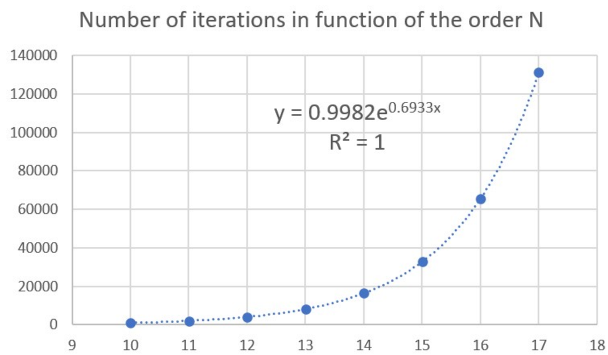
Using Excel, we drew the trend-line and displayed the formula. We compared different possible trend-lines, using the R squared value, which indicates the reliability of a trend-line.



For the time complexity, we found that the exponential function $y = 0.0004e^{0.7048x}$ is the best fit with $R^2 = 0.9983$. As shown in the figure below, the power function has a slightly lower R squared value, $R^2 = 0.9921$.



Using the same methodology for the number of iterations, we found a perfect fit with the exponential function at $R^2 = 1$.



Using these figures, we can determine that the Simplex algorithm has an exponential complexity in time, as well as in number of iterations when solving the Klee-Minty problems.

V. Comparison with the scientific literature

The scientific literature differentiates between "average case" and "worst case". In the average case, the simplex method is known to be very efficient, with polynomial time complexity. But when faced with the worst case -often the Klee-Minty problem-, its time complexity becomes exponential.

According to this information and our results, we can suppose that we are in the worst case scenario with our Klee-Minty problem.

VI. Introducing a change of variable

We operate the following change of variable :

$$u_j = 100^{1-j} x_j$$

and study its influence on the results. The problem becomes :

$$\begin{cases} min - 100u_1 - 1000u_2 - 10000u_3 \\ u_1 \leq 1 \\ 20u_1 + 100u_2 \leq 10^2 \\ 200u_1 + 2000u_2 + 10000u_3 \leq 10^4 \\ u_1 \geq 0 \quad u_2 \geq 0 \quad u_3 \geq 0 \end{cases}$$

When solving this problem using the Python code, we obtain the solution after only 1 iteration, and after only an average of 4 milliseconds. This is true independently of the order of the the problem.

The simplex method usually uses the rule of steepest descent, and this forces the method to go by every corner of the Klee-Minty cube before finding the solution, which was right next to the starting point. However, with the change of variable, the solution is reached from the starting point in only one step. As the solution is always next to the starting point, the order of the problem does not matter.