Why does the simplex algorithm work?

Theorem (Fundamental theorem of Linear Programming)

Take a LP in standard inequality form. Then one out of the following three assertions holds:

- *The LP in* ***infeasible****.*
- *The LP is* ***unbounded****.*
- *The LP has at least one* ***basic optimal solution****, that is a solution obtained by setting non basic variables to 0 in a dictionary.*

Theorem (Fundamental theorem of Linear Programming)

Take a LP in standard inequality form. Then one out of the following three assertions holds:

- *The LP in **infeasible**.*
- *The LP is **unbounded**.*
- *The LP has at least one **basic optimal solution**, that is a solution obtained by setting non basic variables to 0 in a dictionary.*

In particular, if the original LP has n decision variables and m constraints, so that the total number of variables (decision and slack ones) is $n + m$, if there exists an optimal solution then there exists one with at most m non-zero variables.

Using the Two Phase method, to prove the fundamental theorem of Linear Programming only the following needs to be proved.

Theorem

Take a (feasible) dictionary coming from a LP in standard form. Then, providing the rule for choosing entering and leaving variables is well chosen[1]*, one needs to perform only a* ***finite*** *number of pivots to optimality until:*

- *either there is no entering variable and an optimal basic solution is reached;*
- *either there is no leaving variable and the problem is unbounded.*

[1]Anstee's rule can fail, but Bland's rule does not.

Proving the second theorem

The proof of the second theorem relies on two results.

- There are finitely many dictionaries. In fact, each dictionary is entirely characterized by which variables are the basic ones and which ones are the non basic ones.

[2] If $n = 77$ and $m = 9$ (diet's problem), this number is roughly 4.6×10^{11}.
[3] to choose entering and leaving variables

Proving the second theorem

The proof of the second theorem relies on two results.

- There are finitely many dictionaries. In fact, each dictionary is entirely characterized by which variables are the basic ones and which ones are the non basic ones.
  (If there are $n + m$ variables with m basic ones, the number of dictionaries is no more than $\binom{n+m}{m}$.[2])

[2] If $n = 77$ and $m = 9$ (diet's problem), this number is roughly 4.6×10^{11}.
[3] to choose entering and leaving variables

Proving the second theorem

The proof of the second theorem relies on two results.

- There are finitely many dictionaries. In fact, each dictionary is entirely characterized by which variables are the basic ones and which ones are the non basic ones.
  (If there are $n + m$ variables with m basic ones, the number of dictionaries is no more than $\binom{n+m}{m}$.[2])
- When we do successive pivots to optimality, we never go through the same dictionary twice: with a good rule[3] cycling never occurs.

---

[2] If $n = 77$ and $m = 9$ (diet's problem), this number is roughly 4.6×10^{11}.
[3] to choose entering and leaving variables

Proving the second theorem

The proof of the second theorem relies on two results.

- There are finitely many dictionaries. In fact, each dictionary is entirely characterized by which variables are the basic ones and which ones are the non basic ones.
  (If there are $n + m$ variables with m basic ones, the number of dictionaries is no more than $\binom{n+m}{m}$.[2])
- When we do successive pivots to optimality, we never go through the same dictionary twice: with a good rule[3] cycling never occurs.

Hence at some point the pivoting process must stop: it means that we cannot either find a entering variable or a leaving one.

[2] If $n = 77$ and $m = 9$ (diet's problem), this number is roughly 4.6×10^{11}.
[3] to choose entering and leaving variables

Why finitely many dictionaries?

Each dictionary is chacterized by the choice of the basic variables.

- Each dictionary is just a set of equations.
- The pivoting process does not modify the set of solutions of these equations because we only add or subtract one row to the others.

Why finitely many dictionaries?

Each dictionary is chacterized by the choice of the basic variables.

- Each dictionary is just a set of equations.
- The pivoting process does not modify the set of solutions of these equations because we only add or subtract one row to the others.
- Thus, assume that $\mathbf{x}_B$ is the vector of basic variables and $\mathbf{x}_N$ the one of non-basic ones. Assume that we have A and B two matrices such that

$$\mathbf{x}_B = A\mathbf{x}_N \quad \text{and} \quad \mathbf{x}_B = B\mathbf{x}_N$$

are two dictionaries with the same basic variables. As these equations are equivalent, it means that $(\mathbf{x}_B, \mathbf{x}_N)$ satisfy the first equation if and only if it satisfies the second one. In particular, for all $\mathbf{x}_N$ there holds

$$A\mathbf{x}_N = B\mathbf{x}_N.$$

This is enough to say that $A = B$, that is the coefficients of the two dictionaries are the same.

No cycling?

- *A priori* the value of z (when setting non basic variables to 0) strictly increases at each pivot, hence we could not go back to the same dictionary because it would imply the same value for z.

[4] Choosing the entering variable as the one with the smallest subscript, and similarly for the leaving one

No cycling?

- *A priori* the value of z (when setting non basic variables to 0) strictly increases at each pivot, hence we could not go back to the same dictionary because it would imply the same value for z.
- Some dictionaries are degenerate: setting non basic variables to 0 gives the value 0 to some basic variables.
- It leads to degenerate pivots where the value of z does not change. If we follow Anstee's rule, degeneracy can lead to cycling, see for instance `https://www.math.ubc.ca/~anstee/math340/340cycling.pdf`

[4] Choosing the entering variable as the one with the smallest subscript, and similarly for the leaving one

No cycling?

- *A priori* the value of z (when setting non basic variables to 0) strictly increases at each pivot, hence we could not go back to the same dictionary because it would imply the same value for z.
- Some dictionaries are degenerate: setting non basic variables to 0 gives the value 0 to some basic variables.
- It leads to degenerate pivots where the value of z does not change. If we follow Anstee's rule, degeneracy can lead to cycling, see for instance `https://www.math.ubc.ca/~anstee/math340/340cycling.pdf`
- Bland's rule[4] can be proved to avoid cycling even with degeneracy, hence leading to the termination of the simplex algorithm.

[4] Choosing the entering variable as the one with the smallest subscript, and similarly for the leaving one

No cycling?

- *A priori* the value of z (when setting non basic variables to 0) strictly increases at each pivot, hence we could not go back to the same dictionary because it would imply the same value for z.
- Some dictionaries are degenerate: setting non basic variables to 0 gives the value 0 to some basic variables.
- It leads to degenerate pivots where the value of z does not change. If we follow Anstee's rule, degeneracy can lead to cycling, see for instance `https://www.math.ubc.ca/~anstee/math340/340cycling.pdf`
- Bland's rule[4] can be proved to avoid cycling even with degeneracy, hence leading to the termination of the simplex algorithm.

As a remark, for practical problems **degeneracy** is a very common phenomenon but actual occurrences of **cycling** are very rare. Note that degeneracy means that (e.g. with Anstee's rule) we may cycle, not that we necessarily will.

[4]Choosing the entering variable as the one with the smallest subscript, and similarly for the leaving one