Number of Simplex Iterations

Number of Simplex Iterations

Each iteration of Simplex can be implemented in polynomial time.

Number of Simplex Iterations

Each iteration of Simplex can be implemented in polynomial time.

If we use lexicographic pivoting we know that Simplex requires at most $\binom{n}{m}$ iterations, because it will not visit a basis twice.

Number of Simplex Iterations

Each iteration of Simplex can be implemented in polynomial time.

If we use lexicographic pivoting we know that Simplex requires at most $\binom{n}{m}$ iterations, because it will not visit a basis twice.

The input size is $L \cdot n \cdot m$, where n is the number of variables, m is the number of constraints, and L is the length of the binary representation of the largest coefficient in the matrix A.

Number of Simplex Iterations

Each iteration of Simplex can be implemented in polynomial time.

If we use lexicographic pivoting we know that Simplex requires at most $\binom{n}{m}$ iterations, because it will not visit a basis twice.

The input size is $L \cdot n \cdot m$, where n is the number of variables, m is the number of constraints, and L is the length of the binary representation of the largest coefficient in the matrix A.

If we really require $\binom{n}{m}$ iterations then Simplex is not a polynomial time algorithm.

Number of Simplex Iterations

Each iteration of Simplex can be implemented in polynomial time.

If we use lexicographic pivoting we know that Simplex requires at most $\binom{n}{m}$ iterations, because it will not visit a basis twice.

The input size is $L \cdot n \cdot m$, where n is the number of variables, m is the number of constraints, and L is the length of the binary representation of the largest coefficient in the matrix A.

If we really require $\binom{n}{m}$ iterations then Simplex is not a polynomial time algorithm.

Can we obtain a better analysis?

Number of Simplex Iterations

Observation

Simplex visits every feasible basis at most once.

Number of Simplex Iterations

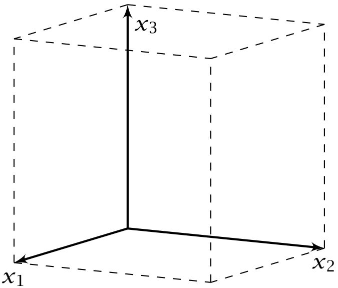
Observation

Simplex visits every feasible basis at most once.

However, also the number of feasible bases can be very large.

Example

$$\begin{aligned} \max\ & c^T x \\ \text{s.t.}\ & 0 \le x_1 \le 1 \\ & 0 \le x_2 \le 1 \\ & \quad\vdots \\ & 0 \le x_n \le 1 \end{aligned}$$

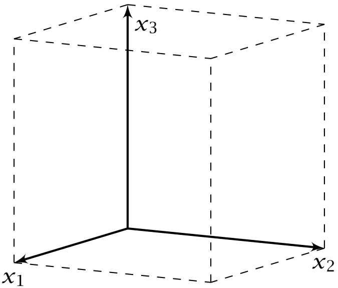


$2n$ constraint on n variables define an n-dimensional hypercube as feasible region.

The feasible region has 2^n vertices.

Example

$$\begin{aligned} \max\ & c^T x \\ \text{s.t.}\ & 0 \le x_1 \le 1 \\ & 0 \le x_2 \le 1 \\ & \vdots \\ & 0 \le x_n \le 1 \end{aligned}$$



However, Simplex may still run quickly as it usually does not visit all feasible bases.

In the following we give an example of a feasible region for which there is a bad Pivoting Rule.

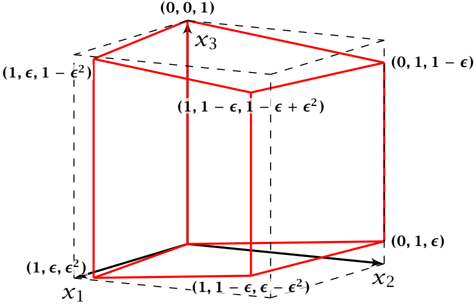
Pivoting Rule

A Pivoting Rule defines how to choose the entering and leaving variable for an iteration of Simplex.

In the non-degenerate case after choosing the entering variable the leaving variable is unique.

Klee Minty Cube

$$\begin{aligned} \max\ & x_n \\ \text{s.t.}\ & 0 \le x_1 \le 1 \\ & \epsilon x_1 \le x_2 \le 1 - \epsilon x_1 \\ & \epsilon x_2 \le x_3 \le 1 - \epsilon x_2 \\ & \vdots \\ & \epsilon x_{n-1} \le x_n \le 1 - \epsilon x_{n-1} \\ & x_i \ge 0 \end{aligned}$$



Observations

- We have $2n$ constraints, and $3n$ variables (after adding slack variables to every constraint).
- Every basis is defined by $2n$ variables, and n non-basic variables.
- There exist degenerate vertices.
- The degeneracies come from the non-negativity constraints, which are superfluous.
- In the following all variables x_i stay in the basis at all times.
- Then, we can uniquely specify a basis by choosing for each variable whether it should be equal to its lower bound, or equal to its upper bound (the slack variable corresponding to the non-tight constraint is part of the basis).
- We can also simply identify each basis/vertex with the corresponding hypercube vertex obtained by letting $\epsilon \to 0$.

Observations

- We have $2n$ constraints, and $3n$ variables (after adding slack variables to every constraint).
- Every basis is defined by $2n$ variables, and n non-basic variables.
- There exist degenerate vertices.
- The degeneracies come from the non-negativity constraints, which are superfluous.
- In the following all variables x_i stay in the basis at all times.
- Then, we can uniquely specify a basis by choosing for each variable whether it should be equal to its lower bound, or equal to its upper bound (the slack variable corresponding to the non-tight constraint is part of the basis).
- We can also simply identify each basis/vertex with the corresponding hypercube vertex obtained by letting $\epsilon \to 0$.

Observations

- We have $2n$ constraints, and $3n$ variables (after adding slack variables to every constraint).
- Every basis is defined by $2n$ variables, and n non-basic variables.
- There exist degenerate vertices.
- The degeneracies come from the non-negativity constraints, which are superfluous.
- In the following all variables x_i stay in the basis at all times.
- Then, we can uniquely specify a basis by choosing for each variable whether it should be equal to its lower bound, or equal to its upper bound (the slack variable corresponding to the non-tight constraint is part of the basis).
- We can also simply identify each basis/vertex with the corresponding hypercube vertex obtained by letting $\epsilon \to 0$.

Observations

- We have $2n$ constraints, and $3n$ variables (after adding slack variables to every constraint).
- Every basis is defined by $2n$ variables, and n non-basic variables.
- There exist degenerate vertices.
- The degeneracies come from the non-negativity constraints, which are superfluous.
- In the following all variables x_i stay in the basis at all times.
- Then, we can uniquely specify a basis by choosing for each variable whether it should be equal to its lower bound, or equal to its upper bound (the slack variable corresponding to the non-tight constraint is part of the basis).
- We can also simply identify each basis/vertex with the corresponding hypercube vertex obtained by letting $\epsilon \to 0$.

Observations

- We have $2n$ constraints, and $3n$ variables (after adding slack variables to every constraint).
- Every basis is defined by $2n$ variables, and n non-basic variables.
- There exist degenerate vertices.
- The degeneracies come from the non-negativity constraints, which are superfluous.
- In the following all variables x_i stay in the basis at all times.
- Then, we can uniquely specify a basis by choosing for each variable whether it should be equal to its lower bound, or equal to its upper bound (the slack variable corresponding to the non-tight constraint is part of the basis).
- We can also simply identify each basis/vertex with the corresponding hypercube vertex obtained by letting $\epsilon \to 0$.

Observations

- We have $2n$ constraints, and $3n$ variables (after adding slack variables to every constraint).
- Every basis is defined by $2n$ variables, and n non-basic variables.
- There exist degenerate vertices.
- The degeneracies come from the non-negativity constraints, which are superfluous.
- In the following all variables x_i stay in the basis at all times.
- Then, we can uniquely specify a basis by choosing for each variable whether it should be equal to its lower bound, or equal to its upper bound (the slack variable corresponding to the non-tight constraint is part of the basis).
- We can also simply identify each basis/vertex with the corresponding hypercube vertex obtained by letting $\epsilon \to 0$.

Observations

- We have $2n$ constraints, and $3n$ variables (after adding slack variables to every constraint).
- Every basis is defined by $2n$ variables, and n non-basic variables.
- There exist degenerate vertices.
- The degeneracies come from the non-negativity constraints, which are superfluous.
- In the following all variables x_i stay in the basis at all times.
- Then, we can uniquely specify a basis by choosing for each variable whether it should be equal to its lower bound, or equal to its upper bound (the slack variable corresponding to the non-tight constraint is part of the basis).
- We can also simply identify each basis/vertex with the corresponding hypercube vertex obtained by letting $\epsilon \to 0$.

Analysis

- In the following we specify a sequence of bases (identified by the corresponding hypercube node) along which the objective function strictly increases.
- The basis $(0, \ldots, 0, 1)$ is the unique optimal basis.
- Our sequence S_n starts at $(0, \ldots, 0)$ ends with $(0, \ldots, 0, 1)$ and visits every node of the hypercube.
- An unfortunate Pivoting Rule may choose this sequence, and, hence, require an exponential number of iterations.

Analysis

- In the following we specify a sequence of bases (identified by the corresponding hypercube node) along which the objective function strictly increases.
- The basis $(0, \dots, 0, 1)$ is the unique optimal basis.
- Our sequence S_n starts at $(0, \dots, 0)$ ends with $(0, \dots, 0, 1)$ and visits every node of the hypercube.
- An unfortunate Pivoting Rule may choose this sequence, and, hence, require an exponential number of iterations.

Analysis

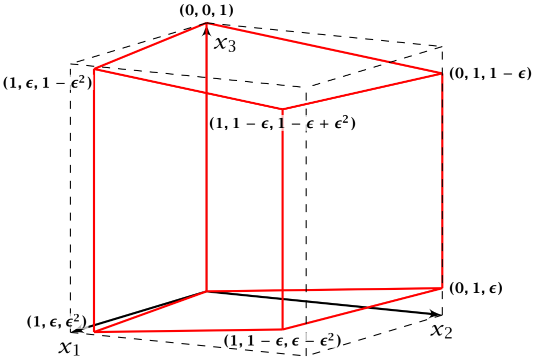
- In the following we specify a sequence of bases (identified by the corresponding hypercube node) along which the objective function strictly increases.
- The basis $(0, \ldots, 0, 1)$ is the unique optimal basis.
- Our sequence S_n starts at $(0, \ldots, 0)$ ends with $(0, \ldots, 0, 1)$ and visits every node of the hypercube.
- An unfortunate Pivoting Rule may choose this sequence, and, hence, require an exponential number of iterations.

Analysis

- In the following we specify a sequence of bases (identified by the corresponding hypercube node) along which the objective function strictly increases.
- The basis $(0, \ldots, 0, 1)$ is the unique optimal basis.
- Our sequence S_n starts at $(0, \ldots, 0)$ ends with $(0, \ldots, 0, 1)$ and visits every node of the hypercube.
- An unfortunate Pivoting Rule may choose this sequence, and, hence, require an exponential number of iterations.

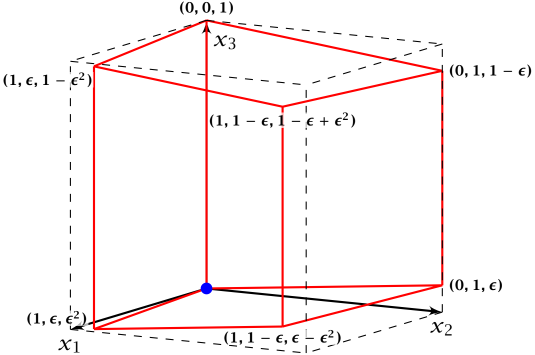
Klee Minty Cube

$$\begin{aligned} &\max\ x_n \\ &\quad \text{s.t.} \quad 0 \le x_1 \le 1 \\ &\qquad\qquad \epsilon x_1 \le x_2 \le 1 - \epsilon x_1 \\ &\qquad\qquad \epsilon x_2 \le x_3 \le 1 - \epsilon x_2 \end{aligned}$$



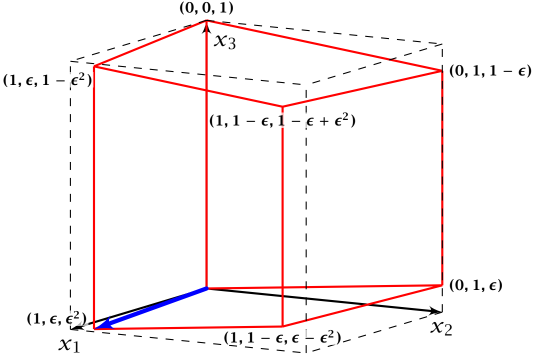
Klee Minty Cube

$$\begin{aligned} \max\ & x_n \\ \text{s.t.}\quad & 0 \le x_1 \le 1 \\ & \epsilon x_1 \le x_2 \le 1 - \epsilon x_1 \\ & \epsilon x_2 \le x_3 \le 1 - \epsilon x_2 \end{aligned}$$



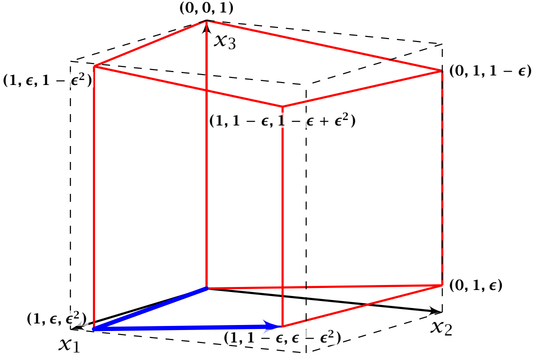
Klee Minty Cube

$$\begin{aligned} \max\ & x_n \\ \text{s.t.}\quad & 0 \le x_1 \le 1 \\ & \epsilon x_1 \le x_2 \le 1 - \epsilon x_1 \\ & \epsilon x_2 \le x_3 \le 1 - \epsilon x_2 \end{aligned}$$



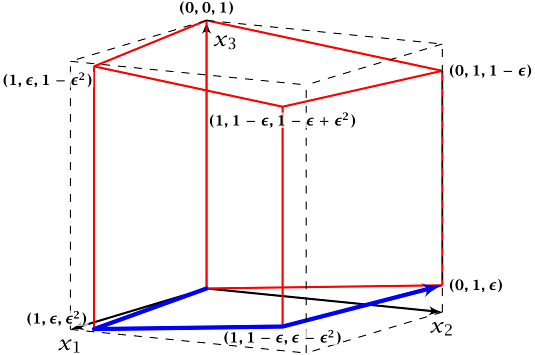
Klee Minty Cube

$$\begin{aligned} \max\ & x_n \\ \text{s.t.}\quad & 0 \le x_1 \le 1 \\ & \epsilon x_1 \le x_2 \le 1 - \epsilon x_1 \\ & \epsilon x_2 \le x_3 \le 1 - \epsilon x_2 \end{aligned}$$



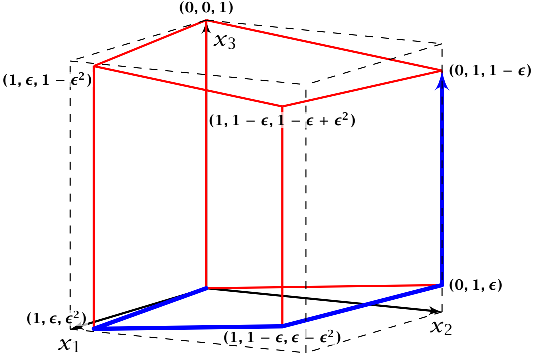
Klee Minty Cube

$$\begin{aligned} \max\ & x_n \\ \text{s.t.}\ & 0 \le x_1 \le 1 \\ & \epsilon x_1 \le x_2 \le 1 - \epsilon x_1 \\ & \epsilon x_2 \le x_3 \le 1 - \epsilon x_2 \end{aligned}$$



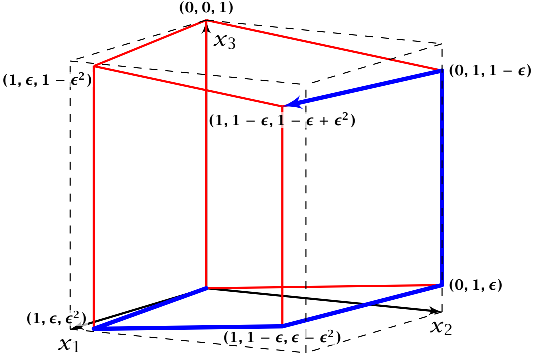
Klee Minty Cube

$$\begin{aligned} &\max\ x_n \\ &\text{s.t.} \quad 0 \le x_1 \le 1 \\ &\qquad \epsilon x_1 \le x_2 \le 1 - \epsilon x_1 \\ &\qquad \epsilon x_2 \le x_3 \le 1 - \epsilon x_2 \end{aligned}$$



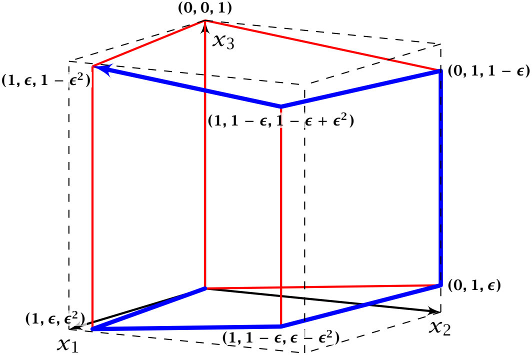
Klee Minty Cube

$$\begin{aligned} &\max x_n \\ &\text{s.t.} \quad 0 \le x_1 \le 1 \\ &\qquad \epsilon x_1 \le x_2 \le 1 - \epsilon x_1 \\ &\qquad \epsilon x_2 \le x_3 \le 1 - \epsilon x_2 \end{aligned}$$



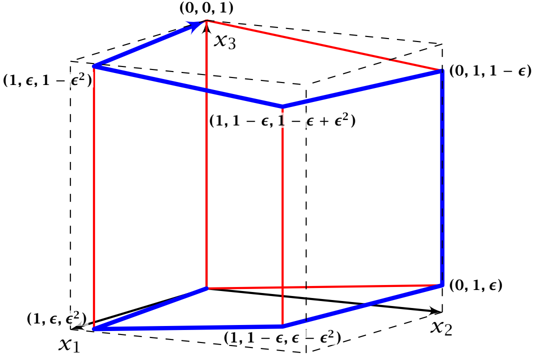
Klee Minty Cube

$$\max\ x_n$$
$$\begin{aligned} \text{s.t.} \quad 0 &\le x_1 \le 1 \\ \epsilon x_1 &\le x_2 \le 1 - \epsilon x_1 \\ \epsilon x_2 &\le x_3 \le 1 - \epsilon x_2 \end{aligned}$$



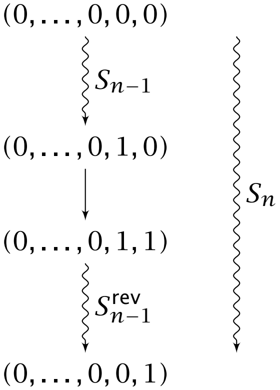
Klee Minty Cube

$$
\begin{aligned}
\max\ & x_n \\
\text{s.t.}\quad & 0 \le x_1 \le 1 \\
& \epsilon x_1 \le x_2 \le 1 - \epsilon x_1 \\
& \epsilon x_2 \le x_3 \le 1 - \epsilon x_2
\end{aligned}
$$



Analysis

The sequence S_n that visits every node of the hypercube is defined recursively



The non-recursive case is $S_1 = 0 \to 1$

Analysis

Lemma 2

The objective value x_n is increasing along path S_n.

Proof by induction:

$n = 1$: obvious, since $S_1 = 0 \to 1$, and $1 > 0$.

$n - 1 \to n$

- For the first part the value of $x_n = \epsilon x_{n-1}$.
- By induction hypothesis x_{n-1} is increasing along S_{n-1}, hence, also x_n.
- Going from $(0, \ldots, 0, 1, 0)$ to $(0, \ldots, 0, 1, 1)$ increases x_n for small enough ϵ.
- For the remaining path S_{n-1}^{rev} we have $x_n = 1 - \epsilon x_{n-1}$.
- By induction hypothesis x_{n-1} is increasing along S_{n-1}, hence $-\epsilon x_{n-1}$ is increasing along S_{n-1}^{rev}.

Analysis

Lemma 2

The objective value x_n is increasing along path S_n.

Proof by induction:

$n = 1$: obvious, since $S_1 = 0 \to 1$, and $1 > 0$.

$n - 1 \to n$

- For the first part the value of $x_n = \epsilon x_{n-1}$.
- By induction hypothesis x_{n-1} is increasing along S_{n-1}, hence, also x_n.
- Going from $(0, \ldots, 0, 1, 0)$ to $(0, \ldots, 0, 1, 1)$ increases x_n for small enough ϵ.
- For the remaining path S_{n-1}^{rev} we have $x_n = 1 - \epsilon x_{n-1}$.
- By induction hypothesis x_{n-1} is increasing along S_{n-1}, hence $-\epsilon x_{n-1}$ is increasing along S_{n-1}^{rev}.

Analysis

Lemma 2

The objective value x_n is increasing along path S_n.

Proof by induction:

$\boldsymbol{n = 1}$: obvious, since $S_1 = 0 \to 1$, and $1 > 0$.

$\boldsymbol{n - 1 \to n}$

- For the first part the value of $x_n = \epsilon x_{n-1}$.
- By induction hypothesis x_{n-1} is increasing along S_{n-1}, hence, also x_n.
- Going from $(0, \ldots, 0, 1, 0)$ to $(0, \ldots, 0, 1, 1)$ increases x_n for small enough ϵ.
- For the remaining path S_{n-1}^{rev} we have $x_n = 1 - \epsilon x_{n-1}$.
- By induction hypothesis x_{n-1} is increasing along S_{n-1}, hence $-\epsilon x_{n-1}$ is increasing along S_{n-1}^{rev}.

Analysis

Lemma 2

The objective value x_n is increasing along path S_n.

Proof by induction:

$\boldsymbol{n = 1}$: obvious, since $S_1 = 0 \to 1$, and $1 > 0$.

$\boldsymbol{n - 1 \to n}$

- For the first part the value of $x_n = \epsilon x_{n-1}$.
- By induction hypothesis x_{n-1} is increasing along S_{n-1}, hence, also x_n.
- Going from $(0, \ldots, 0, 1, 0)$ to $(0, \ldots, 0, 1, 1)$ increases x_n for small enough ϵ.
- For the remaining path S_{n-1}^{rev} we have $x_n = 1 - \epsilon x_{n-1}$.
- By induction hypothesis x_{n-1} is increasing along S_{n-1}, hence $-\epsilon x_{n-1}$ is increasing along S_{n-1}^{rev}.

Analysis

Lemma 2
The objective value x_n is increasing along path S_n.

Proof by induction:

$n = 1$: obvious, since $S_1 = 0 \to 1$, and $1 > 0$.

$n - 1 \to n$

- For the first part the value of $x_n = \epsilon x_{n-1}$.
- By induction hypothesis x_{n-1} is increasing along S_{n-1}, hence, also x_n.
- Going from $(0, \ldots, 0, 1, 0)$ to $(0, \ldots, 0, 1, 1)$ increases x_n for small enough ϵ.
- For the remaining path S^{rev}_{n-1} we have $x_n = 1 - \epsilon x_{n-1}$.
- By induction hypothesis x_{n-1} is increasing along S_{n-1}, hence $-\epsilon x_{n-1}$ is increasing along S^{rev}_{n-1}.

Analysis

Lemma 2

The objective value x_n is increasing along path S_n.

Proof by induction:

$\boldsymbol{n = 1}$: obvious, since $S_1 = 0 \to 1$, and $1 > 0$.

$\boldsymbol{n - 1 \to n}$

- For the first part the value of $x_n = \epsilon x_{n-1}$.
- By induction hypothesis x_{n-1} is increasing along S_{n-1}, hence, also x_n.
- Going from $(0, \dots, 0, 1, 0)$ to $(0, \dots, 0, 1, 1)$ increases x_n for small enough ϵ.
- For the remaining path S_{n-1}^{rev} we have $x_n = 1 - \epsilon x_{n-1}$.
- By induction hypothesis x_{n-1} is increasing along S_{n-1}, hence $-\epsilon x_{n-1}$ is increasing along S_{n-1}^{rev}.

Analysis

Lemma 2

The objective value x_n is increasing along path S_n.

Proof by induction:

$n = 1$: obvious, since $S_1 = 0 \to 1$, and $1 > 0$.

$n - 1 \to n$

- For the first part the value of $x_n = \epsilon x_{n-1}$.
- By induction hypothesis x_{n-1} is increasing along S_{n-1}, hence, also x_n.
- Going from $(0, \dots, 0, 1, 0)$ to $(0, \dots, 0, 1, 1)$ increases x_n for small enough ϵ.
- For the remaining path S_{n-1}^{rev} we have $x_n = 1 - \epsilon x_{n-1}$.
- By induction hypothesis x_{n-1} is increasing along S_{n-1}, hence $-\epsilon x_{n-1}$ is increasing along S_{n-1}^{rev}.

Analysis

Lemma 2

The objective value x_n is increasing along path S_n.

Proof by induction:

$\boldsymbol{n = 1}$: obvious, since $S_1 = 0 \to 1$, and $1 > 0$.

$\boldsymbol{n - 1 \to n}$

- For the first part the value of $x_n = \epsilon x_{n-1}$.
- By induction hypothesis x_{n-1} is increasing along S_{n-1}, hence, also x_n.
- Going from $(0, \ldots, 0, 1, 0)$ to $(0, \ldots, 0, 1, 1)$ increases x_n for small enough ϵ.
- For the remaining path S_{n-1}^{rev} we have $x_n = 1 - \epsilon x_{n-1}$.
- By induction hypothesis x_{n-1} is increasing along S_{n-1}, hence $-\epsilon x_{n-1}$ is increasing along S_{n-1}^{rev}.

Remarks about Simplex

Observation

The simplex algorithm takes at most $\binom{n}{m}$ iterations. Each iteration can be implemented in time $\mathcal{O}(mn)$.

In practise it usually takes a linear number of iterations.

Remarks about Simplex

Theorem
For almost all known deterministic pivoting rules (rules for choosing entering and leaving variables) there exist lower bounds that require the algorithm to have exponential running time ($\Omega(2^{\Omega(n)})$) (e.g. Klee Minty 1972).

Remarks about Simplex

Theorem

For some standard randomized pivoting rules there exist subexponential lower bounds ($\Omega(2^{\Omega(n^\alpha)})$ for $\alpha > 0$) (Friedmann, Hansen, Zwick 2011).

Remarks about Simplex

Conjecture (Hirsch 1957)
The edge-vertex graph of an m-facet polytope in d-dimensional Euclidean space has diameter no more than $m - d$.

The conjecture has been proven wrong in 2010.

But the question whether the diameter is perhaps of the form $\mathcal{O}(\text{poly}(m, d))$ is open.