A Pivot based approach to Vertex Enumeration of Polytopes and Polyhedra

Tilman Hinnerichs

Hauptseminar „Selected Topics in Logic and Verification“ – TU Dresden

July 28, 2020

The paper

„A Pivoting Algorithm for Convex Hulls and Vertex Enumeration of Arrangements and Polyhedra“

Authors: David Avis and Komei Fukuda
Publication year: 1992
Journal: Discrete & Computational Geometry

A Pivot based approach to Vertex Enumeration of Polytopes and Polyhedra 2 / 29

The paper

„A Pivoting Algorithm for Convex Hulls and Vertex Enumeration of Arrangements and Polyhedra“

Authors: David Avis and Komei Fukuda
Publication year: 1992
Journal: Discrete & Computational Geometry
Cited 751 times (according to Google Scholar)

A Pivot based approach to Vertex Enumeration of Polytopes and Polyhedra
2 / 29

Outline

1. The Vertex Enumeration Problem
2. The SIMPLEX Algorithm
3. From pivot step to VEP solution
   - Degeneracy of Vertices
   - Bland's rule
   - Avis-Fukuda Algoritm

The Vertex Enumeration Problem

Definition (Convex Polyhedron)

Given a matrix $A = (a_{ij}) \in \mathbb{R}^{m \times n}$ and a vector $b \in \mathbb{R}^m$.
A *convex polyhedron* P is defined as

$$P = \{x \in \mathbb{R}^n : Ax + b \geq 0\} \tag{1}$$

The Vertex Enumeration Problem

Definition (Convex Polyhedron)

Given a matrix $A = (a_{ij}) \in \mathbb{R}^{m \times n}$ and a vector $b \in \mathbb{R}^m$.
A *convex polyhedron* P is defined as

$$P = \{x \in \mathbb{R}^n : Ax + b \geq 0\} \tag{1}$$

Definition (Polyhedron Vertex)

A point $x \in P$ is a *vertex* of P **iff**
it cannot be written as a convex combination of two other solutions **iff**
it is the unique solution to a subset of m inequalities solved as equations.

The Vertex Enumeration Problem (VEP)

Definition (Vertex Enumeration Problem)

For a given polytope/polyhedron, hyperplane arrangement P return all vertices of P.

[1] Khachiyan et al. [2008]

The Vertex Enumeration Problem (VEP)

Definition (Vertex Enumeration Problem)

For a given polytope/polyhedron, hyperplane arrangement P return all vertices of P.

Theorem[1]

The VEP is NP-complete.

[1] Khachiyan et al. [2008]

An Example Problem

Example: All meals for 1 Euro

Var	Food type	Energy (kcal/100g)	Protein (g)	Calcium (mg)	Price (Euro cents)	Maximum
x_1	Oatmeal[2]	250	15	40	10	500g
x_2	Milk[3]	80	4	130	4	1000g
x_3	DD Stollen[4]	340	7	30	20	200g
	Min. daily	**2000**	**55**	**800**		

[2] Regular LIDL oatmeal
[3] Regular Bio LIDL milk
[4] https://fddb.info

Putting data into inequalities

Example: Meals for 1 Euro

$$A = \begin{bmatrix} 350 & 80 & 340 \\ 15 & 4 & 7 \\ 40 & 130 & 30 \\ -10 & -4 & -20 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, -b = \begin{bmatrix} 2000 \\ 55 \\ 800 \\ -100 \\ -5 \\ -10 \\ -2 \end{bmatrix}, \begin{pmatrix} Energy \\ Protein \\ Calcium \\ Price \\ x_1 Max \\ x_2 Max \\ x_3 Max \end{pmatrix}$$

Polytope visualized

Example: All meals for 1 Euro

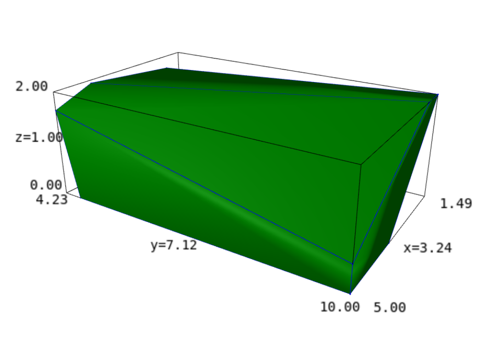


Figure: The exact shape is unknown/expensive

An Example Problem

- Use convex combination to yield infinitely many food combinations

An Example Problem

- Use convex combination to yield infinitely many food combinations

Definition (Convex combination)

A convex combination is a linear combination of points.
Let $x_1, x_2, \ldots, x_n$ be a finite set of points. A convex combination is a point of the form

$$y = a_1 x_1 + a_2 x_2 + \cdots + a_n x_n$$

while $a_1, \ldots, a_n \in \mathbb{R}^+$, and

$$a_1 + \cdots + a_n = 1$$

Linear optimization problems

Definition

Let $G \subseteq \mathbb{R}^n$ be a polyhedron and $c \in \mathbb{R}^n$. Then a problem of the form

$$z = f(x) = c^T \cdot x \to min \qquad with\ x \in G$$

is called a *linear* optimization problem.

Linear optimization problems

Definition

Let $G \subseteq \mathbb{R}^n$ be a polyhedron and $c \in \mathbb{R}^n$. Then a problem of the form

$$z = f(x) = c^T \cdot x \rightarrow min \qquad with\ x \in G$$

is called a *linear* optimization problem.

Protein rich meals for 1 Euro

We are trying to maximize our protein intake:

$$c = \begin{bmatrix} 15 \\ 4 \\ 7 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$z = c^T x \rightarrow max$$

The SIMPLEX Algorithm

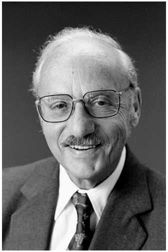


Figure: George Dantzig (1914 - 2005)

Algorithm 1 The SIMPLEX Algorithm

1: Find a first feasible vertex
2: Calculate the optimal vertex

From $\geq$ to $=$

The inequalities

$$Ax + b \geq 0$$

describing P are transformed to the equivalent equality system

$$Mx + b = x_p$$

with $x \in \mathbb{R}_+^n$, $x_p \in \mathbb{R}_+^m$, and $M \in \mathbb{R}^{m \times n}$ (if $x \geq 0$)

From $\geq$ to $=$

The inequalities

$$Ax + b \geq 0$$

describing P are transformed to the equivalent equality system

$$Mx + b = x_p$$

with $x \in \mathbb{R}^n_+$, $x_p \in \mathbb{R}^m_+$, and $M \in \mathbb{R}^{m \times n}$ (if $x \geq 0$)

Definition (Tableau form)

A linear program of the form

$$z = f'(x') = c'^T \cdot x' \to min \qquad \text{with } Mx_N + b = x_B \text{ with } \begin{array}{l} x_N := x, \\ x_B := x_p, \\ x' \leftrightarrow \begin{pmatrix} x_N \\ x_B \end{pmatrix} \\ b \geq 0 \end{array}$$

is said to be in **tableau form**.

Another example

$$-z = -30x_1 - 45x_2 \rightarrow min$$

$$\text{with} \quad \begin{array}{rcrcrcr} x_3 & = & -4x_1 & - & 3x_2 & + & 100 \\ x_4 & = & -x_1 & - & 2x_2 & + & 50 \\ x_3, x_4 & , & x_1 & , & x_2 & \geq & 0 \end{array}$$

non-basic variables, with indices $N = \{1, 2\}, |N| = n$
basic variables, with indices $B = \{3, 4\}, |B| = m$

SIMPLEX-Tableau:

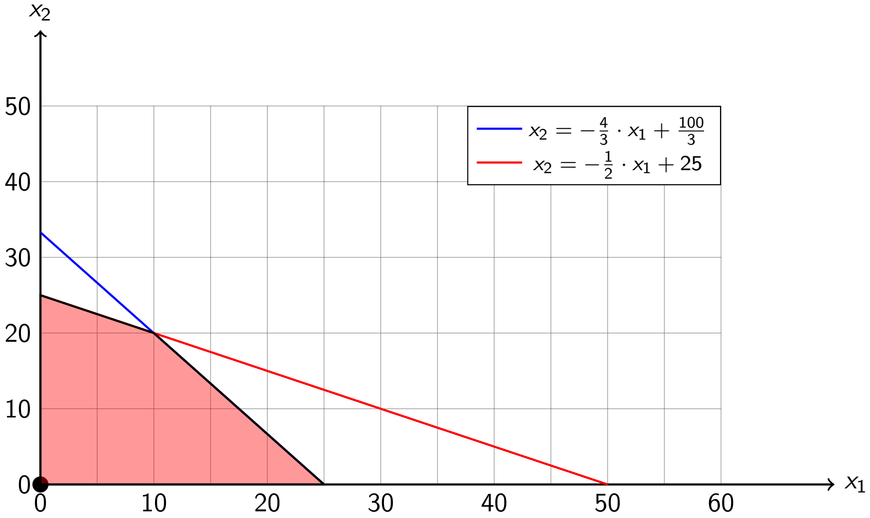
ST_0	x_1	x_2	1
$x_3 =$	-4	-3	100
$x_4 =$	-1	-2	50
$-z =$	-30	-45	0

ST_0	x_N	1
$x_B =$	M	p
$z =$	q^T	q_0

ST_0	x_N	1
$x_B =$	M	p
$z =$	q^T	q_0

x_N results from x and N
x_B results from x and B
M results from A
$p := b$
$q := c'$
$q_0 := 0$

Visualization of new polytope P



Evaluating the SIMPLEX tableau

ST_0	x_1	x_2	1
$x_3 =$	-4	-3	100
$x_4 =$	-1	-2	50
$-z =$	-30	-45	0

Phase 2: Determine optimal vertex

Algorithm 2 Phase 2: Determine optimal vertex

1: **while** $\tau \in N$ with $q_\tau < 0$ exists **do**
2: Choose PIVOT element $M_{\sigma\tau}$
3: Calculate the new tableau under usage of swapping rules
4: **end while**

Phase 2: Determine optimal vertex

Algorithm 3 Phase 2: Determine optimal vertex

1: **while** $\tau \in N$ with $q_\tau < 0$ exists **do**
2: Choose PIVOT element $M_{\sigma\tau}$
3: Calculate the new tableau under usage of swapping rules
4: **end while**

Theorem

The SIMPLEX tableau is optimal **iff** $p \geq 0$ and $q \geq 0$

Choosing a pivot element

ST_0	x_N	1
$x_B =$	P	p
$z =$	q^T	q_0

We choose an arbitrary $\tau \in N$ with $q_\tau < 0$, and calculate
$-\frac{p_\sigma}{P_{\sigma\tau}} := \min\left\{-\frac{p_i}{P_{i\tau}} \mid -\frac{p_i}{P_{i\tau}} < 0, i \in B\right\}$:

ST_0	x_1	x_2	1
$x_3 =$	-4	-3	100
$x_4 =$	-1	-2	50
$-z =$	-30	-45	0

$\implies \tau := 2, \sigma := 4$

ST_0	x_1	x_2	1
$x_3 =$	-4	-3	100
$x_4 =$	-1	-2	50
$-z =$	-30	-45	0

ST_0	x_1	x_2	1
$x_3 =$	-4	-3	100
$x_4 =$	-1	$\boxed{-2}$	50
$-z =$	-30	-45	0

ST_1	x_1	x_4	1
$x_3 =$	$-\frac{5}{2}$	$\frac{3}{2}$	25
$x_2 =$	$-\frac{1}{2}$	$-\frac{1}{2}$	25
$-z =$	$-\frac{15}{2}$	$\frac{45}{2}$	-1125

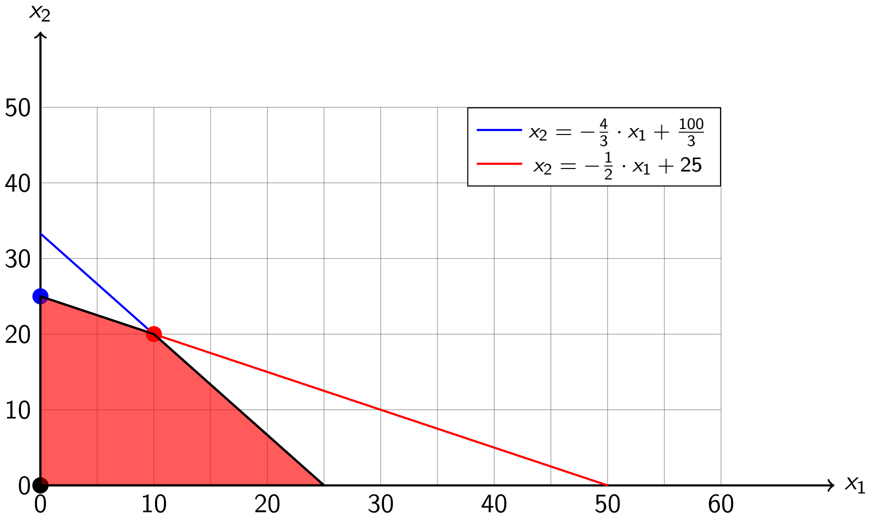
ST_0	x_1	x_2	1
$x_3 =$	-4	-3	100
$x_4 =$	-1	-2	50
$-z =$	-30	-45	0

ST_1	x_1	x_4	1
$x_3 =$	$-\frac{5}{2}$	$\frac{3}{2}$	25
$x_2 =$	$-\frac{1}{2}$	$-\frac{1}{2}$	25
$-z =$	$-\frac{15}{2}$	$\frac{45}{2}$	-1125

ST_0	x_1	x_2	1
$x_3 =$	-4	-3	100
$x_4 =$	-1	-2	50
$-z =$	-30	-45	0

ST_1	x_1	x_4	1
$x_3 =$	$-\frac{5}{2}$	$\frac{3}{2}$	25
$x_2 =$	$-\frac{1}{2}$	$-\frac{1}{2}$	25
$-z =$	$-\frac{15}{2}$	$\frac{45}{2}$	-1125

ST_2	x_3	x_4	1
$x_1 =$	$-\frac{2}{5}$	$\frac{3}{5}$	10
$x_2 =$	$\frac{1}{5}$	$-\frac{8}{10}$	20
$-z =$	3	18	-1200


x_2
x_1
0
10
20
30
40
50
60
$x_2 = -\frac{4}{3} \cdot x_1 + \frac{100}{3}$
$x_2 = -\frac{1}{2} \cdot x_1 + 25$

Some thoughts

1. What if we choose a pair of indices different from the optimal one in a simplex step?

Some thoughts

1. What if we choose a pair of indices different from the optimal one in a simplex step?
2. How does this help to build a solution for the VEP?

Degeneracy of Vertices

Definition (Degeneracy)

Let $P \subseteq \mathbb{R}^d$.
We will call a vertex A that satisfies $d+1$ given inequalities with equality **degenerate**.

Degeneracy of Vertices

Definition (Degeneracy)

Let $P \subseteq \mathbb{R}^d$.
We will call a vertex A that satisfies $d+1$ given inequalities with equality **degenerate**.

Example

ST_0	x_1	x_2	1
$x_3 =$	-4	-3	0
$x_4 =$	-1	-2	50
$-z =$	-30	-45	0

A Visual Example

Example

Example: From Avis' lecture notes

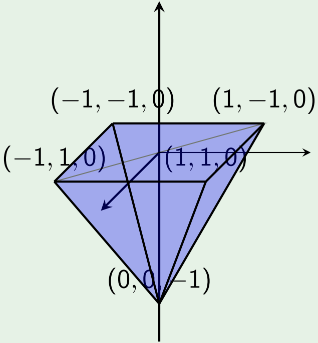


Figure: P is a regular pyramid

Properties of the SIMPLEX algorithm

- Does not always terminate

Properties of the SIMPLEX algorithm

- Does not always terminate

Definition (Bland's rule)

During a Simplex step, if several options are available choose the lowest-numbered non-basic column with negative (reduced) cost and the lowest-numbered basic variable.

Properties of the SIMPLEX algorithm

- Does not always terminate

Definition (Bland's rule)

During a Simplex step, if several options are available choose the lowest-numbered non-basic column with negative (reduced) cost and the lowest-numbered basic variable.

Theorem

For a given basic feasible solution (BFS) v Bland's rule deterministically determines a path from v along the vertices to the optimal vertex v^*.

Example: Meals for 1 Euro

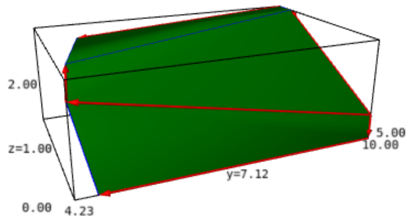


Figure: The paths span a tree of all vertices

Algorithm: Avis-Fukuda 1992

Idea:

- reverse search this spanning tree starting from the root

Algorithm: Avis-Fukuda 1992

Idea:

- reverse search this spanning tree starting from the root

Definition

Let A be a basic feasible solution (vertex).
Let (τ, σ) be the pivot obtained by applying Bland's rule to A yielding A'.
We call (σ, τ) the *reverse pivot* for A'.

Algorithm: Avis-Fukuda 1992

Idea:

- reverse search this spanning tree starting from the root

Definition

Let A be a basic feasible solution (vertex).
Let (τ, σ) be the pivot obtained by applying Bland's rule to A yielding A'.
We call (σ, τ) the *reverse pivot* for A'.

Definition (Neighbour)

$$Adj(N, i, j) := \begin{cases} \emptyset & M_{ij} = 0 \\ \emptyset & (N \cup \{i\}) - \{j\} \text{ infeasible dictionary} \\ (N \cup \{i\}) - \{j\} & \text{feasible dictionary} \end{cases}$$

Algorithm: Avis-Fukuda 1992

Problem

For any given vertex/BFS A we know its successor A'.
When to introduce an edge in the tree between A and any other neighbour $Adj(A_N, i, j)$ for $i \in A_B, j \in A_N$?

Algorithm: Avis-Fukuda 1992

Problem

For any given vertex/BFS A we know its successor A'.
When to introduce an edge in the tree between A and any other neighbour $Adj(A_N, i, j)$ for $i \in A_B, j \in A_N$?

- Perform conditional depth first search

Algorithm: Avis-Fukuda 1992

Algorithm 4 Phase 2: Determine optimal vertex

```
SEARCH(B,N,M):
for all i, j in NEIGHBOURS(B, N, M) do
    if REVERSE(B, N, M, i, j) then
        PIVOT(B, N, M, i, j)
        if LEX_MIN(B, N, M) then
            print(B)
        end if
        SEARCH(B, N, M)
        SELECT-PIVOT(M)
        PIVOT(B, N, M, i, j)
    end if
end for
```

Complexity and features

The algorithm has the following properties:

- Virtually no additional storage is required beyond the input data
- The output list is free of duplicates
- The algorithm is extremely simple, requires no data structures, and handles all degenerate cases
- The running time is output sensitive for nondegenerate cases
- The algorithm is easy to parallelize efficiently

Complexity and features

The algorithm has the following properties:

- Virtually no additional storage is required beyond the input data
- The output list is free of duplicates
- The algorithm is extremely simple, requires no data structures, and handles all degenerate cases
- The running time is output sensitive for nondegenerate cases
- The algorithm is easy to parallelize efficiently

Complexity (v vertices of $P \subseteq \mathbb{R}^d$, n inequalities):

- in time $\mathcal{O}(ndv)$
- in space $\mathcal{O}(nd)$

Sources

- L. Khachiyan, E. Boros, K. Borys, K. Elbassioni, and V. Gurvich. Generating all verticesof a polyhedron is hard.Discrete and Computational Geometry, 39(1):174–190, 2008.
- D. Avis, K. Fukuda. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra . Discrete Comput. Geom., 8 (1992), pp. 295-313
- Fukuda lecture nodes at Tokio University
- Avis lecture nodes at Montreal University

Approaches to the VEP

1. Double-description/Motzkin-based method
2. Pivot-based methods
3. Newer methods

A visual example

Example: From Avis' lecture notes

$$A = \begin{bmatrix} -1 & 0 & 1 \\ 0 & -1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & -1 \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, -b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \begin{aligned} 1 - x_1 + x_3 &\geq 0 \\ 1 - x_2 + x_3 &\geq 0 \\ 1 + x_1 + x_3 &\geq 0 \\ 1 + x_2 + x_3 &\geq 0 \\ -x_3 &\geq 0 \end{aligned}$$

A visual example

Example

Example: From Avis' lecture notes

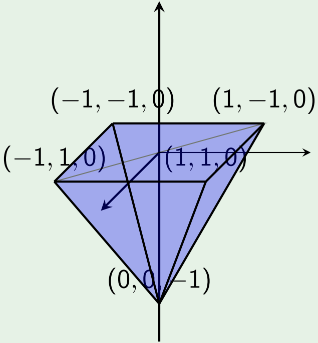


Figure: P is a regular pyramid