CS156: The Calculus of Computation

Zohar Manna

Winter 2008

Chapter 8: Quantifier-free Linear Arithmetic

Decision Procedures for Quantifier-free Fragments

For theory T with signature Σ and axioms $\mathcal{A}$, decide if

$$F[x_1, \ldots, x_n] \quad \text{or} \quad \exists x_1, \ldots, x_n.\ F[x_1, \ldots, x_n] \quad \text{is} \quad T\text{-satisfiable}$$

$$\left[\begin{array}{l}\text{Decide if} \\ \qquad F[x_1, \ldots, x_n] \quad \text{or} \quad \forall x_1, \ldots, x_n.\ F[x_1, \ldots, x_n] \quad \text{is} \quad T\text{-valid}\end{array}\right]$$

where F is quantifier-free and $\text{free}(F) = \{x_1, \ldots, x_n\}$

Note: no quantifier alternations

Conjunctive Quantifier-free Fragment

We consider only conjunctive quantifier-free Σ-formulae, i.e., conjunctions of Σ-literals (Σ-atoms or negations of Σ-atoms).

For given arbitrary quantifier-free Σ-formula F, convert it into DNF Σ-formula

$$F_1 \vee \ldots \vee F_k$$

where each F_i conjunctive.

F is T-satisfiable iff at least one F_i is T-satisfiable.

Preliminary Concepts

Vector

variable n-vector

$$\overline{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

n-vector $\overline{a} \in \mathbb{Q}^n$

$$\overline{a} = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}$$

transpose

$$\overline{a}^{\mathsf{T}} = \begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix}$$

Matrix

$m \times n$-matrix

$A \in \mathbb{Q}^{m \times n}$

$$A = \begin{bmatrix} a_{11} \cdots a_{1n} \\ \vdots \quad \ddots \quad \vdots \\ a_{m1} \cdots a_{mn} \end{bmatrix}$$

transpose

$$A^{\mathsf{T}} = \begin{bmatrix} a_{11} \cdots a_{m1} \\ \vdots \quad \ddots \quad \vdots \\ a_{1n} \cdots a_{mn} \end{bmatrix}$$

column

row

$$\begin{bmatrix} & a_{1j} & \\ & \vdots & \\ a_{i1} \cdots & a_{ij} & \cdots a_{in} \\ & \vdots & \\ & a_{mj} & \end{bmatrix}$$

Multiplication I

vector-vector

$$\overline{a}^{\mathsf{T}}\overline{b} = [a_1 \;\; \cdots \;\; a_n] \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} = \sum_{i=1}^{n} a_i b_i$$

matrix-vector

$$A\overline{x} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} a_{1i} x_i \\ \vdots \\ \sum_{i=1}^{n} a_{mi} x_i \end{bmatrix}$$

Multiplication II

matrix-matrix

$$\underbrace{\begin{bmatrix} & \vdots & \\ \cdots & a_{ik} & \cdots \\ & \vdots & \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} & \vdots & \\ \cdots & b_{kj} & \cdots \\ & \vdots & \end{bmatrix}}_{B} = \underbrace{\begin{bmatrix} & \vdots & \\ \cdots & p_{ij} & \cdots \\ & \vdots & \end{bmatrix}}_{P}$$

where

$$p_{ij} = \bar{a}_i \bar{b}_j = \begin{bmatrix} a_{i1} & \cdots & a_{in} \end{bmatrix} \begin{bmatrix} b_{1j} \\ \vdots \\ b_{nj} \end{bmatrix} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

Special Vectors and Matrices

$\overline{0}$ - vector (column) of 0s

$\overline{1}$ - vector of 1s

Thus $\overline{1}^{\mathsf{T}}\overline{x} = \sum_{i=1}^{n} x_i$

$$I = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix}$$ identity matrix ($n \times n$)

Thus $IA = AI = A$, for $n \times n$ matrix A.

unit vector $e_i = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \longleftarrow$ ith (Note: matrix indices start at 1)

Vector Space - set S of vectors closed under addition and scaling of vectors. That is,

$$\text{if } \overline{v}_1, \ldots, \overline{v}_k \in S \quad \text{then} \quad \begin{array}{l} \lambda_1 \overline{v}_1 + \cdots + \lambda_k \overline{v}_k \in S \\ \text{for } \lambda_1, \ldots, \lambda_n \in \mathbb{Q} \end{array}$$

Linear Equation

$$F \; : \; A\overline{x} \;=\; \overline{b}$$

$m \times n$-matrix   variable n-vector m-vector

represents the $\Sigma_{\mathbb{Q}}$-formula

$$F : (a_{11}x_1 + \cdots + a_{1n}x_n = b_1) \wedge \cdots \wedge (a_{m1}x_1 + \cdots + a_{mn}x_n = b_m)$$

Gaussian Elimination

Find $\overline{x}$ s.t. $A\overline{x} = \overline{b}$ by elementary row operations

- Swap two rows
- Multiply a row by a nonzero scalar
- Add one row to another

Example 4 I

Solve

$$\begin{bmatrix} 3 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 6 \\ 1 \\ 2 \end{bmatrix}$$

Construct the augmented matrix

$$\left[\begin{array}{ccc|c} 3 & 1 & 2 & 6 \\ 1 & 0 & 1 & 1 \\ 2 & 2 & 1 & 2 \end{array}\right]$$

Apply the row operations as follows:

Example 4 II

1. Add $-2\bar{a}_1 + 4\bar{a}_2$ to $\bar{a}_3$

$$\left[\begin{array}{ccc|c} 3 & 1 & 2 & 6 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & -6 \end{array}\right]$$

2. Add $-\bar{a}_1 + 2\bar{a}_2$ to $\bar{a}_2$

$$\left[\begin{array}{ccc|c} 3 & 1 & 2 & 6 \\ 0 & -1 & 1 & -3 \\ 0 & 0 & 1 & -6 \end{array}\right]$$

This augmented matrix is in triangular form.

Example 4 III

Solving

$$\begin{aligned} & & x_3 &= -6 \\ -x_2 + x_3 = -3 & \Rightarrow & x_2 &= -3 \\ 3x_1 + x_2 + 2x_3 = 6 & \Rightarrow & x_1 &= 7 \end{aligned}$$

The solution is $\bar{x} = \begin{bmatrix} 7 & -3 & -6 \end{bmatrix}^T$

Inverse Matrix

A^{-1} is the inverse matrix of square matrix A if

$$AA^{-1} = A^{-1}A = I$$

Square matrix A is nonsingular (invertible) if its inverse A^{-1} exists.

How to compute A^{-1} of A?

$$[A \mid I] \xrightarrow[\text{row operations}]{\text{elementary}} [I \mid A^{-1}]$$

How to compute kth column of A^{-1}?
Solve $A\bar{y} = e_k$, i.e.

$$\left[\begin{array}{c|c} & 0 \\ & \vdots \\ A & 1 \\ & \vdots \\ & 0 \end{array}\right] \xrightarrow[\text{solve using elementary row operations}]{} \begin{array}{c}\text{solve triangular matrix} \\ \bar{y} = \dots \\ (k\text{th column of } A^{-1})\end{array}$$

Linear Inequalities I

Polyhedral Space

For $m \times n$-matrix A, variable n-vector $\overline{x}$, and m-vector $\overline{b}$, the $\Sigma_{\mathbb{Q}}$-formula

$$G : A\overline{x} \leq \overline{b}, \quad \text{i.e.,} \quad G : \bigwedge_{i=1}^{m} a_{i1}x_1 + \cdots + a_{in}x_n \leq b_i$$

describes a subset (space) of $\mathbb{Q}^n$, called a **polyhedron**.

Linear Inequalities II

Convex Space

An n-dimensional space $S \subseteq \mathbb{R}^n$ is **convex** if for all pairs of points $\bar{v}_1, \bar{v}_2 \in S$,

$$\lambda \bar{v}_1 + (1 - \lambda)\bar{v}_2 \in S \quad \text{for } \lambda \in [0, 1] \ .$$

$A\overline{x} \leq \overline{b}$ defines a **convex space**. For suppose $A\bar{v}_1 \leq \overline{b}$ and $A\bar{v}_2 \leq \overline{b}$; then also

$$A(\lambda \bar{v}_1 + (1 - \lambda)\bar{v}_2) \leq \overline{b} \ .$$

Linear Inequalities III

Vertex

Consider $m \times n$-matrix A where $m \geq n$.

An n-vector $\bar{v}$ is a **vertex** of $A\bar{x} \leq \bar{b}$ if there is

- a nonsingular $n \times n$-submatrix A_0 of A and
- corresponding n-subvector $\bar{b}_0$ of $\bar{b}$

such that

$$A_0 \bar{v} = \bar{b}_0 \ .$$

The rows a_{0_i} in A_0 and corresponding values b_{0_i} of $\bar{b}_0$ are the set of **defining constraints** of the vertex $\bar{v}$.

Two vertices are **adjacent** if they have defining constraint sets that differ in only one constraint.

Example I

Consider the linear inequality

$$\underbrace{\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{-1} \\ \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & \mathbf{-1} & \mathbf{0} \\ 0 & 1 & 0 & -1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x \\ y \\ z_1 \\ z_2 \end{bmatrix}}_{\overline{x}} \leq \underbrace{\begin{bmatrix} 0 \\ 0 \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{3} \\ \mathbf{2} \\ 2 \end{bmatrix}}_{\overline{b}}$$

A is a 7×4-matrix, $\overline{b}$ is a 7-vector, and
$\overline{x}$ is a variable 4-vector representing the four variables $\{x, y, z_1, z_2\}$.

Example II

$\overline{v} = [2\ 1\ 0\ 0]^T$ is a vertex of the constraints. For the nonsingular submatrix A_0 (rows 3, 4, 5, 6 of A: defining constraints of $\overline{v}$),

$$\underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{0} & -\mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{1} \\ \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{0} & -\mathbf{1} & \mathbf{0} \end{bmatrix}}_{A_0} \underbrace{\begin{bmatrix} 2 \\ 1 \\ 0 \\ 0 \end{bmatrix}}_{\overline{v}} = \underbrace{\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{3} \\ \mathbf{2} \end{bmatrix}}_{b_0}$$

Example III

Another vertex: $\overline{v}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T$, since

$$\underbrace{\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}}_{A_0} \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\overline{v}_0} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{b_0}$$

(rows 1,2,3,4 of A: defining constraints of $\overline{v}_0$)
Note: $\overline{v}$ and $\overline{v}_0$ are not adjacent; they are different in 2 defining constraints.

Linear Programming I

Optimization Problem

$$\begin{array}{llll} \textbf{max} & \overline{c}^{\mathsf{T}}\overline{x} & & \ldots \text{objective function} \\ \textbf{subject to} & & & \\ & A\overline{x} \leq \overline{b} & & \ldots \text{constraints} \end{array}$$

$$\textit{Maximize} \quad \sum_{i=1}^{n} c_i x_i$$

$$\textit{subject to} \quad \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \leq \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

Linear Programming II

Solution:

Find vertex $\overline{v}^*$ satisfying $A\overline{x} \leq \overline{b}$ and maximizing $\overline{c}^\mathsf{T}\overline{x}$.
That is,

$A\overline{v}^* \leq \overline{b}$ and
$\overline{c}^\mathsf{T}\overline{v}^*$ is maximal: $\overline{c}^\mathsf{T}\overline{v}^* \geq \overline{c}^\mathsf{T}\overline{u}$ for all $\overline{u}$ satisfying $A\overline{u} \leq \overline{b}$

- If $A\overline{x} \leq \overline{b}$ is unsatisfiable, then maximum is $-\infty$
- It's possible that the maximum is unbounded, then maximum is ∞

Example: Consider optimization problem:

$$\textbf{max} \quad \underbrace{\begin{bmatrix} 1 & 1 & -1 & -1 \end{bmatrix}}_{\overline{c}^T} \underbrace{\begin{bmatrix} x \\ y \\ z_1 \\ z_2 \end{bmatrix}}_{\overline{x}}$$

subject to

$$\underbrace{\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x \\ y \\ z_1 \\ z_2 \end{bmatrix}}_{\overline{x}} \leq \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 3 \\ 2 \\ 2 \end{bmatrix}}_{\overline{b}}$$

Example (cont):

The objective function is

$$(x - z_1) + (y - z_2) \ .$$

The constraints are equivalent to the $\Sigma_{\mathbb{Q}}$-formula

$$\begin{array}{c} x \geq 0 \ \wedge \ y \geq 0 \ \wedge \ z_1 \geq 0 \ \wedge \ z_2 \geq 0 \\ \wedge\, x + y \leq 3 \ \wedge \ x - z_1 \leq 2 \ \wedge \ y - z_2 \quad \leq \quad 2 \end{array}$$

Example: Linear Programming I

A company is producing two different products using three machines A, B, and C.

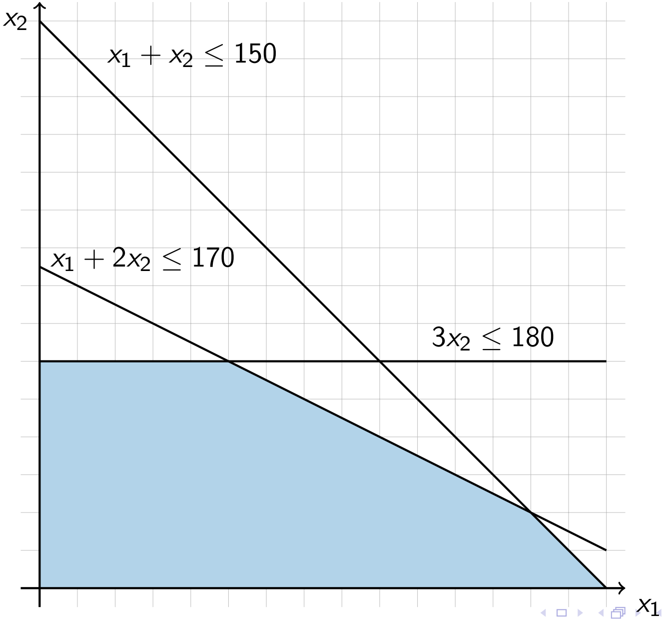
- Product 1 needs A for one, and B for one hour.
- Product 2 needs A for two, B for one, and C for three hours.
- Product 1 can be sold for \$300; Product 2 for \$500.
- Monthly availability of machines:
  A: 170 hours, B: 150 hours, C 180 hours.

Example: Linear Programming II

Let x_1 and x_2 denote the amount of product 1 and product 2, resp. We want to optimize $300x_1 + 500x_2$ subject to:

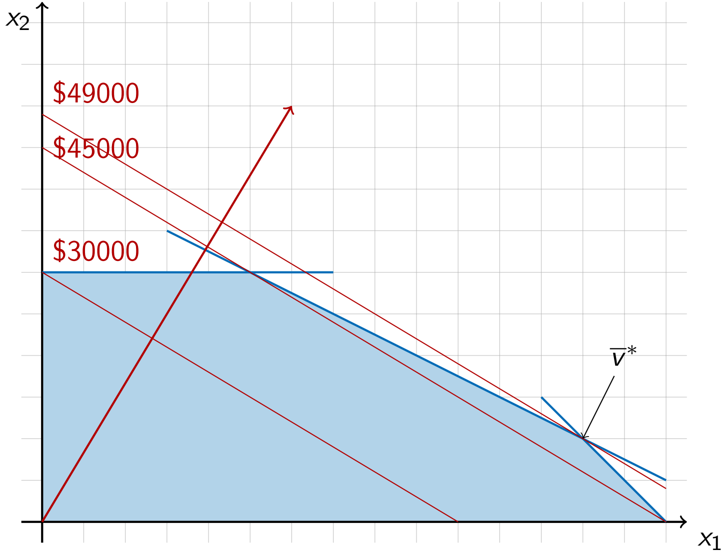
$$
\begin{aligned}
1x_1 + 2x_2 &\leq 170 && \text{Machine (A)} \\
1x_1 + 1x_2 &\leq 150 && \text{Machine (B)} \\
0x_1 + 3x_2 &\leq 180 && \text{Machine (C)} \\
x_1 \geq 0 \wedge x_2 &\geq 0
\end{aligned}
$$

Example: Linear Programming III



Example: Linear Programming IV

Optimize $300x_1 + 500x_2$:



Duality Theorem

For $m \times n$-matrix A, m-vector $\overline{b}$ and n-vector $\overline{c}$:

$$\max\{\overline{c}^{\mathsf{T}}\overline{x} \mid A\overline{x} \leq \overline{b} \ \wedge \ \overline{x} \geq \overline{0}\} = \min\{\overline{b}^{\mathsf{T}}\overline{y} \mid A^{\mathsf{T}}\overline{y} \geq \overline{c} \ \wedge \ \overline{y} \geq \overline{0}\}$$

if the constraints are satisfiable.

That is,

maximizing the function $c^{\mathsf{T}}\overline{x}$ over $A\overline{x} \leq \overline{b}$, $\overline{x} \geq \overline{0}$
(the primal form of the optimization problem)

is equivalent to

minimizing the function $\overline{b}^{\mathsf{T}}\overline{y}$ over $A^{\mathsf{T}}\overline{y} \geq \overline{c}$, $\overline{y} \geq \overline{0}$
(the dual form of the optimization problem)

By convention: when $A\overline{x} \leq b \ \wedge \ \overline{x} \geq 0$ unsatisfiable, the max is $-\infty$ and the min is ∞.

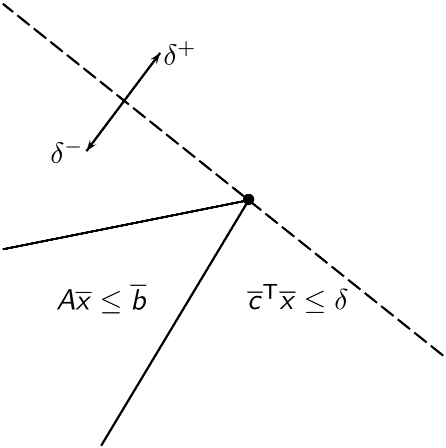


Figure: Visualization of the duality theorem

The region labeled $A\overline{x} \leq \overline{b}$ satisfies the inequality. The objective function $\overline{c}^{\mathsf{T}}\overline{x}$ is represented by the dashed line. Its value increases in the direction of the arrow labeled δ^+ and decreases in the direction of the arrow labeled δ^-.

Example: A Dual Problem

What is the value of a machine hour?
Let y_A, y_B, y_C be the values of machine A, B, and C.
The value of the machine hours to produce something $\geq$ the value of the product ($>$ if that product should not be produced).

$$y_A \geq 0 \ \wedge \ y_B \geq 0 \ \wedge \ y_C \geq 0$$
$$1y_A + 1y_B + 0y_C \geq 300$$
$$2y_A + 1y_B + 3y_C \geq 500$$

We minimize the value $170y_A + 150y_B + 180y_C$ to get the value of a machine hour:

$$y_A = 200 \ \wedge \ y_B = 100 \ \wedge \ y_C = 0$$
$$170y_A + 150y_B + 180y_C = 49000$$

This is the dual problem. It has the same optimal value.

The Simplex Method

Consider linear program

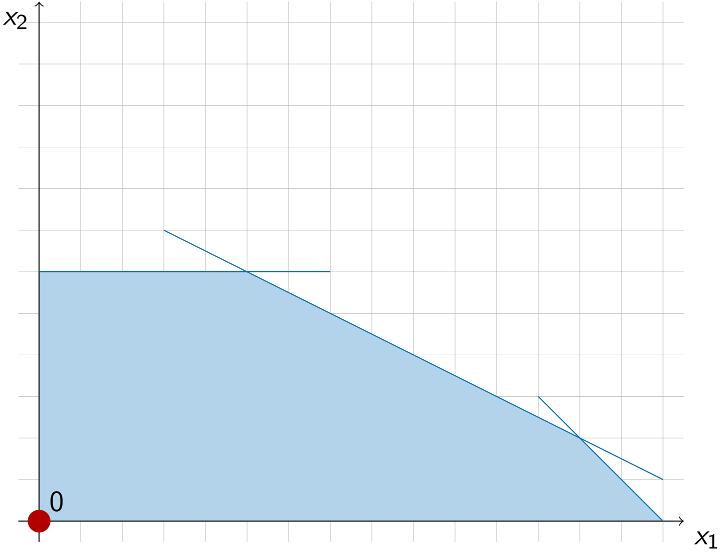
$$M \quad : \quad \textbf{max } \bar{c}^{\mathsf{T}}\bar{x}$$
$$\textbf{subject to } G : A\bar{x} \leq \bar{b}$$

The **simplex method** solves the linear program in two main steps:

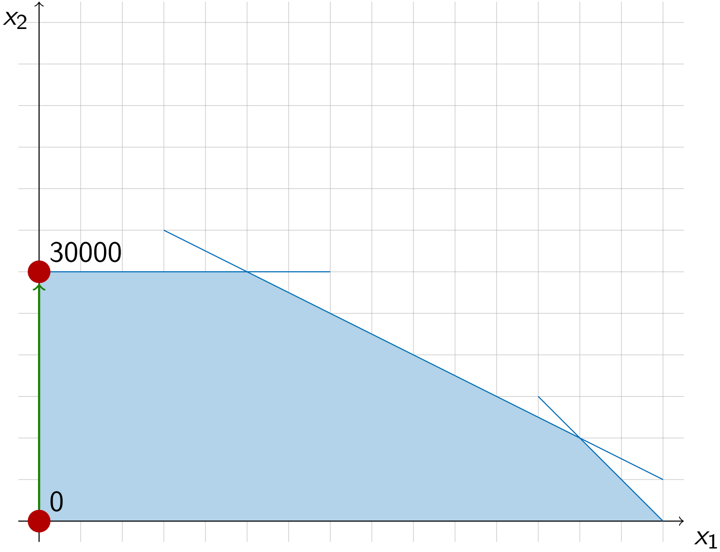
1. Obtain an initial vertex $\bar{v}_1$ of $A\bar{x} \leq \bar{b}$.
2. Iteratively traverse the vertices of $A\bar{x} \leq \bar{b}$, beginning at $\bar{v}_1$, in search of the vertex that maximizes $\bar{c}^{\mathsf{T}}\bar{x}$. On each iteration determine if $\bar{c}^{\mathsf{T}}\bar{v}_i > \bar{c}^{\mathsf{T}}\bar{v}_i'$ for the vertices $\bar{v}_i'$ adjacent to $\bar{v}_i$:
   - If not, move to one of the adjacent vertices $\bar{v}_i'$ with a greater objective value.
   - If so, halt and report $\bar{v}_i$ as the optimum point with value $\bar{c}^{\mathsf{T}}\bar{v}_i$.

The final vertex $\bar{v}_i$ is a **local optimum** since its adjacent vertices have lesser objective values. But because the space defined by $A\bar{x} \leq \bar{b}$ is convex, $\bar{v}_i$ is also the **global optimum**: it is the highest value attained by any point that satisfies the constraints.

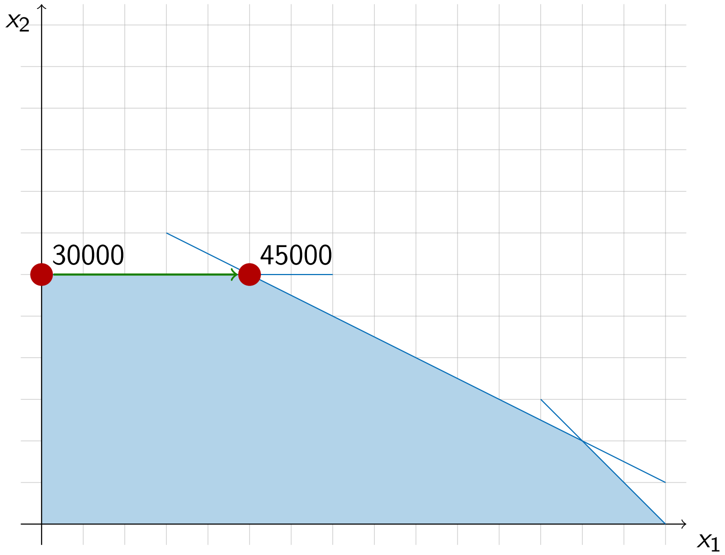
Example



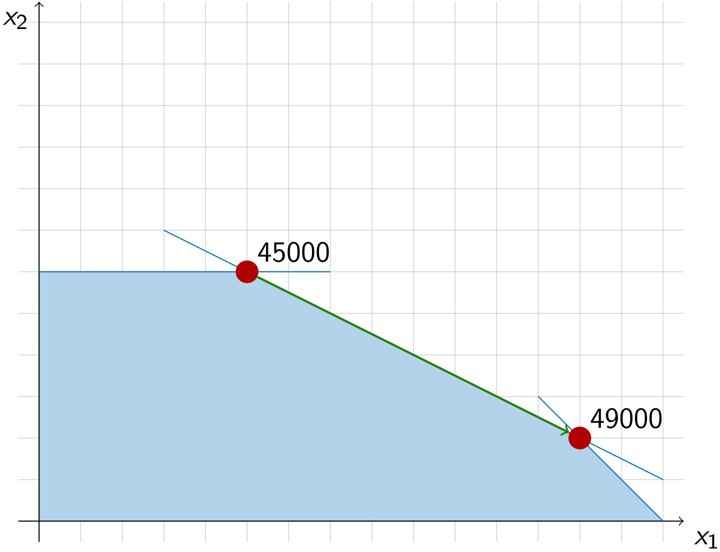
Example



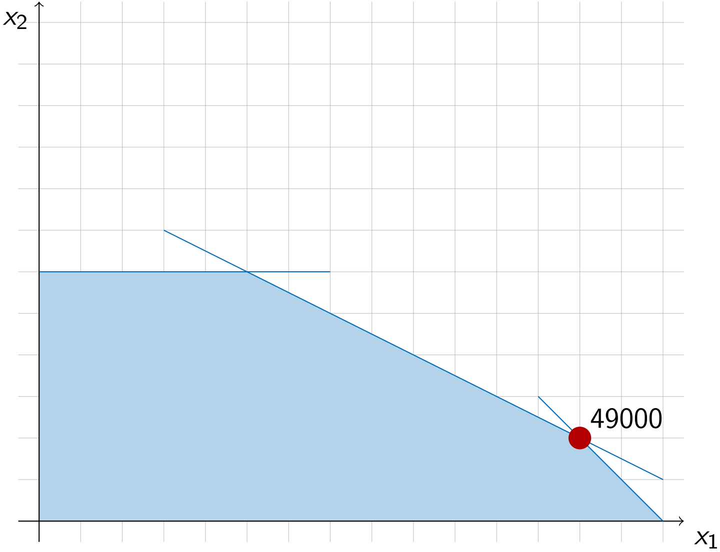
Example



Example



Example



How do we use optimization to determine satisfiability?

We are not interested in an *optimal* solution $\overline{x}$ such that

$$F : A\overline{x} \leq \overline{b} \;;$$

we want *some* solution. However, this hard to find.

Idea: Transform F into an *optimization* problem with an initial (not-optimal) vertex $\overline{v}_1$ and a desired optimum v_F.

Apply the Simplex Method until an optimal vertex $\overline{v}^*$ is obtained.

The optimum value for $\overline{v}^*$ is v_F iff $F : Ax \leq b$ is satisfiable.

The solution can be computed from the optimal solution $\overline{x}$ of the optimization problem.

Outline of the Algorithm I

Determine if $\Sigma_{\mathbb{Q}}$-formula

$$\begin{array}{rl} F: & \bigwedge_{i=1}^{m} a_{i1}x_1 + \ldots + a_{in}x_n \leq b_i \\ \wedge & \bigwedge_{i=1}^{\ell} \alpha_{i1}x_1 + \ldots + \alpha_{in}x_n < \beta_i \end{array}$$

is satisfiable.

<u>Note</u>: Equations

$$a_{i1}x_1 + \ldots + a_{in}x_n = b_i$$

are allowed; break them into two inequalities:

$$\begin{array}{rcl} a_{i1}x_1 + \ldots + a_{in}x_n & \leq & b_i \\ -a_{i1}x_1 + \ldots + -a_{in}x_n & \leq & -b_i \end{array}$$

Outline of the Algorithm II

F is $T_{\mathbb{Q}}$-equivalent to the $\Sigma_{\mathbb{Q}}$-formula

$$
\begin{array}{lll}
F' : & & \bigwedge_{i=1}^{m} a_{i1}x_1 + \ldots + a_{in}x_n \leq b_i \\
 & \wedge & \bigwedge_{i=1}^{\ell} \alpha_{i1}x_1 + \ldots + \alpha_{in}x_n + z \leq \beta_i \\
 & \wedge & z > 0
\end{array}
$$

Outline of the Algorithm III

To decide the $T_{\mathbb{Q}}$-satisfiability of F', solve the linear program

max z
subject to

$$
\begin{array}{rcl}
\bigwedge_{i=1}^{m} a_{i1}x_1 + \ldots + a_{in}x_n & \leq & b_i \\
\bigwedge_{i=1}^{\ell} \alpha_{i1}x_1 + \ldots + \alpha_{in}x_n + z & \leq & \beta_i
\end{array}
$$

F' is $T_{\mathbb{Q}}$-satisfiable iff the optimum is positive.

Outline of the Algorithm IV

When F does not contain any strict inequality literals, the corresponding linear program

max 1
subject to

$$\bigwedge_{i=1}^{m} a_{i1}x_1 + \ldots + a_{in}x_n \quad \leq \quad b_i$$

has optimum $-\infty$ iff the constraints are $T_{\mathbb{Q}}$-unsatisfiable,
1 iff the constraints are $T_{\mathbb{Q}}$-satisfiable.

Outline of the Algorithm V

To determine the satisfiability of $F : A\overline{x} \le \overline{b}$,

$\underline{M \to M_0}$

reformulate the satisfiability of F as an optimization problem:

$$M_0 : \max\{\bar{c}^{\mathsf{T}}\bar{x}' \mid A'\bar{x}' \le \bar{b}'\}$$

such that F is $T_{\mathbb{Q}}$-satisfiable iff the optimal value of M_0 is a particular value v_F (derived from the structure of F).

Simplex Method

vertex traversal until termination

Outline of the Algorithm VI

The simplex method traverses the vertices of $A'\overline{x}' \leq \overline{b}'$ searching for the maximum of the objective function $\overline{c}^{\mathsf{T}}\overline{x}'$.

If $\overline{v}_1, \overline{v}_2, \ldots$ are the traversed vertices in the iteration, then

$$\overline{c}^{\mathsf{T}}\overline{v}_1 < \overline{c}^{\mathsf{T}}\overline{v}_2 < \cdots \ .$$

The simplex method terminates at some vertex $\overline{v}_{i^*}$ where $\overline{c}^{\mathsf{T}}\overline{v}_{i^*}$ is the global optimum

Final step: Compare the discovered optimal value $\overline{c}^{\mathsf{T}}\overline{v}_{i^*}$ to the desired value v_F.

- if equal, then F is $T_{\mathbb{Q}}$-satisfiable
- otherwise, F is $T_{\mathbb{Q}}$-unsatisfiable

Step 0: From Satisfiability to Optimization

Given $\Sigma_{\mathbb{Q}}$-formula

$$F: \; A\overline{x} \leq \overline{b} \tag{8.1}$$

reformulate to new constraint system (new A, $\overline{x}$, $\overline{b}$)

$$F': \; \overline{x} \geq 0, \; A\overline{x} \leq \overline{b}$$

such that F' is $T_{\mathbb{Q}}$-equisatisfiable to F

The trick: replace each variable x in F by $x_1 - x_2$ and add $\bar{x} \geq 0$

Step 0: From Satisfiability to Optimization

Making the b_i positive

Collect the lines where b_i is negative:

$$A\overline{x} = \begin{bmatrix} D_1 \\ -D_2 \end{bmatrix} \overline{x} \quad \leq \quad \begin{bmatrix} \overline{g}_1 \\ -\overline{g}_2 \end{bmatrix} = \overline{b}$$

where

$$\begin{aligned} \overline{g}_1 &\geq 0 \\ \overline{g}_2 &> 0 \end{aligned}$$

Multiply the bottom rows with -1:

$$\begin{aligned} D_1\overline{x} \leq \overline{g}_1 \\ D_2\overline{x} \geq \overline{g}_2 \end{aligned}$$

Example 1: $x + y \geq 1 \land x - y \geq -1$

$\Sigma_{\mathbb{Q}}$-formula

$$F : \; x + y \geq 1 \;\land\; x - y \geq -1 \; .$$

To convert it to the form $\overline{x} \geq \overline{0} \;\land\; A\overline{x} \leq \overline{b}$, introduce nonnegative x_1, x_2 for x and y_1, y_2 for y:

$$F' : \; \begin{array}{l} (x_1 - x_2) + (y_1 - y_2) \geq 1 \;\land\; (x_1 - x_2) - (y_1 - y_2) \geq -1 \\ \;\land\; x_1, x_2, y_1, y_2 \geq 0 \end{array}$$

F is $T_{\mathbb{Q}}$-equisatisfiable to F'. In matrix form (with $\overline{x} \geq 0$),

$$F' : \; \underbrace{\begin{bmatrix} -1 & 1 & -1 & 1 \\ -1 & 1 & 1 & -1 \end{bmatrix}}_{A} \begin{bmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{bmatrix} \leq \underbrace{\begin{bmatrix} -1 \\ 1 \end{bmatrix}}_{\overline{b}}$$

Example 1: $x + y \geq 1 \wedge x - y \geq -1$

$$F' : \begin{array}{l} (x_1 - x_2) + (y_1 - y_2) \geq 1 \;\wedge\; (x_1 - x_2) - (y_1 - y_2) \geq -1 \\ \;\wedge\; x_1, x_2, y_1, y_2 \geq 0 \end{array}$$

Since $b_1 < 0$ and $b_2 > 0$, separating constraints yields

$$\underbrace{\begin{bmatrix} -1 & 1 & 1 & -1 \end{bmatrix}}_{D_1} \begin{bmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{bmatrix} \leq \underbrace{[1]}_{\overline{g}_1}$$

$$\underbrace{\begin{bmatrix} 1 & -1 & 1 & -1 \end{bmatrix}}_{D_2} \begin{bmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{bmatrix} \geq \underbrace{[1]}_{\overline{g}_2}$$

Step 0: From Satisfiability to Optimization

$$D_1\overline{x} \leq \overline{g}_1 \qquad \overline{g}_1 \geq 0$$
$$D_2\overline{x} \geq \overline{g}_2 \qquad \overline{g}_2 > 0$$

Generate the optimization problem:

$$M_0: \textbf{max} \quad \overline{1}^{\mathsf{T}}(D_2\overline{x} - \overline{z}) \tag{8.2}$$

$$\textbf{subject to}$$

$$\begin{array}{rcll} \overline{x}, \overline{z} & \geq & \overline{0} & (1) \\ D_1\overline{x} & \leq & \overline{g}_1 & (2) \\ D_2\overline{x} - \overline{z} & \leq & \overline{g}_2 & (3) \end{array}$$

length of variable vector $\overline{z} = \#$ of rows of D_2

- The point $\overline{x} = \overline{0}$, $\overline{z} = \overline{0}$ satisfies constraints (1) – (3). It's a vertex.
- The optimum v_F equals $\overline{1}^{\mathsf{T}}\overline{g}_2$ (the equality in (3) holds) iff F is $T_{\mathbb{Q}}$-satisfiable. (proof on p. 220)

The $\overline{x}$ part of the optimal solution $\overline{v}^*$ satisfies F.

Step 0: From Satisfiability to Optimization

M_F can be written in standard form as

$$M_F : \quad \textbf{max} \quad \underbrace{\bar{1}^{\mathsf{T}} \begin{bmatrix} D_2 & -I \end{bmatrix}}_{\bar{c}^{\mathsf{T}}} \underbrace{\begin{bmatrix} \bar{x} \\ \bar{z} \end{bmatrix}}_{\bar{y}} \tag{8.3}$$

subject to

$$\underbrace{\begin{bmatrix} -I & \\ & -I \\ D_1 & \\ D_2 & -I \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \bar{x} \\ \bar{z} \end{bmatrix}}_{\bar{y}} \leq \underbrace{\begin{bmatrix} \bar{0} \\ \bar{0} \\ \bar{g}_1 \\ \bar{g}_2 \end{bmatrix}}_{\bar{b}}$$

Example 1: $x + y \geq 1 \wedge x - y \geq -1$

$$\underbrace{\begin{bmatrix} -1 & 1 & 1 & -1 \end{bmatrix}}_{D_1} \begin{bmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{bmatrix} \leq \underbrace{[1]}_{\overline{g}_1} \text{ and } \underbrace{\begin{bmatrix} 1 & -1 & 1 & -1 \end{bmatrix}}_{D_2} \begin{bmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{bmatrix} \geq \underbrace{[1]}_{\overline{g}_2}$$

D_2 has only one row, so $\overline{z} = [z]$.

Pose the following optimization problem:

$$\textbf{max} \quad \begin{bmatrix} 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{bmatrix} - [z]$$

subject to

$\ldots$

Example 1: $x + y \geq 1 \land x - y \geq -1$

$$
\begin{aligned}
x_1, x_2, y_1, y_2, z &\geq 0 \\
\begin{bmatrix} -1 & 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{bmatrix} &\leq [1] \\
\begin{bmatrix} 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \end{bmatrix} - [z] &\leq [1]
\end{aligned}
$$

F is $T_{\mathbb{Q}}$-satisfiable iff the optimum is $\overline{1}^{\mathsf{T}}\overline{g}_2 = 1$.
$[x_1 \ x_2 \ y_1 \ y_2 \ z] = [0 \ 0 \ 0 \ 0 \ 0]$ is a vertex.

Example 1: $x + y \geq 1 \wedge x - y \geq -1$

Rewriting the optimization problem

$$\textbf{max} \quad \underbrace{\begin{bmatrix} 1 & -1 & 1 & -1 & -1 \end{bmatrix}}_{\bar{c}^{T}} \begin{bmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \\ z \end{bmatrix}$$

subject to

$$\overbrace{\begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ -1 & 1 & 1 & -1 & 0 \\ 1 & -1 & 1 & -1 & -1 \end{bmatrix}}^{A} \begin{bmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \\ z \end{bmatrix} \leq \overbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}}^{\bar{b}}$$

From $<$ to $\leq$ (reminder)

If we have some strict inequalities:

$$\begin{aligned} \overline{x} &\geq 0 \\ A_0\overline{x} &\leq \overline{b}_0 \\ A_1\overline{x} &< \overline{b}_1 \end{aligned}$$

introduce a new variable $z \geq 0$ and maximize z, such that

$$\begin{aligned} \overline{x} \geq 0 \wedge z &\geq 0 \\ A_0\overline{x} &\leq \overline{b}_0 \\ A_1\overline{x} + z \cdot \overline{1} &\leq \overline{b}_1 \end{aligned}$$

The maximum is greater than 0 iff the original constraint is satisfiable.
Note: In this case, one can stop the simplex algorithm after the first time z increases. Why?

Example 1A: $x + y > 1 \wedge x - y > -1$

Normal form:

$$\begin{aligned} x_1, x_2, y_1, y_2 &\geq 0 \\ -x_1 + x_2 + y_1 - y_2 &< 1 \\ -x_1 + x_2 - y_1 + y_2 &< -1 \end{aligned}$$

Introduce z_1 for the strictness: Maximize z_1 subject to

$$\begin{aligned} x_1, x_2, y_1, y_2, z_1 &\geq 0 \\ -x_1 + x_2 + y_1 - y_2 + z_1 &\leq 1 \\ -x_1 + x_2 - y_1 + y_2 + z_1 &\leq -1 \end{aligned}$$

Introduce z_2 to get rid of negative bound:

Example 1A: $x + y > 1 \wedge x - y > -1$

Maximize $x_1 - x_2 + y_1 - y_2 - z_1 - z_2$ subject to

$$x_1, x_2, y_1, y_2, z_1 \geq 0$$
$$-x_1 + x_2 + y_1 - y_2 + z_1 \leq 1$$
$$x_1 - x_2 + y_1 - y_2 - z_1 - z_2 \leq 1$$

Example 1A: $x + y > 1 \wedge x - y > -1$

In matrix form:

max $[1\ -1\ 1\ -1\ -1\ -1]\overline{x}$

subject to

$$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & 1 & 1 & -1 & 1 & 0 \\ 1 & -1 & 1 & -1 & -1 & -1 \end{bmatrix} \overline{x} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

From Satisfiability to Optimization: Summary

1. Adding the constraints $\overline{x} \geq 0$
   Replace each variable x by $x_1 - x_2$, then add $\overline{x} \geq 0$.
2. Getting rid of strict inequality $<$
   Add variable $z \geq 0$, replace $Ax < \overline{b}$ with $A\overline{x} + z \leq \overline{b}$, optimize z.
   Strict inequality satisfiable iff optimum > 0.
3. Making the b_i positive

Vertex Traversal: Find a Better Vertex

Optimization problem of form

$$\begin{aligned} \textbf{max} \quad & \overline{c}^{\mathsf{T}}\overline{x} \qquad (8.3) \\ \textbf{subject to} & \\ A\overline{x} & \leq \overline{b} \end{aligned}$$

we are given satisfying vertex $\overline{v}_i$.

- The simplex method traverses vertices of the space defined by $A\overline{x} \leq \overline{b}$ to find the vertex $\overline{v}^*$ that maximizes $\overline{c}^{\mathsf{T}}\overline{x}$.
- One iteration seeks vertex $\overline{v}_{i+1}$ "adjacent" $(n-1$ shared defining constraints) to $\overline{v}_i$ s.t. $\overline{c}^{\mathsf{T}}\overline{v}_{i+1} > \overline{c}^{\mathsf{T}}\overline{v}_i$
- For $i = 1$, the initial vertex $\overline{v}_1$ of M_0 is $\overline{x} = \overline{0}$, $\overline{z} = \overline{0}$

Example (cont):

$$\overline{v}_1 = [x_1 \ x_2 \ y_1 \ y_2 \ z]^{\mathsf{T}} = [0\ 0\ 0\ 0\ 0]^{\mathsf{T}}$$

Vertex Traversal

<u>Find $\overline{u}$</u>

Construct vector $\overline{u}$ s.t.

$$\overline{u}^{\mathsf{T}} A = \overline{c}^{\mathsf{T}} \qquad (8.4)$$

If $\overline{u} \geq \overline{0}$ then by the Duality Theorem $\overline{v}_i$ is optimal.

- Given $\overline{v}_i$
- Construct $n \times n$ nonsingular submatrix A_i with corresponding rows $\overline{b}_i$ s.t.

$$A_i \overline{v}_i = \overline{b}_i$$

- Let $R =$ rows of A in A_i
- Solve

$$\boxed{{A_i}^{\mathsf{T}} \overline{u}_i = \overline{c}} \qquad (8.5)$$

- Let $\overline{u}$ be $\overline{u}_i$ for indices in R and
  0's for indices not in R ($\overline{u}_i$ suffices!)

Example 1: $x + y \geq 1 \wedge x - y \geq -1$

Choose the first five rows of A and $\overline{b}$ ($R = [1; 2; 3; 4; 5]$) since

$$\underbrace{\begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}}_{A_1} \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\overline{v}_1} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\overline{b}_1}$$

i.e. $-I\overline{v}_1 = \overline{b}_1$. Solving (by Gaussian elimination):

$$\underbrace{\begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}}_{{A_1}^{\mathsf{T}}} \overline{u}_1 = \underbrace{\begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \\ -1 \end{bmatrix}}_{\overline{c}}$$

Example 1: $x + y \geq 1 \wedge x - y \geq -1$

(i.e. $-I\overline{u}_1 = \overline{c}$, and thus $\overline{u}_1 = -\overline{c}$) yields

$$\overline{u}_1{}^{\mathsf{T}} = \begin{bmatrix} -1 & 1 & -1 & 1 & 1 \end{bmatrix} .$$

Then

$$\overline{u} = \begin{bmatrix} -1 & 1 & -1 & 1 & 1 & 0 & 0 \end{bmatrix}^{\mathsf{T}}$$

Vertex Traversal

Case 1: $\overline{u} \geq \overline{0}$

In this case, $\overline{v}_i$ is actually the optimal point with optimal value $\overline{c}^{\mathsf{T}}\overline{v}_i$. (proof on p. 226)

Case 2: $\overline{u} \ngeq \overline{0}$, i.e. there exists some $u_k < 0$

In this case, $\overline{v}_i$ is not the optimal point. We need to move along an edge to an adjacent vertex to increase the value of the objective function.

- Let k be the lowest index of $\overline{u}$ s.t. $u_k < 0$ (must be $k \in R$)
- Let k' be the index of the corresponding row of $\overline{u}_i$ and A_i and the corresponding column of $-A_i^{-1}$

Vertex Traversal

Find $\overline{y}$

- Let $\overline{y}$ be the k'th column of $-A_i^{-1}$. Solve

$$\boxed{A_i\overline{y} = -\mathrm{e}_{k'}} \tag{8.8}$$

That is,

$$\begin{aligned} \overline{a}_\ell \overline{y} &= 0 && \text{for every row } \overline{a}_\ell \text{ of } A_i,\ \ell \neq k' \\ \overline{a}_{k'} \overline{y} &= -1 && \text{for the } k'\text{th row } \overline{a}_{k'} \text{ of } A_i \end{aligned}$$

The vector $\overline{y}$ provides the direction along which to move to the next vertex.

Example 1: $x + y \geq 1 \wedge x - y \geq -1$

We found so far

$$\overline{u}_1 = \begin{bmatrix} -1 & 1 & -1 & 1 & 1 \end{bmatrix}^T \text{ and } \overline{u} = \begin{bmatrix} -1 & 1 & -1 & 1 & 1 & 0 & 0 \end{bmatrix}^T$$

$k = 1$ since the first row of $\overline{u}$ is -1. $k' = 1$ since it is also the first row of $\overline{u}_i$.

Thus, solve

$$\underbrace{\begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}}_{A_1} \overline{y} = \underbrace{\begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{-e_1}$$

i.e. $-I\overline{y} = -e_1$, yielding $\overline{y} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix}^T$.

Vertex Traversal

Find λ and v_{i+1}

We move along edge $\overline{y}$ to better vertex $\overline{v}_{i+1}$.

- Let S = indices ℓ s.t. $\overline{a}_\ell \overline{y} > 0$
- Find greatest $\lambda_i \geq 0$ such that

$$A(\overline{v}_i + \lambda_i \overline{y}) \leq \overline{b}$$

Choose $\lambda_i > 0$ such that

$$\begin{aligned} \overline{a}_\ell(\overline{v}_i + \lambda_i \overline{y}) &= b_\ell && \text{for some } \ell \in S \\ \overline{a}_m(\overline{v}_i + \lambda_i \overline{y}) &\leq b_m && \text{for } m \in S - \{\ell\} \end{aligned}$$

Vertex Traversal

- Set $\boxed{\overline{v}_{i+1} = \overline{v}_i + \lambda_i \overline{y}}$ (8.12)

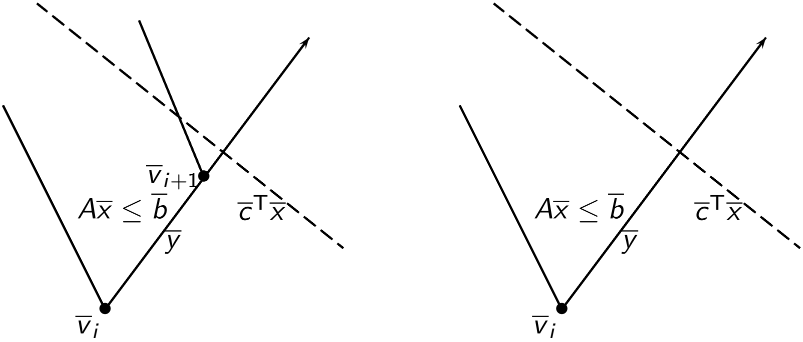
  Vertex $\overline{v}_{i+1}$ is discovered by moving along ray $\overline{y}$ as far as possible without violating the constraints. Moreover,

$$\overline{c}^{\mathsf{T}} \overline{v}_{i+1} > \overline{c}^{\mathsf{T}} \overline{v}_i \ .$$

- Construct A_{i+1} from A_i for next iteration by substituting row $\overline{a}_\ell$ of A for row $\overline{a}_{k'}$ of A_i

Since there are only finite number of vertices to examine, Case 1 eventually occurs.

Vertex Traversal



(a) bounded **(b) unbounded**

- **(a)** depicts the discovery of vertex $\overline{v}_{i+1}$ by moving along ray $\overline{y}$ as far as possible without violating the constraints.
- **(b)** illustrates what happens when all points along the ray laybeled $\overline{y}$ satisfy the constraints: moving along the ray increases $\overline{c}^{\mathsf{T}}\overline{x}$ without bound.

Example 1: $x + y \geq 1 \wedge x - y \geq -1$

We found in Step 1

$$\bar{y} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix}^{\mathsf{T}}$$

where

$$\underbrace{\begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}}_{A_1} \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\bar{y}} = \underbrace{\begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{-\mathrm{e}_1}$$

Example 1: $x + y \geq 1 \wedge x - y \geq -1$

Compute $A\overline{y}$

$$\underbrace{\begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ -1 & 1 & 1 & -1 & 0 \\ 1 & -1 & 1 & -1 & -1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\overline{y}} = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ 1 \end{bmatrix}$$

Example 1: $x + y \geq 1 \wedge x - y \geq -1$

$S = [7]$ since $\overline{a}_7\overline{y} = 1 > 0$. Examining the 7th row of the constraints, choose the greatest λ_1 such that (8.7b)

$$\underbrace{\begin{bmatrix} 1 & -1 & 1 & -1 & -1 \end{bmatrix}}_{\overline{a}_7}(\overline{v}_1 + \lambda_1\overline{y}) \quad =$$

$$\begin{bmatrix} 1 & -1 & 1 & -1 & -1 \end{bmatrix}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \lambda_1 \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}\right) \quad = \quad \underbrace{1}_{b_7}$$

that is, choose $\lambda_1 = 1$. Therefore, (8.7c)

$$\boxed{\overline{v}_2 = \overline{v}_1 + \lambda_1\overline{y} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix}^T}$$

Example 1: $x + y \geq 1 \wedge x - y \geq -1$

Form A_2 from A_1 replacing the 1st row ($k' = 1$) of A_1 by the 7th row ($\ell = 7$) of A.

$$A_2 = \begin{bmatrix} 1 & -1 & 1 & -1 & -1 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix} \qquad \overline{b}_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Thus, $A_2 \overline{v}_2 = \overline{b}_2$. This move to vertex $\overline{v}_2$ makes progress:

$$\underbrace{\begin{bmatrix} 1 & -1 & 1 & -1 & -1 \end{bmatrix}}_{\overline{c}^\mathsf{T}} \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\overline{v}_1} = 0 < \underbrace{\begin{bmatrix} 1 & -1 & 1 & -1 & -1 \end{bmatrix}}_{\overline{c}^\mathsf{T}} \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\overline{v}_2} = 1 \, .$$

Example 1: $x + y \geq 1 \wedge x - y \geq -1$

Now $R = [7;2;3;4;5]$ (rows of A in A_2).

Solve

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 & -1 \end{bmatrix}}_{{A_2}^{\mathsf{T}}} \overline{u}_2 = \underbrace{\begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \\ -1 \end{bmatrix}}_{\overline{c}}$$

for $\overline{u}_2$ yielding $\overline{u}_2 = [1\ 0\ 0\ 0\ 0]^{\mathsf{T}}$. Since $\overline{u}_2 \geq 0$, we are in Case 1: we have found an optimum point, $\overline{v}_2$, with optimal value 1.

Since we have that $v_F = \overline{1}^{\mathsf{T}} \overline{g}_2 = 1$, the equality of the optimial point and v_F implies that

Example 1: $x + y \geq 1 \wedge x - y \geq -1$

$$F:\ x + y \geq 1\ \wedge\ x - y \geq -1$$

is $T_{\mathbb{Q}}$-satisfiable. In particular, extract from

$$\begin{bmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \\ z \end{bmatrix} = \overline{v}_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

the assignment

$$x = x_1 - x_2 = 1 - 0 = 1 \quad \text{and} \quad y = y_1 - y_2 = 0 - 0 = 0\ ,$$

which indeed satisfies F.

Example 2

Consider optimization problem of the form (8.3)

$$\textbf{max} \underbrace{\begin{bmatrix} -1 & 1 \end{bmatrix}}_{\overline{c}^{\mathsf{T}}} \overline{x}$$

subject to

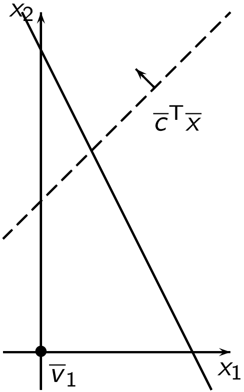
$$\underbrace{\begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 2 & 1 \end{bmatrix}}_{A} \overline{x} \leq \underbrace{\begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}}_{\overline{b}}$$

$\overline{v}_1 = \begin{bmatrix} 0 & 0 \end{bmatrix}^{\mathsf{T}}$ is a vertex.

The first two constraints are the defining constraints of $\overline{v}_1$, so choose $R = [1; 2]$:

$$A_1 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \qquad \overline{b}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} .$$

Thus $A_1 \overline{v}_1 = \overline{b}_1$.



The solid lines represent the constraints. The dashed line indicates $\overline{c}^{\mathsf{T}}\overline{x}$; the arrow points in the direction of increasing value.

Example 2

First Iteration

From (8.5), solving

$$\underbrace{\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}}_{A_1^\mathsf{T}} \overline{u}_1 = \underbrace{\begin{bmatrix} -1 \\ 1 \end{bmatrix}}_{\overline{c}} \qquad \text{i.e.,} \; -I\overline{u}_1 = \overline{c}$$

for $\overline{u}_1$ yields

$$\overline{u}_1 = -\overline{c} = \begin{bmatrix} 1 & -1 \end{bmatrix}^\mathsf{T} .$$

Adding 0s for rows not in R produces

$$\overline{u} = \begin{bmatrix} 1 & -1 & 0 \end{bmatrix}^\mathsf{T} .$$

This $\overline{u}$ satisfies $\overline{u}^\mathsf{T} A = \overline{c}^\mathsf{T}$ of (8.6).

Example 2

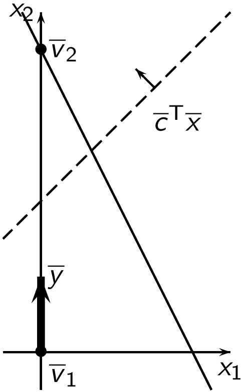
Since the 2nd row of $\overline{u}$ is -1, we are in Case 2 ($\overline{u} \ngeq 0$) with $k = 2$ of $\overline{u}$, corresponding to row $k' = 2$ of $\overline{u}_1$.

Let $\overline{y}$ be the 2nd column of $-A_1^{-1}$, and solve (8.8)

$$\underbrace{\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}}_{A_1} \overline{y} = \underbrace{\begin{bmatrix} 0 \\ -1 \end{bmatrix}}_{-e_2}$$

for $\overline{y}$, yielding

$$\overline{y} = \begin{bmatrix} 0 & 1 \end{bmatrix}^{\mathsf{T}} .$$



The $\overline{y}$ is visualized by the dark solid arrow that points up from $\overline{v}_1$. The vertical and horizontal lines are the defining constraints of $\overline{v}_1$; in moving in the direction $\overline{y}$, we keep the vertical constraint for the next vertex $\overline{v}_2$ but drop the horizontal constraint. The diagonal constraint will become the second of $\overline{v}_2$'s defining constraints.

Example 2

Choose λ_1 such that

$$\underbrace{\begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 2 & 1 \end{bmatrix}}_{A} \left(\underbrace{\begin{bmatrix} 0 \\ 0 \end{bmatrix}}_{\overline{v}_1} + \lambda_1 \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{\overline{y}} \right) \leq \underbrace{\begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}}_{\overline{b}} .$$

Example 2

We have

$$\underbrace{\begin{bmatrix}-1 & 0\end{bmatrix}}_{(A)_1}\underbrace{\begin{bmatrix}0\\1\end{bmatrix}}_{\bar{y}} = 0$$

$$\underbrace{\begin{bmatrix}0 & -1\end{bmatrix}}_{(A)_2}\underbrace{\begin{bmatrix}0\\1\end{bmatrix}}_{\bar{y}} < 0$$

$$\underbrace{\begin{bmatrix}2 & 1\end{bmatrix}}_{(A)_3}\underbrace{\begin{bmatrix}0\\1\end{bmatrix}}_{\bar{y}} > 0 \quad \Rightarrow \quad \begin{bmatrix}2 & 1\end{bmatrix}\left(\begin{bmatrix}0\\0\end{bmatrix} + \lambda_1\begin{bmatrix}0\\1\end{bmatrix}\right) = 2$$

$$\Rightarrow \quad \lambda_1 = 2$$

Thus $\lambda_1 = 2$, $\ell = 3$.

Example 2

From (8.12),

$$\overline{v}_2 = \overline{v}_1 + \lambda_1 \overline{y} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + 2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \end{bmatrix} .$$

Choosing $R = [1; 3]$ and replacing the 2nd row of A_1 and $\overline{b}_1$ ($k' = 2$) with the 3rd row ($\ell_3 = 3$) of $A\overline{x} \leq \overline{b}$ yields

$$A_2 = \begin{bmatrix} -1 & 0 \\ 2 & 1 \end{bmatrix} \quad \text{and} \quad \overline{b}_2 = \begin{bmatrix} 0 \\ 2 \end{bmatrix} ; \quad \text{i.e., } A_2 \overline{v}_2 = \overline{b}_2$$

The vertical and diagonal constraints are the defining constraints of $\overline{v}_2$.

Example 2

<u>Next Iteration</u>

In the next iteration, solving

$$\underbrace{\begin{bmatrix} -1 & 2 \\ 0 & 1 \end{bmatrix}}_{A_2^{\mathsf{T}}} \overline{u}_2 = \underbrace{\begin{bmatrix} -1 \\ 1 \end{bmatrix}}_{\overline{c}}$$

yields $\overline{u}_2 = \begin{bmatrix} 3 & 1 \end{bmatrix}^{\mathsf{T}}$. Adding 0s for rows not in R produces

$$\overline{u} = \begin{bmatrix} 3 & 0 & 1 \end{bmatrix}^{\mathsf{T}} .$$

Since $\overline{u} \geq \overline{0}$, we are in <u>Case 1</u>. The max is

$$\overline{c}^{\mathsf{T}} \overline{v}_2 = \begin{bmatrix} -1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} = 2$$

at vertex $\overline{v}_2^{\mathsf{T}} = \begin{bmatrix} 0 & 2 \end{bmatrix}$.

Example 3: $x \geq 0 \land y \geq 0 \land x \geq 2 \land y \geq 2 \land x + y \leq 3$

$\Sigma_{\mathbb{Q}}$-formula (8.1)

$$F: \; x \geq 0 \quad \land \quad y \geq 0 \quad \land \quad x \geq 2 \quad \land \quad y \geq 2 \quad \land \quad x + y \leq 3 \,,$$

or, in matrix form,

$$F: \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ -1 & 0 \\ 0 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ -2 \\ -2 \\ 3 \end{bmatrix}$$

Is F $T_{\mathbb{Q}}$-satisfiable?

Example 3: $x \geq 0 \wedge y \geq 0 \wedge x \geq 2 \wedge y \geq 2 \wedge x + y \leq 3$

<u>Step 0</u>

Because x and y are already constrained to be nonnegative, we do not need to introduce new x_1, x_2, y_1, y_2. Rewrite:

$$\underbrace{[1\ \ 1]}_{D_1} \begin{bmatrix} x \\ y \end{bmatrix} \leq \underbrace{[3]}_{\overline{g}_1} \quad \text{and} \quad \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{D_2} \begin{bmatrix} x \\ y \end{bmatrix} \geq \underbrace{\begin{bmatrix} 2 \\ 2 \end{bmatrix}}_{\overline{g}_2}$$

so that $\overline{g}_1 \geq 0$ and $\overline{g}_2 > 0$.

Then (8.2):

$$\begin{array}{ll} \textbf{max} & \overline{1}^{\mathsf{T}}(D_2\overline{x} - \overline{z}) \\ \textbf{subject to} & \\ & \begin{array}{rcl} \overline{x}, \overline{z} & \geq & \overline{0} \\ D_1\overline{x} & \leq & \overline{g}_1 \\ D_2\overline{x} - \overline{z} & \leq & \overline{g}_2 \end{array} \end{array}$$

Example 3: $x \geq 0 \wedge y \geq 0 \wedge x \geq 2 \wedge y \geq 2 \wedge x + y \leq 3$

Expanding, we have

$$
\begin{aligned}
\bar{c}^{\mathsf{T}}\bar{x} &= \bar{1}^{\mathsf{T}} \begin{bmatrix} D_2 & -I \end{bmatrix} \begin{bmatrix} x \\ y \\ z_1 \\ z_2 \end{bmatrix} \\
&= [1 \;\; 1] \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z_1 \\ z_2 \end{bmatrix} \\
&= \underbrace{\begin{bmatrix} 1 & 1 & -1 & -1 \end{bmatrix}}_{\bar{c}^{\mathsf{T}}} \begin{bmatrix} x \\ y \\ z_1 \\ z_2 \end{bmatrix} .
\end{aligned}
$$

Example 3: $x \geq 0 \wedge y \geq 0 \wedge x \geq 2 \wedge y \geq 2 \wedge x + y \leq 3$

obtaining the optimization problem (8.3)

$$\textbf{max} \quad \underbrace{[1 \; 1 \; -1 \; -1]}_{\overline{c}^{\mathsf{T}}} \begin{bmatrix} x \\ y \\ z_1 \\ z_2 \end{bmatrix}$$

$$\textbf{subject to}$$

$$\underbrace{\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}}_{A} \begin{bmatrix} x \\ y \\ z_1 \\ z_2 \end{bmatrix} \leq \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 3 \\ 2 \\ 2 \end{bmatrix}}_{\overline{b}}$$

Example 3: $x \geq 0 \wedge y \geq 0 \wedge x \geq 2 \wedge y \geq 2 \wedge x + y \leq 3$

Use the initial vertex

$$\overline{v}_1 = \begin{bmatrix} x \\ y \\ z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

in Step 1.

F is satisfiable iff the optimal value v_F is equal to

$$\overline{1}^{\mathsf{T}} \overline{g}_2 = [1\ 1] \begin{bmatrix} 2 \\ 2 \end{bmatrix} = 4\ .$$

We use the simplex algorithm to find the optimum.

Example 3: $x \geq 0 \wedge y \geq 0 \wedge x \geq 2 \wedge y \geq 2 \wedge x + y \leq 3$

Step 1

Choose rows $R = [1;2;3;4]$ of A and $\overline{b}$, giving

$$\underbrace{\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}}_{A_1} \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\overline{v}_1} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\overline{b}_1}$$

Solving

$$\underbrace{\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}}_{{A_1}^{\mathsf{T}}} \overline{u}_1 = \underbrace{\begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}}_{\overline{c}}$$

Example 3: $x \geq 0 \wedge y \geq 0 \wedge x \geq 2 \wedge y \geq 2 \wedge x + y \leq 3$

yields $\overline{u}_1 = [-1 \ -1 \ 1 \ 1]^\mathsf{T}$. Adding 0s for the rows not in R produces $\overline{u}$:

$$\overline{u} = \begin{bmatrix} -1 & -1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}^\mathsf{T}.$$

Since $u_1, u_2 < 0$, we are in Case 2 with $k = k' = 1$. Let $\overline{y}$ be the first column of $-A_1^{-1}$: solve

$$\underbrace{\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}}_{A_1} \overline{y} = \underbrace{\begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{-\bar{e}_1}$$

to yield $\overline{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^\mathsf{T}$. Then $S = [5; 6]$; *i.e.*, the 5th and 6th rows $\overline{a}$ of A are such that $\overline{ay} > 0$. Choose the largest λ_1 such that $A(\overline{v}_1 + \lambda_1 \overline{y}) \leq \overline{b}$.

Example 3: $x \geq 0 \wedge y \geq 0 \wedge x \geq 2 \wedge y \geq 2 \wedge x + y \leq 3$

Focusing on the 5th and 6th rows of A (since $S' = [5; 6]$), choose the largest λ_1 such that

$$\underbrace{\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 \end{bmatrix}}_{\text{rows 5,6 of } A} \left(\underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\overline{v}_1} + \lambda_1 \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\overline{y}} \right) \leq \underbrace{\begin{bmatrix} 3 \\ 2 \end{bmatrix}}_{\text{rows 5,6 of } \overline{b}}$$

Namely, choose $\lambda_1 = 2$ (and $\ell = 6$). Then

$$\overline{v}_2 = \overline{v}_1 + \lambda_1 \overline{y} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 2 \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Example 3: $x \geq 0 \wedge y \geq 0 \wedge x \geq 2 \wedge y \geq 2 \wedge x + y \leq 3$

Replace the 1st row of A_1 (since $k' = 1$) by the 6th row of A (since $\ell = 6$) to produce

$$A_2 = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \quad \text{and} \quad \overline{b}_2 = \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Have we made progress? Yes, for

$$\overline{c}^{\mathsf{T}}\overline{v}_1 = 0 \quad < \quad 2 = \overline{c}^{\mathsf{T}}\overline{v}_2 \ .$$

The objective function has increased from 0 to 2.

Example 3: $x \geq 0 \wedge y \geq 0 \wedge x \geq 2 \wedge y \geq 2 \wedge x + y \leq 3$

Step 2

Now $R = [6; 2; 3; 4]$ (the indices of rows of A in A_2). Solve

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}}_{{A_2}^{\mathsf{T}}} \overline{u}_2 = \underbrace{\begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}}_{\overline{c}}$$

to yield

$$\overline{u}_2 = \begin{bmatrix} \underset{6}{1} & \underset{2}{-1} & \underset{3}{0} & \underset{4}{1} \end{bmatrix}^{\mathsf{T}} .$$

Example 3: $x \geq 0 \wedge y \geq 0 \wedge x \geq 2 \wedge y \geq 2 \wedge x + y \leq 3$

Then filling in 0s for the other rows of A produces:

$$\overline{u} = \begin{bmatrix} 0 & \underset{2}{-1} & \underset{3}{0} & \underset{4}{1} & 0 & \underset{6}{1} & 0 \end{bmatrix}^{\mathsf{T}}$$

$u_2 < 0$, so $k = 2$, which corresponds to row $k' = 2$ of $\overline{u}_2$.
According to Case 2, let $\overline{y}$ be the 2nd column of $-A_2^{-1}$: solve $A_2\overline{y} = -e_2$ to yield $\overline{y} = [0\ 1\ 0\ 0]^{\mathsf{T}}$. Then the 5th and 7th rows $\overline{a}$ of A are such that $\overline{ay} > 0$ so that $S = [5; 7]$.

Focusing on the 5th and 7th rows of A, choose the largest λ_2 such that

$$\underbrace{\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}}_{\text{rows 5,7 of } A} \left(\underbrace{\begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\overline{v}_2} + \lambda_2 \underbrace{\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}}_{\overline{y}} \right) \leq \underbrace{\begin{bmatrix} 3 \\ 2 \end{bmatrix}}_{\text{rows 5,7 of } \overline{b}}$$

Example 3: $x \geq 0 \wedge y \geq 0 \wedge x \geq 2 \wedge y \geq 2 \wedge x + y \leq 3$

Choose $\lambda_2 = 1$ (and $\ell = 5$). Then

$$\overline{v}_3 = \overline{v}_2 + \lambda_2 \overline{y} = \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \end{bmatrix} + 1 \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Replace the 2nd row of A_2 (since $k' = 2$) by the 5th row of A (since $\ell = 5$) to produce

$$A_3 = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \quad \text{and} \quad \overline{b}_3 = \begin{bmatrix} 2 \\ 3 \\ 0 \\ 0 \end{bmatrix}$$

Example 3: $x \geq 0 \wedge y \geq 0 \wedge x \geq 2 \wedge y \geq 2 \wedge x + y \leq 3$

Have we made progress? Yes, for

$$\begin{aligned} & \overline{c}^{\mathsf{T}}\overline{v}_1 = 0 \\ < \quad & \overline{c}^{\mathsf{T}}\overline{v}_2 = 2 \\ < \quad & \overline{c}^{\mathsf{T}}\overline{v}_3 = 3 \; . \end{aligned}$$

The objective function has increased from 2 to 3.

Example 3: $x \geq 0 \land y \geq 0 \land x \geq 2 \land y \geq 2 \land x + y \leq 3$

Step 3

Now $R = [6; 5; 3; 4]$. Solve ${A_3}^{\mathsf{T}} \overline{u}_3 = \overline{c}$, yielding $\overline{u}_3 = [0\ 1\ 1\ 1]^{\mathsf{T}}$. Now $\overline{u}_3 \geq \overline{0}$, so we are in Case 1: $\overline{v}_3$ is the optimum with objective value

$$\underbrace{[1\ 1\ \ -1\ \ -1]}_{\overline{c}^{\mathsf{T}}} \underbrace{\begin{bmatrix} 2 \\ 1 \\ 0 \\ 0 \end{bmatrix}}_{\overline{v}_3} = 3\ .$$

Final Step: Satisfiability

The optimal value of the constructed optimization problem is 3, which is less than the required $v_F = 4$ of Step 0. Hence, F is $T_{\mathbb{Q}}$-unsatisfiable.

Linear Programming (Dantzig 1940s)

A *linear programming problem* involves the optimization of a *linear objective function*, subject to *linear inequality constraints*.

max $\overline{c}^{\mathsf{T}}\overline{x}$ (objective function)

subject to $A\overline{x} \leq \overline{b}$ (constraints)

$\overline{x}$ denotes a vector:

max $$\sum_{i=1}^{n} c_i x_i$$

subject to $$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \leq \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

Example: Linear Programming

A company is producing two different products using three machines A, B, and C.

- Product 1 needs A for one, and B for one hour.
- Product 2 needs A for two, B for one, and C for three hours.
- Product 1 can be sold for $300; Product 2 for $500.
- Monthly availability of machines:
  A: 170 hours, B: 150 hours, C 180 hours.

Let x_1 and x_2 denote the projected monthly sale of product 1 and product 2, respectively.
We want to optimize $300x_1 + 500x_2$ subject to:

$$
\begin{aligned}
1x_1 + 2x_2 &\leq 170 && \text{Machine (A)}\\
1x_1 + 1x_2 &\leq 150 && \text{Machine (B)}\\
0x_1 + 3x_2 &\leq 180 && \text{Machine (C)}\\
x_1 \geq 0 \wedge x_2 &\geq 0
\end{aligned}
$$

The Simplex Algorithm

To find the optimal solution proceed as follows:

- start at some vertex of the solution space,
- proceed along adjacent edge to reach a vertex with better cost,
- continue until local optimum is found.

The solution space forms a convex polyhedron.
Therefore local optimum is global optimum.

A Problem with a Simple Vertex

If the problem is of the following shape:

$$
\begin{aligned}
x_1 &\geq 0 \\
&\vdots \\
x_n &\geq 0 \\
A\overline{x} &\leq \overline{b}, \text{where } \overline{b} \geq \overline{0}
\end{aligned}
$$

or (in matrix form)

$$
\begin{bmatrix} -1 & & 0 \\ & \ddots & \\ 0 & & -1 \\ & & \\ & A & \\ & & \end{bmatrix} \overline{x} \leq \begin{bmatrix} 0 \\ \vdots \\ 0 \\ b_1 \\ \dots \\ b_m \end{bmatrix}, \text{where } b_1, \dots, b_m \geq 0,
$$

then a simple (initial) vertex of solution space is $\overline{x} = 0$.

Vertex of $A\overline{x} \leq \overline{b}$ and its dual

An n-vector $\overline{v}$ is a vertex of $A\overline{x} \leq \overline{b}$ if there is nonsingular $n \times n$-submatrix A_0 and corresponding n-subvector $\overline{b}_0$ s.t.

$$A_0\overline{v} = \overline{b}_0 \text{ and } A\overline{v} \leq \overline{b}$$

Move the rows corresponding to A_0 in A and $\overline{b}_0$ in $\overline{b}$ upwards:

$$A = \begin{bmatrix} A_0 \\ * \end{bmatrix} \text{ and } \overline{b} = \begin{bmatrix} \overline{b}_0 \\ * \end{bmatrix}$$

Construct solution $\overline{u}$ of the dual problem $A^{\mathsf{T}}\overline{y} \geq \overline{c}$ as follows:
Since A_0 is invertible, we can solve

$$\boxed{{A_0}^{\mathsf{T}}\overline{u}_0 = \overline{c}}$$

to get $\overline{u}_0$. Set $\overline{u} := \begin{bmatrix} \overline{u}_0 \\ \overline{0} \end{bmatrix}$, then:

$$A^{\mathsf{T}}\overline{u} = \begin{bmatrix} {A_0}^{\mathsf{T}} & * \end{bmatrix} \begin{bmatrix} \overline{u}_0 \\ \overline{0} \end{bmatrix} = {A_0}^{\mathsf{T}}\overline{u}_0 + \overline{0} = \overline{c}\,.$$

Case $\overline{u} \geq \overline{0}$

If $\overline{u} \geq \overline{0}$, then $\overline{v}$ is optimal:
We have

$$\begin{aligned} \overline{c}^{\mathsf{T}}\overline{v} &= (A^{\mathsf{T}}\overline{u})^{\mathsf{T}}\overline{v} \\ &= \overline{u}^{\mathsf{T}}A\overline{v} \\ &= \overline{u}^{\mathsf{T}}\begin{bmatrix} A_0 \\ * \end{bmatrix}\overline{v} \\ &= \begin{bmatrix} \overline{u}_0{}^{\mathsf{T}} & \overline{0} \end{bmatrix}\begin{bmatrix} \overline{b}_0 \\ * \end{bmatrix} \\ &= \overline{u}^{\mathsf{T}}\overline{b} \end{aligned}$$

Let $\overline{x}$ be an arbitrary vector that satisfies $A\overline{x} \leq b$, then:

$$\overline{c}^{\mathsf{T}}\overline{x} = (A^{\mathsf{T}}\overline{u})^{\mathsf{T}}\overline{x} = \overline{u}^{\mathsf{T}}A\overline{x} \underset{\overline{u}\geq\overline{0}}{\leq} \overline{u}^{\mathsf{T}}\overline{b} = \overline{c}^{\mathsf{T}}\overline{v}\ .$$

Hence, $\overline{c}^{\mathsf{T}}\overline{v}$ is maximal.

Case $\overline{u} \ngeq \overline{0}$

If $\overline{u} \ngeq \overline{0}$, there is some coordinate k s.t. $u_k < 0$.
This corresponds to some row of matrix A_0.

Find $\overline{y}$

Solve for $\overline{y}$ in equation

$$\boxed{A_0 \overline{y} = -\bar{e}_k} \ .$$

This is the direction in which we move.
Set $\overline{v}' = \overline{v} + \lambda \overline{y}$, where $\lambda \geq 0$. Then

$$\begin{aligned} A_0 \overline{v}' &= A_0(\overline{v} + \lambda \overline{y}) \\ &= \overline{b}_0 - \lambda \bar{e}_k \\ &\leq \overline{b}_0 \end{aligned}$$

and equality holds for all but the kth row.

Case $\overline{u} \not\geq \overline{0}$

Moreover, $\overline{v}'$ is better than $\overline{v}$:

$$\begin{aligned} \overline{c}^{\mathsf{T}}\overline{y} &= \overline{u}_0^{\mathsf{T}} A_0 \overline{y} \\ &= \overline{u}_0^{\mathsf{T}} (-\overline{e}_k) \\ &= -u_k \\ &> 0 \,. \end{aligned}$$

Hence,

$$\overline{c}^{\mathsf{T}}\overline{v}' \quad = \quad \overline{c}^{\mathsf{T}}\overline{v} + \lambda \underbrace{\overline{c}^{\mathsf{T}}\overline{y}}_{>0} \quad \geq \quad \overline{c}^{\mathsf{T}}\overline{v}$$

How to find λ

Find λ

Now choose λ such that still $A(\overline{v} + \lambda\overline{y}) \leq b$ and equality holds for some constraint $(A)_\ell(\overline{v} + \lambda\overline{y}) = b_\ell$, $\ell > n$.
This gives a better vertex.

For each row $\ell > n$ with $(A)_\ell\overline{y} > 0$, solve λ_ℓ in the equation

$$(A)_\ell(\overline{v} + \lambda_\ell\overline{y}) = b_\ell$$

From $(A)_\ell\overline{v} \leq b_\ell$:

$$0 \quad \leq \quad b_\ell - (A)_\ell\overline{v} \quad = \quad \lambda_\ell(A)_\ell\overline{y}$$

Since $(A)_\ell\overline{y} > 0$, we have $\lambda_\ell \geq 0$.

Choose as λ the smallest λ_ℓ.

The cases for λ

Since $A_0\overline{y} = -\overline{e}_k$,

$$A(\overline{v} + \lambda\overline{y}) \quad \leq \quad \overline{b} + \lambda A\overline{y} \quad = \quad \overline{b} + \lambda \begin{bmatrix} -\overline{e}_k \\ (A)_{n+1}\overline{y} \\ \vdots \\ (A)_m\overline{y} \end{bmatrix}$$

<u>Case 1</u>

There is no $\ell > n$ with $(A)_\ell\overline{y} > 0$. Then $A(\overline{v} + \lambda\overline{y}) \leq b$ holds for all $\lambda \geq 0$ and the maximum value of $\overline{c}^\mathsf{T}x$ is unbounded:

$$\lim_{\lambda\to\infty} \overline{c}^\mathsf{T}(\overline{v} + \lambda\overline{y}) = \lim_{\lambda\to\infty} \left(\overline{c}^\mathsf{T}\overline{v} + \lambda \underbrace{\overline{c}^\mathsf{T}\overline{y}}_{\geq 0} \right) = \infty\,.$$

The cases for λ

Case 2

If λ is the smallest λ_ℓ with $(A)_\ell \overline{y} > 0$, then

$$(A)_\ell(\overline{v} + \lambda\overline{y}) = b_\ell \quad \text{and} \quad A(\overline{v} + \lambda\overline{y}) \leq \overline{b}$$

Thus $\overline{v} + \lambda\overline{y}$ is a better vertex.

Example 4: Linear Programming

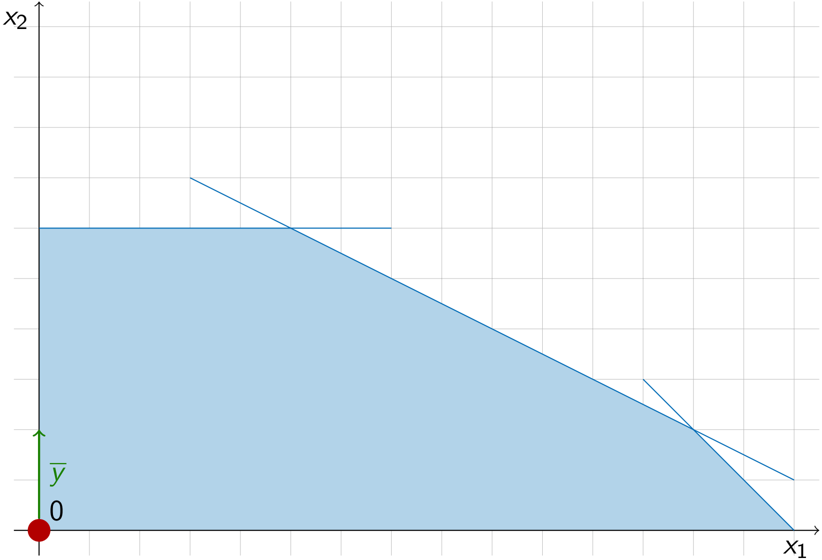
max

$$\underbrace{\begin{bmatrix} 300 & 500 \end{bmatrix}}_{\overline{c}} \overline{x}$$

subject to

$$\underbrace{\begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 1 & 2 \\ 1 & 1 \\ 0 & 3 \end{bmatrix}}_{A} \overline{x} \leq \underbrace{\begin{bmatrix} 0 \\ 0 \\ 170 \\ 150 \\ 180 \end{bmatrix}}_{\overline{b}}$$

Example 4: Linear Programming



Example 4: Linear Programming

$$\bar{v} = \begin{bmatrix} 0 & 0 \end{bmatrix}^{\mathsf{T}} \qquad \underbrace{\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}}_{A_0} \underbrace{\begin{bmatrix} 0 \\ 0 \end{bmatrix}}_{\bar{v}} = \underbrace{\begin{bmatrix} 0 \\ 0 \end{bmatrix}}_{\bar{b}_0}$$

$$\underbrace{\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}}_{A_0^{\mathsf{T}}} \bar{u}_0 = \underbrace{\begin{bmatrix} 300 \\ 500 \end{bmatrix}}_{\bar{c}} \quad \Rightarrow \quad \bar{u} = \begin{bmatrix} -300 & -500 & 0 & 0 & 0 \end{bmatrix}^{\mathsf{T}}$$

$$u_2 = -500 < 0 \quad \Rightarrow \quad \text{choose } k = 2$$

$$\underbrace{\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}}_{A_0} \bar{y} = \underbrace{\begin{bmatrix} 0 \\ -1 \end{bmatrix}}_{-\bar{e}_2} \quad \Rightarrow \quad \bar{y} = \begin{bmatrix} 0 & 1 \end{bmatrix}^{\mathsf{T}}$$

Example 4: Linear Programming

$$\underbrace{\begin{bmatrix}1 & 2\end{bmatrix}}_{(A)_3} \underbrace{\begin{bmatrix}0\\1\end{bmatrix}}_{\bar{y}} > 0 \quad \Rightarrow \quad \begin{bmatrix}1 & 2\end{bmatrix}\left(\begin{bmatrix}0\\0\end{bmatrix} + \lambda_3 \begin{bmatrix}0\\1\end{bmatrix}\right) = 170$$

$$\Rightarrow \quad \lambda_3 = 85$$

$$\underbrace{\begin{bmatrix}1 & 1\end{bmatrix}}_{(A)_4} \underbrace{\begin{bmatrix}0\\1\end{bmatrix}}_{\bar{y}} > 0 \quad \Rightarrow \quad \begin{bmatrix}1 & 1\end{bmatrix}\left(\begin{bmatrix}0\\0\end{bmatrix} + \lambda_4 \begin{bmatrix}0\\1\end{bmatrix}\right) = 150$$

$$\Rightarrow \quad \lambda_4 = 150$$

$$\underbrace{\begin{bmatrix}0 & 3\end{bmatrix}}_{(A)_5} \underbrace{\begin{bmatrix}0\\1\end{bmatrix}}_{\bar{y}} > 0 \quad \Rightarrow \quad \begin{bmatrix}0 & 3\end{bmatrix}\left(\begin{bmatrix}0\\0\end{bmatrix} + \lambda_5 \begin{bmatrix}0\\1\end{bmatrix}\right) = 180$$

$$\Rightarrow \quad \lambda_5 = 60$$

Example 4: Linear Programming

Thus $\lambda = \lambda_5 = 60$, $\ell = 5$, and

$$\overline{v}' = \underbrace{\begin{bmatrix} 0 \\ 0 \end{bmatrix}}_{\overline{v}} + \underbrace{60}_{\lambda} \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{\overline{y}} = \begin{bmatrix} 0 \\ 60 \end{bmatrix} .$$

Example 4: Linear Programming

max

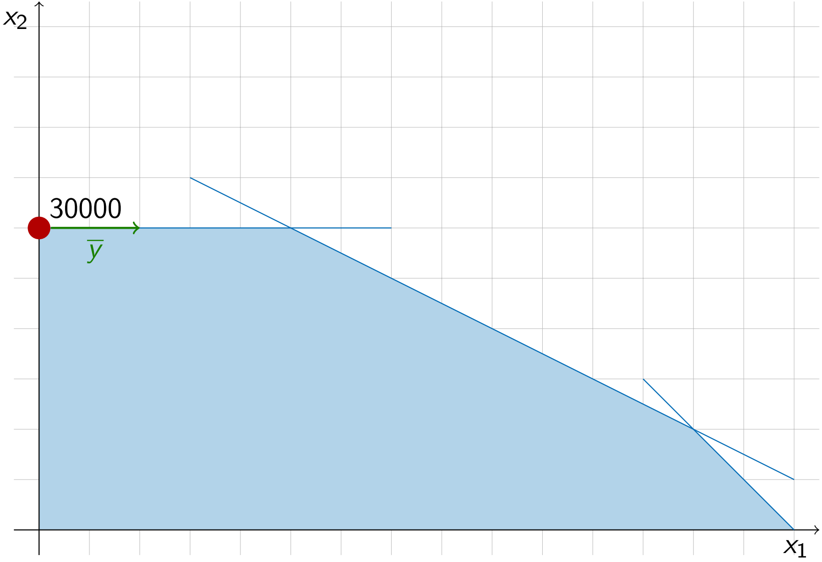
$$\begin{bmatrix} 300 & 500 \end{bmatrix} \overline{x}$$

subject to

$$\begin{bmatrix} -1 & 0 \\ 0 & 3 \\ 0 & -1 \\ 1 & 2 \\ 1 & 1 \end{bmatrix} \overline{x} \leq \begin{bmatrix} 0 \\ 180 \\ 0 \\ 170 \\ 150 \end{bmatrix}$$

$\ell = 5 \Rightarrow k = 2$
(not swap, but okay)

Example 4: Linear Programming



Example 4: Linear Programming

$$\bar{v} = \begin{bmatrix} 0 & 60 \end{bmatrix}^{\mathsf{T}} \qquad \underbrace{\begin{bmatrix} -1 & 0 \\ 0 & 3 \end{bmatrix}}_{A_0} \underbrace{\begin{bmatrix} 0 \\ 60 \end{bmatrix}}_{\bar{v}} = \underbrace{\begin{bmatrix} 0 \\ 180 \end{bmatrix}}_{\bar{b}_0}$$

$$\underbrace{\begin{bmatrix} -1 & 0 \\ 0 & 3 \end{bmatrix}}_{A_0^{\mathsf{T}}} \bar{u}_0 = \underbrace{\begin{bmatrix} 300 \\ 500 \end{bmatrix}}_{\bar{c}} \quad \Rightarrow \quad \bar{u} = \begin{bmatrix} -300 & 166\frac{2}{3} & 0 & 0 & 0 \end{bmatrix}^{\mathsf{T}}$$

$$u_1 = -300 < 0 \quad \Rightarrow \quad \text{choose } k = 1$$

$$\underbrace{\begin{bmatrix} -1 & 0 \\ 0 & 3 \end{bmatrix}}_{A_0} \bar{y} = \underbrace{\begin{bmatrix} -1 \\ 0 \end{bmatrix}}_{-\bar{e}_1} \quad \Rightarrow \quad \bar{y} = \begin{bmatrix} 1 & 0 \end{bmatrix}^{\mathsf{T}}$$

Example 4: Linear Programming

$$\underbrace{\begin{bmatrix} 0 & -1 \end{bmatrix}}_{(A)_3} \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{\bar{y}} = 0$$

$$\underbrace{\begin{bmatrix} 1 & 2 \end{bmatrix}}_{(A)_4} \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{\bar{y}} > 0 \quad \Rightarrow \quad \begin{bmatrix} 1 & 2 \end{bmatrix} \left(\begin{bmatrix} 0 \\ 60 \end{bmatrix} + \lambda_4 \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) = 170$$

$$\Rightarrow \quad \lambda_4 = 50$$

$$\underbrace{\begin{bmatrix} 1 & 1 \end{bmatrix}}_{(A)_5} \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{\bar{y}} > 0 \quad \Rightarrow \quad \begin{bmatrix} 1 & 1 \end{bmatrix} \left(\begin{bmatrix} 0 \\ 60 \end{bmatrix} + \lambda_5 \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) = 150$$

$$\Rightarrow \quad \lambda_5 = 90$$

Example 4: Linear Programming

Since $(A)_3\overline{y} = 0$, $\lambda_4 = 50$, and $\lambda_5 = 90$, we have $\lambda = 50$ and $\ell = 4$, so

$$\overline{v}' = \underbrace{\begin{bmatrix} 0 \\ 60 \end{bmatrix}}_{\overline{v}} + \underbrace{50}_{\lambda} \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{\overline{y}} = \begin{bmatrix} 50 \\ 60 \end{bmatrix} .$$

Example 4: Linear Programming

max

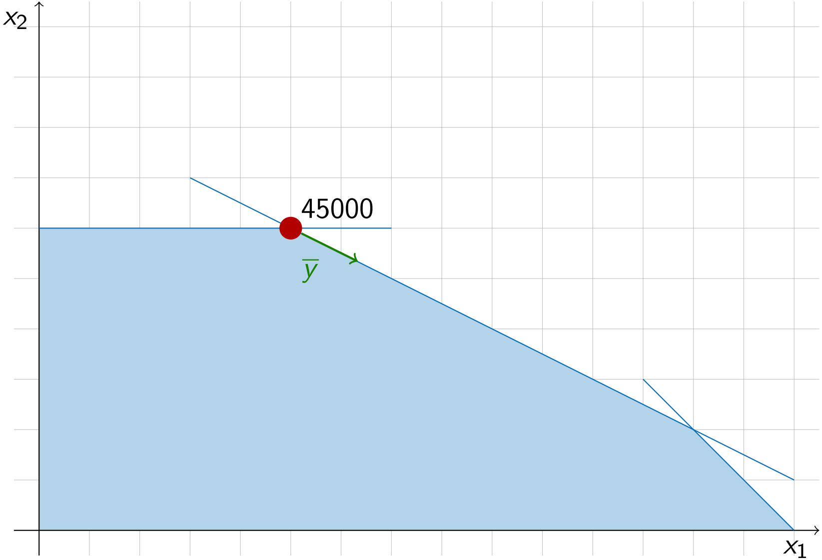
$$\begin{bmatrix} 300 & 500 \end{bmatrix} \overline{x}$$

subject to

$$\begin{bmatrix} 1 & 2 \\ 0 & 3 \\ -1 & 0 \\ 0 & -1 \\ 1 & 1 \end{bmatrix} \overline{x} \leq \begin{bmatrix} 170 \\ 180 \\ 0 \\ 0 \\ 150 \end{bmatrix}$$

$\ell = 4 \Leftrightarrow k = 1$ (swap)

Example 4: Linear Programming



Example 4: Linear Programming

$$\bar{v} = \begin{bmatrix} 50 & 60 \end{bmatrix}^{\mathsf{T}} \qquad \underbrace{\begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}}_{A_0} \underbrace{\begin{bmatrix} 50 \\ 60 \end{bmatrix}}_{\bar{v}} = \underbrace{\begin{bmatrix} 170 \\ 180 \end{bmatrix}}_{\bar{b}_0}$$

$$\underbrace{\begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix}}_{A_0^{\mathsf{T}}} \bar{u}_0 = \underbrace{\begin{bmatrix} 300 \\ 500 \end{bmatrix}}_{\bar{c}} \quad \Rightarrow \quad \bar{u} = \begin{bmatrix} 300 & -33\frac{1}{3} & 0 & 0 & 0 \end{bmatrix}^{\mathsf{T}}$$

$$u_2 = -33\frac{1}{3} < 0 \quad \Rightarrow \quad \text{choose } k = 2$$

$$\underbrace{\begin{bmatrix} 1 & 2 \\ 0 & 3 \end{bmatrix}}_{A_0} \bar{y} = \underbrace{\begin{bmatrix} 0 \\ -1 \end{bmatrix}}_{-\bar{e}_2} \quad \Rightarrow \quad \bar{y} = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} \end{bmatrix}^{\mathsf{T}}$$

Example 4: Linear Programming

$$\underbrace{\begin{bmatrix} -1 & 0 \end{bmatrix}}_{(A)_3} \underbrace{\begin{bmatrix} \frac{2}{3} \\ -\frac{1}{3} \end{bmatrix}}_{\bar{y}} < 0$$

$$\underbrace{\begin{bmatrix} 0 & -1 \end{bmatrix}}_{(A)_4} \underbrace{\begin{bmatrix} \frac{2}{3} \\ -\frac{1}{3} \end{bmatrix}}_{\bar{y}} > 0 \quad \Rightarrow \quad \underbrace{\begin{bmatrix} 0 & -1 \end{bmatrix}}_{(A)_4} \left(\underbrace{\begin{bmatrix} 50 \\ 60 \end{bmatrix}}_{\bar{v}} + \lambda_4 \underbrace{\begin{bmatrix} \frac{2}{3} \\ -\frac{1}{3} \end{bmatrix}}_{\bar{y}} \right) = \underbrace{0}_{b_4}$$

$$\Rightarrow \quad \lambda_4 = 180$$

$$\underbrace{\begin{bmatrix} 1 & 1 \end{bmatrix}}_{(A)_5} \underbrace{\begin{bmatrix} \frac{2}{3} \\ -\frac{1}{3} \end{bmatrix}}_{\bar{y}} > 0 \quad \Rightarrow \quad \underbrace{\begin{bmatrix} 1 & 1 \end{bmatrix}}_{(A)_5} \left(\underbrace{\begin{bmatrix} 50 \\ 60 \end{bmatrix}}_{\bar{v}} + \lambda_5 \underbrace{\begin{bmatrix} \frac{2}{3} \\ -\frac{1}{3} \end{bmatrix}}_{\bar{y}} \right) = \underbrace{150}_{b_5}$$

$$\Rightarrow \quad \lambda_5 = 120$$

Example 4: Linear Programming

Since $(A)_3\overline{y} < 0$, $\lambda_4 = 180$, and $\lambda_5 = 120$,
we have $\lambda = 120$ and $\ell = 5$, so

$$\overline{v}' = \underbrace{\begin{bmatrix} 50 \\ 60 \end{bmatrix}}_{\overline{v}} + \underbrace{120}_{\lambda} \underbrace{\begin{bmatrix} \frac{2}{3} \\ -\frac{1}{3} \end{bmatrix}}_{\overline{y}} = \begin{bmatrix} 130 \\ 20 \end{bmatrix} .$$

Example 4: Linear Programming

max

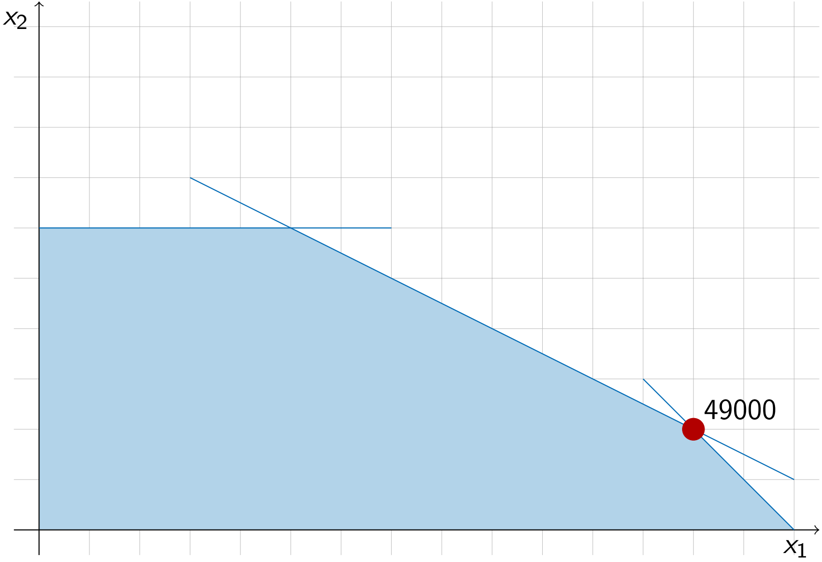
$$\begin{bmatrix} 300 & 500 \end{bmatrix} \overline{x}$$

subject to

$$\begin{bmatrix} 1 & 2 \\ 1 & 1 \\ -1 & 0 \\ 0 & -1 \\ 0 & 3 \end{bmatrix} \overline{x} \leq \begin{bmatrix} 170 \\ 150 \\ 0 \\ 0 \\ 180 \end{bmatrix}$$

$\ell = 5 \Leftrightarrow k = 2$ (swap)

Example 4: Linear Programming



Example 4: Linear Programming

$$\overline{v} = \begin{bmatrix}130 & 20\end{bmatrix}^{\mathsf{T}}$$

$$\underbrace{\begin{bmatrix}1 & 2\\ 1 & 1\end{bmatrix}}_{A_0}\underbrace{\begin{bmatrix}130\\ 20\end{bmatrix}}_{\overline{v}} = \underbrace{\begin{bmatrix}170\\ 150\end{bmatrix}}_{\overline{b}_0}$$

$$\underbrace{\begin{bmatrix}1 & 1\\ 2 & 1\end{bmatrix}}_{A_0^{\mathsf{T}}}\overline{u}_0 = \underbrace{\begin{bmatrix}300\\ 500\end{bmatrix}}_{\overline{c}} \quad \Rightarrow \quad \overline{u} = \begin{bmatrix}200 & 100 & 0 & 0 & 0\end{bmatrix}^{\mathsf{T}}$$

Since $\overline{u} \geq 0$, we have reached the maximum, with

$$\overline{x} = \begin{bmatrix}130\\ 20\end{bmatrix} .$$

Example 4: Linear Programming

Finally, therefore,

$$\textbf{max} = \underbrace{\begin{bmatrix} 300 & 500 \end{bmatrix}}_{\overline{c}^{\mathsf{T}}} \underbrace{\begin{bmatrix} 130 \\ 20 \end{bmatrix}}_{\overline{x}} = 49000 \; .$$