CS257: Introduction to Automated Reasoning

QF_LRA

Stanford University

Overview

SMT solvers can be used to solve arithmetic problems

Linear Programs (LPs) are a particularly interesting class of arithmetic problems, with stand-alone solvers

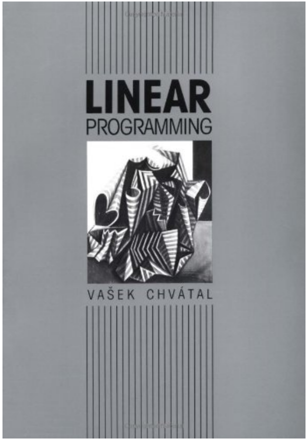
Many interesting applications: robotic planning, formal verification, operations research

Some of the slides are contributed by Guy Katz.

Outline

- QF_LRA
- Linear Programming
- The Simplex algorithm

Readings: DP 5.1-5.2 and optionally...



=

Linear Programs: Historical Context

- Dates back at least to 19^{th} century
  - A procedure now called Fourier-Motzkin elimination first proposed by Joseph Fourier in 1826 and re-discovered by Theodore Motzkin in 1936
- More interests during and after WW2
  - 1939: Leonid Kantorovich formulated the problem of Linear Programming and developed a decision procedure (won Nobel prize in economics in 1975)
  - 1946: George Dantzig (Stanford professor 1966–2005) invented the **Simplex method**
    - Simplex still used extensively (in Operations Research)
    - Our focus today!
- 1979: first shown to be solvable in polynomial time by Leonid Khachiyan
  - 1984: Interior-point method invented by Narendra Karmarkar

Review: Theory of Real Arithmetics ($\mathcal{T}_{RA}$)

Equality: Yes

$\Sigma^S = \{\mathsf{R}\}$

$\Sigma^F = \{+, -, *, \leq, q_i$ for each rational number constant $i\}$

$\mathcal{S}$ is the class of structures that interprets R as the set of real numbers, and the functions in the usual way ($sort(q_i) = \langle \mathsf{R} \rangle$)

Quantifier-free linear real arithmetic (QF_LRA): 1) no quantifiers; 2) $*$ can only appear if at least one of the two operands is a rational constant.

Many SMT solvers (e.g., Z3, cvc5) implement Simplex as the theory solver for $\mathcal{T}_{RA}$

Linear Programming

A **linear programming (LP)** instance includes:

- An $m \times n$ matrix A called the constraint matrix
- An m-dimensional vector b
- An n-dimensional vector c (the objective function)

The goal: find a solution x that maximizes $c^T x$ subject to the linear inequality constraints $Ax \leq b$

Example and Terminology

Maximize $2x_2 - x_1$ subject to:

$$x_1 + x2 \leq 3$$
$$2x_1 - x_2 \leq -5$$

Here:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad A = \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix} \quad b = \begin{bmatrix} 3 \\ -5 \end{bmatrix} \quad c = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$$

Find x that maximizes $c^T x$, subject to $Ax \leq b$

Example and Terminology

Maximize $2x_2 - x_1$ subject to:

$$x_1 + x2 \leq 3$$
$$2x_1 - x_2 \leq -5$$

If a particular assignment of x satisfies $Ax \leq b$, we call it a **feasible solution**

Otherwise, it is an **infeasible solution**

Is $\langle 0, 0 \rangle$ a feasible solution?

Is $\langle -2, 1 \rangle$ a feasible solution?

For a given assignment of x, the value of $c^T x$ is the **objective value** (or cost) of x

What is the objective value of $\langle -2, 1 \rangle$?

Example and Terminology

A feasible solution with a maximal objective value (over all feasible solutions) is called an **optimal solution**

If a linear program has no feasible solutions, the linear program is **infeasible**

If the optimal solution's objective value is ∞, the linear program is called **unbounded**

Geometric Interpretation

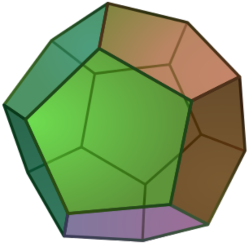
For an $m \times n$ constraint matrix A, the set of points $P = \{x \mid Ax \leq b\}$ form a **convex polytope** in n-dimensional space

Polytope: the generalization of polyhedron from 3 dimensional space to higher dimensions

Convexity: for all $v_1, v_2 \in R^n$, if $v_1, v_2 \in P$, then for all $\lambda \in [0, 1], \lambda v_1 + (1 - \lambda) v_2 \in P$

In other words, every point on the line segment connecting two points in P is also in P

Goal: find a point in the polytope that maximizes $c^T x$

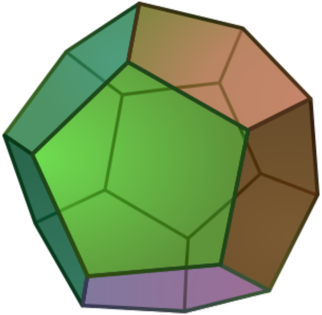


Geometric Interpretation

The LP is infeasible if the polytope is empty

The LP is unbounded if the polytope is open in the direction of the objective function

The optimal solution for a bounded LP must lie on a vertex of the polytope



Satisfiability as Linear Programming

Goal: use LP to check the satisfiability of quantifier-free conjunctive $\mathcal{T}_{RA}$-formulas

Step 1: convert equalities to inequalities

A $\mathcal{T}_{LRA}$-equality can be written in the form $a^T x_i = b$

We rewrite this as $a^T x_i \geq b \wedge a^T x_i \leq b$

Satisfiability as Linear Programming

Goal: use LP to check the satisfiability of quantifier-free conjunctive $\mathcal{T}_{RA}$-formulas

Step 2: handle strict inequalities

A $\mathcal{T}_{LRA}$-literal is of the form $a^T x_i \leq b$ or $\neg a^T x_i \leq b$

$a^T x_i \leq b$ is already in the desired form

For the latter:

$\neg a^T x_i \leq b$
$\Leftrightarrow a^T x_i > b$
$\Leftrightarrow -a^T x_i < -b$
$\Leftrightarrow -a^T x_i + y \leq -b \wedge y > 0$

Note: y is a new variable and the same y is used in all atoms

Example: What is the result of rewriting $\neg(2x_1 - x_2 \leq 3)$?

Now, the formula is of the form $Ax \leq b \wedge y > 0$

Satisfiability as Linear Programming

Step 3: To check the satisfiability of $Ax \leq b \wedge y > 0$, encode the following LP:

Maximize y subject to $Ax \leq b$

The formula is satisfiable if and only if the optimal value is positive

Methods for solving LPs:

- Ellipsoid method (Khachian, 1979) Polynomial time
- Interior-point algorithm (Karmarkar, 1984) Polynomial time
- Simplex algorithm (Dantzig, 1949) Exponential time (probably)

Still, Simplex remains the most popular

Standard Form

The general form of LP is to maximize y subject to a system of inequalities.

However, the algorithm is easier to present if we make the additional assumption that all variables are non-negative:

$$\begin{aligned}
\text{maximize} \quad & \sum_{j=1}^{n} c_j x_j \\
\text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j \le b_i \quad \text{for } i = 1, 2, \ldots, m \\
& x_j \ge 0 \quad \text{for } j = 1, 2, \ldots, n
\end{aligned}$$

We call this the **standard form**.

The algorithm we present is still general because any LP can be transformed to standard form.

Standard Form

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{n} c_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j \le b_i \quad \text{for } i = 1, 2, \ldots, m \\ & x_j \ge 0 \quad \text{for } j = 1, 2, \ldots, n \end{aligned}$$

Running example:

$$\begin{aligned} \max \quad & 5x_1 + 4x_2 + 3x_3 \\ \text{s.t.} \quad & 2x_1 + 3x_2 + x_3 \le 5 \\ & 4x_1 + x_2 + 2x_3 \le 11 \\ & 3x_1 + 4x_2 + 2x_3 \le 8 \\ & x_1, x_2, x_3 \ge 0 \end{aligned}$$

Slack Variables

Observe the first equation

$$2x_1 + 3x_2 + x_3 \leq 5$$

Define a new variable to represent the **slack**:

$$x_4 = 5 - 2x_1 - 3x_2 - x_3, \quad x_4 \geq 0$$

Do this to every each constraint so everything becomes equalities

Define a new variable to represent the objective value:

$$z = 5x_1 + 4x_2 + 3x_3$$

$$\begin{aligned} \max \quad & 5x_1 + 4x_2 + 3x_3 \\ \text{s.t.} \quad & 2x_1 + 3x_2 + x_3 \leq 5 \\ & 4x_1 + x_2 + 2x_3 \leq 11 \\ & 3x_1 + 4x_2 + 2x_3 \leq 8 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

Slack Variables

$$\begin{aligned} \max \quad & 5x_1 + 4x_2 + 3x_3 \\ \text{s.t.} \quad & 2x_1 + 3x_2 + x_3 \le 5 \\ & 4x_1 + x_2 + 2x_3 \le 11 \\ & 3x_1 + 4x_2 + 2x_3 \le 8 \\ & x_1, x_2, x_3 \ge 0 \end{aligned} \quad \Rightarrow \quad \begin{aligned} \max \quad & z \\ \text{s.t.} \quad & x_4 = 5 - 2x_1 - 3x_2 - x_3 \\ & x_5 = 11 - 4x_1 - x_2 - 2x_3 \\ & x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \\ & z = 0 + 5x_1 + 4x_2 + 3x_3 \\ & x_1, x_2, x_3, x_4, x_5, x_6 \ge 0 \end{aligned}$$

New variables are called **slack variables**

Optimal solution remains optimal for the new problem

The Simplex Strategy

- Start with a feasible solution
  - For our example, set all original variables to 0
  - $x_4 = 5, x_5 = 11, x_6 = 8, x_1, x_2, x_3, z = 0$
- Iteratively improve the objective value
  - Go from x to x' only if $z(x) \leq z(x')$

$$\begin{aligned} x_4 &= 5 - 2x_1 - 3x_2 - x_3 \\ x_5 &= 11 - 4x_1 - x_2 - 2x_3 \\ x_6 &= 8 - 3x_1 - 4x_2 - 2x_3 \\ z &= 0 + 5x_1 + 4x_2 + 3x_3 \end{aligned}$$

What can we improve here?

One option: make x_1 larger, leave x_2, x_3 as is

- $x_1 \mapsto 1 \Rightarrow z \mapsto 5$
- $x_1 \mapsto 2 \Rightarrow z \mapsto 10$
- $x_1 \mapsto 3 \Rightarrow z \mapsto 15$

But x_4, x_5, x_6 become negative now, so the solution is no longer feasible

The Simplex Strategy

Moral of the story:

- Can't increase x_1 too much
- Increase it as much as possible, without harming feasibility

$$\begin{aligned} x_4 &= 5 - 2x_1 - 3x_2 - x_3 \\ x_5 &= 11 - 4x_1 - x_2 - 2x_3 \\ x_6 &= 8 - 3x_1 - 4x_2 - 2x_3 \\ z &= 0 + 5x_1 + 4x_2 + 3x_3 \end{aligned} \quad \Longrightarrow \quad x_1 \leq \frac{5}{2}, x_1 \leq \frac{11}{4}, x_1 \leq \frac{8}{3}$$

Select the tightest bound, $x_1 \leq \frac{5}{2}$

- New assignment:$x_1 = \frac{5}{2}, x_2 = x_3 = x_4 = 0, x_5 = 1, x_6 = \frac{1}{2}$
- This gives $z = \frac{25}{2}$, which is indeed an improvement

The Simplex Strategy

Currently, $x_1 = \frac{5}{2}, x_2 = x_3 = x_4 = 0, x_5 = 1, x_6 = \frac{1}{2}$ and $z = \frac{25}{2}$

How do we continue?

For the first iteration we had:

- A feasible solution ✓
- An equation system, where
  - variables with positive value are expressed in terms of variables with 0 values

$$x_4 = 5 - 2x_1 - 3x_2 - x_3$$
$$x_5 = 11 - 4x_1 - x_2 - 2x_3$$
$$x_6 = 8 - 3x_1 - 4x_2 - 2x_3$$
$$z = 0 + 5x_1 + 4x_2 + 3x_3$$

Does the current equation system satisfy this property?

Need to update the equations

The Simplex Strategy

$$\begin{aligned} x_4 &= 5 - 2x_1 - 3x_2 - x_3 \\ x_5 &= 11 - 4x_1 - x_2 - 2x_3 \\ x_6 &= 8 - 3x_1 - 4x_2 - 2x_3 \\ z &= 0 + 5x_1 + 4x_2 + 3x_3 \end{aligned}$$

What should we change?

- Initially: x_1 was 0, x_4 was positive
- Now: x_4 is 0, x_1 is positive

Isolate x_1, eliminate from right-hand-side

$$x_4 = 5 - 2x_1 - 3x_2 - x_3 \Rightarrow x_1 = \tfrac{5}{2} - \tfrac{3}{2}x_2 - \tfrac{1}{2}x_3 - \tfrac{1}{2}x_4$$

$$\begin{aligned} x_4 &= 5 - 2x_1 - 3x_2 - x_3 \\ x_5 &= 11 - 4x_1 - x_2 - 2x_3 \\ x_6 &= 8 - 3x_1 - 4x_2 - 2x_3 \\ z &= 0 + 5x_1 + 4x_2 + 3x_3 \end{aligned} \quad\Longrightarrow\quad \begin{aligned} x_1 &= \tfrac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4 \\ x_5 &= 1 + 5x_2 + 2x_4 \\ x_6 &= \tfrac{1}{2} + \frac{1}{2}x_2 - \frac{1}{2}x_3 + \frac{3}{2}x_4 \\ z &= \tfrac{25}{2} - \frac{7}{2}x_2 + \frac{1}{2}x_3 - \frac{5}{2}x_4 \end{aligned}$$

The Simplex Strategy

$$
\begin{aligned}
x_1 &= \tfrac{5}{2} && -\frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4 \\
x_5 &= 1 && +5x_2 \qquad\qquad\quad + 2x_4 \\
x_6 &= \tfrac{1}{2} && +\frac{1}{2}x_2 - \frac{1}{2}x_3 + \frac{3}{2}x_4 \\
z &= \tfrac{25}{2} && -\frac{7}{2}x_2 + \frac{1}{2}x_3 - \frac{5}{2}x_4
\end{aligned}
$$

How can we improve z further?

- Option 1: decrease x_2 or x_4 but we can't since $x_2, x_4 \geq 0$
- Option 2: increase x_3 By how much?

Bounds of x_3: $x_3 \leq 5, x_3 \leq 1, x_3 \leq \infty$

So we increase x_3 to 1

- New assignment:$x_1 = 2, x_2 = 0, x_3 = 1, x_4 = 0, x_5 = 1, x_6 = 0$
- This gives $z = 13$, which is again an improvement

The Simplex Strategy

As before, we switch x_6 and x_3, and eliminate x_3 from right-hand-side.

$$
\begin{aligned}
x_1 &= \tfrac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4 \\
x_5 &= 1 + 5x_2 + 2x_4 \\
x_6 &= \tfrac{1}{2} + \frac{1}{2}x_2 - \frac{1}{2}x_3 + \frac{3}{2}x_4 \\
z &= \tfrac{25}{2} - \frac{7}{2}x_2 + \frac{1}{2}x_3 - \frac{5}{2}x_4
\end{aligned}
\quad \Rightarrow \quad
\begin{aligned}
x_1 &= 2 - 2x_2 - 2x_4 + x_6 \\
x_5 &= 1 + 5x_2 + 2x_4 \\
x_3 &= 1 + x_2 + 3x_4 - 2x_6 \\
z &= 13 - 3x_2 - x_4 - x_6
\end{aligned}
$$

The Simplex Strategy

Can we improve z further?

- No, because $x_2, x_4, x_6 \geq 0$
- And all appear with negative signs in the objective function

So we are done, and maximal value of z is 13

Optimal solution is $x_1 = 2, x_2 = 0, x_3 = 1, x_4 = 0$

$$
\begin{aligned}
x_1 &= 2 \quad -2x_2 - 2x_4 + x_6 \\
x_5 &= 1 \quad +5x_2 + 2x_4 \\
x_3 &= 1 \quad +x_2 + 3x_4 - 2x_6 \\
z &= 13 \quad -3x_2 - x_4 - x_6
\end{aligned}
$$

The Simplex Algorithm

$$
\begin{aligned}
\text{maximize} \quad & \sum_{j=1}^{n} c_j x_j \\
\text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j \le b_i \quad \text{for } i = 1, 2, \ldots, m \\
& x_j \ge 0 \quad \text{for } j = 1, 2, \ldots, n
\end{aligned}
$$

1. Introduce slack variables $x_{n+1}, \ldots, x_{n+m}$
2. Set $x_{n+i} = b_i - \sum_{j=1}^{n} a_{ij} x_j$
3. Start with initial feasible solution $\overline{x}_0$
4. If some addends in the current objective function have positive coefficients, update the feasible solution (to improve the objective value). Otherwise, the current solution is the optimal.
5. Update the equations
6. Go to step 4

Updating the Equations: Pivoting

As we progress towards the optimal solution, equations are updated
This computational process of constructing the new equation system is called **pivoting**

$$
\begin{aligned}
x_4 &= 5 - 2x_1 - 3x_2 - x_3 \\
x_5 &= 11 - 4x_1 - x_2 - 2x_3 \\
x_6 &= 8 - 3x_1 - 4x_2 - 2x_3 \\
z &= 0 + 5x_1 + 4x_2 + 3x_3
\end{aligned}
$$

Invariants:

- Number of equations (m) never changes
- Variables are eiter left hand side or right hand side, never both
  - Left hand side variables are called **basic**
  - Right hand side variables are called **non-basic**
- Non-basic variables always pressed against their bounds (always 0)
- Basic variable assignment determined by non-basic assignment and equations

Updating the Equations: Pivoting

$$
\begin{aligned}
x_4 &= 5 - 2x_1 - 3x_2 - x_3 \\
x_5 &= 11 - 4x_1 - x_2 - 2x_3 \\
x_6 &= 8 - 3x_1 - 4x_2 - 2x_3 \\
z &= 0 + 5x_1 + 4x_2 + 3x_3
\end{aligned}
$$

The set of basic variables is called **the basis**

In the pivoting step:

- A non-basic variable enters the basis (the entering variable)
- A basic variable leaves the basis (the leaving variable)

How is the entering variable chosen? To increase the value of z

One strategy (**Dantzig's rule**) picks the variable with the largest coefficient

How is the leaving variable chosen? To maintain feasibility

Select the basic variable corresponding to the tightest upper-bound

Tableau and Implementation

We have presented the equation system as a "dictionary"

A more popular version is called a tableau:

$$\begin{aligned} x_4 &= 5 - 2x_1 - 3x_2 - x_3 \\ x_5 &= 11 - 4x_1 - x_2 - 2x_3 \\ x_6 &= 8 - 3x_1 - 4x_2 - 2x_3 \\ z &= 0 + 5x_1 + 4x_2 + 3x_3 \end{aligned} \quad \Rightarrow$$

x_1	x_2	x_3	x_4	x_5	x_6	RHS
2	3	1	1	0	0	5
4	1	2	1	1	0	11
3	4	2	1	0	1	8
5	4	3	0	0	0	0

The pivoting process can be understood as a series of matrix operations

Some Pitfalls

Possible problems of the procedure that we described so far:

- Initialization: how to obtain an initial feasible solution?
- Termination: can we encounter an endless sequence of dictionaries without reaching an optimal z?

Pitfalls: initialization

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{n} c_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j \le b_i \quad \text{for } i = 1, 2, \ldots, m \\ & x_j \ge 0 \quad \text{for } j = 1, 2, \ldots, n \end{aligned}$$

Easy when all b_i's are non-negative

What can we do for negative b_i's?

Pitfalls: initialization

Solution: switch to an auxiliary problem with a known feasible solution

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{n} c_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j \le b_i \quad \text{for } i = 1, 2, \ldots, m \\ & x_j \ge 0 \quad \text{for } j = 1, 2, \ldots, n \end{aligned}$$

becomes

$$\begin{aligned} \text{minimize} \quad & x_0 \\ \text{s.t.} \quad & \left(\sum_{j=1}^{n} a_{ij} x_j \right) - x_0 \le b_i \quad \text{for } i = 1, 2, \ldots, m \\ & x_j \ge 0 \quad \text{for } j = 0, 1, 2, \ldots, n \end{aligned}$$

Pitfalls: initialization

$$\begin{aligned} \text{minimize} \quad & x_0 \\ \text{s.t.} \quad & \left(\sum_{j=1}^{n} a_{ij} x_j\right) - x_0 \le b_i \quad \text{for } i = 1, 2, \ldots, m \\ & x_j \ge 0 \quad \text{for } j = 0, 1, 2, \ldots, n \end{aligned}$$

For the auxiliary problem, a feasible solution is easy to find: set $x_1, \ldots, x_n = 0$, and make x_0 sufficiently large

Original problem has a solution if and only if the optimal solution for the auxiliary problem has $x_0 = 0$

Initialization: example

$$\begin{aligned} \max \quad & x_1 + 2x_2 \\ \text{s.t.} \quad & 2x_1 - 3x_2 \le -2 \\ & 4x_1 - x_2 \le -4 \\ & x_1, x_2 \ge 0 \end{aligned} \quad \Rightarrow \quad \begin{aligned} \max \quad & -x_0 \\ \text{s.t.} \quad & 2x_1 - 3x_2 - x_0 \le -2 \\ & 4x_1 - x_2 - x_0 \le -4 \\ & x_0, x_1, x_2 \ge 0 \end{aligned}$$

Initial feasible solution: $x_0 = 4, x_1 = 0, x_2 = 0$

The dictionary of the auxilliary problem:

$$\begin{aligned} x_3 &= -2 - 2x_1 + 3x_2 + x_0 \\ x_4 &= -4 - 4x_1 + x_2 + x_0 \\ z &= -x_0 \end{aligned}$$

Any issues? Variables on the right-hand side need to be 0

Solution: perform a pivot step to move x_0 into the basis

$$\begin{aligned} x_3 &= 2 + 2x_1 + 2x_2 + x_4 \\ x_0 &= 4 + 4x_1 - x_2 + x_4 \\ z &= -4 - 4x_1 + x_2 - x_4 \end{aligned}$$

The Two Steps of Simplex

Traditionally, the optimization problem is divided into two phases:

Phase I: Find a feasible solution

Phase II: Optimize the objective function

But behind the scenes, there is only Phase II

Pitfalls: Termination

Recall the goal of every iteration is to increase z

In each pivoting step, we swap a non-basic variable with a basic variable:

- The non-basic (entering) variable has a positive coefficient in the objective function
- If no such variable exists, the objective function is optimal and we can stop
- The leaving variable is the one imposing the tightest constraint

An iteration will never make z worse

So when might we not converge to the optimal z?

Pitfalls: Terminations

Theorem: if the simplex method fails to terminate, it must be **cycling** (i.e., same dictionary is repeated infinitely often

Proof sketch:

1. there are only finitely many bases;
2. each bases uniquely defines the dictionary;
3. Therefore, there are only finitely many values of z to try

If simplex is cycling, then z has to stop increasing

Degenerate Pivots

Consider the following case:

$$\begin{aligned} x_1 &= \quad -2x_2 + 3x_3 \\ z &= 5x_2 - x_3 + 4x_4 \end{aligned}$$

Dantzig's rule: pick x_2 as the entering variable

Leaving variable is x_1, but its value cannot increase

So the value of z doesn't change after this iteration

A pivot is called **degenerate** iff it does not change the objective value

Note: empirically rare in practice

Cycling can only occur in the presence of degenerate pivot.

Pivoting Strategies

There exist variable selection strategies that guarantee termination

Bland's Rule (1977): the simplex method terminates as long as the entering and leaving variables are selected by the smallest-subscript rule in each iteration

Example: $z = -5x_1 - 3x_2 + 4x_3 + 40x_4$

The entering variable is: x_3

Leaving variable: still the one imposing the tightest constraint, but break tie by picking the smaller subscript

Modern solvers use more sophisticated heuristics (e.g., Steepest Edge) that might not prevent cycling

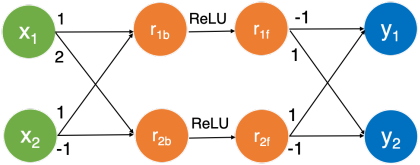
When cycling is detected: switch to Bland's rule for a while

Complexity: the common strategies all have worse-case exponential time

Possible optimizations

- More sophisticated pivoting strategy
- Use rational number representation (to handle numerical instability)
- Handle general Linear Program (variables can have non-zero lower bounds and/or finite upper bounds)
- Extract irreducible infeasible subset in case of infeasibility (theory explanations)
- ...

Application: Neural Network Verification



Property to verify: $\forall x_1 . x_2 . ((x_1 \in [-2, 1] \wedge x_2 \in [-2, 2]) \rightarrow y_1 < y_2)$

1. Encoding of the neural network ϕ_n (linear + ReLUs):

   $r_{1b} = x_1 + x_2 \quad r_{2b} = 2x_1 - x_2$ $\qquad (r_{1b} \leq 0 \wedge r_{1f} = 0) \vee (r_{1b} \geq 0 \wedge r_{1f} = r_{1b})$

   $y_1 = -r_{1f} + r_{2f} \quad y_2 = r_{1f} - r_{2f}$ $\qquad (r_{2b} \leq 0 \wedge r_{2f} = 0) \vee (r_{2b} \geq 0 \wedge r_{2f} = r_{2b})$

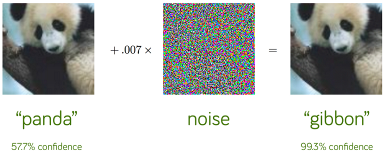
2. Encoding of the the property ϕ_p:

   Submit your answer to `https://pollev.com/andreww095`

3. Property holds iff $\phi_n \wedge \phi_p$ is unsatisfiable

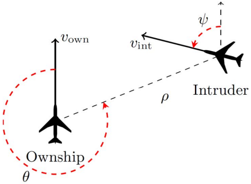
Practical properties

- Robustness property: $\forall x', \|\mathbf{x} - x'\| < \epsilon \Rightarrow \|N(\mathbf{x}) - N(x')\| < \delta$



"There is no adversarial input within ϵ distance"

- Reachability property: $\forall x, x \in [x_l, x_u] \Rightarrow y \in [y_l, y_u]$



"Whenever intruder is near and to the right advise strong left."

A lot of attentions in recent years.