

Randomised Algorithms

Lecture 7: Linear Programming: Simplex Algorithm

Thomas Sauerwald (tms41@cam.ac.uk)

Lent 2025



UNIVERSITY OF
CAMBRIDGE

Simplex Algorithm by Example

Details of the Simplex Algorithm

Finding an Initial Solution

Appendix: Cycling and Termination (non-examinable)

Simplex Algorithm: Introduction

Simplex Algorithm

- classical method for solving linear programs (Dantzig, 1947)
- usually fast in practice although worst-case runtime not polynomial
- iterative procedure somewhat similar to Gaussian elimination

Simplex Algorithm

- classical method for solving linear programs (Dantzig, 1947)
- usually fast in practice although worst-case runtime not polynomial
- iterative procedure somewhat similar to Gaussian elimination

Basic Idea:

- Each iteration corresponds to a “basic solution” of the slack form
- All non-basic variables are 0, and the basic variables are determined from the equality constraints
- Each iteration converts one slack form into an equivalent one while the objective value will not decrease
- Conversion (“pivoting”) is achieved by switching the roles of one basic and one non-basic variable

Simplex Algorithm: Introduction

Simplex Algorithm

- classical method for solving linear programs (Dantzig, 1947)
- usually fast in practice although worst-case runtime not polynomial
- iterative procedure somewhat similar to Gaussian elimination

Basic Idea:

- Each iteration corresponds to a “basic solution” of the slack form
- All non-basic variables are 0, and the basic variables are determined from the equality constraints
- Each iteration converts one slack form into an equivalent one while the objective value will not decrease
- Conversion (“pivoting”) is achieved by switching the roles of one basic and one non-basic variable

In that sense, it is a **greedy algorithm**.

Extended Example: Conversion into Slack Form

$$\begin{array}{llllllll} \text{maximise} & 3x_1 & + & x_2 & + & 2x_3 & & \\ \text{subject to} & & & & & & & \\ & x_1 & + & x_2 & + & 3x_3 & \leq & 30 \\ & 2x_1 & + & 2x_2 & + & 5x_3 & \leq & 24 \\ & 4x_1 & + & x_2 & + & 2x_3 & \leq & 36 \\ & & & x_1, x_2, x_3 & & & \geq & 0 \end{array}$$

Extended Example: Conversion into Slack Form

$$\begin{array}{llllllll} \text{maximise} & 3x_1 & + & x_2 & + & 2x_3 & & \\ \text{subject to} & & & & & & & \\ & x_1 & + & x_2 & + & 3x_3 & \leq & 30 \\ & 2x_1 & + & 2x_2 & + & 5x_3 & \leq & 24 \\ & 4x_1 & + & x_2 & + & 2x_3 & \leq & 36 \\ & & & x_1, x_2, x_3 & & & \geq & 0 \end{array}$$

↓
Conversion into slack form

Extended Example: Conversion into Slack Form

$$\begin{array}{llllll} \text{maximise} & 3x_1 & + & x_2 & + & 2x_3 \\ \text{subject to} & & & & & \\ & x_1 & + & x_2 & + & 3x_3 & \leq & 30 \\ & 2x_1 & + & 2x_2 & + & 5x_3 & \leq & 24 \\ & 4x_1 & + & x_2 & + & 2x_3 & \leq & 36 \\ & & & x_1, x_2, x_3 & & & \geq & 0 \end{array}$$

Conversion into slack form


$$\begin{array}{rcllcllcl} Z & = & & 3x_1 & + & x_2 & + & 2x_3 \\ x_4 & = & 30 & - & x_1 & - & x_2 & - & 3x_3 \\ x_5 & = & 24 & - & 2x_1 & - & 2x_2 & - & 5x_3 \\ x_6 & = & 36 & - & 4x_1 & - & x_2 & - & 2x_3 \end{array}$$

Extended Example: Iteration 1

$$z = 3x_1 + x_2 + 2x_3$$

$$x_4 = 30 - x_1 - x_2 - 3x_3$$

$$x_5 = 24 - 2x_1 - 2x_2 - 5x_3$$

$$x_6 = 36 - 4x_1 - x_2 - 2x_3$$

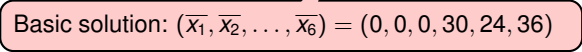
Extended Example: Iteration 1

$$z = 3x_1 + x_2 + 2x_3$$

$$x_4 = 30 - x_1 - x_2 - 3x_3$$

$$x_5 = 24 - 2x_1 - 2x_2 - 5x_3$$

$$x_6 = 36 - 4x_1 - x_2 - 2x_3$$



Basic solution: $(\overline{x_1}, \overline{x_2}, \dots, \overline{x_6}) = (0, 0, 0, 30, 24, 36)$

Extended Example: Iteration 1

$$Z = 3x_1 + x_2 + 2x_3$$

$$x_4 = 30 - x_1 - x_2 - 3x_3$$

$$x_5 = 24 - 2x_1 - 2x_2 - 5x_3$$

$$x_6 = 36 - 4x_1 - x_2 - 2x_3$$

Basic solution: $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_6) = (0, 0, 0, 30, 24, 36)$

This basic solution is **feasible**

Extended Example: Iteration 1

$$Z = 3x_1 + x_2 + 2x_3$$

$$x_4 = 30 - x_1 - x_2 - 3x_3$$

$$x_5 = 24 - 2x_1 - 2x_2 - 5x_3$$

$$x_6 = 36 - 4x_1 - x_2 - 2x_3$$

Basic solution: $(\overline{x_1}, \overline{x_2}, \dots, \overline{x_6}) = (0, 0, 0, 30, 24, 36)$

This basic solution is **feasible**

Objective value is 0.

Extended Example: Iteration 1

Increasing the value of x_1 would increase the objective value.

$$z = 3x_1 + x_2 + 2x_3$$

$$x_4 = 30 - x_1 - x_2 - 3x_3$$

$$x_5 = 24 - 2x_1 - 2x_2 - 5x_3$$

$$x_6 = 36 - 4x_1 - x_2 - 2x_3$$

Basic solution: $(\overline{x_1}, \overline{x_2}, \dots, \overline{x_6}) = (0, 0, 0, 30, 24, 36)$

This basic solution is **feasible**

Objective value is 0.

Extended Example: Iteration 1

Increasing the value of x_1 would increase the objective value.

$$z = 3x_1 + x_2 + 2x_3$$

$$x_4 = 30 - x_1 - x_2 - 3x_3$$

$$x_5 = 24 - 2x_1 - 2x_2 - 5x_3$$

$$x_6 = 36 - 4x_1 - x_2 - 2x_3$$

The third constraint is the tightest and limits how much we can increase x_1 .

Extended Example: Iteration 1

Increasing the value of x_1 would increase the objective value.

$$z = 3x_1 + x_2 + 2x_3$$

$$x_4 = 30 - x_1 - x_2 - 3x_3$$

$$x_5 = 24 - 2x_1 - 2x_2 - 5x_3$$

$$x_6 = 36 - 4x_1 - x_2 - 2x_3$$

The third constraint is the tightest and limits how much we can increase x_1 .

Switch roles of x_1 and x_6 :

Extended Example: Iteration 1

Increasing the value of x_1 would increase the objective value.

$$z = 3x_1 + x_2 + 2x_3$$

$$x_4 = 30 - x_1 - x_2 - 3x_3$$

$$x_5 = 24 - 2x_1 - 2x_2 - 5x_3$$

$$x_6 = 36 - 4x_1 - x_2 - 2x_3$$

The third constraint is the tightest and limits how much we can increase x_1 .

Switch roles of x_1 and x_6 :

- Solving for x_1 yields:

$$x_1 = 9 - \frac{x_2}{4} - \frac{x_3}{2} - \frac{x_6}{4}.$$

Extended Example: Iteration 1

Increasing the value of x_1 would increase the objective value.

$$z = 3x_1 + x_2 + 2x_3$$

$$x_4 = 30 - x_1 - x_2 - 3x_3$$

$$x_5 = 24 - 2x_1 - 2x_2 - 5x_3$$

$$x_6 = 36 - 4x_1 - x_2 - 2x_3$$

The third constraint is the tightest and limits how much we can increase x_1 .

Switch roles of x_1 and x_6 :

- Solving for x_1 yields:

$$x_1 = 9 - \frac{x_2}{4} - \frac{x_3}{2} - \frac{x_6}{4}.$$

- Substitute this into x_1 in the other three equations

Extended Example: Iteration 2

$$\begin{array}{rclclclcl} z & = & 27 & + & \frac{x_2}{4} & + & \frac{x_3}{2} & - & \frac{3x_6}{4} \\ x_1 & = & 9 & - & \frac{x_2}{4} & - & \frac{x_3}{2} & - & \frac{x_6}{4} \\ x_4 & = & 21 & - & \frac{3x_2}{4} & - & \frac{5x_3}{2} & + & \frac{x_6}{4} \\ x_5 & = & 6 & - & \frac{3x_2}{2} & - & 4x_3 & + & \frac{x_6}{2} \end{array}$$

Extended Example: Iteration 2

$$\begin{array}{rclclclcl} z & = & 27 & + & \frac{x_2}{4} & + & \frac{x_3}{2} & - & \frac{3x_6}{4} \\ x_1 & = & 9 & - & \frac{x_2}{4} & - & \frac{x_3}{2} & - & \frac{x_6}{4} \\ x_4 & = & 21 & - & \frac{3x_2}{4} & - & \frac{5x_3}{2} & + & \frac{x_6}{4} \\ x_5 & = & 6 & - & \frac{3x_2}{2} & - & 4x_3 & + & \frac{x_6}{2} \end{array}$$

Basic solution: $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_6) = (9, 0, 0, 21, 6, 0)$ with objective value 27

Extended Example: Iteration 2

Increasing the value of x_3 would increase the objective value.

$$\begin{array}{rclclcl} z & = & 27 & + & \frac{x_2}{4} & + & \frac{x_3}{2} & - & \frac{3x_6}{4} \\ x_1 & = & 9 & - & \frac{x_2}{4} & - & \frac{x_3}{2} & - & \frac{x_6}{4} \\ x_4 & = & 21 & - & \frac{3x_2}{4} & - & \frac{5x_3}{2} & + & \frac{x_6}{4} \\ x_5 & = & 6 & - & \frac{3x_2}{2} & - & 4x_3 & + & \frac{x_6}{2} \end{array}$$

Basic solution: $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_6) = (9, 0, 0, 21, 6, 0)$ with objective value 27

Extended Example: Iteration 2

Increasing the value of x_3 would increase the objective value.

$$\begin{array}{rclclcl} z & = & 27 & + & \frac{x_2}{4} & + & \frac{x_3}{2} & - & \frac{3x_6}{4} \\ x_1 & = & 9 & - & \frac{x_2}{4} & - & \frac{x_3}{2} & - & \frac{x_6}{4} \\ x_4 & = & 21 & - & \frac{3x_2}{4} & - & \frac{5x_3}{2} & + & \frac{x_6}{4} \\ x_5 & = & 6 & - & \frac{3x_2}{2} & - & 4x_3 & + & \frac{x_6}{2} \end{array}$$

The third constraint is the tightest and limits how much we can increase x_3 .

Extended Example: Iteration 2

Increasing the value of x_3 would increase the objective value.

$$\begin{array}{rclclcl} z & = & 27 & + & \frac{x_2}{4} & + & \frac{x_3}{2} & - & \frac{3x_6}{4} \\ x_1 & = & 9 & - & \frac{x_2}{4} & - & \frac{x_3}{2} & - & \frac{x_6}{4} \\ x_4 & = & 21 & - & \frac{3x_2}{4} & - & \frac{5x_3}{2} & + & \frac{x_6}{4} \\ x_5 & = & 6 & - & \frac{3x_2}{2} & - & 4x_3 & + & \frac{x_6}{2} \end{array}$$

The third constraint is the tightest and limits how much we can increase x_3 .

Switch roles of x_3 and x_5 :

Extended Example: Iteration 2

Increasing the value of x_3 would increase the objective value.

$$\begin{array}{rclclcl} z & = & 27 & + & \frac{x_2}{4} & + & \frac{x_3}{2} & - & \frac{3x_6}{4} \\ x_1 & = & 9 & - & \frac{x_2}{4} & - & \frac{x_3}{2} & - & \frac{x_6}{4} \\ x_4 & = & 21 & - & \frac{3x_2}{4} & - & \frac{5x_3}{2} & + & \frac{x_6}{4} \\ x_5 & = & 6 & - & \frac{3x_2}{2} & - & 4x_3 & + & \frac{x_6}{2} \end{array}$$

The third constraint is the tightest and limits how much we can increase x_3 .

Switch roles of x_3 and x_5 :

- Solving for x_3 yields:

$$x_3 = \frac{3}{2} - \frac{3x_2}{8} - \frac{x_5}{4} - \frac{x_6}{8}.$$

Extended Example: Iteration 2

Increasing the value of x_3 would increase the objective value.

$$\begin{array}{rclclcl} z & = & 27 & + & \frac{x_2}{4} & + & \frac{x_3}{2} & - & \frac{3x_6}{4} \\ x_1 & = & 9 & - & \frac{x_2}{4} & - & \frac{x_3}{2} & - & \frac{x_6}{4} \\ x_4 & = & 21 & - & \frac{3x_2}{4} & - & \frac{5x_3}{2} & + & \frac{x_6}{4} \\ x_5 & = & 6 & - & \frac{3x_2}{2} & - & 4x_3 & + & \frac{x_6}{2} \end{array}$$

The third constraint is the tightest and limits how much we can increase x_3 .

Switch roles of x_3 and x_5 :

- Solving for x_3 yields:

$$x_3 = \frac{3}{2} - \frac{3x_2}{8} - \frac{x_5}{4} - \frac{x_6}{8}.$$

- Substitute this into x_3 in the other three equations

Extended Example: Iteration 3

$$\begin{aligned} z &= \frac{111}{4} + \frac{x_2}{16} - \frac{x_5}{8} - \frac{11x_6}{16} \\ x_1 &= \frac{33}{4} - \frac{x_2}{16} + \frac{x_5}{8} - \frac{5x_6}{16} \\ x_3 &= \frac{3}{2} - \frac{3x_2}{8} - \frac{x_5}{4} + \frac{x_6}{8} \\ x_4 &= \frac{69}{4} + \frac{3x_2}{16} + \frac{5x_5}{8} - \frac{x_6}{16} \end{aligned}$$

Extended Example: Iteration 3

$$\begin{aligned} z &= \frac{111}{4} + \frac{x_2}{16} - \frac{x_5}{8} - \frac{11x_6}{16} \\ x_1 &= \frac{33}{4} - \frac{x_2}{16} + \frac{x_5}{8} - \frac{5x_6}{16} \\ x_3 &= \frac{3}{2} - \frac{3x_2}{8} - \frac{x_5}{4} + \frac{x_6}{8} \\ x_4 &= \frac{69}{4} + \frac{3x_2}{16} + \frac{5x_5}{8} - \frac{x_6}{16} \end{aligned}$$

Basic solution: $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_6) = (\frac{33}{4}, 0, \frac{3}{2}, \frac{69}{4}, 0, 0)$ with objective value $\frac{111}{4} = 27.75$

Extended Example: Iteration 3

Increasing the value of x_2 would increase the objective value.

$$\begin{aligned}z &= \frac{111}{4} + \frac{x_2}{16} - \frac{x_5}{8} - \frac{11x_6}{16} \\x_1 &= \frac{33}{4} - \frac{x_2}{16} + \frac{x_5}{8} - \frac{5x_6}{16} \\x_3 &= \frac{3}{2} - \frac{3x_2}{8} - \frac{x_5}{4} + \frac{x_6}{8} \\x_4 &= \frac{69}{4} + \frac{3x_2}{16} + \frac{5x_5}{8} - \frac{x_6}{16}\end{aligned}$$

Basic solution: $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_6) = (\frac{33}{4}, 0, \frac{3}{2}, \frac{69}{4}, 0, 0)$ with objective value $\frac{111}{4} = 27.75$

Extended Example: Iteration 3

Increasing the value of x_2 would increase the objective value.

$$\begin{aligned} z &= \frac{111}{4} + \frac{x_2}{16} - \frac{x_5}{8} - \frac{11x_6}{16} \\ x_1 &= \frac{33}{4} - \frac{x_2}{16} + \frac{x_5}{8} - \frac{5x_6}{16} \\ x_3 &= \frac{3}{2} - \frac{3x_2}{8} - \frac{x_5}{4} + \frac{x_6}{8} \\ x_4 &= \frac{69}{4} + \frac{3x_2}{16} + \frac{5x_5}{8} - \frac{x_6}{16} \end{aligned}$$

The second constraint is the tightest and limits how much we can increase x_2 .

Extended Example: Iteration 3

Increasing the value of x_2 would increase the objective value.

$$\begin{aligned} z &= \frac{111}{4} + \frac{x_2}{16} - \frac{x_5}{8} - \frac{11x_6}{16} \\ x_1 &= \frac{33}{4} - \frac{x_2}{16} + \frac{x_5}{8} - \frac{5x_6}{16} \\ x_3 &= \frac{3}{2} - \frac{3x_2}{8} - \frac{x_5}{4} + \frac{x_6}{8} \\ x_4 &= \frac{69}{4} + \frac{3x_2}{16} + \frac{5x_5}{8} - \frac{x_6}{16} \end{aligned}$$

The second constraint is the tightest and limits how much we can increase x_2 .

Switch roles of x_2 and x_3 :

Extended Example: Iteration 3

Increasing the value of x_2 would increase the objective value.

$$\begin{aligned} z &= \frac{111}{4} + \frac{x_2}{16} - \frac{x_5}{8} - \frac{11x_6}{16} \\ x_1 &= \frac{33}{4} - \frac{x_2}{16} + \frac{x_5}{8} - \frac{5x_6}{16} \\ x_3 &= \frac{3}{2} - \frac{3x_2}{8} - \frac{x_5}{4} + \frac{x_6}{8} \\ x_4 &= \frac{69}{4} + \frac{3x_2}{16} + \frac{5x_5}{8} - \frac{x_6}{16} \end{aligned}$$

The second constraint is the tightest and limits how much we can increase x_2 .

Switch roles of x_2 and x_3 :

- Solving for x_2 yields:

$$x_2 = 4 - \frac{8x_3}{3} - \frac{2x_5}{3} + \frac{x_6}{3}.$$

Extended Example: Iteration 3

Increasing the value of x_2 would increase the objective value.

$$\begin{aligned} z &= \frac{111}{4} + \frac{x_2}{16} - \frac{x_5}{8} - \frac{11x_6}{16} \\ x_1 &= \frac{33}{4} - \frac{x_2}{16} + \frac{x_5}{8} - \frac{5x_6}{16} \\ x_3 &= \frac{3}{2} - \frac{3x_2}{8} - \frac{x_5}{4} + \frac{x_6}{8} \\ x_4 &= \frac{69}{4} + \frac{3x_2}{16} + \frac{5x_5}{8} - \frac{x_6}{16} \end{aligned}$$

The second constraint is the tightest and limits how much we can increase x_2 .

Switch roles of x_2 and x_3 :

- Solving for x_2 yields:

$$x_2 = 4 - \frac{8x_3}{3} - \frac{2x_5}{3} + \frac{x_6}{3}.$$

- Substitute this into x_2 in the other three equations

Extended Example: Iteration 4

$$\begin{array}{rclclclcl} z & = & 28 & - & \frac{x_3}{6} & - & \frac{x_5}{6} & - & \frac{2x_6}{3} \\ x_1 & = & 8 & + & \frac{x_3}{6} & + & \frac{x_5}{6} & - & \frac{x_6}{3} \\ x_2 & = & 4 & - & \frac{8x_3}{3} & - & \frac{2x_5}{3} & + & \frac{x_6}{3} \\ x_4 & = & 18 & - & \frac{x_3}{2} & + & \frac{x_5}{2} & & \end{array}$$

Extended Example: Iteration 4

$$\begin{array}{rclclclcl} z & = & 28 & - & \frac{x_3}{6} & - & \frac{x_5}{6} & - & \frac{2x_6}{3} \\ x_1 & = & 8 & + & \frac{x_3}{6} & + & \frac{x_5}{6} & - & \frac{x_6}{3} \\ x_2 & = & 4 & - & \frac{8x_3}{3} & - & \frac{2x_5}{3} & + & \frac{x_6}{3} \\ x_4 & = & 18 & - & \frac{x_3}{2} & + & \frac{x_5}{2} & & \end{array}$$

Basic solution: $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_6) = (8, 4, 0, 18, 0, 0)$ with objective value 28

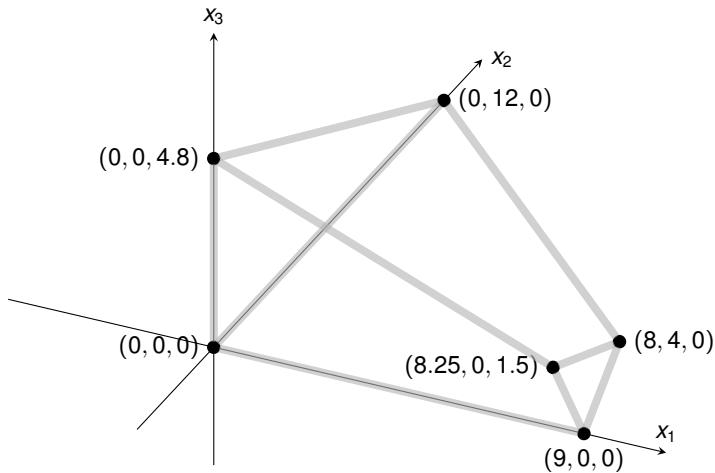
Extended Example: Iteration 4

All coefficients are negative, and hence this basic solution is **optimal!**

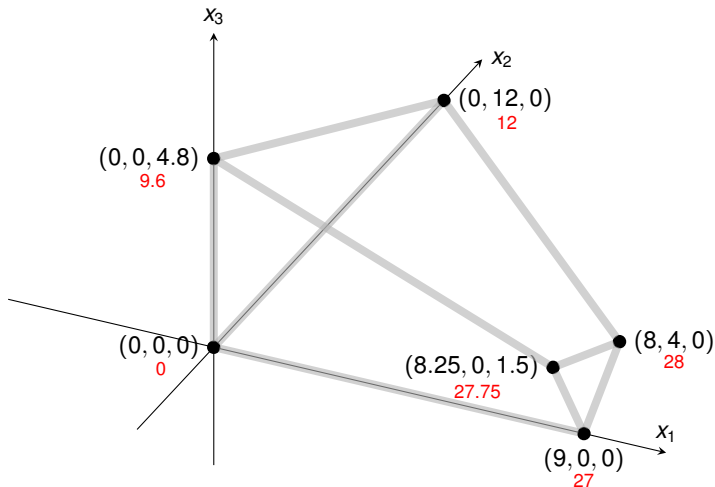
$$\begin{array}{rclclcl} z & = & 28 & - & \frac{x_3}{6} & - & \frac{x_5}{6} & - & \frac{2x_6}{3} \\ x_1 & = & 8 & + & \frac{x_3}{6} & + & \frac{x_5}{6} & - & \frac{x_6}{3} \\ x_2 & = & 4 & - & \frac{8x_3}{3} & - & \frac{2x_5}{3} & + & \frac{x_6}{3} \\ x_4 & = & 18 & - & \frac{x_3}{2} & + & \frac{x_5}{2} & & \end{array}$$

Basic solution: $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_6) = (8, 4, 0, 18, 0, 0)$ with objective value 28

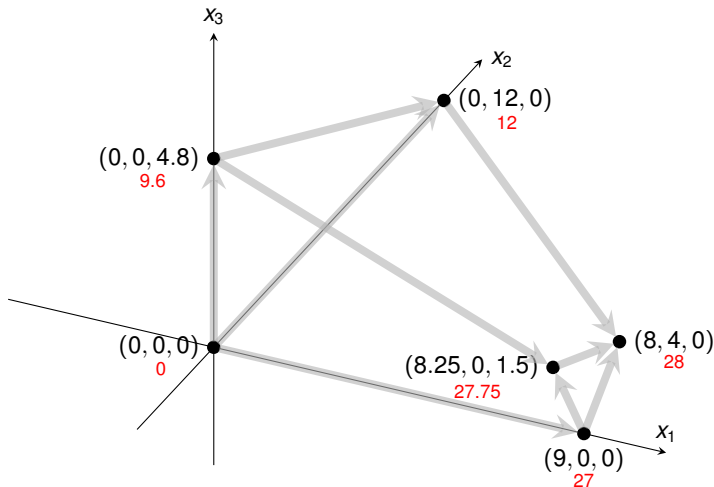
Extended Example: Visualization of SIMPLEX



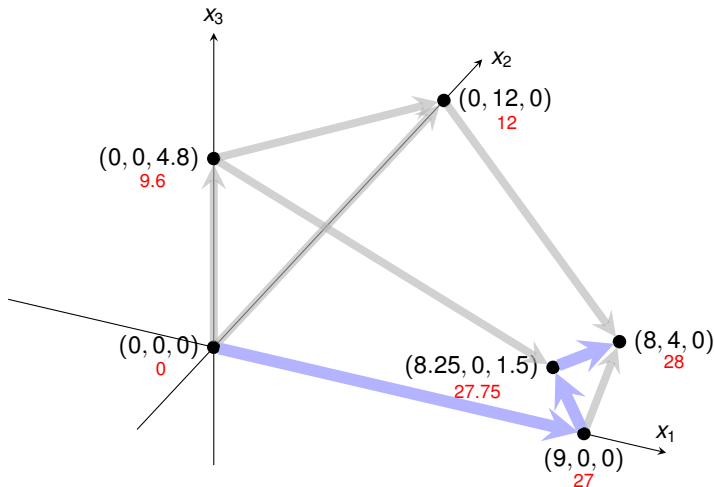
Extended Example: Visualization of SIMPLEX



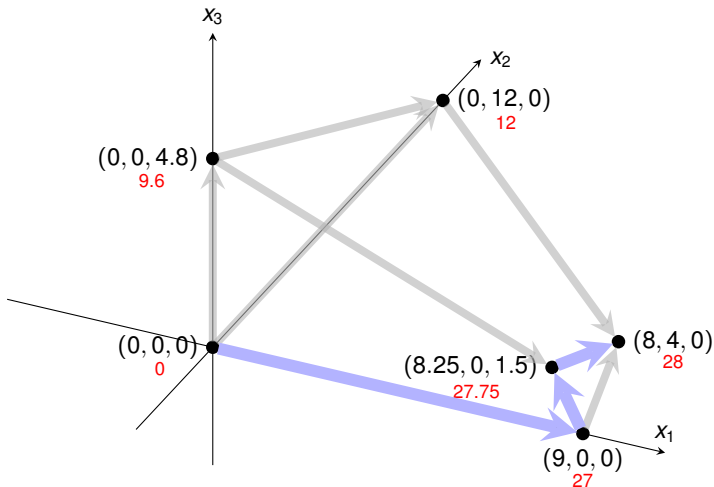
Extended Example: Visualization of SIMPLEX



Extended Example: Visualization of SIMPLEX



Extended Example: Visualization of SIMPLEX



Question: How many basic solutions (including non-feasible ones) are there?

Extended Example: Alternative Runs (1/2)

$$\begin{array}{rclclclcl} z & = & & 3x_1 & + & x_2 & + & 2x_3 \\ x_4 & = & 30 & - & x_1 & - & x_2 & - & 3x_3 \\ x_5 & = & 24 & - & 2x_1 & - & 2x_2 & - & 5x_3 \\ x_6 & = & 36 & - & 4x_1 & - & x_2 & - & 2x_3 \end{array}$$

Extended Example: Alternative Runs (1/2)

$$\begin{array}{rclclclcl} z & = & & 3x_1 & + & x_2 & + & 2x_3 \\ x_4 & = & 30 & - & x_1 & - & x_2 & - & 3x_3 \\ x_5 & = & 24 & - & 2x_1 & - & 2x_2 & - & 5x_3 \\ x_6 & = & 36 & - & 4x_1 & - & x_2 & - & 2x_3 \end{array}$$

↓ Switch roles of x_2 and x_5

Extended Example: Alternative Runs (1/2)

$$\begin{array}{rclclclcl} z & = & & 3x_1 & + & x_2 & + & 2x_3 \\ x_4 & = & 30 & - & x_1 & - & x_2 & - & 3x_3 \\ x_5 & = & 24 & - & 2x_1 & - & 2x_2 & - & 5x_3 \\ x_6 & = & 36 & - & 4x_1 & - & x_2 & - & 2x_3 \end{array}$$

Switch roles of x_2 and x_5
↓

$$\begin{array}{rclclclcl} z & = & 12 & + & 2x_1 & - & \frac{x_3}{2} & - & \frac{x_5}{2} \\ x_2 & = & 12 & - & x_1 & - & \frac{5x_3}{2} & - & \frac{x_5}{2} \\ x_4 & = & 18 & - & x_2 & - & \frac{x_3}{2} & + & \frac{x_5}{2} \\ x_6 & = & 24 & - & 3x_1 & + & \frac{x_3}{2} & + & \frac{x_5}{2} \end{array}$$

Extended Example: Alternative Runs (1/2)

$$\begin{array}{rclclclcl} z & = & & 3x_1 & + & x_2 & + & 2x_3 \\ x_4 & = & 30 & - & x_1 & - & x_2 & - & 3x_3 \\ x_5 & = & 24 & - & 2x_1 & - & 2x_2 & - & 5x_3 \\ x_6 & = & 36 & - & 4x_1 & - & x_2 & - & 2x_3 \end{array}$$

Switch roles of x_2 and x_5
↓

$$\begin{array}{rclclclcl} z & = & 12 & + & 2x_1 & - & \frac{x_3}{2} & - & \frac{x_5}{2} \\ x_2 & = & 12 & - & x_1 & - & \frac{5x_3}{2} & - & \frac{x_5}{2} \\ x_4 & = & 18 & - & x_2 & - & \frac{x_3}{2} & + & \frac{x_5}{2} \\ x_6 & = & 24 & - & 3x_1 & + & \frac{x_3}{2} & + & \frac{x_5}{2} \end{array}$$

Switch roles of x_1 and x_6
↓

Extended Example: Alternative Runs (1/2)

$$\begin{array}{rclclcl} z & = & & 3x_1 & + & x_2 & + & 2x_3 \\ x_4 & = & 30 & - & x_1 & - & x_2 & - & 3x_3 \\ x_5 & = & 24 & - & 2x_1 & - & 2x_2 & - & 5x_3 \\ x_6 & = & 36 & - & 4x_1 & - & x_2 & - & 2x_3 \end{array}$$

Switch roles of x_2 and x_5
↓

$$\begin{array}{rclclcl} z & = & 12 & + & 2x_1 & - & \frac{x_3}{2} & - & \frac{x_5}{2} \\ x_2 & = & 12 & - & x_1 & - & \frac{5x_3}{2} & - & \frac{x_5}{2} \\ x_4 & = & 18 & - & x_2 & - & \frac{x_3}{2} & + & \frac{x_5}{2} \\ x_6 & = & 24 & - & 3x_1 & + & \frac{x_3}{2} & + & \frac{x_5}{2} \end{array}$$

Switch roles of x_1 and x_6
↓

$$\begin{array}{rclclcl} z & = & 28 & - & \frac{x_3}{6} & - & \frac{x_5}{6} & - & \frac{2x_6}{3} \\ x_1 & = & 8 & + & \frac{x_3}{6} & + & \frac{x_5}{6} & - & \frac{x_6}{3} \\ x_2 & = & 4 & - & \frac{8x_3}{3} & - & \frac{2x_5}{3} & + & \frac{x_6}{3} \\ x_4 & = & 18 & - & \frac{x_3}{2} & + & \frac{x_5}{2} & & \end{array}$$

Extended Example: Alternative Runs (2/2)

$$\begin{array}{rclclclcl} z & = & & 3x_1 & + & x_2 & + & 2x_3 \\ x_4 & = & 30 & - & x_1 & - & x_2 & - & 3x_3 \\ x_5 & = & 24 & - & 2x_1 & - & 2x_2 & - & 5x_3 \\ x_6 & = & 36 & - & 4x_1 & - & x_2 & - & 2x_3 \end{array}$$

Extended Example: Alternative Runs (2/2)

$$\begin{array}{rclclclcl} z & = & & 3x_1 & + & x_2 & + & 2x_3 \\ x_4 & = & 30 & - & x_1 & - & x_2 & - & 3x_3 \\ x_5 & = & 24 & - & 2x_1 & - & 2x_2 & - & 5x_3 \\ x_6 & = & 36 & - & 4x_1 & - & x_2 & - & 2x_3 \end{array}$$

↓ Switch roles of x_3 and x_5

Extended Example: Alternative Runs (2/2)

$$\begin{array}{rclclclcl} z & = & & 3x_1 & + & x_2 & + & 2x_3 \\ x_4 & = & 30 & - & x_1 & - & x_2 & - & 3x_3 \\ x_5 & = & 24 & - & 2x_1 & - & 2x_2 & - & 5x_3 \\ x_6 & = & 36 & - & 4x_1 & - & x_2 & - & 2x_3 \end{array}$$

↓ Switch roles of x_3 and x_5

$$\begin{array}{rclclclcl} z & = & \frac{48}{5} & + & \frac{11x_1}{5} & + & \frac{x_2}{5} & - & \frac{2x_5}{5} \\ x_4 & = & \frac{78}{5} & + & \frac{x_1}{5} & + & \frac{x_2}{5} & + & \frac{3x_5}{5} \\ x_3 & = & \frac{24}{5} & - & \frac{2x_1}{5} & - & \frac{2x_2}{5} & - & \frac{x_5}{5} \\ x_6 & = & \frac{132}{5} & - & \frac{16x_1}{5} & - & \frac{x_2}{5} & + & \frac{2x_3}{5} \end{array}$$

Extended Example: Alternative Runs (2/2)

$$\begin{array}{rclclclcl} z & = & & 3x_1 & + & x_2 & + & 2x_3 \\ x_4 & = & 30 & - & x_1 & - & x_2 & - & 3x_3 \\ x_5 & = & 24 & - & 2x_1 & - & 2x_2 & - & 5x_3 \\ x_6 & = & 36 & - & 4x_1 & - & x_2 & - & 2x_3 \end{array}$$

↓ Switch roles of x_3 and x_5

$$\begin{array}{rclclclcl} z & = & \frac{48}{5} & + & \frac{11x_1}{5} & + & \frac{x_2}{5} & - & \frac{2x_5}{5} \\ x_4 & = & \frac{78}{5} & + & \frac{x_1}{5} & + & \frac{x_2}{5} & + & \frac{3x_5}{5} \\ x_3 & = & \frac{24}{5} & - & \frac{2x_1}{5} & - & \frac{2x_2}{5} & - & \frac{x_5}{5} \\ x_6 & = & \frac{132}{5} & - & \frac{16x_1}{5} & - & \frac{x_2}{5} & + & \frac{2x_3}{5} \end{array}$$

Switch roles of x_1 and x_6

Extended Example: Alternative Runs (2/2)

$$\begin{array}{rclclclcl}
 z & = & & 3x_1 & + & x_2 & + & 2x_3 \\
 x_4 & = & 30 & - & x_1 & - & x_2 & - & 3x_3 \\
 x_5 & = & 24 & - & 2x_1 & - & 2x_2 & - & 5x_3 \\
 x_6 & = & 36 & - & 4x_1 & - & x_2 & - & 2x_3
 \end{array}$$

$\downarrow$ Switch roles of x_3 and x_5

$$\begin{array}{rclclclcl}
 z & = & \frac{48}{5} & + & \frac{11x_1}{5} & + & \frac{x_2}{5} & - & \frac{2x_5}{5} \\
 x_4 & = & \frac{78}{5} & + & \frac{x_1}{5} & + & \frac{x_2}{5} & + & \frac{3x_5}{5} \\
 x_3 & = & \frac{24}{5} & - & \frac{2x_1}{5} & - & \frac{2x_2}{5} & - & \frac{x_5}{5} \\
 x_6 & = & \frac{132}{5} & - & \frac{16x_1}{5} & - & \frac{x_2}{5} & + & \frac{2x_3}{5}
 \end{array}$$

$\swarrow$ Switch roles of x_1 and x_6

$$\begin{array}{rclclclcl}
 z & = & \frac{111}{4} & + & \frac{x_2}{16} & - & \frac{x_5}{8} & - & \frac{11x_6}{16} \\
 x_1 & = & \frac{33}{4} & - & \frac{x_2}{16} & + & \frac{x_5}{8} & - & \frac{5x_6}{16} \\
 x_3 & = & \frac{3}{2} & - & \frac{3x_2}{8} & - & \frac{x_5}{4} & + & \frac{x_6}{8} \\
 x_4 & = & \frac{69}{4} & + & \frac{3x_2}{16} & + & \frac{5x_5}{8} & - & \frac{x_6}{16}
 \end{array}$$

Extended Example: Alternative Runs (2/2)

$$\begin{array}{rclclcl}
 z & = & & 3x_1 & + & x_2 & + & 2x_3 \\
 x_4 & = & 30 & - & x_1 & - & x_2 & - & 3x_3 \\
 x_5 & = & 24 & - & 2x_1 & - & 2x_2 & - & 5x_3 \\
 x_6 & = & 36 & - & 4x_1 & - & x_2 & - & 2x_3
 \end{array}$$

↓ Switch roles of x_3 and x_5

$$\begin{array}{rclclcl}
 z & = & \frac{48}{5} & + & \frac{11x_1}{5} & + & \frac{x_2}{5} & - & \frac{2x_5}{5} \\
 x_4 & = & \frac{78}{5} & + & \frac{x_1}{5} & + & \frac{x_2}{5} & + & \frac{3x_5}{5} \\
 x_3 & = & \frac{24}{5} & - & \frac{2x_1}{5} & - & \frac{2x_2}{5} & - & \frac{x_5}{5} \\
 x_6 & = & \frac{132}{5} & - & \frac{16x_1}{5} & - & \frac{x_2}{5} & + & \frac{2x_3}{5}
 \end{array}$$

Switch roles of x_1 and x_6

Switch roles of x_2 and x_3

$$\begin{array}{rclclcl}
 z & = & \frac{111}{4} & + & \frac{x_2}{16} & - & \frac{x_5}{8} & - & \frac{11x_6}{16} \\
 x_1 & = & \frac{33}{4} & - & \frac{x_2}{16} & + & \frac{x_5}{8} & - & \frac{5x_6}{16} \\
 x_3 & = & \frac{3}{2} & - & \frac{3x_2}{8} & - & \frac{x_5}{4} & + & \frac{x_6}{8} \\
 x_4 & = & \frac{69}{4} & + & \frac{3x_2}{16} & + & \frac{5x_5}{8} & - & \frac{x_6}{16}
 \end{array}$$

Extended Example: Alternative Runs (2/2)

$$\begin{array}{rclclcl}
 z & = & & 3x_1 & + & x_2 & + & 2x_3 \\
 x_4 & = & 30 & - & x_1 & - & x_2 & - & 3x_3 \\
 x_5 & = & 24 & - & 2x_1 & - & 2x_2 & - & 5x_3 \\
 x_6 & = & 36 & - & 4x_1 & - & x_2 & - & 2x_3
 \end{array}$$

↓ Switch roles of x_3 and x_5

$$\begin{array}{rclclcl}
 z & = & \frac{48}{5} & + & \frac{11x_1}{5} & + & \frac{x_2}{5} & - & \frac{2x_5}{5} \\
 x_4 & = & \frac{78}{5} & + & \frac{x_1}{5} & + & \frac{x_2}{5} & + & \frac{3x_5}{5} \\
 x_3 & = & \frac{24}{5} & - & \frac{2x_1}{5} & - & \frac{2x_2}{5} & - & \frac{x_5}{5} \\
 x_6 & = & \frac{132}{5} & - & \frac{16x_1}{5} & - & \frac{x_2}{5} & + & \frac{2x_3}{5}
 \end{array}$$

Switch roles of x_1 and x_6

Switch roles of x_2 and x_3

$$\begin{array}{rclclcl}
 z & = & \frac{111}{4} & + & \frac{x_2}{16} & - & \frac{x_5}{8} & - & \frac{11x_6}{16} \\
 x_1 & = & \frac{33}{4} & - & \frac{x_2}{16} & + & \frac{x_5}{8} & - & \frac{5x_6}{16} \\
 x_3 & = & \frac{3}{2} & - & \frac{3x_2}{8} & - & \frac{x_5}{4} & + & \frac{x_6}{8} \\
 x_4 & = & \frac{69}{4} & + & \frac{3x_2}{16} & + & \frac{5x_5}{8} & - & \frac{x_6}{16}
 \end{array}$$

$$\begin{array}{rclclcl}
 z & = & 28 & - & \frac{x_3}{6} & - & \frac{x_5}{6} & - & \frac{2x_6}{3} \\
 x_1 & = & 8 & + & \frac{x_3}{6} & + & \frac{x_5}{6} & - & \frac{x_6}{3} \\
 x_2 & = & 4 & - & \frac{8x_3}{3} & - & \frac{2x_5}{3} & + & \frac{x_6}{3} \\
 x_4 & = & 18 & - & \frac{x_3}{2} & + & \frac{x_5}{2}
 \end{array}$$

Simplex Algorithm by Example

Details of the Simplex Algorithm

Finding an Initial Solution

Appendix: Cycling and Termination (non-examinable)

The Pivot Step Formally

PIVOT(N, B, A, b, c, v, l, e)

```
1  // Compute the coefficients of the equation for new basic variable  $x_e$ .
2  let  $\hat{A}$  be a new  $m \times n$  matrix
3   $\hat{b}_e = b_l / a_{le}$ 
4  for each  $j \in N - \{e\}$ 
5       $\hat{a}_{ej} = a_{lj} / a_{le}$ 
6   $\hat{a}_{el} = 1 / a_{le}$ 
7  // Compute the coefficients of the remaining constraints.
8  for each  $i \in B - \{l\}$ 
9       $\hat{b}_i = b_i - a_{ie} \hat{b}_e$ 
10     for each  $j \in N - \{e\}$ 
11          $\hat{a}_{ij} = a_{ij} - a_{ie} \hat{a}_{ej}$ 
12      $\hat{a}_{il} = -a_{ie} \hat{a}_{el}$ 
13 // Compute the objective function.
14  $\hat{v} = v + c_e \hat{b}_e$ 
15 for each  $j \in N - \{e\}$ 
16      $\hat{c}_j = c_j - c_e \hat{a}_{ej}$ 
17  $\hat{c}_l = -c_e \hat{a}_{el}$ 
18 // Compute new sets of basic and nonbasic variables.
19  $\hat{N} = N - \{e\} \cup \{l\}$ 
20  $\hat{B} = B - \{l\} \cup \{e\}$ 
21 return ( $\hat{N}, \hat{B}, \hat{A}, \hat{b}, \hat{c}, \hat{v}$ )
```

The Pivot Step Formally

PIVOT(N, B, A, b, c, v, l, e)

```
1  // Compute the coefficients of the equation for new basic variable  $x_e$ .
2  let  $\hat{A}$  be a new  $m \times n$  matrix
3   $\hat{b}_e = b_l / a_{le}$ 
4  for each  $j \in N - \{e\}$ 
5       $\hat{a}_{ej} = a_{lj} / a_{le}$ 
6   $\hat{a}_{el} = 1 / a_{le}$ 
7  // Compute the coefficients of the remaining constraints.
8  for each  $i \in B - \{l\}$ 
9       $\hat{b}_i = b_i - a_{ie} \hat{b}_e$ 
10     for each  $j \in N - \{e\}$ 
11          $\hat{a}_{ij} = a_{ij} - a_{ie} \hat{a}_{ej}$ 
12      $\hat{a}_{il} = -a_{ie} \hat{a}_{el}$ 
13 // Compute the objective function.
14  $\hat{v} = v + c_e \hat{b}_e$ 
15 for each  $j \in N - \{e\}$ 
16      $\hat{c}_j = c_j - c_e \hat{a}_{ej}$ 
17  $\hat{c}_l = -c_e \hat{a}_{el}$ 
18 // Compute new sets of basic and nonbasic variables.
19  $\hat{N} = N - \{e\} \cup \{l\}$ 
20  $\hat{B} = B - \{l\} \cup \{e\}$ 
21 return ( $\hat{N}, \hat{B}, \hat{A}, \hat{b}, \hat{c}, \hat{v}$ )
```

Rewrite “tight” equation
for entering variable x_e .

The Pivot Step Formally

PIVOT(N, B, A, b, c, v, l, e)

```
1  // Compute the coefficients of the equation for new basic variable  $x_e$ .
2  let  $\hat{A}$  be a new  $m \times n$  matrix
3   $\hat{b}_e = b_l / a_{le}$ 
4  for each  $j \in N - \{e\}$ 
5       $\hat{a}_{ej} = a_{lj} / a_{le}$ 
6   $\hat{a}_{el} = 1 / a_{le}$ 
7  // Compute the coefficients of the remaining constraints.
8  for each  $i \in B - \{l\}$ 
9       $\hat{b}_i = b_i - a_{ie} \hat{b}_e$ 
10     for each  $j \in N - \{e\}$ 
11          $\hat{a}_{ij} = a_{ij} - a_{ie} \hat{a}_{ej}$ 
12      $\hat{a}_{il} = -a_{ie} \hat{a}_{el}$ 
13 // Compute the objective function.
14  $\hat{v} = v + c_e \hat{b}_e$ 
15 for each  $j \in N - \{e\}$ 
16      $\hat{c}_j = c_j - c_e \hat{a}_{ej}$ 
17  $\hat{c}_l = -c_e \hat{a}_{el}$ 
18 // Compute new sets of basic and nonbasic variables.
19  $\hat{N} = N - \{e\} \cup \{l\}$ 
20  $\hat{B} = B - \{l\} \cup \{e\}$ 
21 return ( $\hat{N}, \hat{B}, \hat{A}, \hat{b}, \hat{c}, \hat{v}$ )
```

Rewrite “tight” equation
for entering variable x_e .

Substituting x_e into
other equations.

The Pivot Step Formally

PIVOT(N, B, A, b, c, v, l, e)

```
1  // Compute the coefficients of the equation for new basic variable  $x_e$ .
2  let  $\hat{A}$  be a new  $m \times n$  matrix
3   $\hat{b}_e = b_l / a_{le}$ 
4  for each  $j \in N - \{e\}$ 
5       $\hat{a}_{ej} = a_{lj} / a_{le}$ 
6   $\hat{a}_{el} = 1 / a_{le}$ 
7  // Compute the coefficients of the remaining constraints.
8  for each  $i \in B - \{l\}$ 
9       $\hat{b}_i = b_i - a_{ie} \hat{b}_e$ 
10     for each  $j \in N - \{e\}$ 
11          $\hat{a}_{ij} = a_{ij} - a_{ie} \hat{a}_{ej}$ 
12      $\hat{a}_{il} = -a_{ie} \hat{a}_{el}$ 
13 // Compute the objective function.
14  $\hat{v} = v + c_e \hat{b}_e$ 
15 for each  $j \in N - \{e\}$ 
16      $\hat{c}_j = c_j - c_e \hat{a}_{ej}$ 
17  $\hat{c}_l = -c_e \hat{a}_{el}$ 
18 // Compute new sets of basic and nonbasic variables.
19  $\hat{N} = N - \{e\} \cup \{l\}$ 
20  $\hat{B} = B - \{l\} \cup \{e\}$ 
21 return ( $\hat{N}, \hat{B}, \hat{A}, \hat{b}, \hat{c}, \hat{v}$ )
```

Rewrite “tight” equation
for entering variable x_e .

Substituting x_e into
other equations.

Substituting x_e into
objective function.

The Pivot Step Formally

PIVOT(N, B, A, b, c, v, l, e)

```
1  // Compute the coefficients of the equation for new basic variable  $x_e$ .
2  let  $\hat{A}$  be a new  $m \times n$  matrix
3   $\hat{b}_e = b_l / a_{le}$ 
4  for each  $j \in N - \{e\}$ 
5       $\hat{a}_{ej} = a_{lj} / a_{le}$ 
6   $\hat{a}_{el} = 1 / a_{le}$ 
7  // Compute the coefficients of the remaining constraints.
8  for each  $i \in B - \{l\}$ 
9       $\hat{b}_i = b_i - a_{ie} \hat{b}_e$ 
10     for each  $j \in N - \{e\}$ 
11          $\hat{a}_{ij} = a_{ij} - a_{ie} \hat{a}_{ej}$ 
12      $\hat{a}_{il} = -a_{ie} \hat{a}_{el}$ 
13 // Compute the objective function.
14  $\hat{v} = v + c_e \hat{b}_e$ 
15 for each  $j \in N - \{e\}$ 
16      $\hat{c}_j = c_j - c_e \hat{a}_{ej}$ 
17  $\hat{c}_l = -c_e \hat{a}_{el}$ 
18 // Compute new sets of basic and nonbasic variables.
19  $\hat{N} = N - \{e\} \cup \{l\}$ 
20  $\hat{B} = B - \{l\} \cup \{e\}$ 
21 return ( $\hat{N}, \hat{B}, \hat{A}, \hat{b}, \hat{c}, \hat{v}$ )
```

Rewrite “tight” equation for entering variable x_e .

Substituting x_e into other equations.

Substituting x_e into objective function.

Update non-basic and basic variables

The Pivot Step Formally

PIVOT(N, B, A, b, c, v, l, e)

```
1  // Compute the coefficients of the equation for new basic variable  $x_e$ .
2  let  $\hat{A}$  be a new  $m \times n$  matrix
3   $\hat{b}_e = b_l / a_{le}$ 
4  for each  $j \in N - \{e\}$ 
5       $\hat{a}_{ej} = a_{lj} / a_{le}$ 
6   $\hat{a}_{el} = 1 / a_{le}$ 
7  // Compute the coefficients of the remaining constraints.
8  for each  $i \in B - \{l\}$ 
9       $\hat{b}_i = b_i - a_{ie} \hat{b}_e$ 
10     for each  $j \in N - \{e\}$ 
11          $\hat{a}_{ij} = a_{ij} - a_{ie} \hat{a}_{ej}$ 
12          $\hat{a}_{il} = -a_{ie} \hat{a}_{el}$ 
13 // Compute the objective function.
14  $\hat{v} = v + c_e \hat{b}_e$ 
15 for each  $j \in N - \{e\}$ 
16      $\hat{c}_j = c_j - c_e \hat{a}_{ej}$ 
17  $\hat{c}_l = -c_e \hat{a}_{el}$ 
18 // Compute new sets of basic and nonbasic variables.
19  $\hat{N} = N - \{e\} \cup \{l\}$ 
20  $\hat{B} = B - \{l\} \cup \{e\}$ 
21 return ( $\hat{N}, \hat{B}, \hat{A}, \hat{b}, \hat{c}, \hat{v}$ )
```

Need that $a_{le} \neq 0$!

Rewrite “tight” equation for entering variable x_e .

Substituting x_e into other equations.

Substituting x_e into objective function.

Update non-basic and basic variables

Questions:

- How do we determine whether a linear program is feasible?
- What do we do if the linear program is feasible, but the initial basic solution is not feasible?
- How do we determine whether a linear program is unbounded?
- How do we choose the entering and leaving variables?

Questions:

- How do we determine whether a linear program is feasible?
- What do we do if the linear program is feasible, but the initial basic solution is not feasible?
- How do we determine whether a linear program is unbounded?
- How do we choose the entering and leaving variables?

Example before was a particularly nice one!

The formal procedure SIMPLEX

SIMPLEX(A, b, c)

```
1  ( $N, B, A, b, c, v$ ) = INITIALIZE-SIMPLEX( $A, b, c$ )
2  let  $\Delta$  be a new vector of length  $m$ 
3  while some index  $j \in N$  has  $c_j > 0$ 
4      choose an index  $e \in N$  for which  $c_e > 0$ 
5      for each index  $i \in B$ 
6          if  $a_{ie} > 0$ 
7               $\Delta_i = b_i / a_{ie}$ 
8          else  $\Delta_i = \infty$ 
9      choose an index  $l \in B$  that minimizes  $\Delta_i$ 
10     if  $\Delta_l == \infty$ 
11         return “unbounded”
12     else ( $N, B, A, b, c, v$ ) = PIVOT( $N, B, A, b, c, v, l, e$ )
13 for  $i = 1$  to  $n$ 
14     if  $i \in B$ 
15          $\bar{x}_i = b_i$ 
16     else  $\bar{x}_i = 0$ 
17 return ( $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n$ )
```

The formal procedure SIMPLEX

SIMPLEX(A, b, c)

```
1  ( $N, B, A, b, c, v$ ) = INITIALIZE-SIMPLEX( $A, b, c$ )
2  let  $\Delta$  be a new vector of length  $m$ 
3  while some index  $j \in N$  has  $c_j > 0$ 
4      choose an index  $e \in N$  for which  $c_e > 0$ 
5      for each index  $i \in B$ 
6          if  $a_{ie} > 0$ 
7               $\Delta_i = b_i / a_{ie}$ 
8          else  $\Delta_i = \infty$ 
9      choose an index  $l \in B$  that minimizes  $\Delta_i$ 
10     if  $\Delta_l == \infty$ 
11         return “unbounded”
12     else ( $N, B, A, b, c, v$ ) = PIVOT( $N, B, A, b, c, v, l, e$ )
13 for  $i = 1$  to  $n$ 
14     if  $i \in B$ 
15          $\bar{x}_i = b_i$ 
16     else  $\bar{x}_i = 0$ 
17 return ( $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n$ )
```

Returns a slack form with a feasible basic solution (if it exists)

The formal procedure SIMPLEX

SIMPLEX(A, b, c)

```
1  ( $N, B, A, b, c, v$ ) = INITIALIZE-SIMPLEX( $A, b, c$ )
2  let  $\Delta$  be a new vector of length  $m$ 
3  while some index  $j \in N$  has  $c_j > 0$ 
4      choose an index  $e \in N$  for which  $c_e > 0$ 
5      for each index  $i \in B$ 
6          if  $a_{ie} > 0$ 
7               $\Delta_i = b_i / a_{ie}$ 
8          else  $\Delta_i = \infty$ 
9      choose an index  $l \in B$  that minimizes  $\Delta_i$ 
10     if  $\Delta_l == \infty$ 
11         return “unbounded”
12     else ( $N, B, A, b, c, v$ ) = PIVOT( $N, B, A, b, c, v, l, e$ )
13 for  $i = 1$  to  $n$ 
14     if  $i \in B$ 
15          $\bar{x}_i = b_i$ 
16     else  $\bar{x}_i = 0$ 
17 return ( $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n$ )
```

Returns a slack form with a feasible basic solution (if it exists)

The formal procedure SIMPLEX

SIMPLEX(A, b, c)

```
1  ( $N, B, A, b, c, v$ ) = INITIALIZE-SIMPLEX( $A, b, c$ )
2  let  $\Delta$  be a new vector of length  $m$ 
3  while some index  $j \in N$  has  $c_j > 0$ 
4      choose an index  $e \in N$  for which  $c_e > 0$ 
5      for each index  $i \in B$ 
6          if  $a_{ie} > 0$ 
7               $\Delta_i = b_i / a_{ie}$ 
8          else  $\Delta_i = \infty$ 
9      choose an index  $l \in B$  that minimizes  $\Delta_i$ 
10     if  $\Delta_l == \infty$ 
11         return "unbounded"
12     else ( $N, B, A, b, c, v$ ) = PIVOT( $N, B, A, b, c, v, l, e$ )
13 for  $i = 1$  to  $n$ 
14     if  $i \in B$ 
15          $\bar{x}_i = b_i$ 
16     else  $\bar{x}_i = 0$ 
17 return ( $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n$ )
```

Returns a slack form with a feasible basic solution (if it exists)

Main Loop:

The formal procedure SIMPLEX

SIMPLEX(A, b, c)

```
1  ( $N, B, A, b, c, v$ ) = INITIALIZE-SIMPLEX( $A, b, c$ )
2  let  $\Delta$  be a new vector of length  $m$ 
3  while some index  $j \in N$  has  $c_j > 0$ 
4      choose an index  $e \in N$  for which  $c_e > 0$ 
5      for each index  $i \in B$ 
6          if  $a_{ie} > 0$ 
7               $\Delta_i = b_i / a_{ie}$ 
8          else  $\Delta_i = \infty$ 
9      choose an index  $l \in B$  that minimizes  $\Delta_i$ 
10     if  $\Delta_l = \infty$ 
11         return “unbounded”
12     else ( $N, B, A, b, c, v$ ) = PIVOT( $N, B, A, b, c, v, l, e$ )
13 for  $i = 1$  to  $n$ 
14     if  $i \in B$ 
15          $\bar{x}_i = b_i$ 
16     else  $\bar{x}_i = 0$ 
17 return ( $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n$ )
```

Returns a slack form with a feasible basic solution (if it exists)

Main Loop:

- terminates if all coefficients in objective function are **non-positive**
- Line 4 picks entering variable x_e with **positive** coefficient
- Lines 6 – 9 pick the tightest constraint, associated with x_l
- Line 11 returns “unbounded” if there are no constraints
- Line 12 calls PIVOT, switching roles of x_l and x_e

The formal procedure SIMPLEX

SIMPLEX(A, b, c)

```
1  ( $N, B, A, b, c, v$ ) = INITIALIZE-SIMPLEX( $A, b, c$ )
2  let  $\Delta$  be a new vector of length  $m$ 
3  while some index  $j \in N$  has  $c_j > 0$ 
4      choose an index  $e \in N$  for which  $c_e > 0$ 
5      for each index  $i \in B$ 
6          if  $a_{ie} > 0$ 
7               $\Delta_i = b_i / a_{ie}$ 
8          else  $\Delta_i = \infty$ 
9      choose an index  $l \in B$  that minimizes  $\Delta_i$ 
10     if  $\Delta_l = \infty$ 
11         return "unbounded"
12     else ( $N, B, A, b, c, v$ ) = PIVOT( $N, B, A, b, c, v, l, e$ )
13 for  $i = 1$  to  $n$ 
14     if  $i \in B$ 
15          $\bar{x}_i = b_i$ 
16     else  $\bar{x}_i = 0$ 
17 return ( $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n$ )
```

Returns a slack form with a feasible basic solution (if it exists)

Main Loop:

- terminates if all coefficients in objective function are **non-positive**
- Line 4 picks entering variable x_e with **positive** coefficient
- Lines 6 – 9 pick the tightest constraint, associated with x_l
- Line 11 returns "unbounded" if there are no constraints
- Line 12 calls PIVOT, switching roles of x_l and x_e

Return corresponding solution.

The formal procedure **SIMPLEX**

SIMPLEX(A, b, c)

```
1  ( $N, B, A, b, c, v$ ) = INITIALIZE-SIMPLEX( $A, b, c$ )
2  let  $\Delta$  be a new vector of length  $m$ 
3  while some index  $j \in N$  has  $c_j > 0$ 
4      choose an index  $e \in N$  for which  $c_e > 0$ 
5      for each index  $i \in B$ 
6          if  $a_{ie} > 0$ 
7               $\Delta_i = b_i / a_{ie}$ 
8          else  $\Delta_i = \infty$ 
9      choose an index  $l \in B$  that minimizes  $\Delta_i$ 
10     if  $\Delta_l == \infty$ 
11         return “unbounded”
12     else ( $N, B, A, b, c, v$ ) = PIVOT( $N, B, A, b, c, v, l, e$ )
13 for  $i = 1$  to  $n$ 
14     if  $i \in B$ 
15          $\bar{x}_i = b_i$ 
16     else  $\bar{x}_i = 0$ 
17 return ( $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n$ )
```

Returns a slack form with a feasible basic solution (if it exists)

Lemma 29.2

Suppose the call to INITIALIZE-SIMPLEX in line 1 returns a slack form for which the basic solution is feasible. Then if SIMPLEX returns a solution, it is a feasible solution. If SIMPLEX returns “unbounded”, the linear program is unbounded.

The formal procedure **SIMPLEX**

SIMPLEX(A, b, c)

```
1  ( $N, B, A, b, c, v$ ) = INITIALIZE-SIMPLEX( $A, b, c$ )
2  let  $\Delta$  be a new vector of length  $m$ 
3  while some index  $j \in N$  has  $c_j > 0$ 
4      choose an index  $e \in N$  for which  $c_e > 0$ 
5      for each index  $i \in B$ 
6          if  $a_{ie} > 0$ 
7               $\Delta_i = b_i / a_{ie}$ 
8          else  $\Delta_i = \infty$ 
9      choose an index  $l \in B$  that minimizes  $\Delta_i$ 
10     if  $\Delta_l == \infty$ 
11     return “unbounded”
```

Returns a slack form with a feasible basic solution (if it exists)

Proof is based on the following three-part loop invariant:

— Lemma 29.2 —

Suppose the call to INITIALIZE-SIMPLEX in line 1 returns a slack form for which the basic solution is feasible. Then if SIMPLEX returns a solution, it is a feasible solution. If SIMPLEX returns “unbounded”, the linear program is unbounded.

The formal procedure **SIMPLEX**

SIMPLEX(A, b, c)

```
1  ( $N, B, A, b, c, v$ ) = INITIALIZE-SIMPLEX( $A, b, c$ )
2  let  $\Delta$  be a new vector of length  $m$ 
3  while some index  $j \in N$  has  $c_j > 0$ 
4      choose an index  $e \in N$  for which  $c_e > 0$ 
5      for each index  $i \in B$ 
6          if  $a_{ie} > 0$ 
7               $\Delta_i = b_i / a_{ie}$ 
8          else  $\Delta_i = \infty$ 
9      choose an index  $l \in B$  that minimizes  $\Delta_i$ 
10     if  $\Delta_l == \infty$ 
11         return “unbounded”
```

Returns a slack form with a feasible basic solution (if it exists)

Proof is based on the following three-part loop invariant:

1. the slack form is always equivalent to the one returned by **INITIALIZE-SIMPLEX**,
2. for each $i \in B$, we have $b_i \geq 0$,
3. the basic solution associated with the (current) slack form is feasible.

Lemma 29.2

Suppose the call to **INITIALIZE-SIMPLEX** in line 1 returns a slack form for which the basic solution is feasible. Then if **SIMPLEX** returns a solution, it is a feasible solution. If **SIMPLEX** returns “unbounded”, the linear program is unbounded.

Simplex Algorithm by Example

Details of the Simplex Algorithm

Finding an Initial Solution

Appendix: Cycling and Termination (non-examinable)

Finding an Initial Solution

$$\begin{array}{llllll} \text{maximise} & 2x_1 & - & x_2 & & \\ \text{subject to} & & & & & \\ & 2x_1 & - & x_2 & \leq & 2 \\ & x_1 & - & 5x_2 & \leq & -4 \\ & & & x_1, x_2 & \geq & 0 \end{array}$$

Finding an Initial Solution

maximise
subject to

$$2x_1 - x_2$$

$$2x_1 - x_2 \leq 2$$

$$x_1 - 5x_2 \leq -4$$

$$x_1, x_2 \geq 0$$



Conversion into slack form

Finding an Initial Solution

maximise
subject to

$$2x_1 - x_2$$

$$2x_1 - x_2 \leq 2$$

$$x_1 - 5x_2 \leq -4$$

$$x_1, x_2 \geq 0$$

Conversion into slack form

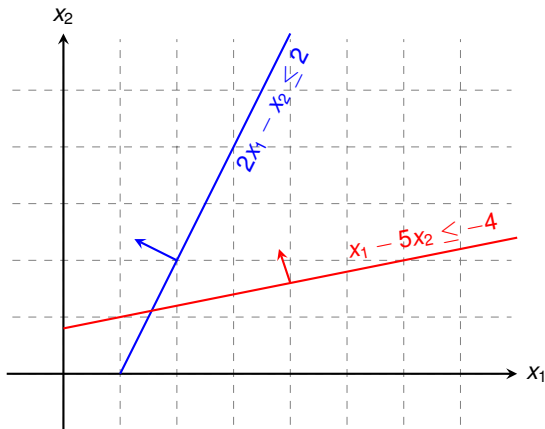
$$\begin{aligned} z &= 2x_1 - x_2 \\ x_3 &= 2 - 2x_1 + x_2 \\ x_4 &= -4 - x_1 + 5x_2 \end{aligned}$$

Basic solution $(x_1, x_2, x_3, x_4) = (0, 0, 2, -4)$ is not feasible!

Geometric Illustration

maximise
subject to

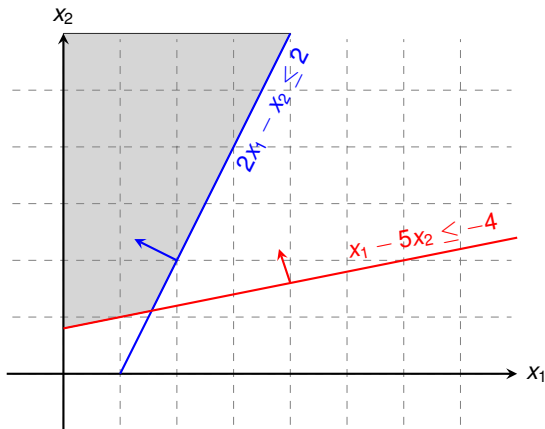
$$\begin{array}{rclcl} 2x_1 & - & x_2 & & \\ 2x_1 & - & x_2 & \leq & 2 \\ x_1 & - & 5x_2 & \leq & -4 \\ x_1, x_2 & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise
subject to

$$\begin{array}{rclcl} 2x_1 & - & x_2 & & \\ 2x_1 & - & x_2 & \leq & 2 \\ x_1 & - & 5x_2 & \leq & -4 \\ x_1, x_2 & \geq & 0 & & \end{array}$$



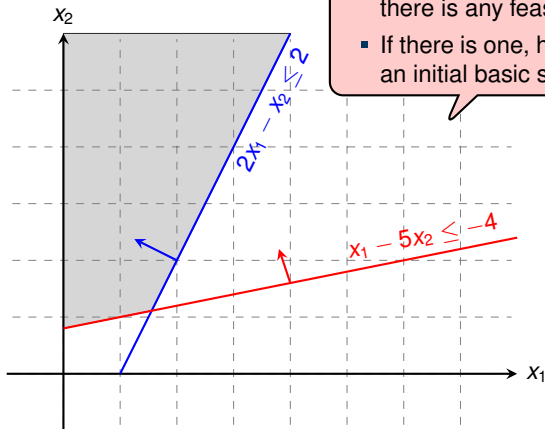
Geometric Illustration

maximise
subject to

$$\begin{array}{rclcl} 2x_1 & - & x_2 & & \\ 2x_1 & - & x_2 & \leq & 2 \\ x_1 & - & 5x_2 & \leq & -4 \\ x_1, x_2 & & & \geq & 0 \end{array}$$

Questions:

- How to determine whether there is any feasible solution?
- If there is one, how to determine an initial basic solution?



Formulating an Auxiliary Linear Program

maximise $\sum_{j=1}^n c_j x_j$
subject to

$$\begin{aligned} \sum_{j=1}^n a_{ij} x_j &\leq b_i && \text{for } i = 1, 2, \dots, m, \\ x_j &\geq 0 && \text{for } j = 1, 2, \dots, n \end{aligned}$$

Formulating an Auxiliary Linear Program

maximise
subject to

$$\sum_{j=1}^n c_j x_j$$

$$\begin{aligned} \sum_{j=1}^n a_{ij} x_j &\leq b_i && \text{for } i = 1, 2, \dots, m, \\ x_j &\geq 0 && \text{for } j = 1, 2, \dots, n \end{aligned}$$

↓
Formulating an Auxiliary Linear Program

Formulating an Auxiliary Linear Program

maximise $\sum_{j=1}^n c_j x_j$
subject to

$$\begin{aligned} \sum_{j=1}^n a_{ij} x_j &\leq b_i && \text{for } i = 1, 2, \dots, m, \\ x_j &\geq 0 && \text{for } j = 1, 2, \dots, n \end{aligned}$$

↓
Formulating an Auxiliary Linear Program

maximise $-x_0$
subject to

$$\begin{aligned} \sum_{j=1}^n a_{ij} x_j - x_0 &\leq b_i && \text{for } i = 1, 2, \dots, m, \\ x_j &\geq 0 && \text{for } j = 0, 1, \dots, n \end{aligned}$$

Formulating an Auxiliary Linear Program

maximise $\sum_{j=1}^n c_j x_j$
subject to

$$\begin{aligned}\sum_{j=1}^n a_{ij} x_j &\leq b_i && \text{for } i = 1, 2, \dots, m, \\ x_j &\geq 0 && \text{for } j = 1, 2, \dots, n\end{aligned}$$

↓ Formulating an Auxiliary Linear Program

maximise $-x_0$
subject to

$$\begin{aligned}\sum_{j=1}^n a_{ij} x_j - x_0 &\leq b_i && \text{for } i = 1, 2, \dots, m, \\ x_j &\geq 0 && \text{for } j = 0, 1, \dots, n\end{aligned}$$

— Lemma 29.11 —

Let L_{aux} be the auxiliary LP of a linear program L in standard form. Then L is feasible if and only if the optimal objective value of L_{aux} is 0.

Formulating an Auxiliary Linear Program

maximise $\sum_{j=1}^n c_j x_j$
subject to

$$\begin{aligned} \sum_{j=1}^n a_{ij} x_j &\leq b_i && \text{for } i = 1, 2, \dots, m, \\ x_j &\geq 0 && \text{for } j = 1, 2, \dots, n \end{aligned}$$

↓ Formulating an Auxiliary Linear Program

maximise $-x_0$
subject to

$$\begin{aligned} \sum_{j=1}^n a_{ij} x_j - x_0 &\leq b_i && \text{for } i = 1, 2, \dots, m, \\ x_j &\geq 0 && \text{for } j = 0, 1, \dots, n \end{aligned}$$

— Lemma 29.11 —

Let L_{aux} be the auxiliary LP of a linear program L in standard form. Then L is feasible if and only if the optimal objective value of L_{aux} is 0.

Proof. Exercise!

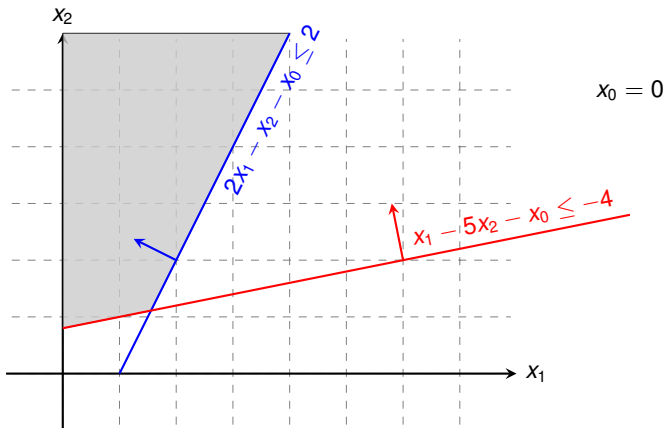
- Let us illustrate the role of x_0 as “distance from feasibility”

- Let us illustrate the role of x_0 as “distance from feasibility”
- We’ll also see that increasing x_0 enlarges the feasible region

Geometric Illustration

maximise $-x_0$
subject to

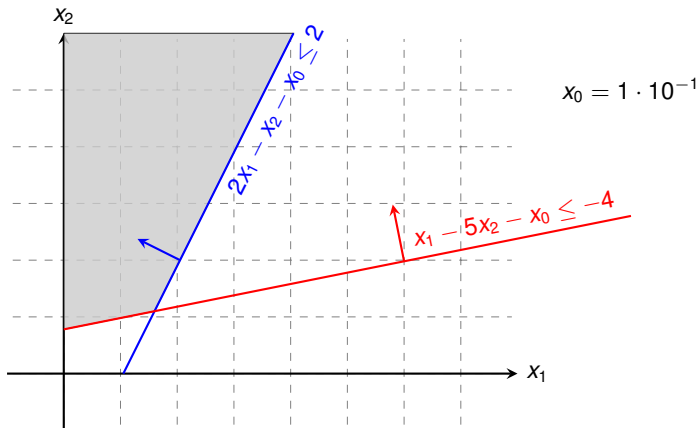
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

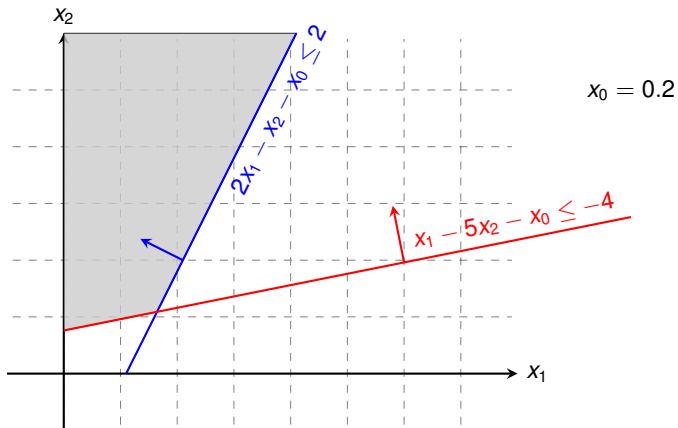
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

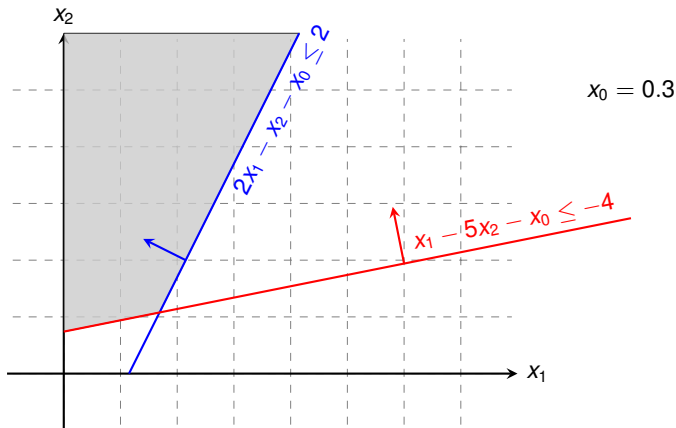
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

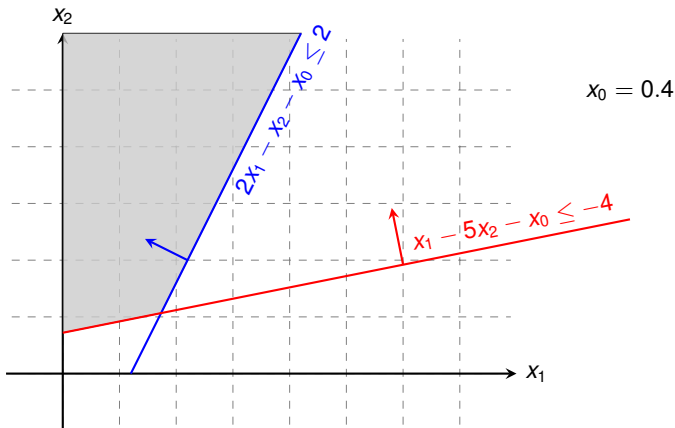
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

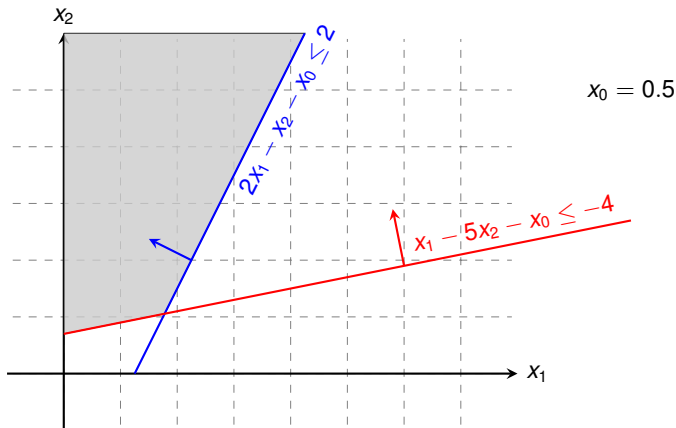
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

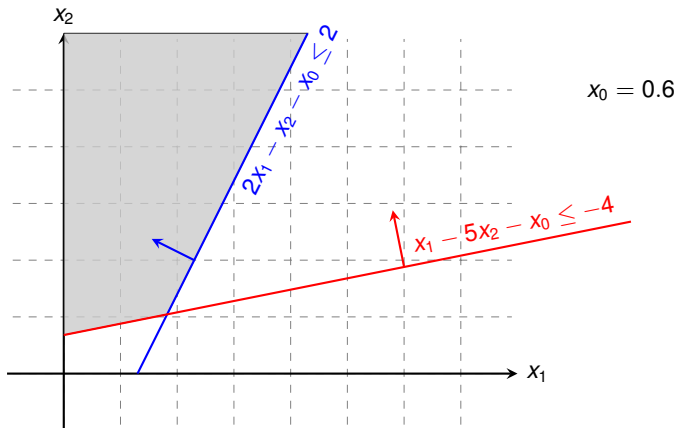
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

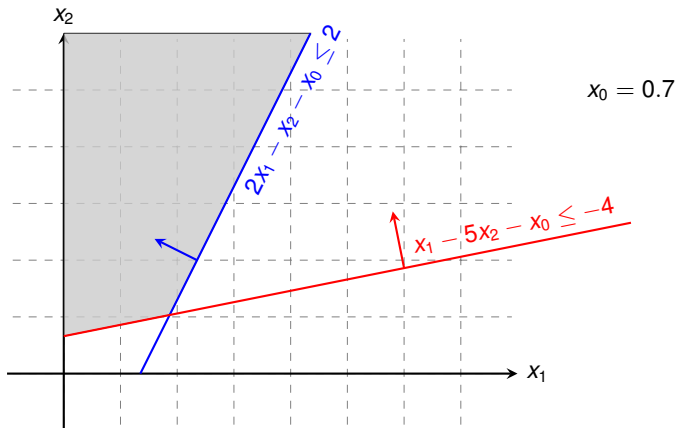
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

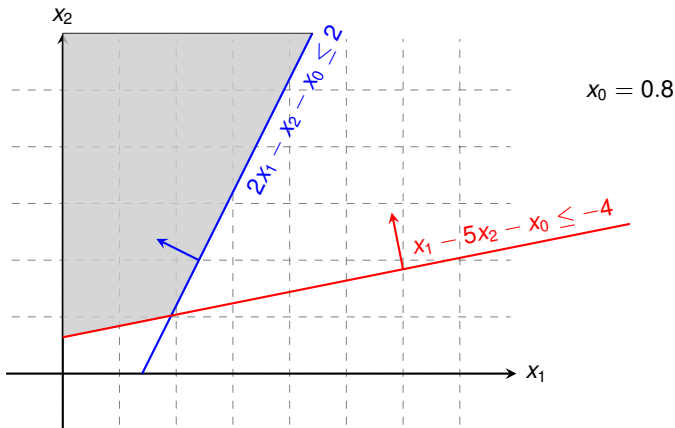
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

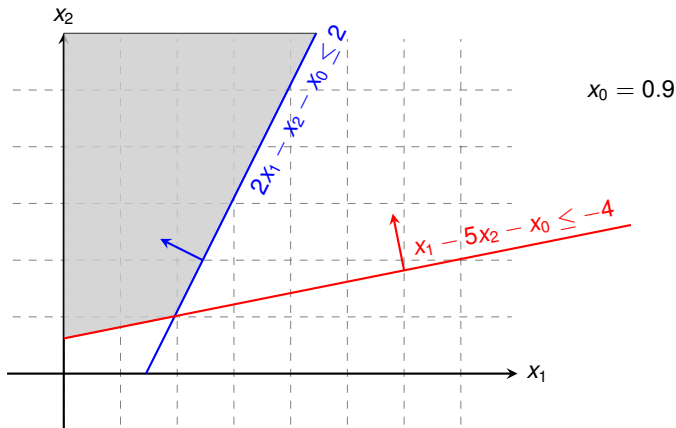
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

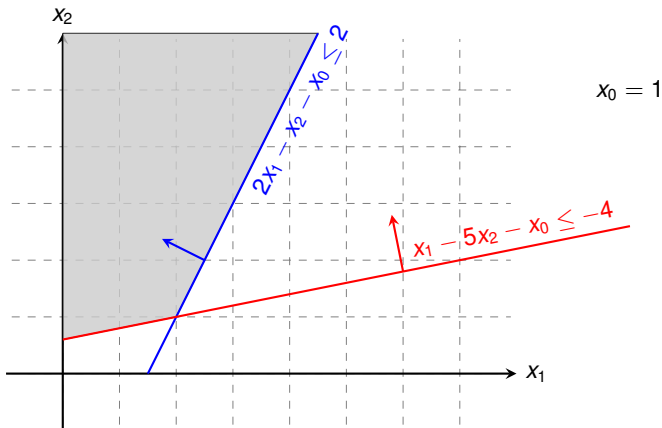
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

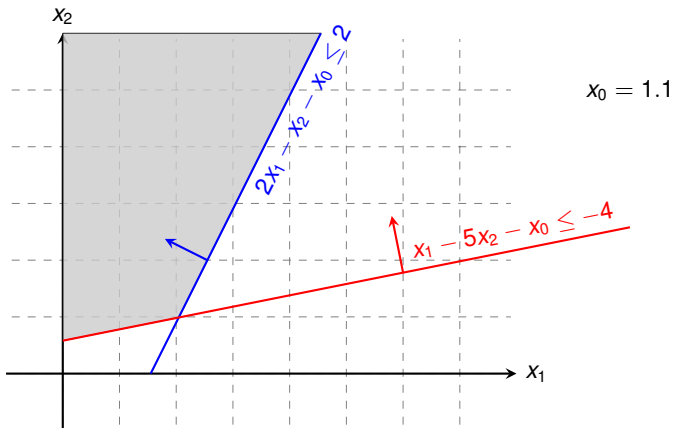
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

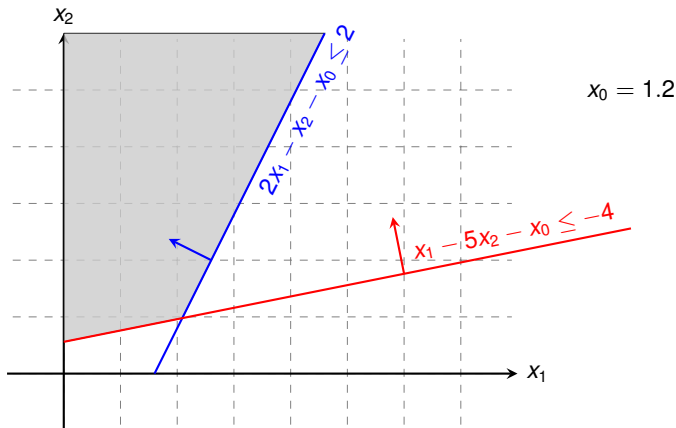
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

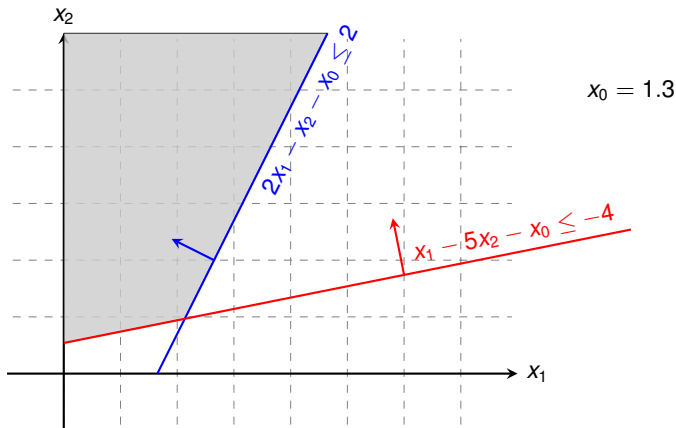
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

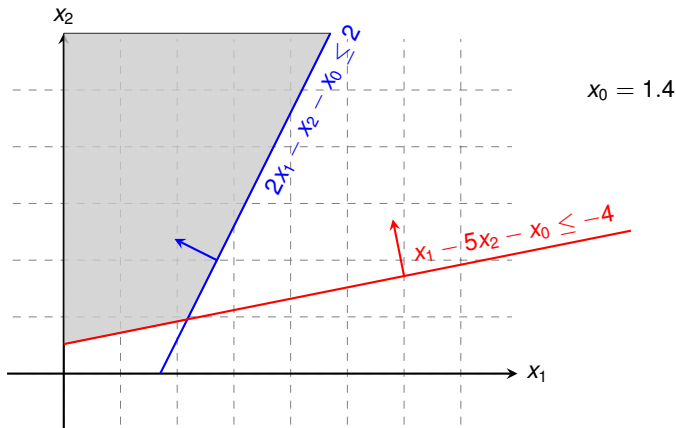
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

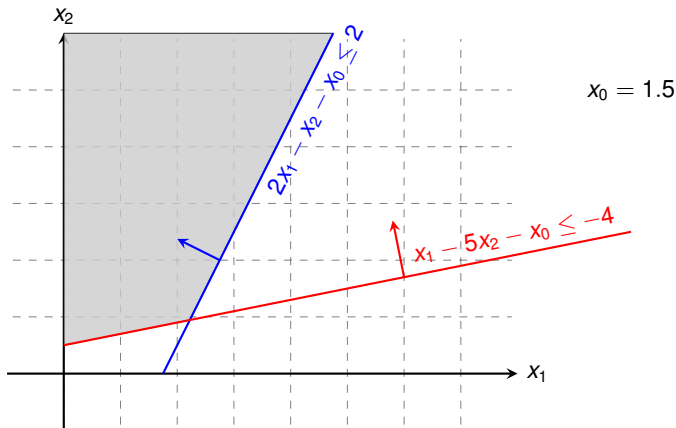
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

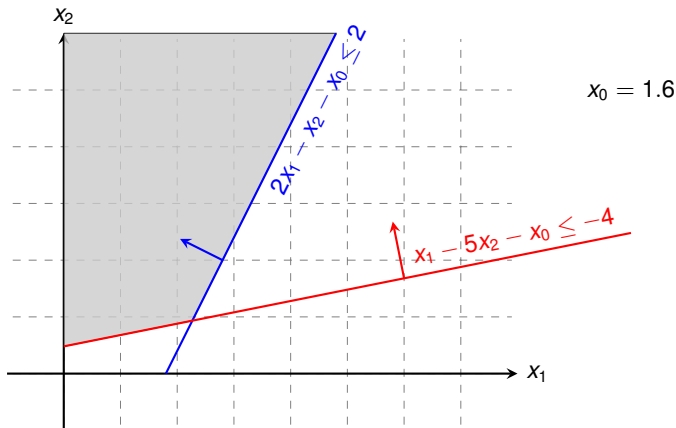
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

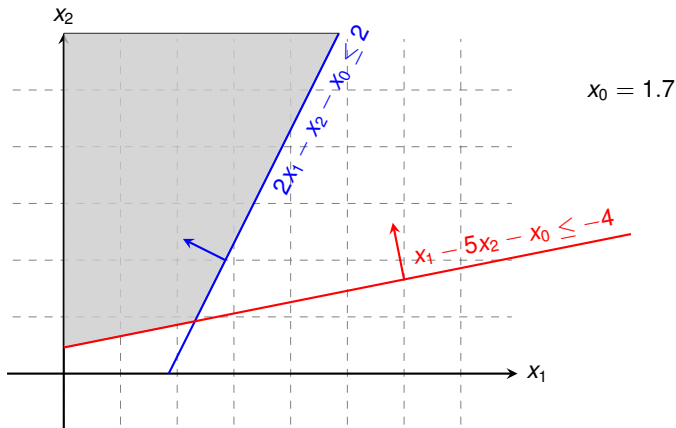
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

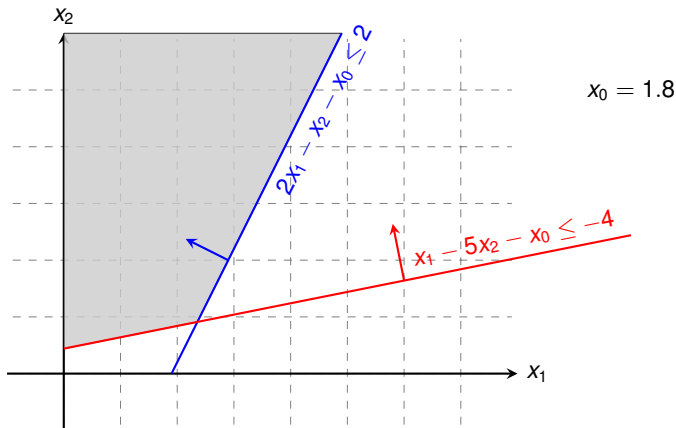
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

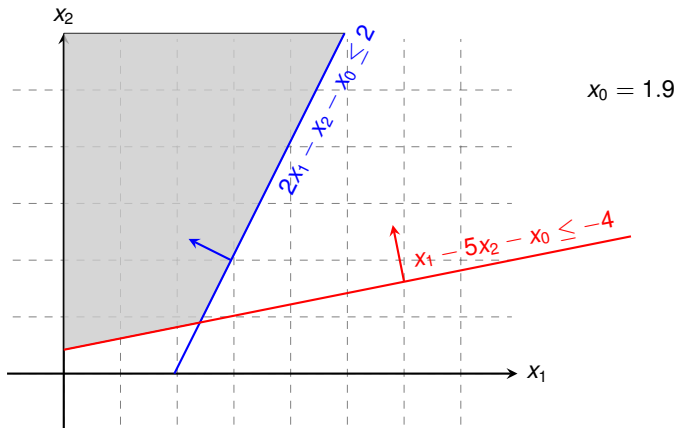
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

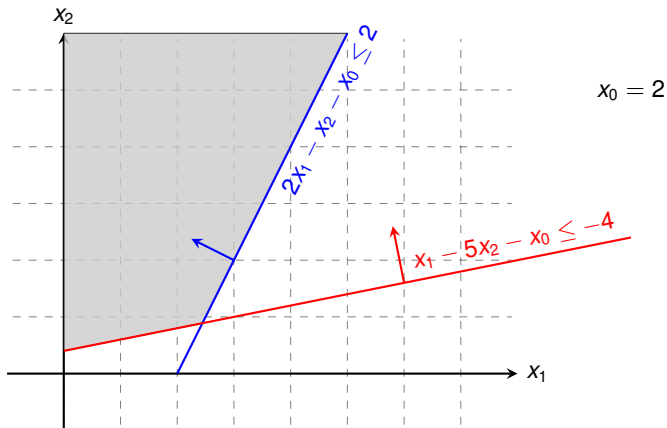
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

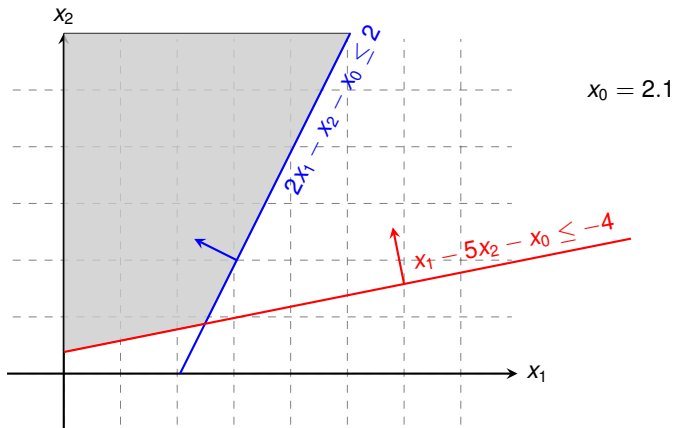
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

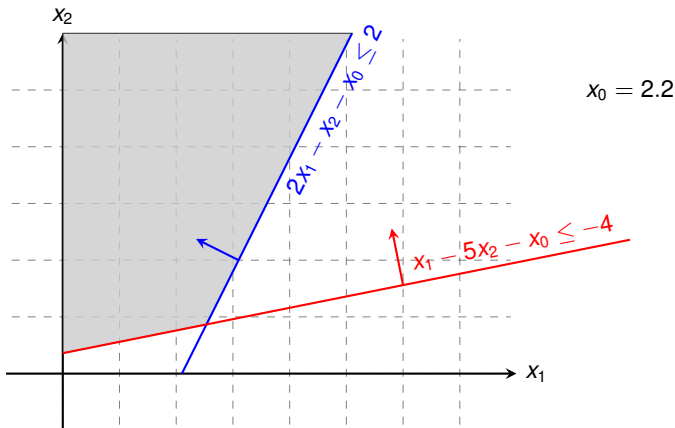
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

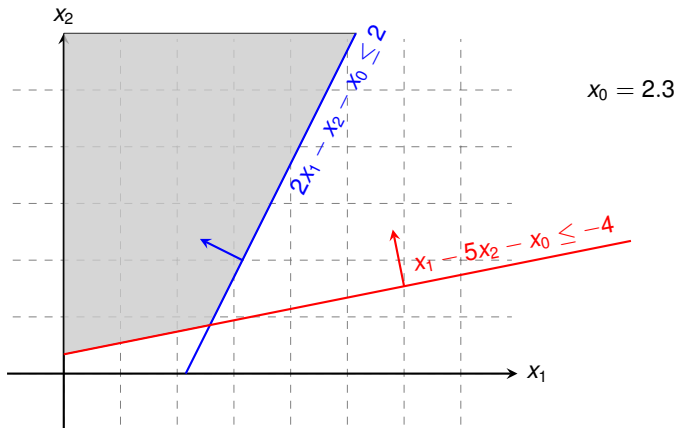
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

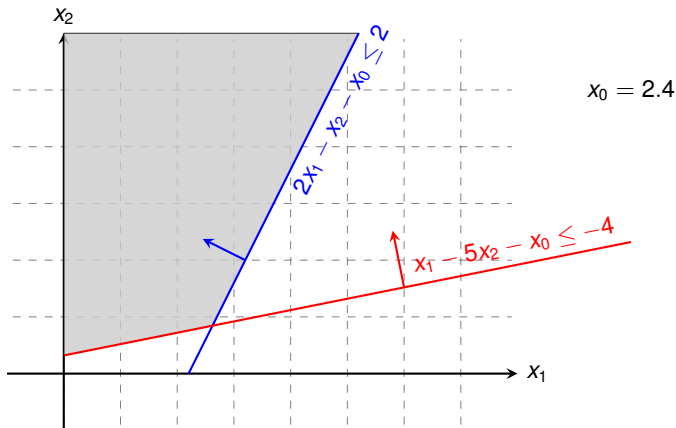
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

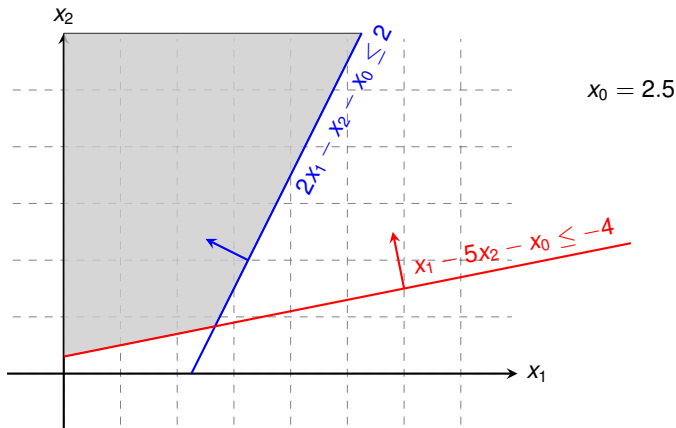
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

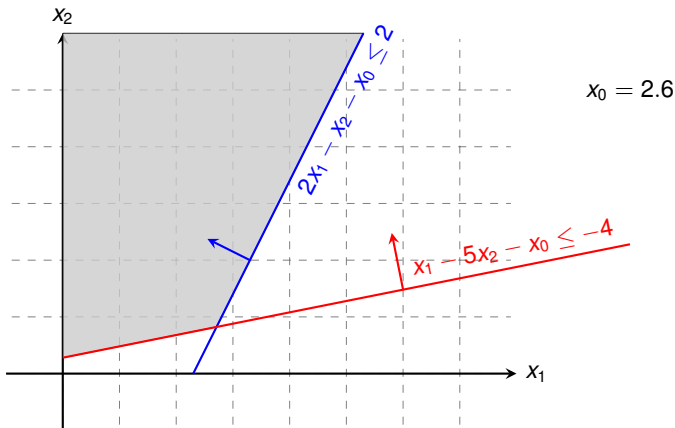
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

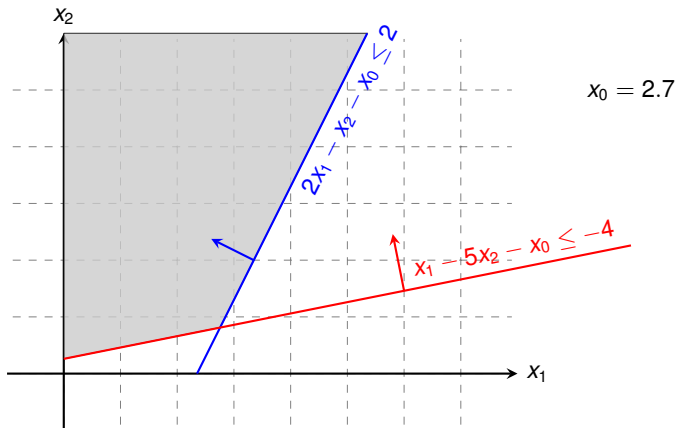
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
 subject to

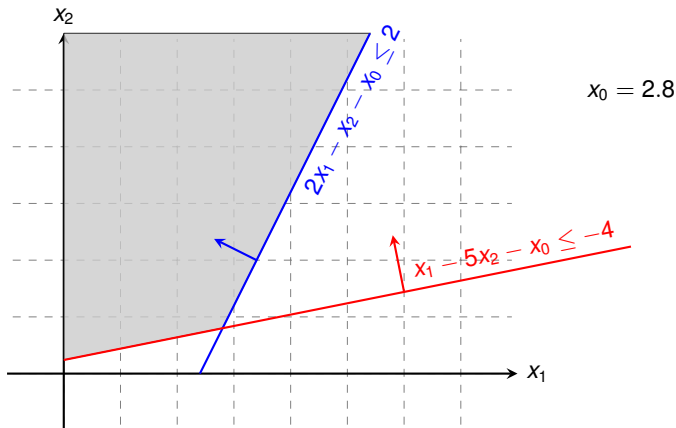
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

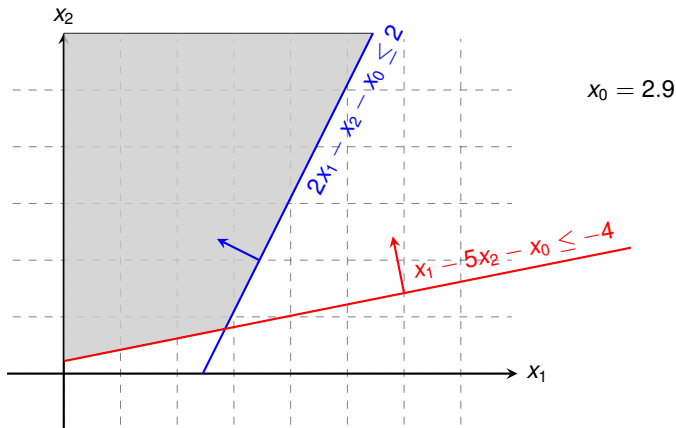
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

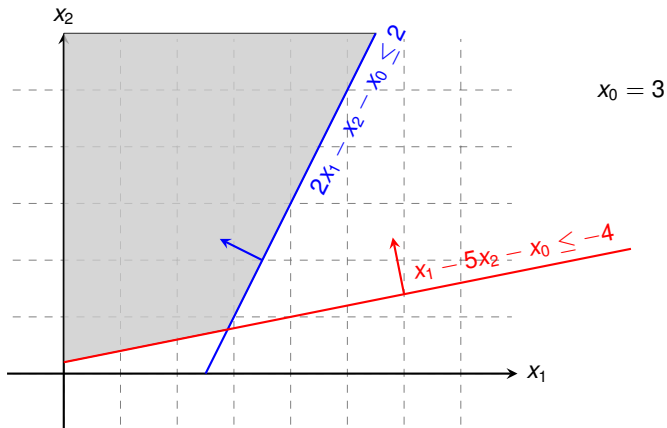
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

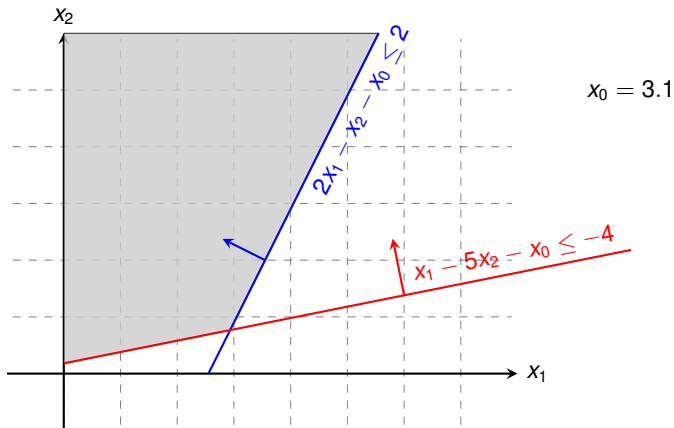
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

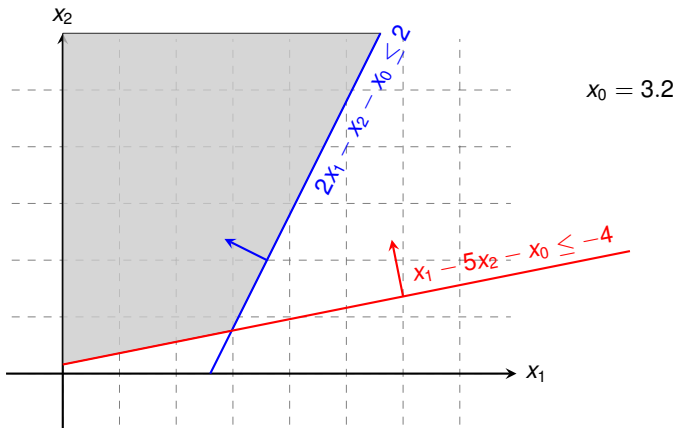
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

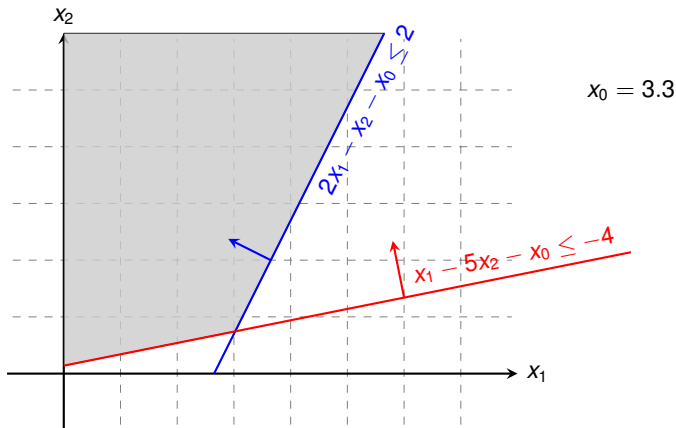
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

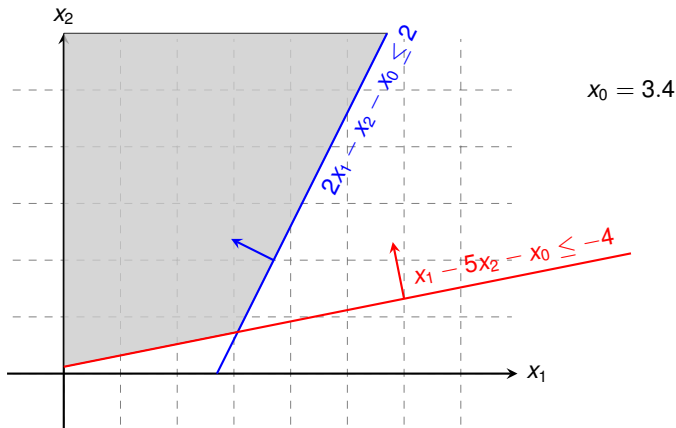
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

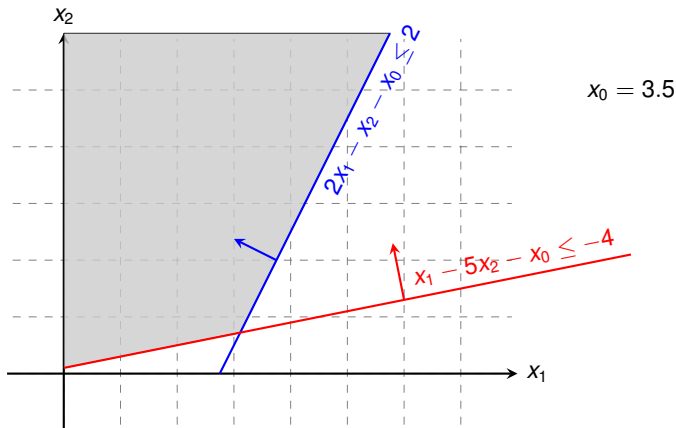
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

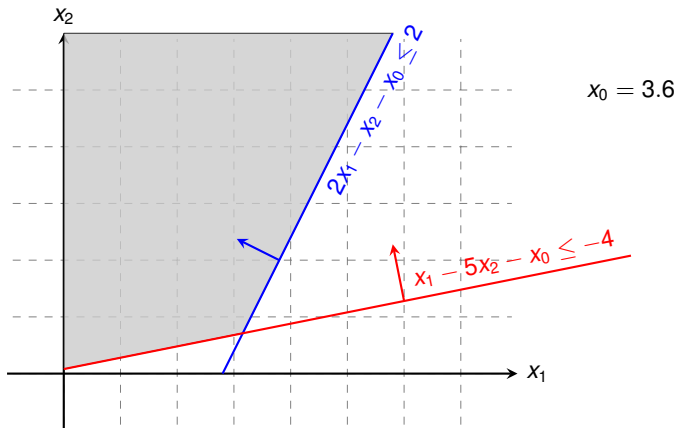
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

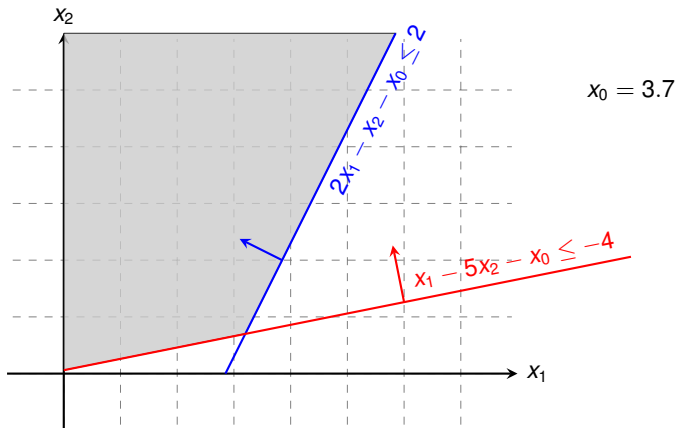
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

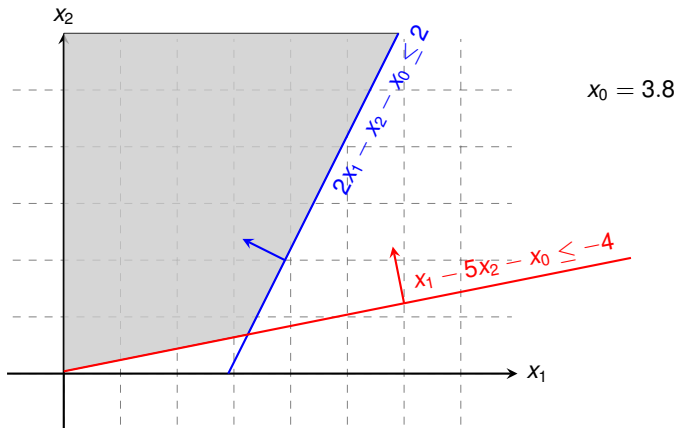
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

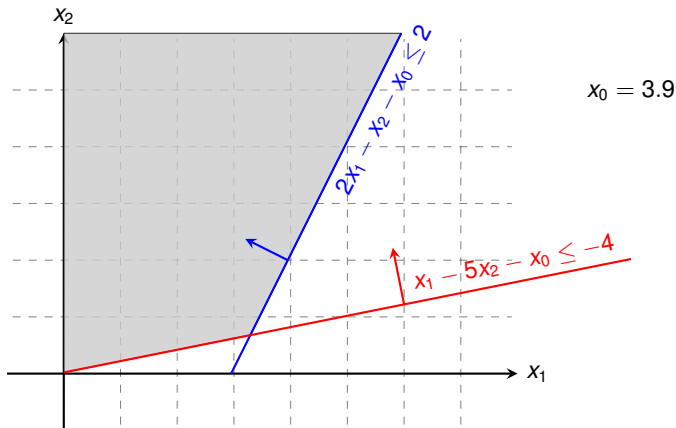
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

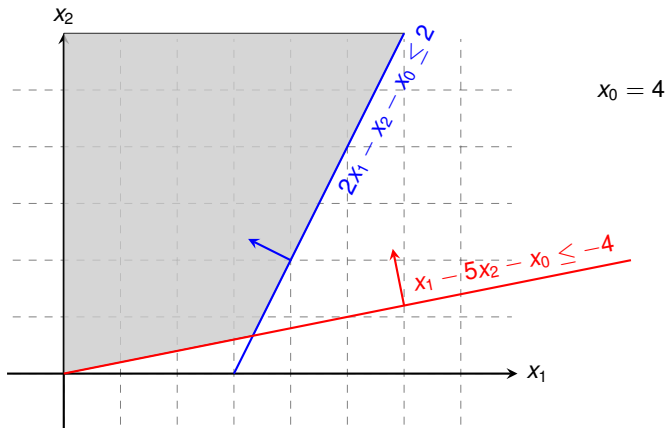
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



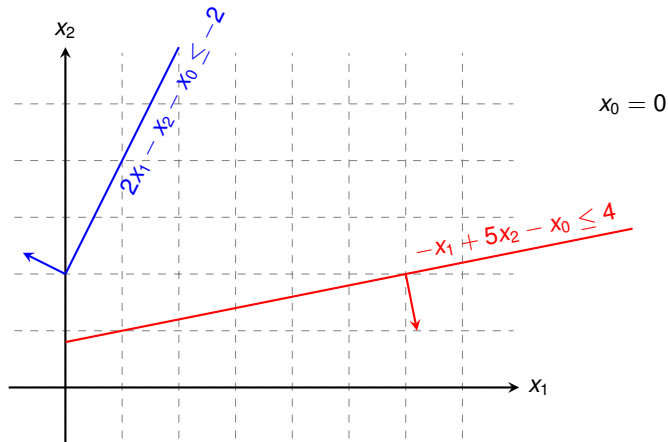
- Let us now modify the original linear program so that it is not feasible

- Let us now modify the original linear program so that it is **not feasible**
- ⇒ Hence the auxiliary linear program has only a solution for a sufficiently large $x_0 > 0$!

Geometric Illustration

maximise $-x_0$
subject to

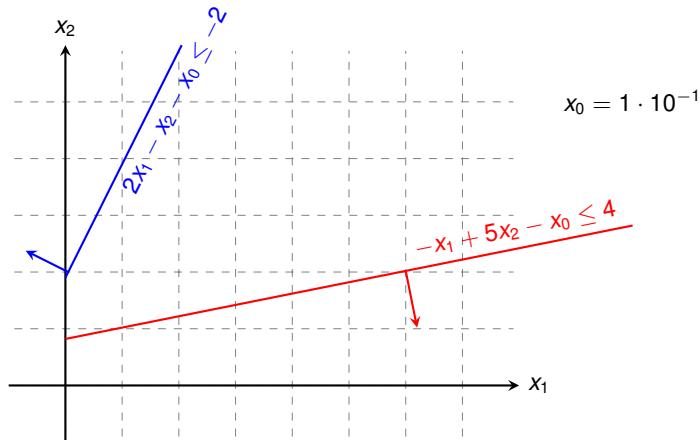
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

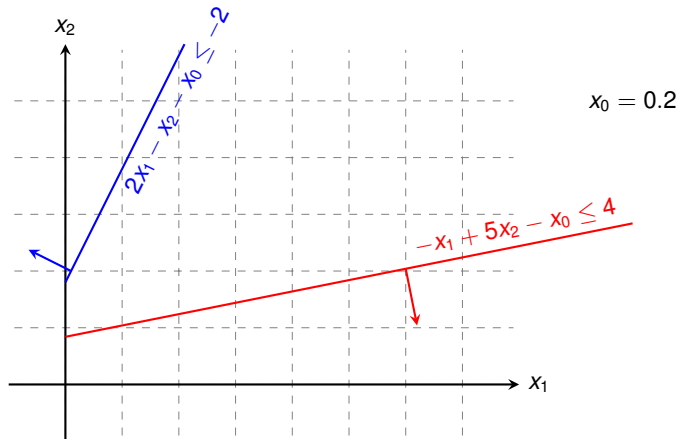
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

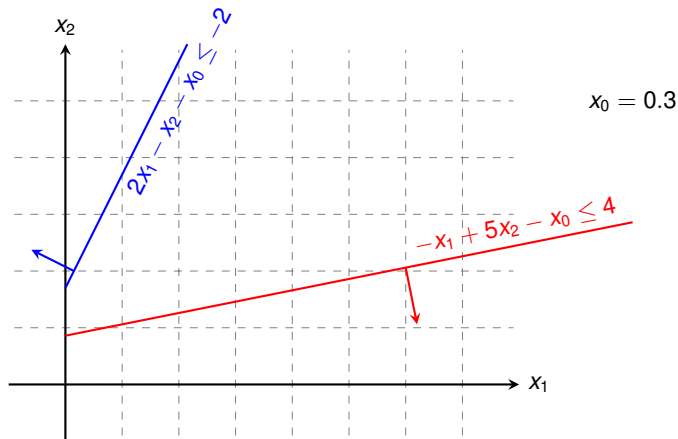
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

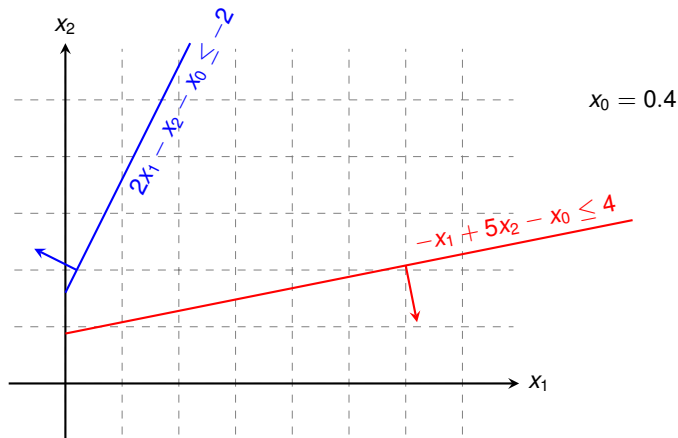
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

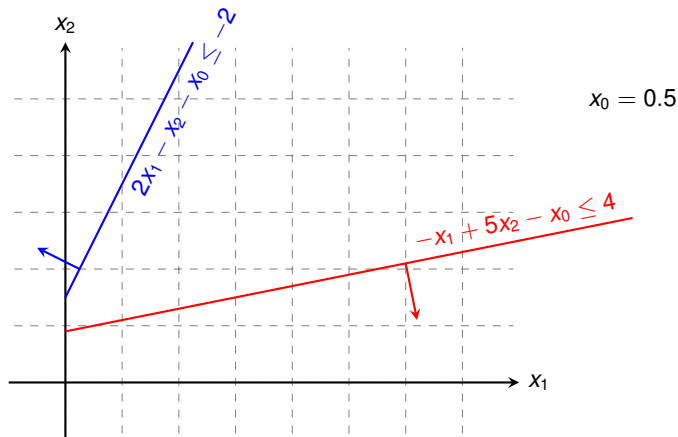
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

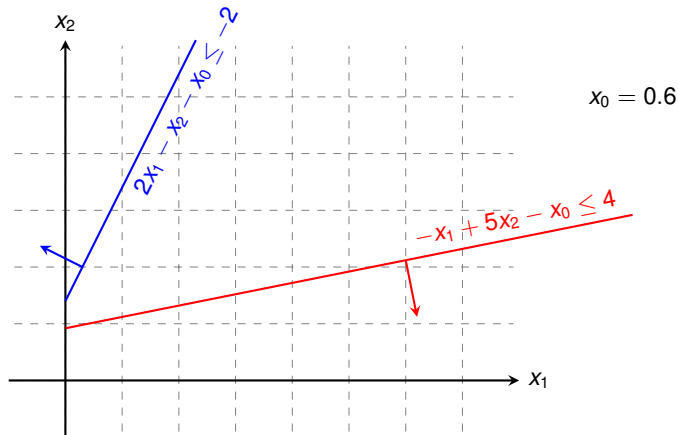
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ & & x_0, x_1, x_2 & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

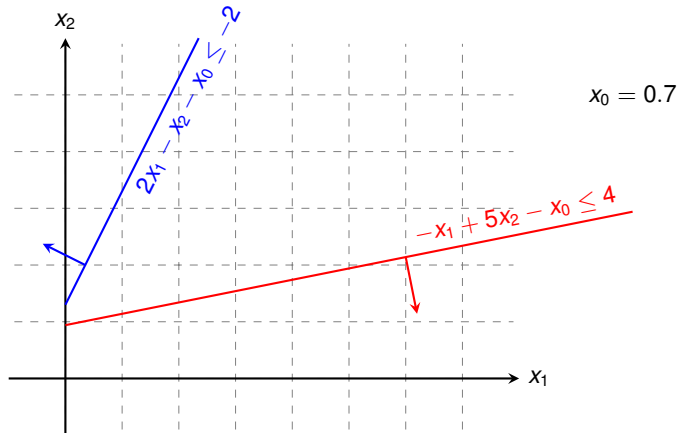
$$\begin{array}{rccccccc} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ & & x_0, x_1, x_2 & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

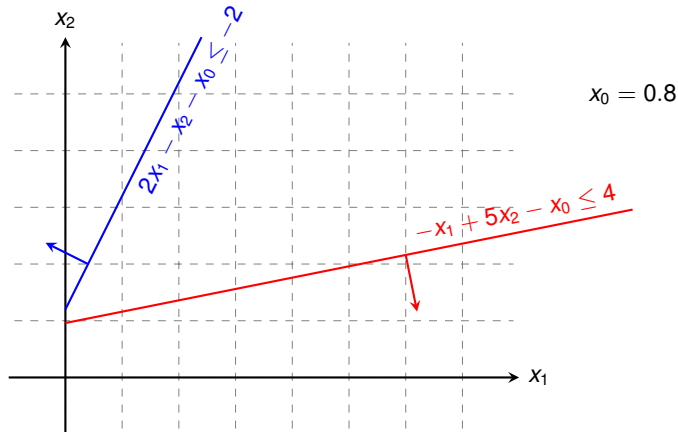
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ & & x_0, x_1, x_2 & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

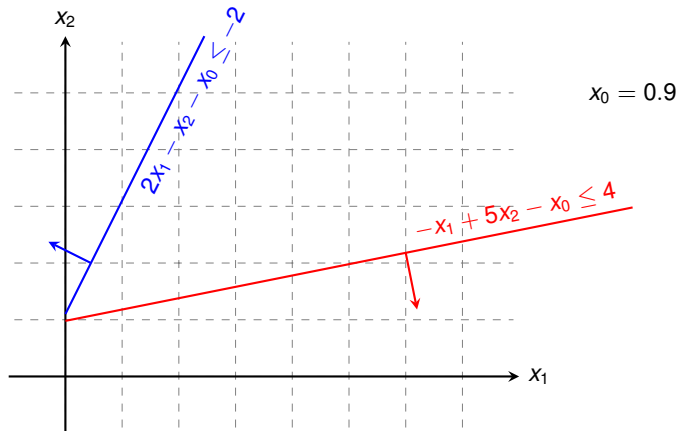
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

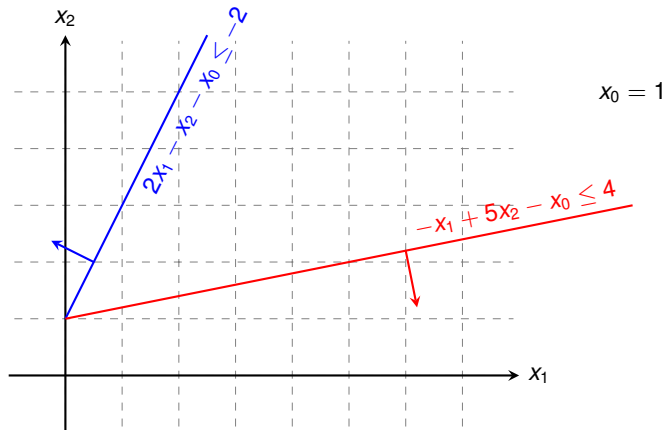
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

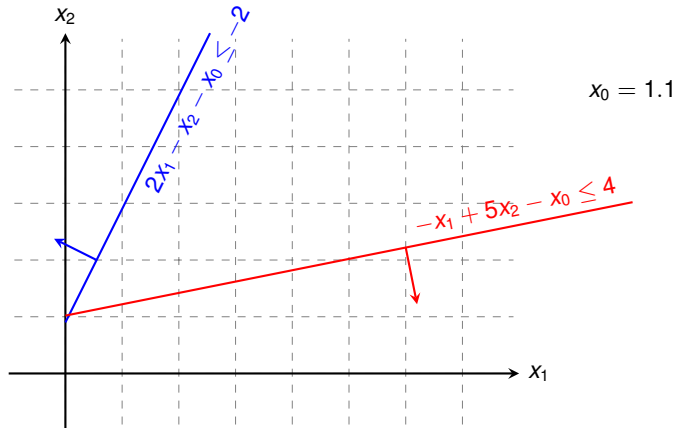
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

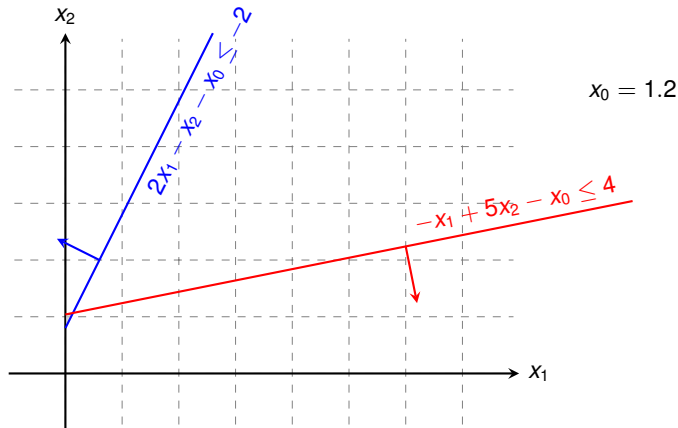
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

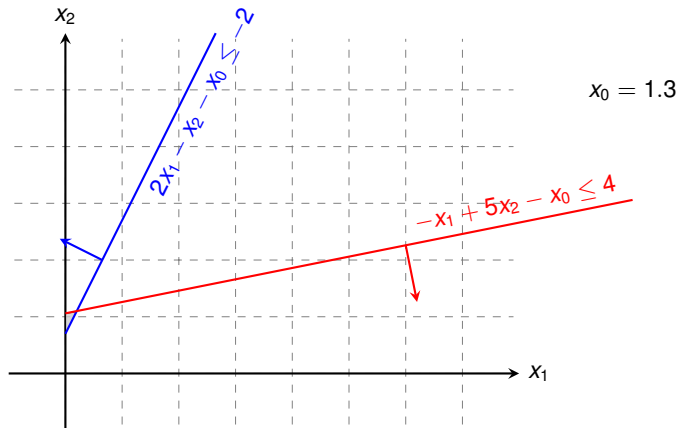
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ & & x_0, x_1, x_2 & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

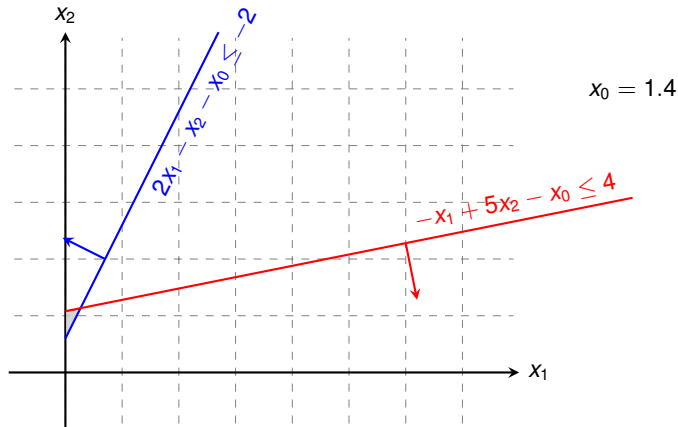
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

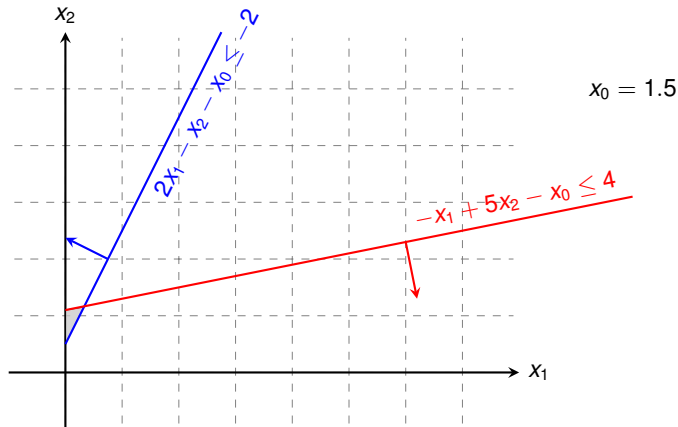
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ & & x_0, x_1, x_2 & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

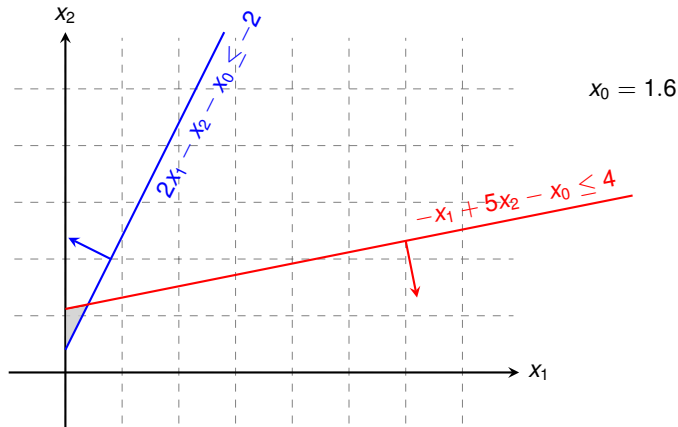
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

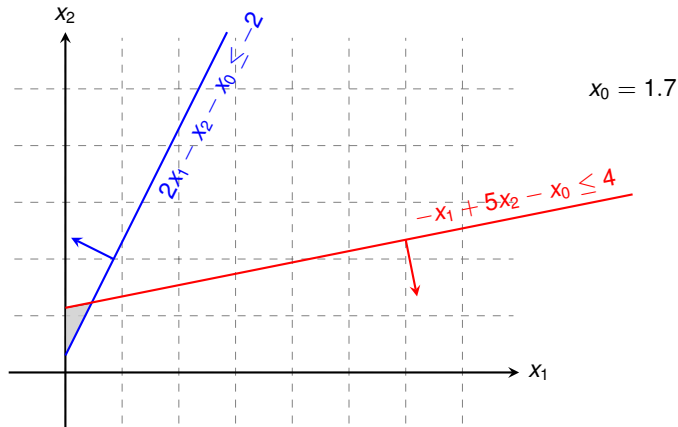
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

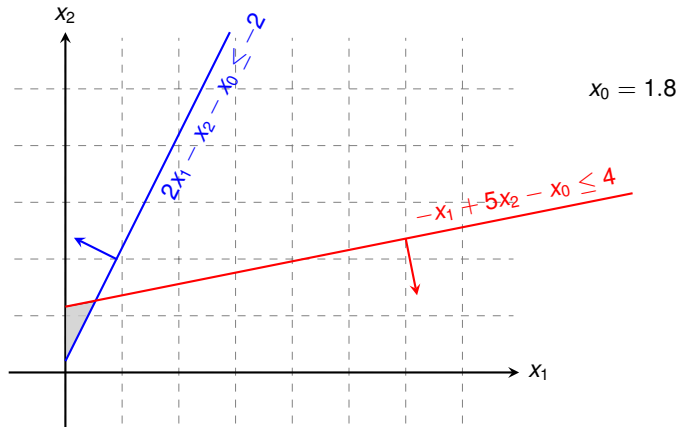
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

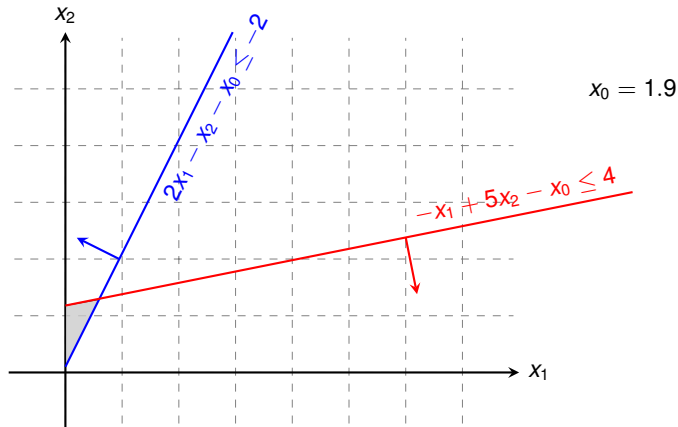
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

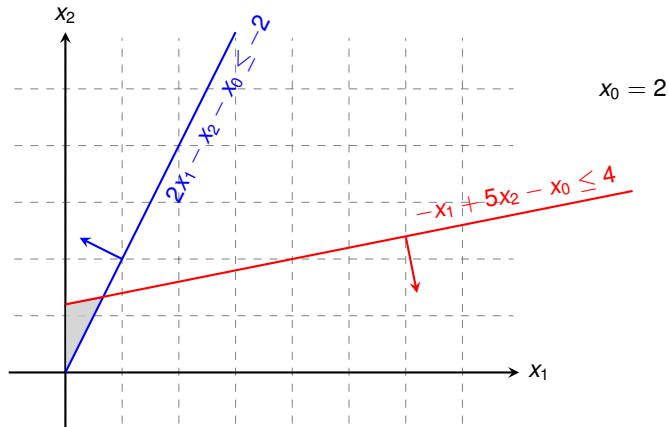
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

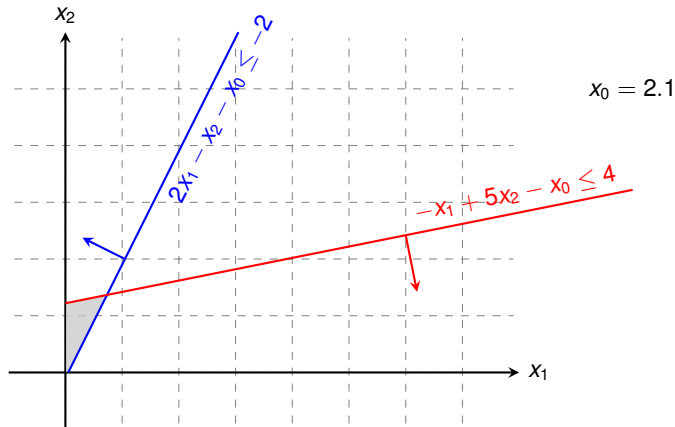
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

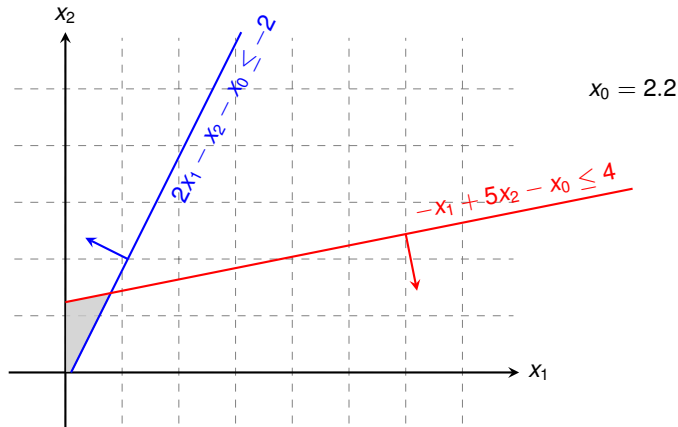
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

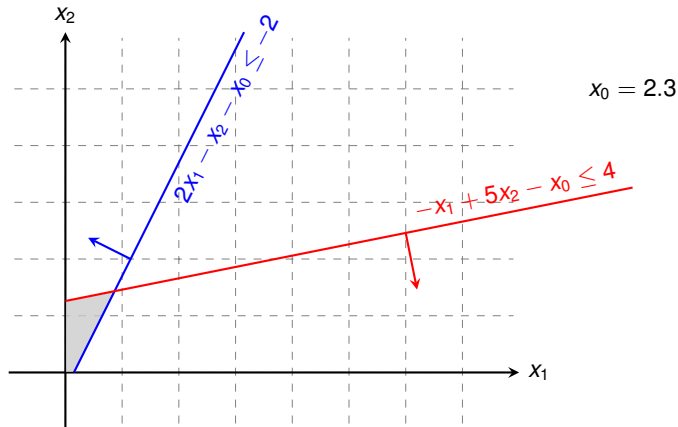
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise
subject to

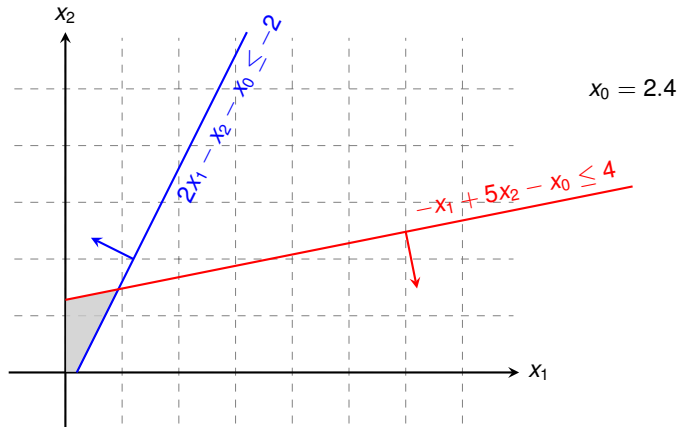
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

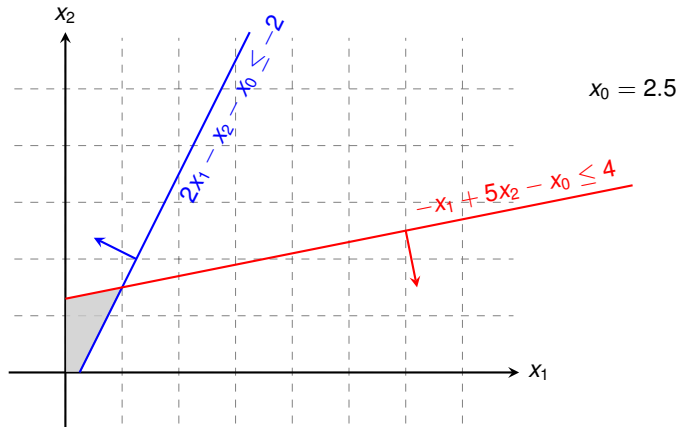
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

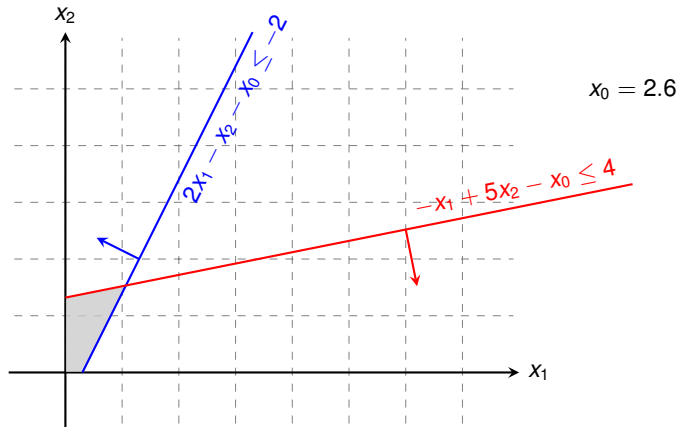
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ & & x_0, x_1, x_2 & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

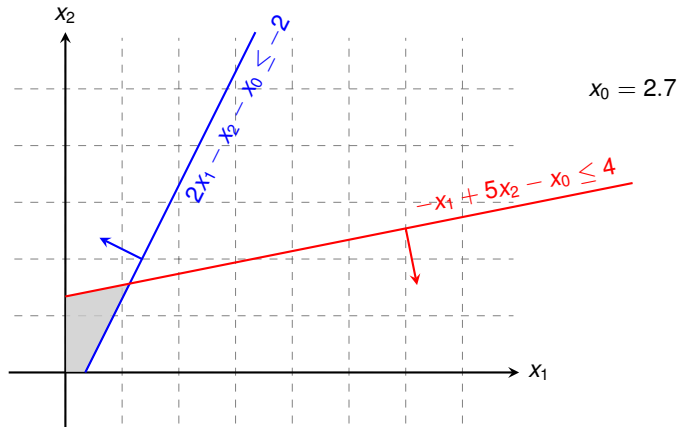
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ & & x_0, x_1, x_2 & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

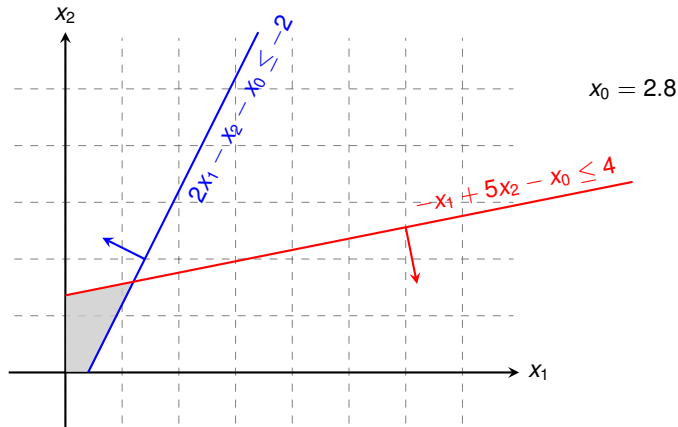
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ & & x_0, x_1, x_2 & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

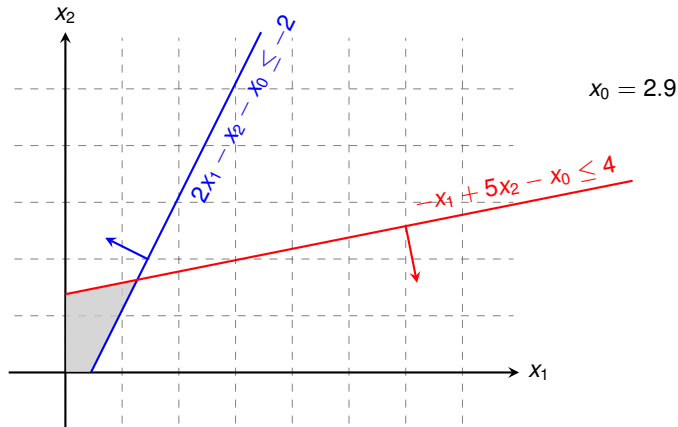
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
 subject to

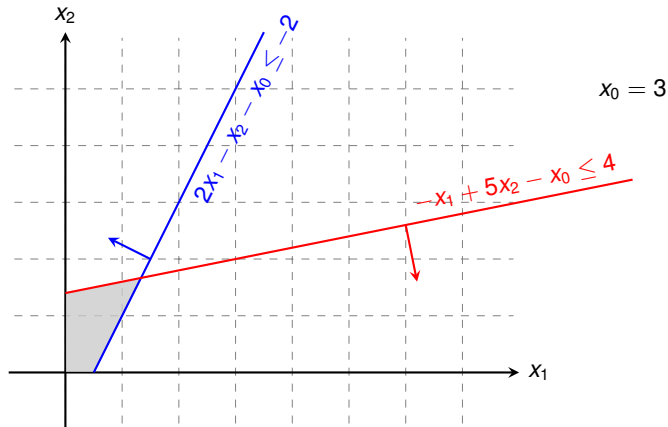
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

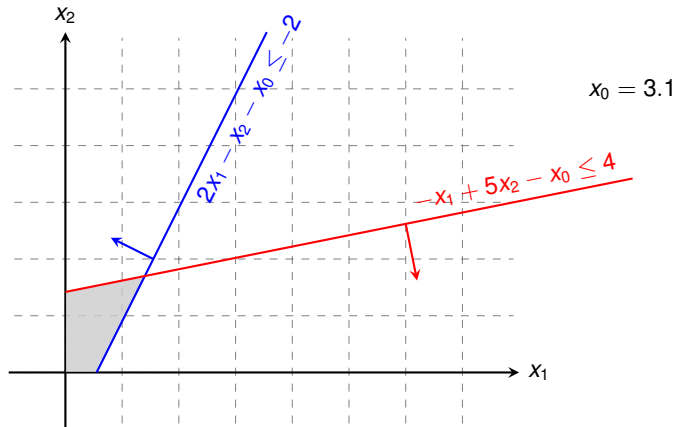
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

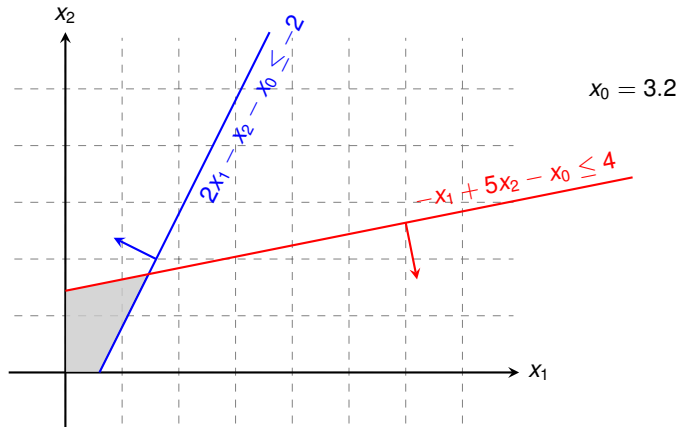
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

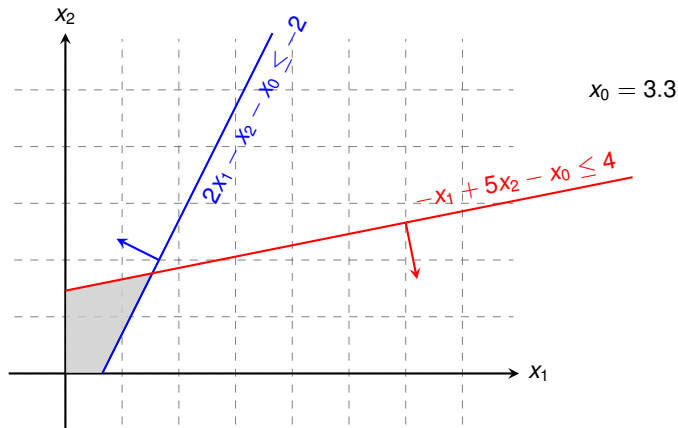
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise
subject to

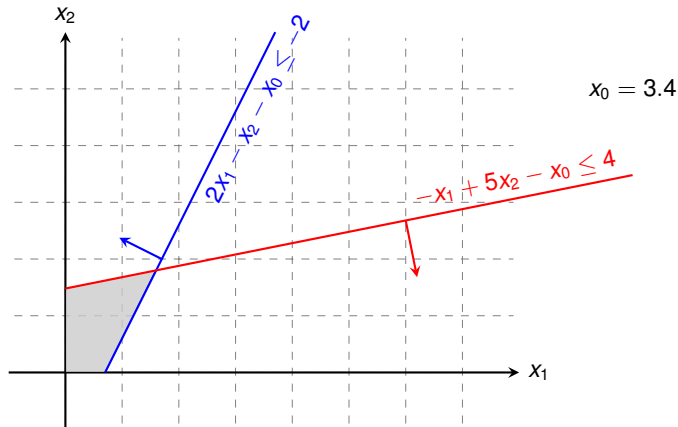
$$\begin{array}{rcccccccl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ & & x_0, x_1, x_2 & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

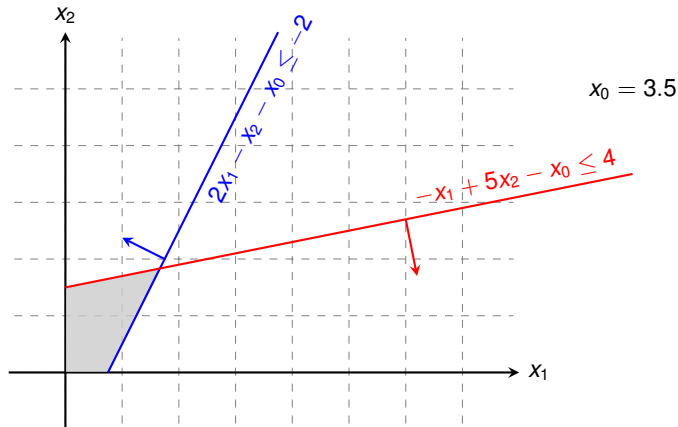
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise
subject to

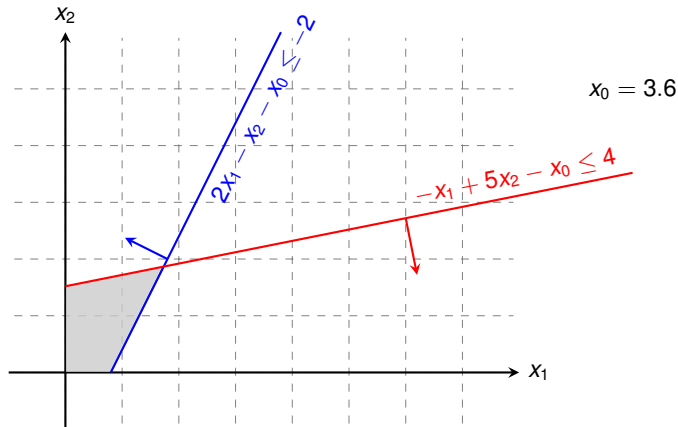
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

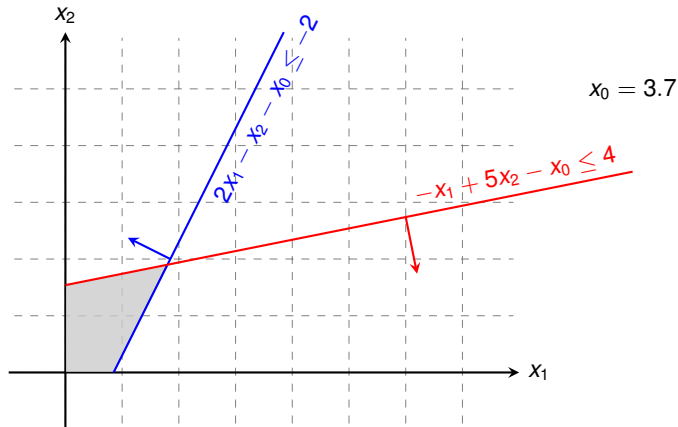
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

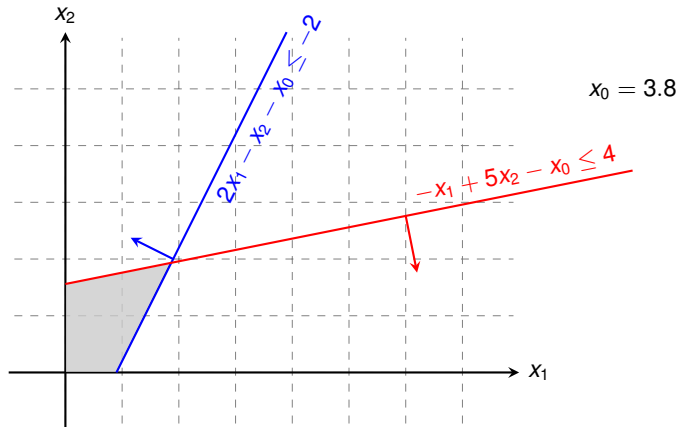
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

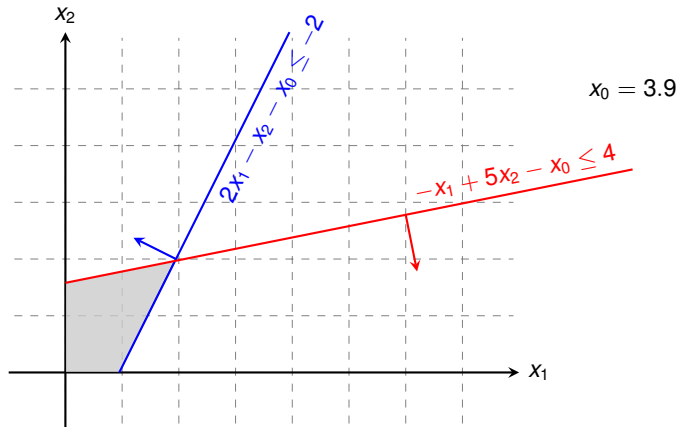
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

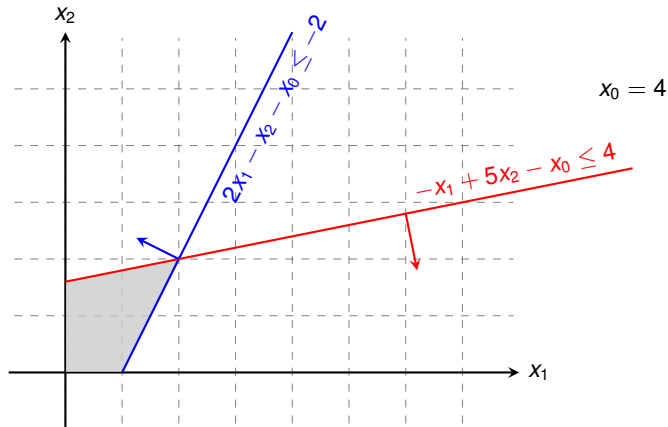
$$\begin{array}{rclclcl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ x_0, x_1, x_2 & & & & & \geq & 0 \end{array}$$



Geometric Illustration

maximise $-x_0$
subject to

$$\begin{array}{rcccccccl} 2x_1 & - & x_2 & - & x_0 & \leq & -2 \\ -x_1 & + & 5x_2 & - & x_0 & \leq & 4 \\ & & x_0, x_1, x_2 & & & \geq & 0 \end{array}$$



```
INITIALIZE-SIMPLEX( $A, b, c$ )
1  let  $k$  be the index of the minimum  $b_i$ 
2  if  $b_k \geq 0$  // is the initial basic solution feasible?
3      return ( $\{1, 2, \dots, n\}, \{n+1, n+2, \dots, n+m\}, A, b, c, 0$ )
4  form  $L_{\text{aux}}$  by adding  $-x_0$  to the left-hand side of each constraint
   and setting the objective function to  $-x_0$ 
5  let  $(N, B, A, b, c, v)$  be the resulting slack form for  $L_{\text{aux}}$ 
6   $l = n + k$ 
7  //  $L_{\text{aux}}$  has  $n + 1$  nonbasic variables and  $m$  basic variables.
8   $(N, B, A, b, c, v) = \text{PIVOT}(N, B, A, b, c, v, l, 0)$ 
9  // The basic solution is now feasible for  $L_{\text{aux}}$ .
10 iterate the while loop of lines 3–12 of SIMPLEX until an optimal solution
    to  $L_{\text{aux}}$  is found
11 if the optimal solution to  $L_{\text{aux}}$  sets  $\bar{x}_0$  to 0
12     if  $\bar{x}_0$  is basic
13         perform one (degenerate) pivot to make it nonbasic
14         from the final slack form of  $L_{\text{aux}}$ , remove  $x_0$  from the constraints and
           restore the original objective function of  $L$ , but replace each basic
           variable in this objective function by the right-hand side of its
           associated constraint
15     return the modified final slack form
16 else return “infeasible”
```

Test solution with $N = \{1, 2, \dots, n\}$, $B = \{n+1, n+2, \dots, n+m\}$, $\bar{x}_i = b_i$ for $i \in B$, $\bar{x}_i = 0$ otherwise.

INITIALIZE-SIMPLEX(A, b, c)

- 1 let k be the index of the minimum b_i
- 2 **if** $b_k \geq 0$ // is the initial basic solution feasible?
- 3 **return** ($\{1, 2, \dots, n\}, \{n+1, n+2, \dots, n+m\}, A, b, c, 0$)
- 4 form L_{aux} by adding $-x_0$ to the left-hand side of each constraint
and setting the objective function to $-x_0$
- 5 let (N, B, A, b, c, v) be the resulting slack form for L_{aux}
- 6 $l = n + k$
- 7 // L_{aux} has $n + 1$ nonbasic variables and m basic variables.
- 8 $(N, B, A, b, c, v) = \text{PIVOT}(N, B, A, b, c, v, l, 0)$
- 9 // The basic solution is now feasible for L_{aux} .
- 10 iterate the **while** loop of lines 3–12 of SIMPLEX until an optimal solution
to L_{aux} is found
- 11 **if** the optimal solution to L_{aux} sets $\bar{x}_0$ to 0
- 12 **if** $\bar{x}_0$ is basic
- 13 perform one (degenerate) pivot to make it nonbasic
- 14 from the final slack form of L_{aux} , remove x_0 from the constraints and
restore the original objective function of L , but replace each basic
variable in this objective function by the right-hand side of its
associated constraint
- 15 **return** the modified final slack form
- 16 **else return** “infeasible”

INITIALIZE-SIMPLEX(A, b, c)

- 1 let k be the index of the minimum b_i
- 2 **if** $b_k \geq 0$ // is the initial basic solution feasible?
- 3 **return** ($\{1, 2, \dots, n\}, \{n+1, n+2, \dots, n+m\}, A, b, c, 0$)
- 4 form L_{aux} by adding $-x_0$ to the left-hand side of each constraint
and setting the objective function to $-x_0$
- 5 let (N, B, A, b, c, v) be the resulting slack form for L_{aux}
- 6 $l = n + k$
- 7 // L_{aux} has $n + 1$ nonbasic variables and m basic variables.
- 8 $(N, B, A, b, c, v) = \text{PIVOT}(N, B, A, b, c, v, l, 0)$
- 9 // The basic solution is now feasible for L_{aux} .
- 10 iterate the **while** loop of lines 3–12 of SIMPLEX until an optimal solution
to L_{aux} is found
- 11 **if** the optimal solution to L_{aux} sets $\bar{x}_0$ to 0
- 12 **if** $\bar{x}_0$ is basic
- 13 perform one (degenerate) pivot to make it nonbasic
- 14 from the final slack form of L_{aux} , remove x_0 from the constraints and
restore the original objective function of L , but replace each basic
variable in this objective function by the right-hand side of its
associated constraint
- 15 **return** the modified final slack form
- 16 **else return** “infeasible”

Test solution with $N = \{1, 2, \dots, n\}$, $B = \{n+1, n+2, \dots, n+m\}$, $\bar{x}_i = b_i$ for $i \in B$, $\bar{x}_i = 0$ otherwise.

ℓ will be the leaving variable so that x_ℓ has the most negative value.

INITIALIZE-SIMPLEX(A, b, c)

```

1  let  $k$  be the index of the minimum  $b_i$ 
2  if  $b_k \geq 0$  // is the initial basic solution feasible?
3      return ( $\{1, 2, \dots, n\}, \{n+1, n+2, \dots, n+m\}, A, b, c, 0$ )
4  form  $L_{\text{aux}}$  by adding  $-x_0$  to the left-hand side of each constraint
   and setting the objective function to  $-x_0$ 
5  let  $(N, B, A, b, c, v)$  be the resulting slack form for  $L_{\text{aux}}$ 
6   $l = n + k$ 
7  //  $L_{\text{aux}}$  has  $n + 1$  nonbasic variables and  $m$  basic variables.
8   $(N, B, A, b, c, v) = \text{PIVOT}(N, B, A, b, c, v, l, 0)$ 
9  // The basic solution is now feasible for  $L_{\text{aux}}$ .
10 iterate the while loop of lines 3–12 of SIMPLEX until an optimal solution
    to  $L_{\text{aux}}$  is found
11 if the optimal solution to  $L_{\text{aux}}$  sets  $\bar{x}_0$  to 0
12     if  $\bar{x}_0$  is basic
13         perform one (degenerate) pivot to make it nonbasic
14         from the final slack form of  $L_{\text{aux}}$ , remove  $x_0$  from the constraints and
         restore the original objective function of  $L$ , but replace each basic
         variable in this objective function by the right-hand side of its
         associated constraint
15     return the modified final slack form
16 else return “infeasible”
    
```

Test solution with $N = \{1, 2, \dots, n\}$, $B = \{n+1, n+2, \dots, n+m\}$, $\bar{x}_i = b_i$ for $i \in B$, $\bar{x}_i = 0$ otherwise.

ℓ will be the leaving variable so that x_ℓ has the most negative value.

Pivot step with x_ℓ leaving and x_0 entering.

INITIALIZE-SIMPLEX(A, b, c)

- 1 let k be the index of the minimum b_i
- 2 **if** $b_k \geq 0$ // is the initial basic solution feasible?
- 3 **return** ($\{1, 2, \dots, n\}, \{n+1, n+2, \dots, n+m\}, A, b, c, 0$)
- 4 form L_{aux} by adding $-x_0$ to the left-hand side of each constraint
and setting the objective function to $-x_0$
- 5 let (N, B, A, b, c, v) be the resulting slack form for L_{aux}
- 6 $l = n + k$
- 7 // L_{aux} has $n + 1$ nonbasic variables and m basic variables.
- 8 $(N, B, A, b, c, v) = \text{PIVOT}(N, B, A, b, c, v, l, 0)$
- 9 // The basic solution is now feasible for L_{aux} .
- 10 **iterate the while loop** of lines 3–12 of SIMPLEX until an optimal solution
to L_{aux} is found
- 11 **if** the optimal solution to L_{aux} sets $\bar{x}_0$ to 0
- 12 **if** $\bar{x}_0$ is basic
- 13 perform one (degenerate) pivot to make it nonbasic
- 14 from the final slack form of L_{aux} , remove x_0 from the constraints and
restore the original objective function of L , but replace each basic
variable in this objective function by the right-hand side of its
associated constraint
- 15 **return** the modified final slack form
- 16 **else return** “infeasible”

Test solution with $N = \{1, 2, \dots, n\}$, $B = \{n+1, n+2, \dots, n+m\}$, $\bar{x}_i = b_i$ for $i \in B$, $\bar{x}_i = 0$ otherwise.

ℓ will be the leaving variable so that x_ℓ has the most negative value.

Pivot step with x_ℓ leaving and x_0 entering.

This pivot step does not change the value of any variable.

Example of INITIALIZE-SIMPLEX (1/3)

$$\begin{array}{llllll} \text{maximise} & 2x_1 & - & x_2 & & \\ \text{subject to} & & & & & \\ & 2x_1 & - & x_2 & \leq & 2 \\ & x_1 & - & 5x_2 & \leq & -4 \\ & & & x_1, x_2 & \geq & 0 \end{array}$$

Example of INITIALIZE-SIMPLEX (1/3)

maximise
subject to

$$2x_1 - x_2$$

$$2x_1 - x_2 \leq 2$$

$$x_1 - 5x_2 \leq -4$$

$$x_1, x_2 \geq 0$$

↓ Formulating the auxiliary linear program
(as basic solution would not be feasible!)

Example of INITIALIZE-SIMPLEX (1/3)

maximise
subject to

$$\begin{array}{rclcl} 2x_1 & - & x_2 & & \\ & & & & \\ 2x_1 & - & x_2 & \leq & 2 \\ x_1 & - & 5x_2 & \leq & -4 \\ & & x_1, x_2 & \geq & 0 \end{array}$$

↓ Formulating the auxiliary linear program
(as basic solution would not be feasible!)

maximise
subject to

$$\begin{array}{rclclcl} & & & - & x_0 & & \\ & & & & & & \\ 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ & & x_1, x_2, x_0 & & & \geq & 0 \end{array}$$

Example of INITIALIZE-SIMPLEX (1/3)

maximise
subject to

$$\begin{array}{rclcl} 2x_1 & - & x_2 & & \\ 2x_1 & - & x_2 & \leq & 2 \\ x_1 & - & 5x_2 & \leq & -4 \\ x_1, x_2 & & & \geq & 0 \end{array}$$

↓ Formulating the auxiliary linear program
(as basic solution would not be feasible!)

maximise
subject to

$$\begin{array}{rclclcl} & & & - & x_0 & \\ 2x_1 & - & x_2 & - & x_0 & \leq & 2 \\ x_1 & - & 5x_2 & - & x_0 & \leq & -4 \\ x_1, x_2, x_0 & & & & & \geq & 0 \end{array}$$

↓ Converting into slack form

Example of INITIALIZE-SIMPLEX (1/3)

$$\begin{array}{llll} \text{maximise} & 2x_1 & - & x_2 \\ \text{subject to} & 2x_1 & - & x_2 \leq 2 \\ & x_1 & - & 5x_2 \leq -4 \\ & x_1, x_2 & & \geq 0 \end{array}$$

↓
Formulating the auxiliary linear program
(as basic solution would not be feasible!)

$$\begin{array}{llllll} \text{maximise} & & & & - & x_0 \\ \text{subject to} & 2x_1 & - & x_2 & - & x_0 \leq 2 \\ & x_1 & - & 5x_2 & - & x_0 \leq -4 \\ & x_1, x_2, x_0 & & & & \geq 0 \end{array}$$

↓
Converting into slack form

$$\begin{array}{llllllll} Z & = & & & & & - & x_0 \\ x_3 & = & 2 & - & 2x_1 & + & x_2 & + & x_0 \\ x_4 & = & -4 & - & x_1 & + & 5x_2 & + & x_0 \end{array}$$

Example of INITIALIZE-SIMPLEX (1/3)

$$\begin{array}{llll}
 \text{maximise} & 2x_1 & - & x_2 \\
 \text{subject to} & & & \\
 & 2x_1 & - & x_2 \leq 2 \\
 & x_1 & - & 5x_2 \leq -4 \\
 & x_1, x_2 & & \geq 0
 \end{array}$$

Formulating the auxiliary linear program
(as basic solution would not be feasible!)

$$\begin{array}{llllll}
 \text{maximise} & & & & - & x_0 \\
 \text{subject to} & & & & & \\
 & 2x_1 & - & x_2 & - & x_0 \leq 2 \\
 & x_1 & - & 5x_2 & - & x_0 \leq -4 \\
 & x_1, x_2, x_0 & & & & \geq 0
 \end{array}$$

Basic solution
(0, 0, 0, 2, -4) not feasible!

Converting into slack form

$$\begin{array}{llllllll}
 Z & = & & & & & - & x_0 \\
 x_3 & = & 2 & - & 2x_1 & + & x_2 & + & x_0 \\
 x_4 & = & -4 & - & x_1 & + & 5x_2 & + & x_0
 \end{array}$$

Example of INITIALIZE-SIMPLEX (2/3)

$$\begin{array}{rclclclcl} Z & = & & & & - & x_0 \\ x_3 & = & 2 & - & 2x_1 & + & x_2 & + & x_0 \\ x_4 & = & -4 & - & x_1 & + & 5x_2 & + & x_0 \end{array}$$

Example of INITIALIZE-SIMPLEX (2/3)

$$\begin{array}{rclclclcl} Z & = & & & & - & x_0 \\ x_3 & = & 2 & - & 2x_1 & + & x_2 & + & x_0 \\ x_4 & = & -4 & - & x_1 & + & 5x_2 & + & x_0 \end{array}$$



Pivot with x_0 entering and x_4 leaving

Example of INITIALIZE-SIMPLEX (2/3)

$$\begin{array}{rclclclcl} Z & = & & & & & - & x_0 \\ x_3 & = & 2 & - & 2x_1 & + & x_2 & + & x_0 \\ x_4 & = & -4 & - & x_1 & + & 5x_2 & + & x_0 \end{array}$$



Pivot with x_0 entering and x_4 leaving

$$\begin{array}{rclclclcl} Z & = & -4 & - & x_1 & + & 5x_2 & - & x_4 \\ x_0 & = & 4 & + & x_1 & - & 5x_2 & + & x_4 \\ x_3 & = & 6 & - & x_1 & - & 4x_2 & + & x_4 \end{array}$$

Example of INITIALIZE-SIMPLEX (2/3)

$$\begin{array}{rclclclcl} Z & = & & & & & - & x_0 \\ x_3 & = & 2 & - & 2x_1 & + & x_2 & + & x_0 \\ x_4 & = & -4 & - & x_1 & + & 5x_2 & + & x_0 \end{array}$$



Pivot with x_0 entering and x_4 leaving

$$\begin{array}{rclclclcl} Z & = & -4 & - & x_1 & + & 5x_2 & - & x_4 \\ x_0 & = & 4 & + & x_1 & - & 5x_2 & + & x_4 \\ x_3 & = & 6 & - & x_1 & - & 4x_2 & + & x_4 \end{array}$$

Basic solution (4, 0, 0, 6, 0) is feasible!

Example of INITIALIZE-SIMPLEX (2/3)

$$\begin{array}{rcllclclcl} Z & = & & & & & - & x_0 \\ x_3 & = & 2 & - & 2x_1 & + & x_2 & + & x_0 \\ x_4 & = & -4 & - & x_1 & + & 5x_2 & + & x_0 \end{array}$$



Pivot with x_0 entering and x_4 leaving

$$\begin{array}{rcllclclcl} Z & = & -4 & - & x_1 & + & 5x_2 & - & x_4 \\ x_0 & = & 4 & + & x_1 & - & 5x_2 & + & x_4 \\ x_3 & = & 6 & - & x_1 & - & 4x_2 & + & x_4 \end{array}$$

Basic solution (4, 0, 0, 6, 0) is feasible!



Pivot with x_2 entering and x_0 leaving

Example of INITIALIZE-SIMPLEX (2/3)

$$\begin{array}{rclclclcl} Z & = & & & & & - & x_0 \\ x_3 & = & 2 & - & 2x_1 & + & x_2 & + & x_0 \\ x_4 & = & -4 & - & x_1 & + & 5x_2 & + & x_0 \end{array}$$



Pivot with x_0 entering and x_4 leaving

$$\begin{array}{rclclclcl} Z & = & -4 & - & x_1 & + & 5x_2 & - & x_4 \\ x_0 & = & 4 & + & x_1 & - & 5x_2 & + & x_4 \\ x_3 & = & 6 & - & x_1 & - & 4x_2 & + & x_4 \end{array}$$

Basic solution (4, 0, 0, 6, 0) is feasible!



Pivot with x_2 entering and x_0 leaving

$$\begin{array}{rclclclcl} Z & = & & - & x_0 & & & \\ x_2 & = & \frac{4}{5} & - & \frac{x_0}{5} & + & \frac{x_1}{5} & + & \frac{x_4}{5} \\ x_3 & = & \frac{14}{5} & + & \frac{4x_0}{5} & - & \frac{9x_1}{5} & + & \frac{x_4}{5} \end{array}$$

Example of INITIALIZE-SIMPLEX (2/3)

$$\begin{array}{rcllclclcl} Z & = & & & & & - & x_0 \\ x_3 & = & 2 & - & 2x_1 & + & x_2 & + & x_0 \\ x_4 & = & -4 & - & x_1 & + & 5x_2 & + & x_0 \end{array}$$

↓
Pivot with x_0 entering and x_4 leaving

$$\begin{array}{rcllclclcl} Z & = & -4 & - & x_1 & + & 5x_2 & - & x_4 \\ x_0 & = & 4 & + & x_1 & - & 5x_2 & + & x_4 \\ x_3 & = & 6 & - & x_1 & - & 4x_2 & + & x_4 \end{array}$$

Basic solution (4, 0, 0, 6, 0) is feasible!

↓
Pivot with x_2 entering and x_0 leaving

$$\begin{array}{rcllclclcl} Z & = & & - & x_0 \\ x_2 & = & \frac{4}{5} & - & \frac{x_0}{5} & + & \frac{x_1}{5} & + & \frac{x_4}{5} \\ x_3 & = & \frac{14}{5} & + & \frac{4x_0}{5} & - & \frac{9x_1}{5} & + & \frac{x_4}{5} \end{array}$$

Optimal solution has $x_0 = 0$, hence the initial problem was feasible!

Example of INITIALIZE-SIMPLEX (3/3)

$$\begin{array}{rclclclcl} Z & = & & - & x_0 & & & \\ x_2 & = & \frac{4}{5} & - & \frac{x_0}{5} & + & \frac{x_1}{5} & + \frac{x_4}{5} \\ x_3 & = & \frac{14}{5} & + & \frac{4x_0}{5} & - & \frac{9x_1}{5} & + \frac{x_4}{5} \end{array}$$

Example of INITIALIZE-SIMPLEX (3/3)

$$\begin{array}{rclclclcl} Z & = & & - & x_0 & & & \\ x_2 & = & \frac{4}{5} & - & \frac{x_0}{5} & + & \frac{x_1}{5} & + \frac{x_4}{5} \\ x_3 & = & \frac{14}{5} & + & \frac{4x_0}{5} & - & \frac{9x_1}{5} & + \frac{x_4}{5} \end{array}$$



Set $x_0 = 0$ and express objective function
by non-basic variables

Example of INITIALIZE-SIMPLEX (3/3)

$$\begin{array}{rclclcl} Z & = & & - & x_0 & & \\ x_2 & = & \frac{4}{5} & - & \frac{x_0}{5} & + & \frac{x_1}{5} + \frac{x_4}{5} \\ x_3 & = & \frac{14}{5} & + & \frac{4x_0}{5} & - & \frac{9x_1}{5} + \frac{x_4}{5} \end{array}$$

$$2x_1 - x_2 = 2x_1 - \left(\frac{4}{5} - \frac{x_0}{5} + \frac{x_1}{5} + \frac{x_4}{5} \right)$$

Set $x_0 = 0$ and express objective function by non-basic variables

$$\begin{array}{rclclcl} Z & = & -\frac{4}{5} & + & \frac{9x_1}{5} & - & \frac{x_4}{5} \\ x_2 & = & \frac{4}{5} & + & \frac{x_1}{5} & + & \frac{x_4}{5} \\ x_3 & = & \frac{14}{5} & - & \frac{9x_1}{5} & + & \frac{x_4}{5} \end{array}$$

Example of INITIALIZE-SIMPLEX (3/3)

$$\begin{array}{rclclcl} Z & = & & - & x_0 & & \\ x_2 & = & \frac{4}{5} & - & \frac{x_0}{5} & + & \frac{x_1}{5} + \frac{x_4}{5} \\ x_3 & = & \frac{14}{5} & + & \frac{4x_0}{5} & - & \frac{9x_1}{5} + \frac{x_4}{5} \end{array}$$

$$2x_1 - x_2 = 2x_1 - \left(\frac{4}{5} - \frac{x_0}{5} + \frac{x_1}{5} + \frac{x_4}{5}\right)$$

Set $x_0 = 0$ and express objective function by non-basic variables

$$\begin{array}{rclclcl} Z & = & -\frac{4}{5} & + & \frac{9x_1}{5} & - & \frac{x_4}{5} \\ x_2 & = & \frac{4}{5} & + & \frac{x_1}{5} & + & \frac{x_4}{5} \\ x_3 & = & \frac{14}{5} & - & \frac{9x_1}{5} & + & \frac{x_4}{5} \end{array}$$

Basic solution $(0, \frac{4}{5}, \frac{14}{5}, 0)$, which is feasible!

Example of INITIALIZE-SIMPLEX (3/3)

$$\begin{array}{rclclcl} Z & = & & - & x_0 & & \\ x_2 & = & \frac{4}{5} & - & \frac{x_0}{5} & + & \frac{x_1}{5} + \frac{x_4}{5} \\ x_3 & = & \frac{14}{5} & + & \frac{4x_0}{5} & - & \frac{9x_1}{5} + \frac{x_4}{5} \end{array}$$

$$2x_1 - x_2 = 2x_1 - \left(\frac{4}{5} - \frac{x_0}{5} + \frac{x_1}{5} + \frac{x_4}{5}\right)$$

Set $x_0 = 0$ and express objective function by non-basic variables

$$\begin{array}{rclclcl} Z & = & -\frac{4}{5} & + & \frac{9x_1}{5} & - & \frac{x_4}{5} \\ x_2 & = & \frac{4}{5} & + & \frac{x_1}{5} & + & \frac{x_4}{5} \\ x_3 & = & \frac{14}{5} & - & \frac{9x_1}{5} & + & \frac{x_4}{5} \end{array}$$

Basic solution $(0, \frac{4}{5}, \frac{14}{5}, 0)$, which is feasible!

Lemma 29.12

If a linear program L has no feasible solution, then INITIALIZE-SIMPLEX returns “infeasible”. Otherwise, it returns a valid slack form for which the basic solution is feasible.

Fundamental Theorem of Linear Programming

Theorem 29.13 (Fundamental Theorem of Linear Programming)

For any linear program L , given in standard form, either:

1. L is infeasible $\Rightarrow$ SIMPLEX returns “infeasible”.
2. L is unbounded $\Rightarrow$ SIMPLEX returns “unbounded”.
3. L has an optimal solution with a finite objective value
 $\Rightarrow$ SIMPLEX returns an optimal solution with a finite objective value.

Small Technicality: need to equip SIMPLEX with an “anti-cycling strategy” (see extra slides)

Fundamental Theorem of Linear Programming

Theorem 29.13 (Fundamental Theorem of Linear Programming)

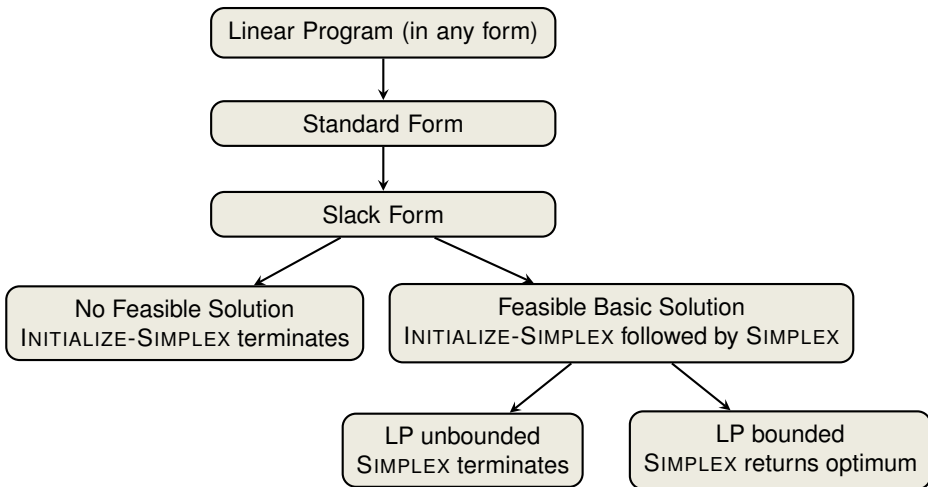
For any linear program L , given in standard form, either:

1. L is infeasible $\Rightarrow$ SIMPLEX returns “infeasible”.
2. L is unbounded $\Rightarrow$ SIMPLEX returns “unbounded”.
3. L has an optimal solution with a finite objective value
 $\Rightarrow$ SIMPLEX returns an optimal solution with a finite objective value.

Small Technicality: need to equip SIMPLEX with an “anti-cycling strategy” (see extra slides)

Proof requires the concept of **duality**, which is not covered in this course (for details see CLRS3, Chapter 29.4)

Workflow for Solving Linear Programs



Linear Programming and Simplex: Summary and Outlook

Linear Programming



Linear Programming and Simplex: Summary and Outlook

Linear Programming

- extremely versatile tool for modelling problems of all kinds

Linear Programming and Simplex: Summary and Outlook

Linear Programming

- extremely versatile tool for modelling problems of all kinds
- basis of [Integer Programming](#), to be discussed in later lectures

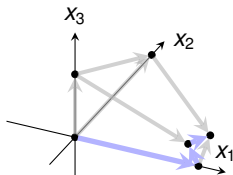
Linear Programming and Simplex: Summary and Outlook

Linear Programming

- extremely versatile tool for modelling problems of all kinds
- basis of [Integer Programming](#), to be discussed in later lectures

Simplex Algorithm

- **In practice:** usually terminates in polynomial time, i.e., $O(m + n)$



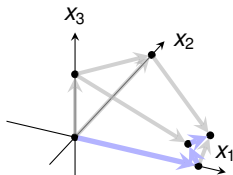
Linear Programming and Simplex: Summary and Outlook

Linear Programming

- extremely versatile tool for modelling problems of all kinds
- basis of **Integer Programming**, to be discussed in later lectures

Simplex Algorithm

- **In practice**: usually terminates in polynomial time, i.e., $O(m + n)$
- **In theory**: even with anti-cycling may need exponential time



Linear Programming and Simplex: Summary and Outlook

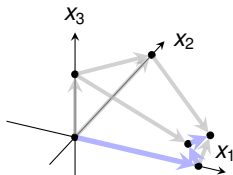
Linear Programming

- extremely versatile tool for modelling problems of all kinds
- basis of **Integer Programming**, to be discussed in later lectures

Simplex Algorithm

- **In practice**: usually terminates in polynomial time, i.e., $O(m + n)$
- **In theory**: even with anti-cycling may need exponential time

Research Problem: Is there a pivoting rule which makes SIMPLEX a polynomial-time algorithm?



Linear Programming and Simplex: Summary and Outlook

Linear Programming

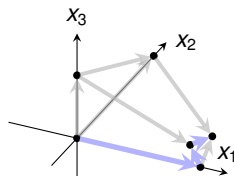
- extremely versatile tool for modelling problems of all kinds
- basis of **Integer Programming**, to be discussed in later lectures

Simplex Algorithm

- **In practice**: usually terminates in polynomial time, i.e., $O(m + n)$
- **In theory**: even with anti-cycling may need exponential time

Research Problem: Is there a pivoting rule which makes SIMPLEX a polynomial-time algorithm?

Polynomial-Time Algorithms



Linear Programming and Simplex: Summary and Outlook

Linear Programming

- extremely versatile tool for modelling problems of all kinds
- basis of **Integer Programming**, to be discussed in later lectures

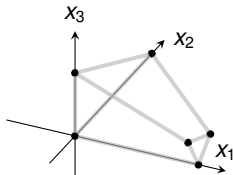
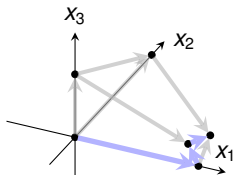
Simplex Algorithm

- **In practice**: usually terminates in polynomial time, i.e., $O(m + n)$
- **In theory**: even with anti-cycling may need exponential time

Research Problem: Is there a pivoting rule which makes SIMPLEX a polynomial-time algorithm?

Polynomial-Time Algorithms

- **Interior-Point Methods**: traverses the interior of the feasible set of solutions (not just vertices!)



Linear Programming and Simplex: Summary and Outlook

Linear Programming

- extremely versatile tool for modelling problems of all kinds
- basis of **Integer Programming**, to be discussed in later lectures

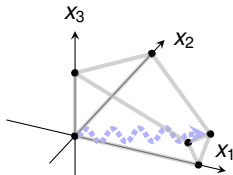
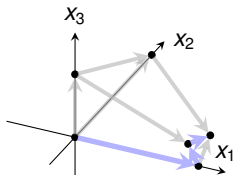
Simplex Algorithm

- **In practice**: usually terminates in polynomial time, i.e., $O(m + n)$
- **In theory**: even with anti-cycling may need exponential time

Research Problem: Is there a pivoting rule which makes SIMPLEX a polynomial-time algorithm?

Polynomial-Time Algorithms

- **Interior-Point Methods**: traverses the interior of the feasible set of solutions (not just vertices!)



1.2 Famous Failures and the Need for Alternatives

For many problems a bit beyond the scope of an undergraduate course, the downside of worst-case analysis rears its ugly head. This section reviews four famous examples in which worst-case analysis gives misleading or useless advice about how to solve a problem. These examples motivate the alternatives to worst-case analysis that are surveyed in Section 1.4 and described in detail in later chapters of the book.

1.2.1 The Simplex Method for Linear Programming

Perhaps the most famous failure of worst-case analysis concerns linear programming, the problem of optimizing a linear function subject to linear constraints (Figure 1.1). Dantzig proposed in the 1940s an algorithm for solving linear programs called the *simplex method*. The simplex method solves linear programs using greedy local

Source: "Beyond the Worst-Case Analysis of Algorithms" by Tim Roughgarden, 2020

Simplex Algorithm by Example

Details of the Simplex Algorithm

Finding an Initial Solution

Appendix: Cycling and Termination (non-examinable)

Termination

Degeneracy: One iteration of SIMPLEX leaves the objective value unchanged.

Termination

Degeneracy: One iteration of SIMPLEX leaves the objective value unchanged.

$$Z = \quad \quad \quad x_1 + x_2 + x_3$$

$$x_4 = 8 - x_1 - x_2$$

$$x_5 = \quad \quad \quad x_2 - x_3$$

Termination

Degeneracy: One iteration of SIMPLEX leaves the objective value unchanged.

$$Z = \quad \quad \quad x_1 \quad + \quad x_2 \quad + \quad x_3$$

$$x_4 = 8 \quad - \quad x_1 \quad - \quad x_2$$

$$x_5 = \quad \quad \quad \quad \quad x_2 \quad - \quad x_3$$

↓ Pivot with x_1 entering and x_4 leaving

Termination

Degeneracy: One iteration of SIMPLEX leaves the objective value unchanged.

$$Z = \quad \quad \quad x_1 + x_2 + x_3$$

$$x_4 = 8 - x_1 - x_2$$

$$x_5 = \quad \quad \quad x_2 - x_3$$

↓ Pivot with x_1 entering and x_4 leaving

$$Z = 8 \quad \quad \quad + x_3 - x_4$$

$$x_1 = 8 - x_2 \quad \quad \quad - x_4$$

$$x_5 = \quad \quad \quad x_2 - x_3$$

Termination

Degeneracy: One iteration of SIMPLEX leaves the objective value unchanged.

$$Z = \quad \quad \quad x_1 + x_2 + x_3$$

$$x_4 = 8 - x_1 - x_2$$

$$x_5 = \quad \quad \quad x_2 - x_3$$

↓ Pivot with x_1 entering and x_4 leaving

$$Z = 8 \quad \quad \quad + x_3 - x_4$$

$$x_1 = 8 - x_2 \quad \quad \quad - x_4$$

$$x_5 = \quad \quad \quad x_2 - x_3$$

↓ Pivot with x_3 entering and x_5 leaving

Termination

Degeneracy: One iteration of SIMPLEX leaves the objective value unchanged.

$$Z = \quad \quad \quad x_1 + x_2 + x_3$$

$$x_4 = 8 - x_1 - x_2$$

$$x_5 = \quad \quad \quad x_2 - x_3$$

↓ Pivot with x_1 entering and x_4 leaving

$$Z = 8 \quad \quad \quad + x_3 - x_4$$

$$x_1 = 8 - x_2 \quad \quad \quad - x_4$$

$$x_5 = \quad \quad \quad x_2 - x_3$$

↓ Pivot with x_3 entering and x_5 leaving

$$Z = 8 + x_2 - x_4 - x_5$$

$$x_1 = 8 - x_2 - x_4$$

$$x_3 = \quad \quad \quad x_2 \quad \quad \quad - x_5$$

Termination

Degeneracy: One iteration of SIMPLEX leaves the objective value unchanged.

$$\begin{array}{rclclcl} Z & = & & x_1 & + & x_2 & + & x_3 \\ x_4 & = & 8 & - & x_1 & - & x_2 & \\ x_5 & = & & & & x_2 & - & x_3 \end{array}$$

Pivot with x_1 entering and x_4 leaving
↓

$$\begin{array}{rclclcl} Z & = & 8 & & + & x_3 & - & x_4 \\ x_1 & = & 8 & - & x_2 & & - & x_4 \\ x_5 & = & & x_2 & - & x_3 & & \end{array}$$

Cycling: If additionally slack form at two iterations are identical, SIMPLEX fails to terminate!

Pivot with x_3 entering and x_5 leaving
↓

$$\begin{array}{rclclcl} Z & = & 8 & + & x_2 & - & x_4 & - & x_5 \\ x_1 & = & 8 & - & x_2 & - & x_4 & \\ x_3 & = & & x_2 & & & - & x_5 \end{array}$$



Exercise: Execute one more step of the Simplex Algorithm on the tableau from the previous slide.

Cycling: SIMPLEX may fail to terminate.

Termination and Running Time

It is theoretically possible, but very rare in practice.

Cycling: SIMPLEX may fail to terminate.

Termination and Running Time

It is theoretically possible, but very rare in practice.

Cycling: SIMPLEX may fail to terminate.

Anti-Cycling Strategies



Termination and Running Time

It is theoretically possible, but very rare in practice.

Cycling: SIMPLEX may fail to terminate.

Anti-Cycling Strategies

1. **Bland's rule:** Choose entering variable with smallest index

Termination and Running Time

It is theoretically possible, but very rare in practice.

Cycling: SIMPLEX may fail to terminate.

Anti-Cycling Strategies

1. **Bland's rule:** Choose entering variable with smallest index
2. **Random rule:** Choose entering variable uniformly at random

Termination and Running Time

It is theoretically possible, but very rare in practice.

Cycling: SIMPLEX may fail to terminate.

— Anti-Cycling Strategies —

1. **Bland's rule:** Choose entering variable with smallest index
2. **Random rule:** Choose entering variable uniformly at random
3. **Perturbation:** Perturb the input slightly so that it is impossible to have two solutions with the same objective value

Termination and Running Time

It is theoretically possible, but very rare in practice.

Cycling: SIMPLEX may fail to terminate.

Anti-Cycling Strategies

1. **Bland's rule:** Choose entering variable with smallest index
2. **Random rule:** Choose entering variable uniformly at random
3. **Perturbation:** Perturb the input slightly so that it is impossible to have two solutions with the same objective value

Replace each b_i by $\hat{b}_i = b_i + \epsilon_i$, where $\epsilon_i \gg \epsilon_{i+1}$ are all small.

Termination and Running Time

It is theoretically possible, but very rare in practice.

Cycling: SIMPLEX may fail to terminate.

Anti-Cycling Strategies

1. **Bland's rule:** Choose entering variable with smallest index
2. **Random rule:** Choose entering variable uniformly at random
3. **Perturbation:** Perturb the input slightly so that it is impossible to have two solutions with the same objective value

Replace each b_i by $\hat{b}_i = b_i + \epsilon_i$, where $\epsilon_i \gg \epsilon_{i+1}$ are all small.

Lemma 29.7

Assuming INITIALIZE-SIMPLEX returns a slack form for which the basic solution is feasible, SIMPLEX either reports that the program is unbounded or returns a feasible solution in at most $\binom{n+m}{m}$ iterations.

Termination and Running Time

It is theoretically possible, but very rare in practice.

Cycling: SIMPLEX may fail to terminate.

Anti-Cycling Strategies

1. **Bland's rule:** Choose entering variable with smallest index
2. **Random rule:** Choose entering variable uniformly at random
3. **Perturbation:** Perturb the input slightly so that it is impossible to have two solutions with the same objective value

Replace each b_i by $\hat{b}_i = b_i + \epsilon_i$, where $\epsilon_i \gg \epsilon_{i+1}$ are all small.

Lemma 29.7

Assuming INITIALIZE-SIMPLEX returns a slack form for which the basic solution is feasible, SIMPLEX either reports that the program is unbounded or returns a feasible solution in at most $\binom{n+m}{m}$ iterations.

Every set B of basic variables uniquely determines a slack form, and there are at most $\binom{n+m}{m}$ unique slack forms.