III. Linear Programming

Thomas Sauerwald

Easter 2018

UNIVERSITY OF CAMBRIDGE

Outline

Introduction

Linear Programming (informal definition)

- maximize or minimize an objective, given limited resources and competing constraint
- constraints are specified as (in)equalities

Introduction

Linear Programming (informal definition)

- maximize or minimize an objective, given limited resources and competing constraint
- constraints are specified as (in)equalities

Example: Political Advertising

Introduction

Linear Programming (informal definition)

- maximize or minimize an objective, given limited resources and competing constraint
- constraints are specified as (in)equalities

Example: Political Advertising

- Imagine you are a politician trying to win an election

Introduction

Linear Programming (informal definition)

- maximize or minimize an objective, given limited resources and competing constraint
- constraints are specified as (in)equalities

Example: Political Advertising

- Imagine you are a politician trying to win an election
- Your district has three different types of areas: Urban, suburban and rural, each with, respectively, 100,000, 200,000 and 50,000 registered voters

Introduction

Linear Programming (informal definition)

- maximize or minimize an objective, given limited resources and competing constraint
- constraints are specified as (in)equalities

Example: Political Advertising

- Imagine you are a politician trying to win an election
- Your district has three different types of areas: Urban, suburban and rural, each with, respectively, 100,000, 200,000 and 50,000 registered voters
- Aim: at least half of the registered voters in each of the three regions should vote for you

Introduction

Linear Programming (informal definition)

- maximize or minimize an objective, given limited resources and competing constraint
- constraints are specified as (in)equalities

Example: Political Advertising

- Imagine you are a politician trying to win an election
- Your district has three different types of areas: Urban, suburban and rural, each with, respectively, 100,000, 200,000 and 50,000 registered voters
- Aim: at least half of the registered voters in each of the three regions should vote for you
- Possible Actions: Advertise on one of the primary issues which are (i) building more roads, (ii) gun control, (iii) farm subsidies and (iv) a gasoline tax dedicated to improve public transit.

Political Advertising Continued

policy	urban	suburban	rural
build roads	−2	5	3
gun control	8	2	−5
farm subsidies	0	0	10
gasoline tax	10	0	−2

The effects of policies on voters. Each entry describes the number of thousands of voters who could be won (lost) over by spending $1,000 on advertising support of a policy on a particular issue.

Political Advertising Continued

policy	urban	suburban	rural
build roads	-2	5	3
gun control	8	2	-5
farm subsidies	0	0	10
gasoline tax	10	0	-2

The effects of policies on voters. Each entry describes the number of thousands of voters who could be won (lost) over by spending $1,000 on advertising support of a policy on a particular issue.

- Possible Solution:
  - $20,000 on advertising to building roads
  - $0 on advertising to gun control
  - $4,000 on advertising to farm subsidies
  - $9,000 on advertising to a gasoline tax

Political Advertising Continued

policy	urban	suburban	rural
build roads	-2	5	3
gun control	8	2	-5
farm subsidies	0	0	10
gasoline tax	10	0	-2

The effects of policies on voters. Each entry describes the number of thousands of voters who could be won (lost) over by spending $1,000 on advertising support of a policy on a particular issue.

- Possible Solution:
  - $20,000 on advertising to building roads
  - $0 on advertising to gun control
  - $4,000 on advertising to farm subsidies
  - $9,000 on advertising to a gasoline tax
- Total cost: $33,000

Political Advertising Continued

policy	urban	suburban	rural
build roads	-2	5	3
gun control	8	2	-5
farm subsidies	0	0	10
gasoline tax	10	0	-2

The effects of policies on voters. Each entry describes the number of thousands of voters who could be won (lost) over by spending $1,000 on advertising support of a policy on a particular issue.

- Possible Solution:
  - $20,000 on advertising to building roads
  - $0 on advertising to gun control
  - $4,000 on advertising to farm subsidies
  - $9,000 on advertising to a gasoline tax
- Total cost: $33,000

What is the best possible strategy?

Towards a Linear Program

policy	urban	suburban	rural
build roads	-2	5	3
gun control	8	2	-5
farm subsidies	0	0	10
gasoline tax	10	0	-2

The effects of policies on voters. Each entry describes the number of thousands of voters who could be won (lost) over by spending $1,000 on advertising support of a policy on a particular issue.

Towards a Linear Program

policy	urban	suburban	rural
build roads	-2	5	3
gun control	8	2	-5
farm subsidies	0	0	10
gasoline tax	10	0	-2

The effects of policies on voters. Each entry describes the number of thousands of voters who could be won (lost) over by spending $1,000 on advertising support of a policy on a particular issue.

- x_1 = number of thousands of dollars spent on advertising on building roads
- x_2 = number of thousands of dollars spent on advertising on gun control
- x_3 = number of thousands of dollars spent on advertising on farm subsidies
- x_4 = number of thousands of dollars spent on advertising on gasoline tax

Towards a Linear Program

policy	urban	suburban	rural
build roads	-2	5	3
gun control	8	2	-5
farm subsidies	0	0	10
gasoline tax	10	0	-2

The effects of policies on voters. Each entry describes the number of thousands of voters who could be won (lost) over by spending $1,000 on advertising support of a policy on a particular issue.

- x_1 = number of thousands of dollars spent on advertising on building roads
- x_2 = number of thousands of dollars spent on advertising on gun control
- x_3 = number of thousands of dollars spent on advertising on farm subsidies
- x_4 = number of thousands of dollars spent on advertising on gasoline tax

Constraints:

Towards a Linear Program

policy	urban	suburban	rural
build roads	-2	5	3
gun control	8	2	-5
farm subsidies	0	0	10
gasoline tax	10	0	-2

The effects of policies on voters. Each entry describes the number of thousands of voters who could be won (lost) over by spending $1,000 on advertising support of a policy on a particular issue.

- x_1 = number of thousands of dollars spent on advertising on building roads
- x_2 = number of thousands of dollars spent on advertising on gun control
- x_3 = number of thousands of dollars spent on advertising on farm subsidies
- x_4 = number of thousands of dollars spent on advertising on gasoline tax

Constraints:

- $-2x_1 + 8x_2 + 0x_3 + 10x_4 \geq 50$

Towards a Linear Program

policy	urban	suburban	rural
build roads	-2	5	3
gun control	8	2	-5
farm subsidies	0	0	10
gasoline tax	10	0	-2

The effects of policies on voters. Each entry describes the number of thousands of voters who could be won (lost) over by spending $1,000 on advertising support of a policy on a particular issue.

- x_1 = number of thousands of dollars spent on advertising on building roads
- x_2 = number of thousands of dollars spent on advertising on gun control
- x_3 = number of thousands of dollars spent on advertising on farm subsidies
- x_4 = number of thousands of dollars spent on advertising on gasoline tax

Constraints:

- $-2x_1 + 8x_2 + 0x_3 + 10x_4 \geq 50$
- $5x_1 + 2x_2 + 0x_3 + 0x_4 \geq 100$

Towards a Linear Program

policy	urban	suburban	rural
build roads	-2	5	3
gun control	8	2	-5
farm subsidies	0	0	10
gasoline tax	10	0	-2

The effects of policies on voters. Each entry describes the number of thousands of voters who could be won (lost) over by spending $1,000 on advertising support of a policy on a particular issue.

- x_1 = number of thousands of dollars spent on advertising on building roads
- x_2 = number of thousands of dollars spent on advertising on gun control
- x_3 = number of thousands of dollars spent on advertising on farm subsidies
- x_4 = number of thousands of dollars spent on advertising on gasoline tax

Constraints:

- $-2x_1 + 8x_2 + 0x_3 + 10x_4 \geq 50$
- $5x_1 + 2x_2 + 0x_3 + 0x_4 \geq 100$
- $3x_1 - 5x_2 + 10x_3 - 2x_4 \geq 25$

Towards a Linear Program

policy	urban	suburban	rural
build roads	-2	5	3
gun control	8	2	-5
farm subsidies	0	0	10
gasoline tax	10	0	-2

The effects of policies on voters. Each entry describes the number of thousands of voters who could be won (lost) over by spending $1,000 on advertising support of a policy on a particular issue.

- x_1 = number of thousands of dollars spent on advertising on building roads
- x_2 = number of thousands of dollars spent on advertising on gun control
- x_3 = number of thousands of dollars spent on advertising on farm subsidies
- x_4 = number of thousands of dollars spent on advertising on gasoline tax

Constraints:

- $-2x_1 + 8x_2 + 0x_3 + 10x_4 \geq 50$
- $5x_1 + 2x_2 + 0x_3 + 0x_4 \geq 100$
- $3x_1 - 5x_2 + 10x_3 - 2x_4 \geq 25$

Objective: Minimize $x_1 + x_2 + x_3 + x_4$

The Linear Program

Linear Program for the Advertising Problem

$$
\begin{array}{llrcrcrcrcr}
\text{minimize} & & x_1 & + & x_2 & + & x_3 & + & x_4 & & \\
\text{subject to} & & & & & & & & & & \\
& & -2x_1 & + & 8x_2 & + & 0x_3 & + & 10x_4 & \geq & 50 \\
& & 5x_1 & + & 2x_2 & + & 0x_3 & + & 0x_4 & \geq & 100 \\
& & 3x_1 & - & 5x_2 & + & 10x_3 & - & 2x_4 & \geq & 25 \\
& & & & x_1, x_2, x_3, x_4 & & & & & \geq & 0
\end{array}
$$

The Linear Program

Linear Program for the Advertising Problem

$$\begin{array}{lrcrcrcrcr}
\text{minimize} & x_1 & + & x_2 & + & x_3 & + & x_4 & & \\
\text{subject to} & & & & & & & & & \\
 & -2x_1 & + & 8x_2 & + & 0x_3 & + & 10x_4 & \geq & 50 \\
 & 5x_1 & + & 2x_2 & + & 0x_3 & + & 0x_4 & \geq & 100 \\
 & 3x_1 & - & 5x_2 & + & 10x_3 & - & 2x_4 & \geq & 25 \\
 & & & x_1, x_2, x_3, x_4 & & & & & \geq & 0
\end{array}$$

The solution of this linear program yields the optimal advertising strategy.

The Linear Program

Linear Program for the Advertising Problem

$$\begin{array}{lrcrcrcrcr}
\text{minimize} & x_1 & + & x_2 & + & x_3 & + & x_4 & & \\
\text{subject to} & & & & & & & & & \\
& -2x_1 & + & 8x_2 & + & 0x_3 & + & 10x_4 & \geq & 50 \\
& 5x_1 & + & 2x_2 & + & 0x_3 & + & 0x_4 & \geq & 100 \\
& 3x_1 & - & 5x_2 & + & 10x_3 & - & 2x_4 & \geq & 25 \\
& & & x_1, x_2, x_3, x_4 & & & & & \geq & 0
\end{array}$$

The solution of this linear program yields the optimal advertising strategy.

Formal Definition of Linear Program

The Linear Program

Linear Program for the Advertising Problem

$$
\begin{array}{lrcrcrcrcr}
\text{minimize} & x_1 & + & x_2 & + & x_3 & + & x_4 & & \\
\text{subject to} & & & & & & & & & \\
& -2x_1 & + & 8x_2 & + & 0x_3 & + & 10x_4 & \geq & 50 \\
& 5x_1 & + & 2x_2 & + & 0x_3 & + & 0x_4 & \geq & 100 \\
& 3x_1 & - & 5x_2 & + & 10x_3 & - & 2x_4 & \geq & 25 \\
& & x_1, x_2, x_3, x_4 & & & & & & \geq & 0
\end{array}
$$

The solution of this linear program yields the optimal advertising strategy.

Formal Definition of Linear Program

- Given $a_1, a_2, \ldots, a_n$ and a set of variables $x_1, x_2, \ldots, x_n$, a linear function f is defined by

$$f(x_1, x_2, \ldots, x_n) = a_1 x_1 + a_2 x_2 + \cdots + a_n x_n.$$

The Linear Program

Linear Program for the Advertising Problem

$$
\begin{array}{lrcrcrcrcr}
\text{minimize} & x_1 & + & x_2 & + & x_3 & + & x_4 & & \\
\text{subject to} & & & & & & & & & \\
& -2x_1 & + & 8x_2 & + & 0x_3 & + & 10x_4 & \geq & 50 \\
& 5x_1 & + & 2x_2 & + & 0x_3 & + & 0x_4 & \geq & 100 \\
& 3x_1 & - & 5x_2 & + & 10x_3 & - & 2x_4 & \geq & 25 \\
& & & x_1, x_2, x_3, x_4 & & & & & \geq & 0
\end{array}
$$

The solution of this linear program yields the optimal advertising strategy.

Formal Definition of Linear Program

- Given $a_1, a_2, \ldots, a_n$ and a set of variables $x_1, x_2, \ldots, x_n$, a linear function f is defined by

$$f(x_1, x_2, \ldots, x_n) = a_1x_1 + a_2x_2 + \cdots + a_nx_n.$$

- Linear Equality: $f(x_1, x_2, \ldots, x_n) = b$
- Linear Inequality: $f(x_1, x_2, \ldots, x_n) \gtreqless b$

The Linear Program

Linear Program for the Advertising Problem

$$\begin{array}{lrcrcrcrcr}
\text{minimize} & x_1 & + & x_2 & + & x_3 & + & x_4 & & \\
\text{subject to} & & & & & & & & & \\
& -2x_1 & + & 8x_2 & + & 0x_3 & + & 10x_4 & \geq & 50 \\
& 5x_1 & + & 2x_2 & + & 0x_3 & + & 0x_4 & \geq & 100 \\
& 3x_1 & - & 5x_2 & + & 10x_3 & - & 2x_4 & \geq & 25 \\
& & & x_1, x_2, x_3, x_4 & & & & & \geq & 0
\end{array}$$

The solution of this linear program yields the optimal advertising strategy.

Formal Definition of Linear Program

- Given $a_1, a_2, \ldots, a_n$ and a set of variables $x_1, x_2, \ldots, x_n$, a linear function f is defined by

$$f(x_1, x_2, \ldots, x_n) = a_1 x_1 + a_2 x_2 + \cdots + a_n x_n.$$

- Linear Equality: $f(x_1, x_2, \ldots, x_n) = b$
- Linear Inequality: $f(x_1, x_2, \ldots, x_n) \substack{\geq \\ \leq} b$

The Linear Program

Linear Program for the Advertising Problem

$$
\begin{array}{lrcrcrcrcr}
\text{minimize} & x_1 & + & x_2 & + & x_3 & + & x_4 & & \\
\text{subject to} & & & & & & & & & \\
& -2x_1 & + & 8x_2 & + & 0x_3 & + & 10x_4 & \geq & 50 \\
& 5x_1 & + & 2x_2 & + & 0x_3 & + & 0x_4 & \geq & 100 \\
& 3x_1 & - & 5x_2 & + & 10x_3 & - & 2x_4 & \geq & 25 \\
& & & x_1, x_2, x_3, x_4 & & & & & \geq & 0
\end{array}
$$

The solution of this linear program yields the optimal advertising strategy.

Formal Definition of Linear Program

- Given $a_1, a_2, \ldots, a_n$ and a set of variables $x_1, x_2, \ldots, x_n$, a linear function f is defined by

$$f(x_1, x_2, \ldots, x_n) = a_1 x_1 + a_2 x_2 + \cdots + a_n x_n.$$

- Linear Equality: $f(x_1, x_2, \ldots, x_n) = b$
- Linear Inequality: $f(x_1, x_2, \ldots, x_n) \gtreqless b$

Linear Constraints

The Linear Program

Linear Program for the Advertising Problem

$$
\begin{array}{lrcrcrcrcr}
\text{minimize} & x_1 & + & x_2 & + & x_3 & + & x_4 & & \\
\text{subject to} & & & & & & & & & \\
& -2x_1 & + & 8x_2 & + & 0x_3 & + & 10x_4 & \geq & 50 \\
& 5x_1 & + & 2x_2 & + & 0x_3 & + & 0x_4 & \geq & 100 \\
& 3x_1 & - & 5x_2 & + & 10x_3 & - & 2x_4 & \geq & 25 \\
& & & x_1, x_2, x_3, x_4 & & & & & \geq & 0
\end{array}
$$

The solution of this linear program yields the optimal advertising strategy.

Formal Definition of Linear Program

- Given $a_1, a_2, \ldots, a_n$ and a set of variables $x_1, x_2, \ldots, x_n$, a linear function f is defined by

$$f(x_1, x_2, \ldots, x_n) = a_1 x_1 + a_2 x_2 + \cdots + a_n x_n.$$

- Linear Equality: $f(x_1, x_2, \ldots, x_n) = b$
- Linear Inequality: $f(x_1, x_2, \ldots, x_n) \gtreqless b$

Linear Constraints

- Linear-Progamming Problem: either minimize or maximize a linear function subject to a set of linear constraints

A Small(er) Example

$$
\begin{array}{lrcrcr}
\text{maximize} & x_1 & + & x_2 & & \\
\text{subject to} & & & & & \\
 & 4x_1 & - & x_2 & \leq & 8 \\
 & 2x_1 & + & x_2 & \leq & 10 \\
 & 5x_1 & - & 2x_2 & \geq & -2 \\
 & x_1, x_2 & & & \geq & 0
\end{array}
$$

A Small(er) Example

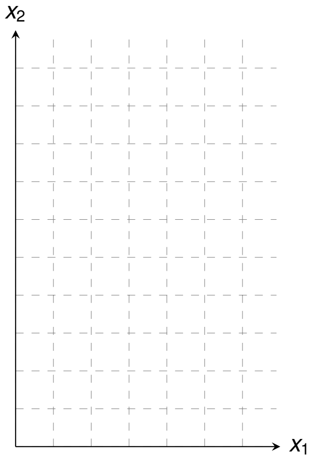
$$\begin{array}{lrcrcr}
\text{maximize} & x_1 & + & x_2 & & \\
\text{subject to} & & & & & \\
& 4x_1 & - & x_2 & \leq & 8 \\
& 2x_1 & + & x_2 & \leq & 10 \\
& 5x_1 & - & 2x_2 & \geq & -2 \\
& x_1, x_2 & & & \geq & 0
\end{array}$$

Any setting of x_1 and x_2 satisfying all constraints is a feasible solution

A Small(er) Example

$$\begin{array}{lrcrcr} \text{maximize} & x_1 & + & x_2 & & \\ \text{subject to} & & & & & \\ & 4x_1 & - & x_2 & \leq & 8 \\ & 2x_1 & + & x_2 & \leq & 10 \\ & 5x_1 & - & 2x_2 & \geq & -2 \\ & x_1, x_2 & & & \geq & 0 \end{array}$$

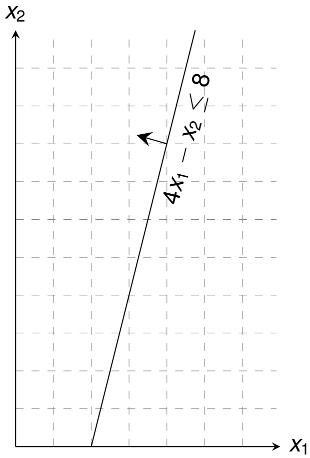
Any setting of x_1 and x_2 satisfying all constraints is a feasible solution



A Small(er) Example

$$\begin{array}{lrcrcr} \text{maximize} & x_1 & + & x_2 & & \\ \text{subject to} & & & & & \\ & 4x_1 & - & x_2 & \leq & 8 \\ & 2x_1 & + & x_2 & \leq & 10 \\ & 5x_1 & - & 2x_2 & \geq & -2 \\ & x_1, x_2 & & & \geq & 0 \end{array}$$

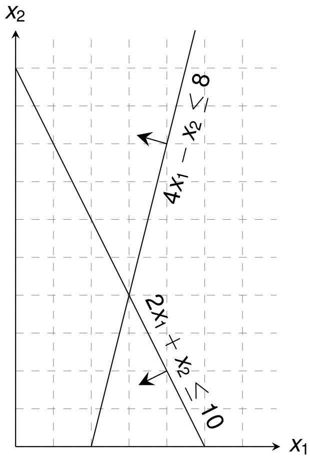
Any setting of x_1 and x_2 satisfying all constraints is a feasible solution



A Small(er) Example

$$\begin{array}{lrcrcr}
\text{maximize} & x_1 & + & x_2 & & \\
\text{subject to} & & & & & \\
& 4x_1 & - & x_2 & \leq & 8 \\
& 2x_1 & + & x_2 & \leq & 10 \\
& 5x_1 & - & 2x_2 & \geq & -2 \\
& x_1, x_2 & & & \geq & 0
\end{array}$$

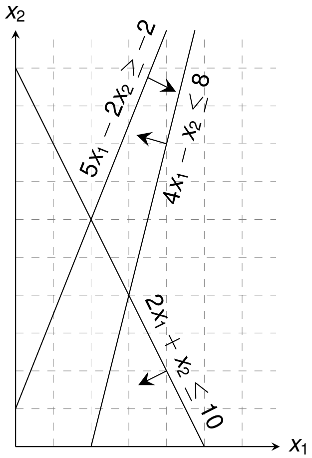
Any setting of x_1 and x_2 satisfying all constraints is a feasible solution



A Small(er) Example

$$
\begin{array}{lrcrcr}
\text{maximize} & x_1 & + & x_2 & & \\
\text{subject to} & & & & & \\
& 4x_1 & - & x_2 & \leq & 8 \\
& 2x_1 & + & x_2 & \leq & 10 \\
& 5x_1 & - & 2x_2 & \geq & -2 \\
& x_1, x_2 & & & \geq & 0
\end{array}
$$

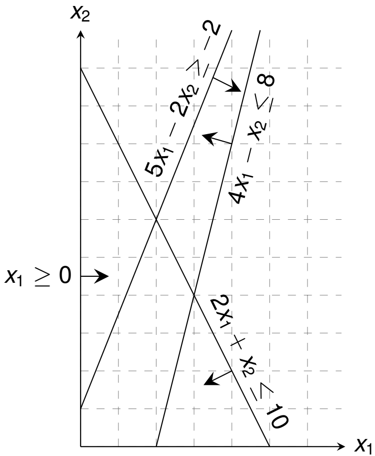
Any setting of x_1 and x_2 satisfying all constraints is a feasible solution



A Small(er) Example

$$
\begin{array}{llllll}
\text{maximize} & x_1 & + & x_2 & & \\
\text{subject to} & & & & & \\
 & 4x_1 & - & x_2 & \leq & 8 \\
 & 2x_1 & + & x_2 & \leq & 10 \\
 & 5x_1 & - & 2x_2 & \geq & -2 \\
 & x_1, x_2 & & & \geq & 0
\end{array}
$$

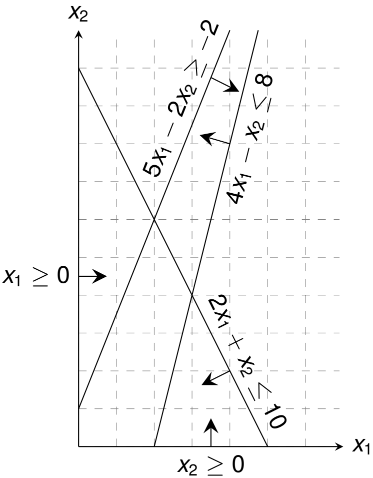
Any setting of x_1 and x_2 satisfying all constraints is a feasible solution



A Small(er) Example

$$\begin{array}{lrcrcr} \text{maximize} & x_1 & + & x_2 & & \\ \text{subject to} & & & & & \\ & 4x_1 & - & x_2 & \leq & 8 \\ & 2x_1 & + & x_2 & \leq & 10 \\ & 5x_1 & - & 2x_2 & \geq & -2 \\ & x_1, x_2 & & & \geq & 0 \end{array}$$

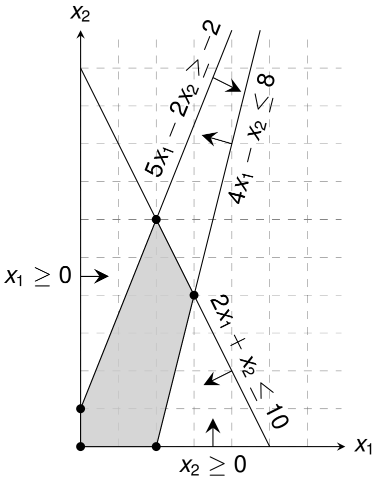
Any setting of x_1 and x_2 satisfying all constraints is a feasible solution



A Small(er) Example

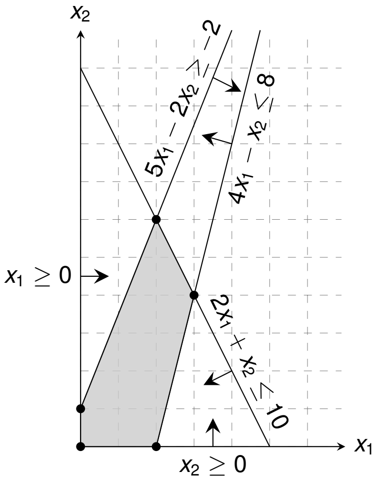
$$\begin{array}{lrcrcr} \text{maximize} & x_1 & + & x_2 & & \\ \text{subject to} & & & & & \\ & 4x_1 & - & x_2 & \leq & 8 \\ & 2x_1 & + & x_2 & \leq & 10 \\ & 5x_1 & - & 2x_2 & \geq & -2 \\ & x_1, x_2 & & & \geq & 0 \end{array}$$

Any setting of x_1 and x_2 satisfying all constraints is a feasible solution



A Small(er) Example

$$\begin{array}{llcrcr} \text{maximize} & x_1 & + & x_2 & & \\ \text{subject to} & & & & & \\ & 4x_1 & - & x_2 & \leq & 8 \\ & 2x_1 & + & x_2 & \leq & 10 \\ & 5x_1 & - & 2x_2 & \geq & -2 \\ & x_1, x_2 & & & \geq & 0 \end{array}$$

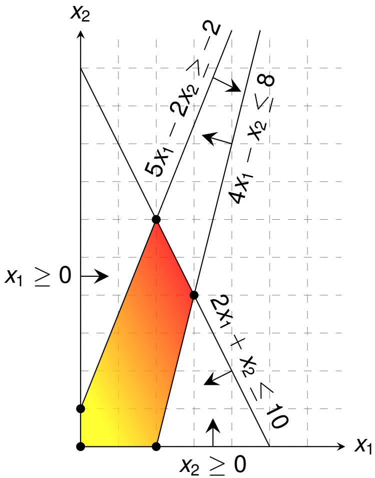


Graphical Procedure: Move the line $x_1 + x_2 = z$ as far up as possible.

A Small(er) Example

$$
\begin{array}{lrcrcr}
\text{maximize} & x_1 & + & x_2 & & \\
\text{subject to} & & & & & \\
& 4x_1 & - & x_2 & \leq & 8 \\
& 2x_1 & + & x_2 & \leq & 10 \\
& 5x_1 & - & 2x_2 & \geq & -2 \\
& x_1, x_2 & & & \geq & 0
\end{array}
$$

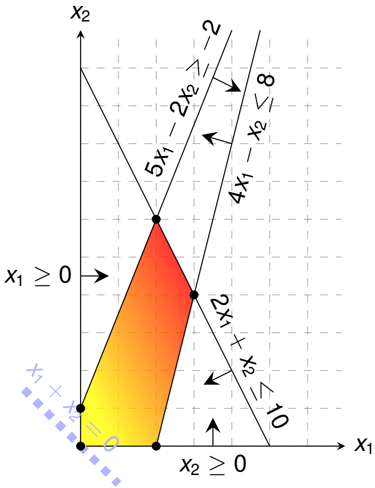
Graphical Procedure: Move the line $x_1 + x_2 = z$ as far up as possible.



A Small(er) Example

$$\begin{array}{lrcrcr} \text{maximize} & x_1 & + & x_2 & & \\ \text{subject to} & & & & & \\ & 4x_1 & - & x_2 & \leq & 8 \\ & 2x_1 & + & x_2 & \leq & 10 \\ & 5x_1 & - & 2x_2 & \geq & -2 \\ & x_1, x_2 & & & \geq & 0 \end{array}$$

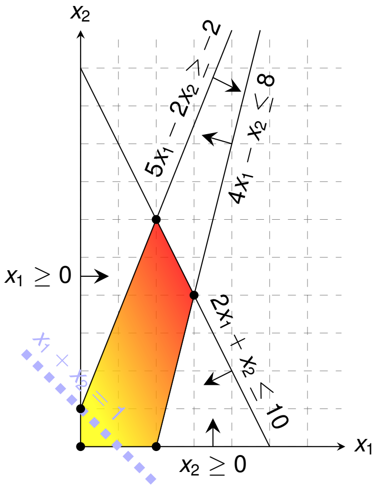
Graphical Procedure: Move the line $x_1 + x_2 = z$ as far up as possible.



A Small(er) Example

$$\begin{array}{lrcrcr}
\text{maximize} & x_1 & + & x_2 & & \\
\text{subject to} & & & & & \\
& 4x_1 & - & x_2 & \leq & 8 \\
& 2x_1 & + & x_2 & \leq & 10 \\
& 5x_1 & - & 2x_2 & \geq & -2 \\
& x_1, x_2 & & & \geq & 0
\end{array}$$

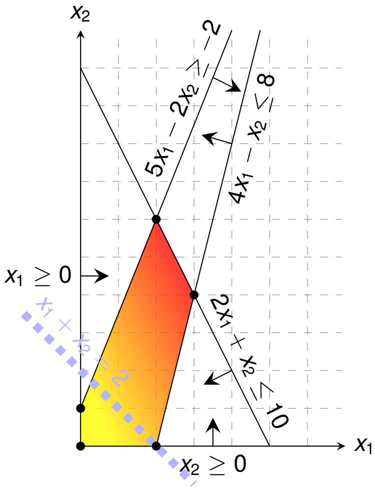
Graphical Procedure: Move the line $x_1 + x_2 = z$ as far up as possible.



A Small(er) Example

$$\begin{array}{lrcrcr}
\text{maximize} & x_1 & + & x_2 & & \\
\text{subject to} & & & & & \\
& 4x_1 & - & x_2 & \leq & 8 \\
& 2x_1 & + & x_2 & \leq & 10 \\
& 5x_1 & - & 2x_2 & \geq & -2 \\
& x_1, x_2 & & & \geq & 0
\end{array}$$

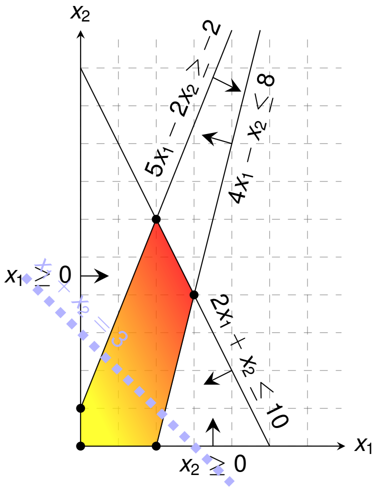
Graphical Procedure: Move the line $x_1 + x_2 = z$ as far up as possible.



A Small(er) Example

$$\begin{array}{llcrcr}
\text{maximize} & x_1 & + & x_2 & & \\
\text{subject to} & & & & & \\
& 4x_1 & - & x_2 & \leq & 8 \\
& 2x_1 & + & x_2 & \leq & 10 \\
& 5x_1 & - & 2x_2 & \geq & -2 \\
& x_1, x_2 & & & \geq & 0
\end{array}$$

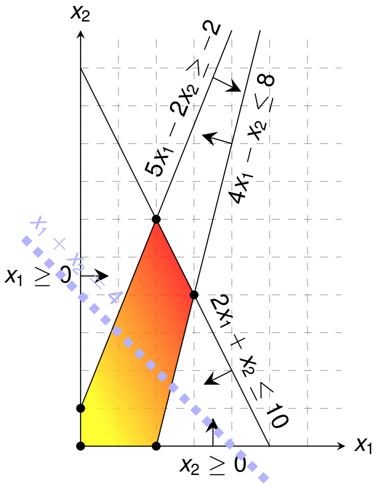
Graphical Procedure: Move the line $x_1 + x_2 = z$ as far up as possible.



A Small(er) Example

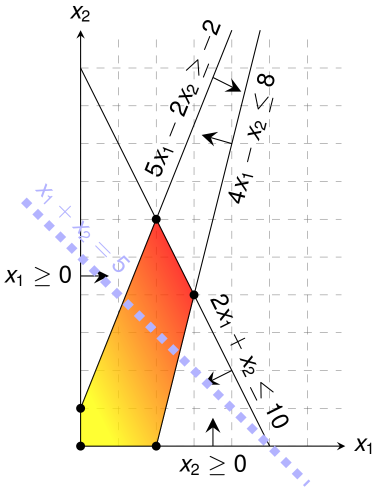
$$
\begin{array}{llcrcr}
\text{maximize} & x_1 & + & x_2 & & \\
\text{subject to} & & & & & \\
& 4x_1 & - & x_2 & \leq & 8 \\
& 2x_1 & + & x_2 & \leq & 10 \\
& 5x_1 & - & 2x_2 & \geq & -2 \\
& x_1, x_2 & & & \geq & 0
\end{array}
$$

Graphical Procedure: Move the line $x_1 + x_2 = z$ as far up as possible.



A Small(er) Example

$$
\begin{array}{llrcrcr}
\text{maximize} & x_1 & + & x_2 & & \\
\text{subject to} & & & & & \\
& 4x_1 & - & x_2 & \leq & 8 \\
& 2x_1 & + & x_2 & \leq & 10 \\
& 5x_1 & - & 2x_2 & \geq & -2 \\
& x_1, x_2 & & & \geq & 0
\end{array}
$$

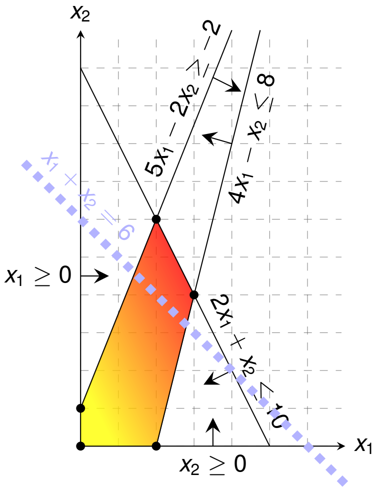


Graphical Procedure: Move the line $x_1 + x_2 = z$ as far up as possible.

A Small(er) Example

$$\begin{array}{lrcrcr} \text{maximize} & x_1 & + & x_2 & & \\ \text{subject to} & & & & & \\ & 4x_1 & - & x_2 & \leq & 8 \\ & 2x_1 & + & x_2 & \leq & 10 \\ & 5x_1 & - & 2x_2 & \geq & -2 \\ & x_1, x_2 & & & \geq & 0 \end{array}$$

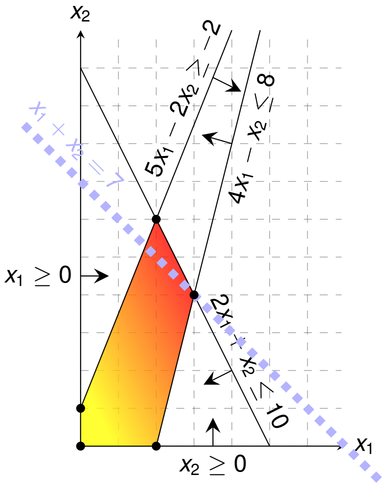
Graphical Procedure: Move the line $x_1 + x_2 = z$ as far up as possible.



A Small(er) Example

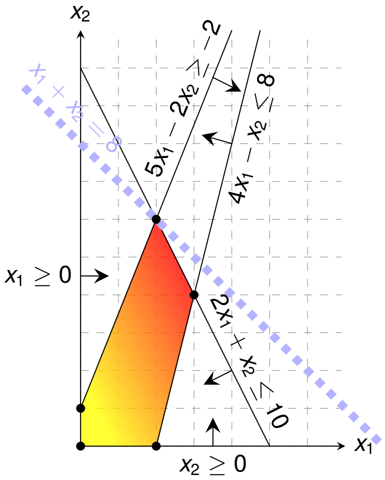
$$\begin{array}{lrcrcr} \text{maximize} & x_1 & + & x_2 & & \\ \text{subject to} & & & & & \\ & 4x_1 & - & x_2 & \leq & 8 \\ & 2x_1 & + & x_2 & \leq & 10 \\ & 5x_1 & - & 2x_2 & \geq & -2 \\ & x_1, x_2 & & & \geq & 0 \end{array}$$

Graphical Procedure: Move the line $x_1 + x_2 = z$ as far up as possible.



A Small(er) Example

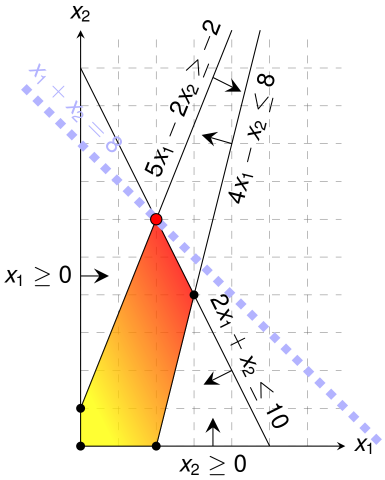
$$\begin{array}{lrcrcr}
\text{maximize} & x_1 & + & x_2 & & \\
\text{subject to} & & & & & \\
& 4x_1 & - & x_2 & \leq & 8 \\
& 2x_1 & + & x_2 & \leq & 10 \\
& 5x_1 & - & 2x_2 & \geq & -2 \\
& x_1, x_2 & & & \geq & 0
\end{array}$$



Graphical Procedure: Move the line $x_1 + x_2 = z$ as far up as possible.

A Small(er) Example

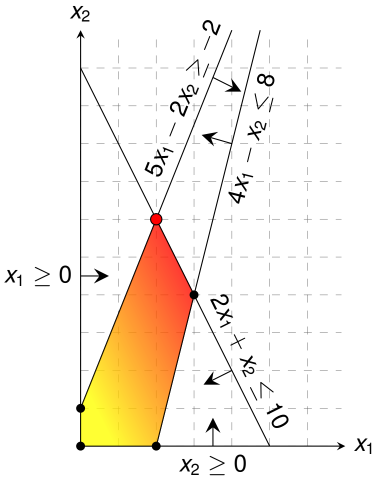
$$
\begin{array}{lrcrcr}
\text{maximize} & x_1 & + & x_2 & & \\
\text{subject to} & & & & & \\
& 4x_1 & - & x_2 & \leq & 8 \\
& 2x_1 & + & x_2 & \leq & 10 \\
& 5x_1 & - & 2x_2 & \geq & -2 \\
& x_1, x_2 & & & \geq & 0
\end{array}
$$



Graphical Procedure: Move the line $x_1 + x_2 = z$ as far up as possible.

A Small(er) Example

$$\begin{array}{llcrcr} \text{maximize} & x_1 & + & x_2 & & \\ \text{subject to} & & & & & \\ & 4x_1 & - & x_2 & \leq & 8 \\ & 2x_1 & + & x_2 & \leq & 10 \\ & 5x_1 & - & 2x_2 & \geq & -2 \\ & x_1, x_2 & & & \geq & 0 \end{array}$$

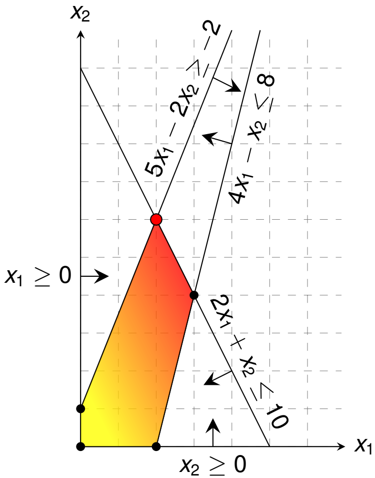


Graphical Procedure: Move the line $x_1 + x_2 = z$ as far up as possible.

A Small(er) Example

$$\begin{array}{llcrcr}
\text{maximize} & x_1 & + & x_2 & & \\
\text{subject to} & & & & & \\
& 4x_1 & - & x_2 & \leq & 8 \\
& 2x_1 & + & x_2 & \leq & 10 \\
& 5x_1 & - & 2x_2 & \geq & -2 \\
& x_1, x_2 & & & \geq & 0
\end{array}$$

Graphical Procedure: Move the line $x_1 + x_2 = z$ as far up as possible.



While the same approach also works for higher-dimensions, we need to take a more systematic and algebraic procedure.

Outline

Introduction

Standard and Slack Forms

Formulating Problems as Linear Programs

Simplex Algorithm

Finding an Initial Solution

Standard and Slack Forms

Standard Form

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

subject to

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \qquad \text{for } i = 1, 2, \ldots, m$$

$$x_j \geq 0 \qquad \text{for } j = 1, 2, \ldots, n$$

Standard and Slack Forms

Standard Form

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

Objective Function

subject to

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \qquad \text{for } i = 1, 2, \ldots, m$$

$$x_j \geq 0 \qquad \text{for } j = 1, 2, \ldots, n$$

Standard and Slack Forms

Standard Form

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

Objective Function

subject to

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \qquad \text{for } i = 1, 2, \ldots, m$$

$$x_j \geq 0 \qquad \text{for } j = 1, 2, \ldots, n$$

$n + m$ Constraints

Standard and Slack Forms

Standard Form

maximize $\sum_{j=1}^{n} c_j x_j$ — Objective Function

subject to

$n + m$ Constraints:

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \qquad \text{for } i = 1, 2, \ldots, m$$

$$x_j \geq 0 \qquad \text{for } j = 1, 2, \ldots, n$$

Non-Negativity Constraints

Standard and Slack Forms

Standard Form

maximize $\sum_{j=1}^{n} c_j x_j$ — Objective Function

subject to

$n + m$ Constraints:

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \text{for } i = 1, 2, \ldots, m$$

$$x_j \geq 0 \quad \text{for } j = 1, 2, \ldots, n$$

Non-Negativity Constraints

Standard Form (Matrix-Vector-Notation)

maximize $c^T x$ — Inner product of two vectors

subject to

$$Ax \leq b$$ — Matrix-vector product

$$x \geq 0$$

Converting Linear Programs into Standard Form

Reasons for a LP not being in standard form:

1. The objective might be a minimization rather than maximization.
2. There might be variables without nonnegativity constraints.
3. There might be equality constraints.
4. There might be inequality constraints (with $\geq$ instead of $\leq$).

Converting Linear Programs into Standard Form

Reasons for a LP not being in standard form:

1. The objective might be a minimization rather than maximization.
2. There might be variables without nonnegativity constraints.
3. There might be equality constraints.
4. There might be inequality constraints (with $\geq$ instead of $\leq$).

Goal: Convert linear program into an equivalent program which is in standard form

Converting Linear Programs into Standard Form

Reasons for a LP not being in standard form:

1. The objective might be a minimization rather than maximization.
2. There might be variables without nonnegativity constraints.
3. There might be equality constraints.
4. There might be inequality constraints (with $\geq$ instead of $\leq$).

Goal: Convert linear program into an equivalent program which is in standard form

Equivalence: a correspondence (not necessarily a bijection) between solutions so that their objective values are identical.

Converting Linear Programs into Standard Form

Reasons for a LP not being in standard form:

1. The objective might be a minimization rather than maximization.
2. There might be variables without nonnegativity constraints.
3. There might be equality constraints.
4. There might be inequality constraints (with $\geq$ instead of $\leq$).

Goal: Convert linear program into an equivalent program which is in standard form

Equivalence: a correspondence (not necessarily a bijection) between solutions so that their objective values are identical.

When switching from maximization to minimization, sign of objective value changes.

Converting into Standard Form (1/5)

Reasons for a LP not being in standard form:

1. The objective might be a minimization rather than maximization.

Converting into Standard Form (1/5)

Reasons for a LP not being in standard form:

1. The objective might be a minimization rather than maximization.

$$
\begin{array}{llcccc}
\text{minimize} & -2x_1 & + & 3x_2 & & \\
\text{subject to} & & & & & \\
& x_1 & + & x_2 & = & 7 \\
& x_1 & - & 2x_2 & \leq & 4 \\
& x_1 & & & \geq & 0
\end{array}
$$

Converting into Standard Form (1/5)

Reasons for a LP not being in standard form:

1. The objective might be a minimization rather than maximization.

$$
\begin{array}{lrcrcr}
\text{minimize} & -2x_1 & + & 3x_2 & & \\
\text{subject to} & & & & & \\
 & x_1 & + & x_2 & = & 7 \\
 & x_1 & - & 2x_2 & \leq & 4 \\
 & x_1 & & & \geq & 0
\end{array}
$$

Negate objective function

Converting into Standard Form (1/5)

Reasons for a LP not being in standard form:

1. The objective might be a minimization rather than maximization.

$$\begin{array}{lrcrcl} \text{minimize} & -2x_1 & + & 3x_2 & & \\ \text{subject to} & & & & & \\ & x_1 & + & x_2 & = & 7 \\ & x_1 & - & 2x_2 & \leq & 4 \\ & x_1 & & & \geq & 0 \end{array}$$

Negate objective function

$$\begin{array}{lrcrcl} \text{maximize} & 2x_1 & - & 3x_2 & & \\ \text{subject to} & & & & & \\ & x_1 & + & x_2 & = & 7 \\ & x_1 & - & 2x_2 & \leq & 4 \\ & x_1 & & & \geq & 0 \end{array}$$

Converting into Standard Form (2/5)

Reasons for a LP not being in standard form:

2. There might be variables without nonnegativity constraints.

Converting into Standard Form (2/5)

Reasons for a LP not being in standard form:

2. There might be variables without nonnegativity constraints.

$$
\begin{array}{llcrcr}
\text{maximize} & 2x_1 & - & 3x_2 & & \\
\text{subject to} & & & & & \\
& x_1 & + & x_2 & = & 7 \\
& x_1 & - & 2x_2 & \leq & 4 \\
& x_1 & & & \geq & 0
\end{array}
$$

Converting into Standard Form (2/5)

Reasons for a LP not being in standard form:

2. There might be variables without nonnegativity constraints.

$$
\begin{array}{llcllcl}
\text{maximize} & 2x_1 & - & 3x_2 & & \\
\text{subject to} & & & & & \\
& x_1 & + & x_2 & = & 7 \\
& x_1 & - & 2x_2 & \leq & 4 \\
& x_1 & & & \geq & 0
\end{array}
$$

Replace x_2 by two non-negative variables x_2' and x_2''

Converting into Standard Form (2/5)

Reasons for a LP not being in standard form:

2. There might be variables without nonnegativity constraints.

$$\begin{array}{llcll} \text{maximize} & 2x_1 & - & 3x_2 & \\ \text{subject to} & & & & \\ & x_1 & + & x_2 & = & 7 \\ & x_1 & - & 2x_2 & \leq & 4 \\ & x_1 & & & \geq & 0 \end{array}$$

Replace x_2 by two non-negative variables x_2' and x_2''

$$\begin{array}{llcllll} \text{maximize} & 2x_1 & - & 3x_2' & + & 3x_2'' & \\ \text{subject to} & & & & & & \\ & x_1 & + & x_2' & - & x_2'' & = & 7 \\ & x_1 & - & 2x_2' & + & 2x_2'' & \leq & 4 \\ & x_1, x_2', x_2'' & & & & & \geq & 0 \end{array}$$

Converting into Standard Form (3/5)

Reasons for a LP not being in standard form:

3. There might be equality constraints.

Converting into Standard Form (3/5)

Reasons for a LP not being in standard form:

3. There might be equality constraints.

$$
\begin{array}{lrcrcrcr}
\text{maximize} & 2x_1 & - & 3x_2' & + & 3x_2'' & & \\
\text{subject to} & & & & & & & \\
& x_1 & + & x_2' & - & x_2'' & = & 7 \\
& x_1 & - & 2x_2' & + & 2x_2'' & \leq & 4 \\
& & & x_1, x_2', x_2'' & & & \geq & 0
\end{array}
$$

Converting into Standard Form (3/5)

Reasons for a LP not being in standard form:

3. There might be equality constraints.

$$\begin{array}{llllllll} \text{maximize} & 2x_1 & - & 3x_2' & + & 3x_2'' & & \\ \text{subject to} & & & & & & & \\ & x_1 & + & x_2' & - & x_2'' & = & 7 \\ & x_1 & - & 2x_2' & + & 2x_2'' & \leq & 4 \\ & x_1, x_2', x_2'' & & & & & \geq & 0 \end{array}$$

Replace each equality by two inequalities.

Converting into Standard Form (3/5)

Reasons for a LP not being in standard form:

3. There might be equality constraints.

$$\begin{array}{llcrcrcl}
\text{maximize} & 2x_1 & - & 3x_2' & + & 3x_2'' & & \\
\text{subject to} & & & & & & & \\
& x_1 & + & x_2' & - & x_2'' & = & 7 \\
& x_1 & - & 2x_2' & + & 2x_2'' & \leq & 4 \\
& x_1, x_2', x_2'' & & & & & \geq & 0
\end{array}$$

Replace each equality by two inequalities.

$$\begin{array}{llcrcrcl}
\text{maximize} & 2x_1 & - & 3x_2' & + & 3x_2'' & & \\
\text{subject to} & & & & & & & \\
& x_1 & + & x_2' & - & x_2'' & \leq & 7 \\
& x_1 & + & x_2' & - & x_2'' & \geq & 7 \\
& x_1 & - & 2x_2' & + & 2x_2'' & \leq & 4 \\
& x_1, x_2', x_2'' & & & & & \geq & 0
\end{array}$$

Converting into Standard Form (4/5)

Reasons for a LP not being in standard form:

4. There might be inequality constraints (with $\geq$ instead of $\leq$).

Converting into Standard Form (4/5)

Reasons for a LP not being in standard form:

4. There might be inequality constraints (with $\geq$ instead of $\leq$).

$$\begin{array}{lrcrcrcr}
\text{maximize} & 2x_1 & - & 3x_2' & + & 3x_2'' & & \\
\text{subject to} & & & & & & & \\
& x_1 & + & x_2' & - & x_2'' & \leq & 7 \\
& x_1 & + & x_2' & - & x_2'' & \geq & 7 \\
& x_1 & - & 2x_2' & + & 2x_2'' & \leq & 4 \\
& \multicolumn{5}{c}{x_1, x_2', x_2''} & \geq & 0
\end{array}$$

Converting into Standard Form (4/5)

Reasons for a LP not being in standard form:

4. There might be inequality constraints (with $\geq$ instead of $\leq$).

$$
\begin{array}{llllllll}
\text{maximize} & 2x_1 & - & 3x_2' & + & 3x_2'' & & \\
\text{subject to} & & & & & & & \\
& x_1 & + & x_2' & - & x_2'' & \leq & 7 \\
& x_1 & + & x_2' & - & x_2'' & \geq & 7 \\
& x_1 & - & 2x_2' & + & 2x_2'' & \leq & 4 \\
& x_1, x_2', x_2'' & & & & & \geq & 0
\end{array}
$$

Negate respective inequalities.

Converting into Standard Form (4/5)

Reasons for a LP not being in standard form:

4. There might be inequality constraints (with $\geq$ instead of $\leq$).

$$
\begin{array}{llllllll}
\text{maximize} & 2x_1 & - & 3x_2' & + & 3x_2'' & & \\
\text{subject to} & & & & & & & \\
& x_1 & + & x_2' & - & x_2'' & \leq & 7 \\
& x_1 & + & x_2' & - & x_2'' & \geq & 7 \\
& x_1 & - & 2x_2' & + & 2x_2'' & \leq & 4 \\
& x_1, x_2', x_2'' & & & & & \geq & 0
\end{array}
$$

Negate respective inequalities.

$$
\begin{array}{llllllll}
\text{maximize} & 2x_1 & - & 3x_2' & + & 3x_2'' & & \\
\text{subject to} & & & & & & & \\
& x_1 & + & x_2' & - & x_2'' & \leq & 7 \\
& -x_1 & - & x_2' & + & x_2'' & \leq & -7 \\
& x_1 & - & 2x_2' & + & 2x_2'' & \leq & 4 \\
& x_1, x_2', x_2'' & & & & & \geq & 0
\end{array}
$$

Converting into Standard Form (5/5)

$$
\begin{array}{lrcrcrcr}
\text{maximize} & 2x_1 & - & 3x_2 & + & 3x_3 & & \\
\text{subject to} & & & & & & & \\
 & x_1 & + & x_2 & - & x_3 & \leq & 7 \\
 & -x_1 & - & x_2 & + & x_3 & \leq & -7 \\
 & x_1 & - & 2x_2 & + & 2x_3 & \leq & 4 \\
 & & x_1, x_2, x_3 & & & & \geq & 0
\end{array}
$$

Converting into Standard Form (5/5)

Rename variable names (for consistency).

$$
\begin{array}{lrcrcrcr}
\text{maximize} & 2x_1 & - & 3x_2 & + & 3x_3 & & \\
\text{subject to} & & & & & & & \\
& x_1 & + & x_2 & - & x_3 & \leq & 7 \\
& -x_1 & - & x_2 & + & x_3 & \leq & -7 \\
& x_1 & - & 2x_2 & + & 2x_3 & \leq & 4 \\
& & & x_1, x_2, x_3 & & & \geq & 0
\end{array}
$$

Converting into Standard Form (5/5)

Rename variable names (for consistency).

$$
\begin{array}{llllllll}
\text{maximize} & 2x_1 & - & 3x_2 & + & 3x_3 & & \\
\text{subject to} & & & & & & & \\
& x_1 & + & x_2 & - & x_3 & \leq & 7 \\
& -x_1 & - & x_2 & + & x_3 & \leq & -7 \\
& x_1 & - & 2x_2 & + & 2x_3 & \leq & 4 \\
& x_1, x_2, x_3 & & & & & \geq & 0
\end{array}
$$

It is always possible to convert a linear program into standard form.

Converting Standard Form into Slack Form (1/3)

Goal: Convert standard form into slack form, where all constraints except for the non-negativity constraints are equalities.

Converting Standard Form into Slack Form (1/3)

Goal: Convert standard form into slack form, where all constraints except for the non-negativity constraints are equalities.

For the simplex algorithm, it is more convenient to work with equality constraints.

Converting Standard Form into Slack Form (1/3)

Goal: Convert standard form into slack form, where all constraints except for the non-negativity constraints are equalities.

For the simplex algorithm, it is more convenient to work with equality constraints.

Introducing Slack Variables

Converting Standard Form into Slack Form (1/3)

Goal: Convert standard form into slack form, where all constraints except for the non-negativity constraints are equalities.

For the simplex algorithm, it is more convenient to work with equality constraints.

Introducing Slack Variables

- Let $\sum_{j=1}^{n} a_{ij}x_j \leq b_i$ be an inequality constraint

Converting Standard Form into Slack Form (1/3)

Goal: Convert standard form into slack form, where all constraints except for the non-negativity constraints are equalities.

For the simplex algorithm, it is more convenient to work with equality constraints.

Introducing Slack Variables

- Let $\sum_{j=1}^{n} a_{ij}x_j \leq b_i$ be an inequality constraint
- Introduce a slack variable s by

Converting Standard Form into Slack Form (1/3)

Goal: Convert standard form into slack form, where all constraints except for the non-negativity constraints are equalities.

For the simplex algorithm, it is more convenient to work with equality constraints.

Introducing Slack Variables

- Let $\sum_{j=1}^{n} a_{ij}x_j \leq b_i$ be an inequality constraint
- Introduce a slack variable s by

$$s = b_i - \sum_{j=1}^{n} a_{ij}x_j$$

Converting Standard Form into Slack Form (1/3)

Goal: Convert standard form into slack form, where all constraints except for the non-negativity constraints are equalities.

For the simplex algorithm, it is more convenient to work with equality constraints.

Introducing Slack Variables

- Let $\sum_{j=1}^{n} a_{ij}x_j \leq b_i$ be an inequality constraint
- Introduce a slack variable s by

$$s = b_i - \sum_{j=1}^{n} a_{ij}x_j$$

$$s \geq 0.$$

Converting Standard Form into Slack Form (1/3)

Goal: Convert standard form into slack form, where all constraints except for the non-negativity constraints are equalities.

For the simplex algorithm, it is more convenient to work with equality constraints.

Introducing Slack Variables

- Let $\sum_{j=1}^{n} a_{ij}x_j \leq b_i$ be an inequality constraint
- Introduce a slack variable s by

$$s = b_i - \sum_{j=1}^{n} a_{ij}x_j$$

$$s \geq 0.$$

s measures the slack between the two sides of the inequality.

Converting Standard Form into Slack Form (1/3)

Goal: Convert standard form into slack form, where all constraints except for the non-negativity constraints are equalities.

For the simplex algorithm, it is more convenient to work with equality constraints.

Introducing Slack Variables

- Let $\sum_{j=1}^{n} a_{ij}x_j \leq b_i$ be an inequality constraint
- Introduce a slack variable s by

$$s = b_i - \sum_{j=1}^{n} a_{ij}x_j$$

$$s \geq 0.$$

s measures the slack between the two sides of the inequality.

- Denote slack variable of the ith inequality by x_{n+i}

Converting Standard Form into Slack Form (2/3)

$$
\begin{array}{lrcrcrcr}
\text{maximize} & 2x_1 & - & 3x_2 & + & 3x_3 & & \\
\text{subject to} & & & & & & & \\
 & x_1 & + & x_2 & - & x_3 & \leq & 7 \\
 & -x_1 & - & x_2 & + & x_3 & \leq & -7 \\
 & x_1 & - & 2x_2 & + & 2x_3 & \leq & 4 \\
 & & x_1, x_2, x_3 & & & & \geq & 0
\end{array}
$$

Converting Standard Form into Slack Form (2/3)

$$
\begin{array}{lrcrcrcr}
\text{maximize} & 2x_1 & - & 3x_2 & + & 3x_3 & & \\
\text{subject to} & & & & & & & \\
 & x_1 & + & x_2 & - & x_3 & \leq & 7 \\
 & -x_1 & - & x_2 & + & x_3 & \leq & -7 \\
 & x_1 & - & 2x_2 & + & 2x_3 & \leq & 4 \\
 & \multicolumn{5}{l}{x_1, x_2, x_3} & \geq & 0
\end{array}
$$

↓ Introduce slack variables

Converting Standard Form into Slack Form (2/3)

$$
\begin{array}{llllllll}
\text{maximize} & 2x_1 & - & 3x_2 & + & 3x_3 & & \\
\text{subject to} & & & & & & & \\
& x_1 & + & x_2 & - & x_3 & \leq & 7 \\
& -x_1 & - & x_2 & + & x_3 & \leq & -7 \\
& x_1 & - & 2x_2 & + & 2x_3 & \leq & 4 \\
& x_1, x_2, x_3 & & & & & \geq & 0
\end{array}
$$

Introduce slack variables

subject to

$$x_4 = 7 - x_1 - x_2 + x_3$$

Converting Standard Form into Slack Form (2/3)

$$\begin{array}{llllllll} \text{maximize} & 2x_1 & - & 3x_2 & + & 3x_3 & & \\ \text{subject to} & & & & & & & \\ & x_1 & + & x_2 & - & x_3 & \leq & 7 \\ & -x_1 & - & x_2 & + & x_3 & \leq & -7 \\ & x_1 & - & 2x_2 & + & 2x_3 & \leq & 4 \\ & x_1, x_2, x_3 & & & & & \geq & 0 \end{array}$$

Introduce slack variables

subject to

$$\begin{array}{lllllllll} x_4 & = & 7 & - & x_1 & - & x_2 & + & x_3 \\ x_5 & = & -7 & + & x_1 & + & x_2 & - & x_3 \end{array}$$

Converting Standard Form into Slack Form (2/3)

$$
\begin{array}{lrcrcrcr}
\text{maximize} & 2x_1 & - & 3x_2 & + & 3x_3 & & \\
\text{subject to} & & & & & & & \\
 & x_1 & + & x_2 & - & x_3 & \leq & 7 \\
 & -x_1 & - & x_2 & + & x_3 & \leq & -7 \\
 & x_1 & - & 2x_2 & + & 2x_3 & \leq & 4 \\
 & & x_1, x_2, x_3 & & & & \geq & 0
\end{array}
$$

Introduce slack variables

subject to

$$
\begin{array}{rcrcrcrcr}
x_4 & = & 7 & - & x_1 & - & x_2 & + & x_3 \\
x_5 & = & -7 & + & x_1 & + & x_2 & - & x_3 \\
x_6 & = & 4 & - & x_1 & + & 2x_2 & - & 2x_3
\end{array}
$$

Converting Standard Form into Slack Form (2/3)

$$
\begin{array}{llllllll}
\text{maximize} & 2x_1 & - & 3x_2 & + & 3x_3 & & \\
\text{subject to} & & & & & & & \\
& x_1 & + & x_2 & - & x_3 & \leq & 7 \\
& -x_1 & - & x_2 & + & x_3 & \leq & -7 \\
& x_1 & - & 2x_2 & + & 2x_3 & \leq & 4 \\
& \multicolumn{5}{c}{x_1, x_2, x_3} & \geq & 0
\end{array}
$$

↓ Introduce slack variables

$$
\begin{array}{lllllllll}
\text{subject to} & & & & & & & & \\
x_4 & = & 7 & - & x_1 & - & x_2 & + & x_3 \\
x_5 & = & -7 & + & x_1 & + & x_2 & - & x_3 \\
x_6 & = & 4 & - & x_1 & + & 2x_2 & - & 2x_3 \\
\multicolumn{4}{c}{x_1, x_2, x_3, x_4, x_5, x_6} & \geq & 0 & & &
\end{array}
$$

Converting Standard Form into Slack Form (2/3)

$$\begin{array}{lrcrcrcr}
\text{maximize} & 2x_1 & - & 3x_2 & + & 3x_3 & & \\
\text{subject to} & & & & & & & \\
& x_1 & + & x_2 & - & x_3 & \leq & 7 \\
& -x_1 & - & x_2 & + & x_3 & \leq & -7 \\
& x_1 & - & 2x_2 & + & 2x_3 & \leq & 4 \\
& \multicolumn{5}{c}{x_1, x_2, x_3} & \geq & 0
\end{array}$$

Introduce slack variables

$$\begin{array}{lrcrcrcrcr}
\text{maximize} & & & & & 2x_1 & - & 3x_2 & + & 3x_3 \\
\text{subject to} & & & & & & & & & \\
& x_4 & = & 7 & - & x_1 & - & x_2 & + & x_3 \\
& x_5 & = & -7 & + & x_1 & + & x_2 & - & x_3 \\
& x_6 & = & 4 & - & x_1 & + & 2x_2 & - & 2x_3 \\
& \multicolumn{5}{c}{x_1, x_2, x_3, x_4, x_5, x_6} & \geq & 0 & &
\end{array}$$

Converting Standard Form into Slack Form (3/3)

$$
\begin{array}{lrcrcrcrcr}
\text{maximize} & & & & & 2x_1 & - & 3x_2 & + & 3x_3 \\
\text{subject to} & & & & & & & & & \\
& x_4 & = & 7 & - & x_1 & - & x_2 & + & x_3 \\
& x_5 & = & -7 & + & x_1 & + & x_2 & - & x_3 \\
& x_6 & = & 4 & - & x_1 & + & 2x_2 & - & 2x_3 \\
& \multicolumn{5}{c}{x_1, x_2, x_3, x_4, x_5, x_6} & \geq & 0 & &
\end{array}
$$

Converting Standard Form into Slack Form (3/3)

$$
\begin{array}{lrrrrrrrrr}
\text{maximize} & & & & & 2x_1 & - & 3x_2 & + & 3x_3 \\
\text{subject to} & & & & & & & & & \\
& x_4 & = & 7 & - & x_1 & - & x_2 & + & x_3 \\
& x_5 & = & -7 & + & x_1 & + & x_2 & - & x_3 \\
& x_6 & = & 4 & - & x_1 & + & 2x_2 & - & 2x_3 \\
& & & \multicolumn{3}{r}{x_1, x_2, x_3, x_4, x_5, x_6} & \geq & 0 & &
\end{array}
$$

Use variable z to denote objective function and omit the nonnegativity constraints.

Converting Standard Form into Slack Form (3/3)

$$
\begin{array}{lrrrrrrrrr}
\text{maximize} & & & & & 2x_1 & - & 3x_2 & + & 3x_3 \\
\text{subject to} & & & & & & & & & \\
 & x_4 & = & 7 & - & x_1 & - & x_2 & + & x_3 \\
 & x_5 & = & -7 & + & x_1 & + & x_2 & - & x_3 \\
 & x_6 & = & 4 & - & x_1 & + & 2x_2 & - & 2x_3 \\
 & \multicolumn{5}{r}{x_1, x_2, x_3, x_4, x_5, x_6} & \geq & 0 & &
\end{array}
$$

Use variable z to denote objective function and omit the nonnegativity constraints.

$$
\begin{array}{rrrrrrrrr}
z & = & & & 2x_1 & - & 3x_2 & + & 3x_3 \\
x_4 & = & 7 & - & x_1 & - & x_2 & + & x_3 \\
x_5 & = & -7 & + & x_1 & + & x_2 & - & x_3 \\
x_6 & = & 4 & - & x_1 & + & 2x_2 & - & 2x_3
\end{array}
$$

Converting Standard Form into Slack Form (3/3)

$$
\begin{array}{lrrrrrrrr}
\text{maximize} & & & & 2x_1 & - & 3x_2 & + & 3x_3 \\
\text{subject to} & & & & & & & & \\
x_4 & = & 7 & - & x_1 & - & x_2 & + & x_3 \\
x_5 & = & -7 & + & x_1 & + & x_2 & - & x_3 \\
x_6 & = & 4 & - & x_1 & + & 2x_2 & - & 2x_3 \\
\multicolumn{5}{r}{x_1, x_2, x_3, x_4, x_5, x_6} & \geq & 0 & &
\end{array}
$$

Use variable z to denote objective function and omit the nonnegativity constraints.

$$
\begin{array}{lrrrrrrrr}
z & = & & & 2x_1 & - & 3x_2 & + & 3x_3 \\
x_4 & = & 7 & - & x_1 & - & x_2 & + & x_3 \\
x_5 & = & -7 & + & x_1 & + & x_2 & - & x_3 \\
x_6 & = & 4 & - & x_1 & + & 2x_2 & - & 2x_3
\end{array}
$$

This is called slack form.

Basic and Non-Basic Variables

$$
\begin{array}{rcrcrcrcr}
z & = & & & 2x_1 & - & 3x_2 & + & 3x_3 \\
x_4 & = & 7 & - & x_1 & - & x_2 & + & x_3 \\
x_5 & = & -7 & + & x_1 & + & x_2 & - & x_3 \\
x_6 & = & 4 & - & x_1 & + & 2x_2 & - & 2x_3
\end{array}
$$

Basic and Non-Basic Variables

$$
\begin{array}{rcrcrcrcr}
z & = & & & 2x_1 & - & 3x_2 & + & 3x_3 \\
x_4 & = & 7 & - & x_1 & - & x_2 & + & x_3 \\
x_5 & = & -7 & + & x_1 & + & x_2 & - & x_3 \\
x_6 & = & 4 & - & x_1 & + & 2x_2 & - & 2x_3
\end{array}
$$

Basic Variables: $B = \{4, 5, 6\}$

Basic and Non-Basic Variables

$$
\begin{array}{rcrcrcrcr}
z & = & & & 2x_1 & - & 3x_2 & + & 3x_3 \\
x_4 & = & 7 & - & x_1 & - & x_2 & + & x_3 \\
x_5 & = & -7 & + & x_1 & + & x_2 & - & x_3 \\
x_6 & = & 4 & - & x_1 & + & 2x_2 & - & 2x_3
\end{array}
$$

Basic Variables: $B = \{4, 5, 6\}$

Non-Basic Variables: $N = \{1, 2, 3\}$

Basic and Non-Basic Variables

$$
\begin{array}{rcrcrcrcr}
z & = & & & 2x_1 & - & 3x_2 & + & 3x_3 \\
x_4 & = & 7 & - & x_1 & - & x_2 & + & x_3 \\
x_5 & = & -7 & + & x_1 & + & x_2 & - & x_3 \\
x_6 & = & 4 & - & x_1 & + & 2x_2 & - & 2x_3
\end{array}
$$

Basic Variables: $B = \{4, 5, 6\}$

Non-Basic Variables: $N = \{1, 2, 3\}$

Slack Form (Formal Definition)

Slack form is given by a tuple (N, B, A, b, c, v) so that

$$z = v + \sum_{j \in N} c_j x_j$$

$$x_i = b_i - \sum_{j \in N} a_{ij} x_j \qquad \text{for } i \in B,$$

and all variables are non-negative.

Basic and Non-Basic Variables

$$
\begin{array}{rcrcrcrcr}
z & = & & & 2x_1 & - & 3x_2 & + & 3x_3 \\
x_4 & = & 7 & - & x_1 & - & x_2 & + & x_3 \\
x_5 & = & -7 & + & x_1 & + & x_2 & - & x_3 \\
x_6 & = & 4 & - & x_1 & + & 2x_2 & - & 2x_3
\end{array}
$$

Basic Variables: $B = \{4, 5, 6\}$

Non-Basic Variables: $N = \{1, 2, 3\}$

Slack Form (Formal Definition)

Slack form is given by a tuple (N, B, A, b, c, v) so that

$$z = v + \sum_{j \in N} c_j x_j$$

$$x_i = b_i - \sum_{j \in N} a_{ij} x_j \qquad \text{for } i \in B,$$

and all variables are non-negative.

Variables/Coefficients on the right hand side are indexed by B and N.

Slack Form (Example)

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

Slack Form (Example)

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

Slack Form Notation

Slack Form (Example)

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

Slack Form Notation

- $B = \{1, 2, 4\}$, $N = \{3, 5, 6\}$

Slack Form (Example)

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

Slack Form Notation

- $B = \{1, 2, 4\}$, $N = \{3, 5, 6\}$
- $$A = \begin{pmatrix} a_{13} & a_{15} & a_{16} \\ a_{23} & a_{25} & a_{26} \\ a_{43} & a_{45} & a_{46} \end{pmatrix} = \begin{pmatrix} -1/6 & -1/6 & 1/3 \\ 8/3 & 2/3 & -1/3 \\ 1/2 & -1/2 & 0 \end{pmatrix}$$

Slack Form (Example)

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

Slack Form Notation

- $B = \{1,2,4\}$, $N = \{3,5,6\}$
- $$A = \begin{pmatrix} a_{13} & a_{15} & a_{16} \\ a_{23} & a_{25} & a_{26} \\ a_{43} & a_{45} & a_{46} \end{pmatrix} = \begin{pmatrix} -1/6 & -1/6 & 1/3 \\ 8/3 & 2/3 & -1/3 \\ 1/2 & -1/2 & 0 \end{pmatrix}$$
- $$b = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} 8 \\ 4 \\ 18 \end{pmatrix},$$

Slack Form (Example)

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

Slack Form Notation

- $B = \{1, 2, 4\}$, $N = \{3, 5, 6\}$
- $$A = \begin{pmatrix} a_{13} & a_{15} & a_{16} \\ a_{23} & a_{25} & a_{26} \\ a_{43} & a_{45} & a_{46} \end{pmatrix} = \begin{pmatrix} -1/6 & -1/6 & 1/3 \\ 8/3 & 2/3 & -1/3 \\ 1/2 & -1/2 & 0 \end{pmatrix}$$
- $$b = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} 8 \\ 4 \\ 18 \end{pmatrix}, \quad c = \begin{pmatrix} c_3 \\ c_5 \\ c_6 \end{pmatrix} = \begin{pmatrix} -1/6 \\ -1/6 \\ -2/3 \end{pmatrix}$$

Slack Form (Example)

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

Slack Form Notation

- $B = \{1, 2, 4\}$, $N = \{3, 5, 6\}$
- $$A = \begin{pmatrix} a_{13} & a_{15} & a_{16} \\ a_{23} & a_{25} & a_{26} \\ a_{43} & a_{45} & a_{46} \end{pmatrix} = \begin{pmatrix} -1/6 & -1/6 & 1/3 \\ 8/3 & 2/3 & -1/3 \\ 1/2 & -1/2 & 0 \end{pmatrix}$$
- $$b = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} 8 \\ 4 \\ 18 \end{pmatrix}, \quad c = \begin{pmatrix} c_3 \\ c_5 \\ c_6 \end{pmatrix} = \begin{pmatrix} -1/6 \\ -1/6 \\ -2/3 \end{pmatrix}$$
- $v = 28$

The Structure of Optimal Solutions

Definition

A point x is a vertex if it cannot be represented as a strict convex combination of two other points in the feasible set.

The Structure of Optimal Solutions

Definition

A point x is a vertex if it cannot be represented as a strict convex combination of two other points in the feasible set.

The set of feasible solutions is a convex set.

The Structure of Optimal Solutions

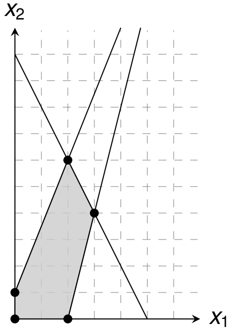
Definition

A point x is a vertex if it cannot be represented as a strict convex combination of two other points in the feasible set.

The set of feasible solutions is a convex set.

Theorem

If the slack form has an optimal solution, one of them occurs at a vertex.



The Structure of Optimal Solutions

Definition

A point x is a vertex if it cannot be represented as a strict convex combination of two other points in the feasible set.

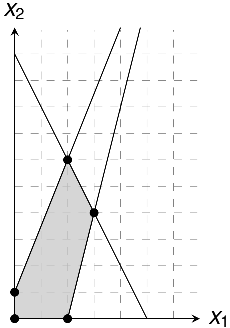
The set of feasible solutions is a convex set.

Theorem

If the slack form has an optimal solution, one of them occurs at a vertex.

Proof Sketch (informal and non-examinable):

- Rewrite LP s.t. $Ax = b$. Let x be optimal but not a vertex



The Structure of Optimal Solutions

Definition

A point x is a vertex if it cannot be represented as a strict convex combination of two other points in the feasible set.

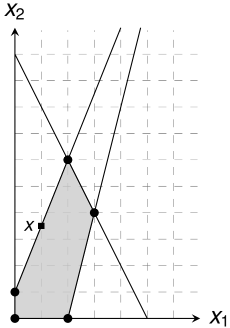
The set of feasible solutions is a convex set.

Theorem

If the slack form has an optimal solution, one of them occurs at a vertex.

Proof Sketch (informal and non-examinable):

- Rewrite LP s.t. $Ax = b$. Let x be optimal but not a vertex



The Structure of Optimal Solutions

Definition

A point x is a vertex if it cannot be represented as a strict convex combination of two other points in the feasible set.

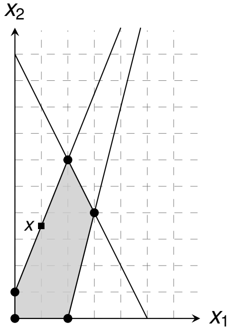
The set of feasible solutions is a convex set.

Theorem

If the slack form has an optimal solution, one of them occurs at a vertex.

Proof Sketch (informal and non-examinable):

- Rewrite LP s.t. $Ax = b$. Let x be optimal but not a vertex
  $\Rightarrow \exists$ vector d s.t. $x - d$ and $x + d$ are feasible



The Structure of Optimal Solutions

Definition

A point x is a vertex if it cannot be represented as a strict convex combination of two other points in the feasible set.

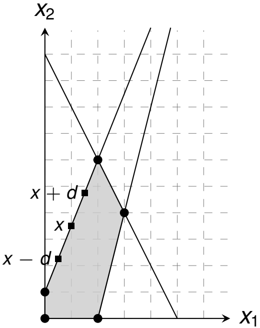
The set of feasible solutions is a convex set.

Theorem

If the slack form has an optimal solution, one of them occurs at a vertex.

Proof Sketch (informal and non-examinable):

- Rewrite LP s.t. $Ax = b$. Let x be optimal but not a vertex $\Rightarrow \exists$ vector d s.t. $x - d$ and $x + d$ are feasible



The Structure of Optimal Solutions

Definition

A point x is a vertex if it cannot be represented as a strict convex combination of two other points in the feasible set.

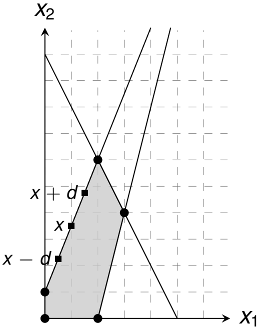
The set of feasible solutions is a convex set.

Theorem

If the slack form has an optimal solution, one of them occurs at a vertex.

Proof Sketch (informal and non-examinable):

- Rewrite LP s.t. $Ax = b$. Let x be optimal but not a vertex $\Rightarrow \exists$ vector d s.t. $x - d$ and $x + d$ are feasible
- Since $A(x + d) = b$ and $Ax = b \Rightarrow Ad = 0$



The Structure of Optimal Solutions

Definition

A point x is a vertex if it cannot be represented as a strict convex combination of two other points in the feasible set.

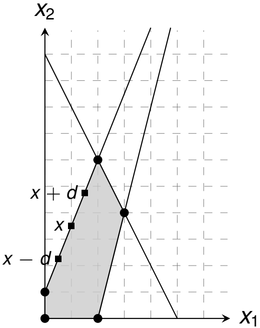
The set of feasible solutions is a convex set.

Theorem

If the slack form has an optimal solution, one of them occurs at a vertex.

Proof Sketch (informal and non-examinable):

- Rewrite LP s.t. $Ax = b$. Let x be optimal but not a vertex $\Rightarrow \exists$ vector d s.t. $x - d$ and $x + d$ are feasible
- Since $A(x + d) = b$ and $Ax = b \Rightarrow Ad = 0$
- W.l.o.g. assume $c^T d \geq 0$ (otherwise replace d by $-d$)



The Structure of Optimal Solutions

Definition

A point x is a vertex if it cannot be represented as a strict convex combination of two other points in the feasible set.

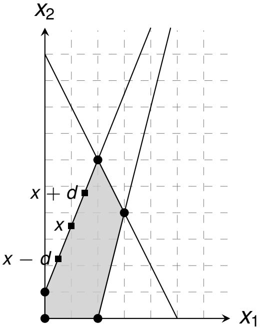
The set of feasible solutions is a convex set.

Theorem

If the slack form has an optimal solution, one of them occurs at a vertex.

Proof Sketch (informal and non-examinable):

- Rewrite LP s.t. $Ax = b$. Let x be optimal but not a vertex $\Rightarrow \exists$ vector d s.t. $x - d$ and $x + d$ are feasible
- Since $A(x + d) = b$ and $Ax = b \Rightarrow Ad = 0$
- W.l.o.g. assume $c^T d \geq 0$ (otherwise replace d by $-d$)
- Consider $x + \lambda d$ as a function of $\lambda \geq 0$



The Structure of Optimal Solutions

Definition

A point x is a vertex if it cannot be represented as a strict convex combination of two other points in the feasible set.

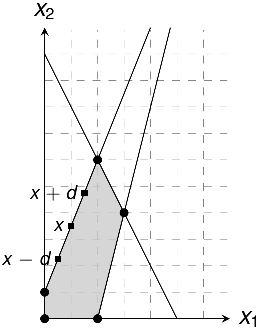
The set of feasible solutions is a convex set.

Theorem

If the slack form has an optimal solution, one of them occurs at a vertex.

Proof Sketch (informal and non-examinable):

- Rewrite LP s.t. $Ax = b$. Let x be optimal but not a vertex $\Rightarrow \exists$ vector d s.t. $x - d$ and $x + d$ are feasible
- Since $A(x + d) = b$ and $Ax = b \Rightarrow Ad = 0$
- W.l.o.g. assume $c^T d \geq 0$ (otherwise replace d by $-d$)
- Consider $x + \lambda d$ as a function of $\lambda \geq 0$
- Case 1: There exists j with $d_j < 0$



The Structure of Optimal Solutions

Definition

A point x is a vertex if it cannot be represented as a strict convex combination of two other points in the feasible set.

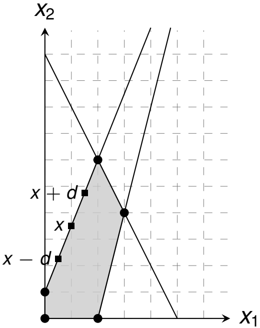
The set of feasible solutions is a convex set.

Theorem

If the slack form has an optimal solution, one of them occurs at a vertex.

Proof Sketch (informal and non-examinable):

- Rewrite LP s.t. $Ax = b$. Let x be optimal but not a vertex $\Rightarrow \exists$ vector d s.t. $x - d$ and $x + d$ are feasible
- Since $A(x + d) = b$ and $Ax = b \Rightarrow Ad = 0$
- W.l.o.g. assume $c^T d \geq 0$ (otherwise replace d by $-d$)
- Consider $x + \lambda d$ as a function of $\lambda \geq 0$
- Case 1: There exists j with $d_j < 0$
  - Increase λ from 0 to λ' until a new entry of $x + \lambda d$ becomes zero



The Structure of Optimal Solutions

Definition

A point x is a vertex if it cannot be represented as a strict convex combination of two other points in the feasible set.

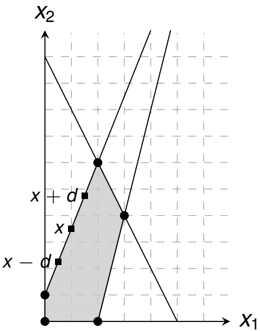
The set of feasible solutions is a convex set.

Theorem

If the slack form has an optimal solution, one of them occurs at a vertex.

Proof Sketch (informal and non-examinable):

- Rewrite LP s.t. $Ax = b$. Let x be optimal but not a vertex $\Rightarrow \exists$ vector d s.t. $x - d$ and $x + d$ are feasible
- Since $A(x + d) = b$ and $Ax = b \Rightarrow Ad = 0$
- W.l.o.g. assume $c^T d \geq 0$ (otherwise replace d by $-d$)
- Consider $x + \lambda d$ as a function of $\lambda \geq 0$
- Case 1: There exists j with $d_j < 0$
  - Increase λ from 0 to λ' until a new entry of $x + \lambda d$ becomes zero
  - $x + \lambda' d$ feasible, since $A(x + \lambda' d) = Ax = b$ and $x + \lambda' d \geq 0$



The Structure of Optimal Solutions

Definition

A point x is a vertex if it cannot be represented as a strict convex combination of two other points in the feasible set.

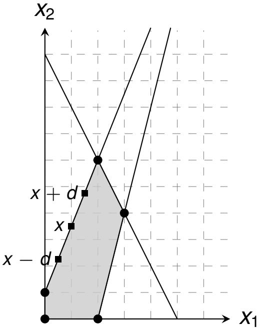
The set of feasible solutions is a convex set.

Theorem

If the slack form has an optimal solution, one of them occurs at a vertex.

Proof Sketch (informal and non-examinable):

- Rewrite LP s.t. $Ax = b$. Let x be optimal but not a vertex $\Rightarrow \exists$ vector d s.t. $x - d$ and $x + d$ are feasible
- Since $A(x + d) = b$ and $Ax = b \Rightarrow Ad = 0$
- W.l.o.g. assume $c^T d \geq 0$ (otherwise replace d by $-d$)
- Consider $x + \lambda d$ as a function of $\lambda \geq 0$
- Case 1: There exists j with $d_j < 0$
  - Increase λ from 0 to λ' until a new entry of $x + \lambda d$ becomes zero
  - $x + \lambda' d$ feasible, since $A(x + \lambda' d) = Ax = b$ and $x + \lambda' d \geq 0$
  - $c^T(x + \lambda' d) = c^T x + c^T \lambda' d \geq c^T x$



The Structure of Optimal Solutions

Definition

A point x is a vertex if it cannot be represented as a strict convex combination of two other points in the feasible set.

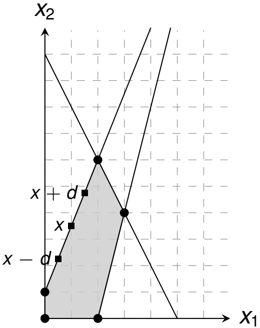
The set of feasible solutions is a convex set.

Theorem

If the slack form has an optimal solution, one of them occurs at a vertex.

Proof Sketch (informal and non-examinable):

- Rewrite LP s.t. $Ax = b$. Let x be optimal but not a vertex $\Rightarrow \exists$ vector d s.t. $x - d$ and $x + d$ are feasible
- Since $A(x+d) = b$ and $Ax = b \Rightarrow Ad = 0$
- W.l.o.g. assume $c^T d \geq 0$ (otherwise replace d by $-d$)
- Consider $x + \lambda d$ as a function of $\lambda \geq 0$
- Case 2: For all j, $d_j \geq 0$



The Structure of Optimal Solutions

Definition

A point x is a vertex if it cannot be represented as a strict convex combination of two other points in the feasible set.

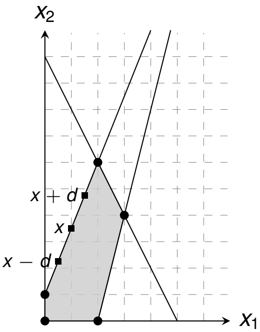
The set of feasible solutions is a convex set.

Theorem

If the slack form has an optimal solution, one of them occurs at a vertex.

Proof Sketch (informal and non-examinable):

- Rewrite LP s.t. $Ax = b$. Let x be optimal but not a vertex $\Rightarrow \exists$ vector d s.t. $x - d$ and $x + d$ are feasible
- Since $A(x + d) = b$ and $Ax = b \Rightarrow Ad = 0$
- W.l.o.g. assume $c^T d \geq 0$ (otherwise replace d by $-d$)
- Consider $x + \lambda d$ as a function of $\lambda \geq 0$
- Case 2: For all j, $d_j \geq 0$
  - $x + \lambda d$ is feasible for all $\lambda \geq 0$: $A(x + \lambda d) = b$ and $x + \lambda d \geq x \geq 0$



The Structure of Optimal Solutions

Definition

A point x is a vertex if it cannot be represented as a strict convex combination of two other points in the feasible set.

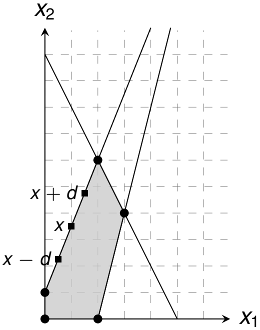
The set of feasible solutions is a convex set.

Theorem

If the slack form has an optimal solution, one of them occurs at a vertex.

Proof Sketch (informal and non-examinable):

- Rewrite LP s.t. $Ax = b$. Let x be optimal but not a vertex $\Rightarrow \exists$ vector d s.t. $x - d$ and $x + d$ are feasible
- Since $A(x + d) = b$ and $Ax = b \Rightarrow Ad = 0$
- W.l.o.g. assume $c^T d \geq 0$ (otherwise replace d by $-d$)
- Consider $x + \lambda d$ as a function of $\lambda \geq 0$
- Case 2: For all j, $d_j \geq 0$
  - $x + \lambda d$ is feasible for all $\lambda \geq 0$: $A(x + \lambda d) = b$ and $x + \lambda d \geq x \geq 0$
  - If $\lambda \to \infty$, then $c^T(x + \lambda d) \to \infty$



The Structure of Optimal Solutions

Definition

A point x is a vertex if it cannot be represented as a strict convex combination of two other points in the feasible set.

The set of feasible solutions is a convex set.

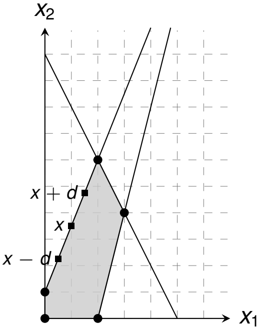
Theorem

If the slack form has an optimal solution, one of them occurs at a vertex.

Proof Sketch (informal and non-examinable):

- Rewrite LP s.t. $Ax = b$. Let x be optimal but not a vertex $\Rightarrow \exists$ vector d s.t. $x - d$ and $x + d$ are feasible
- Since $A(x + d) = b$ and $Ax = b \Rightarrow Ad = 0$
- W.l.o.g. assume $c^T d \geq 0$ (otherwise replace d by $-d$)
- Consider $x + \lambda d$ as a function of $\lambda \geq 0$
- Case 2: For all j, $d_j \geq 0$
  - $x + \lambda d$ is feasible for all $\lambda \geq 0$: $A(x + \lambda d) = b$ and $x + \lambda d \geq x \geq 0$
  - If $\lambda \to \infty$, then $c^T(x + \lambda d) \to \infty$

  $\Rightarrow$ This contradicts the assumption that there exists an optimal solution.



The Structure of Optimal Solutions

Definition

A point x is a vertex if it cannot be represented as a strict convex combination of two other points in the feasible set.

The set of feasible solutions is a convex set.

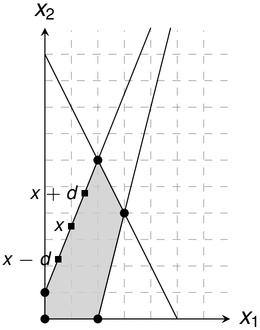
Theorem

If the slack form has an optimal solution, one of them occurs at a vertex.

Proof Sketch (informal and non-examinable):

- Rewrite LP s.t. $Ax = b$. Let x be optimal but not a vertex $\Rightarrow \exists$ vector d s.t. $x - d$ and $x + d$ are feasible
- Since $A(x + d) = b$ and $Ax = b \Rightarrow Ad = 0$
- W.l.o.g. assume $c^T d \geq 0$ (otherwise replace d by $-d$)
- Consider $x + \lambda d$ as a function of $\lambda \geq 0$
- Case 2: For all j, $d_j \geq 0$
  - $x + \lambda d$ is feasible for all $\lambda \geq 0$: $A(x + \lambda d) = b$ and $x + \lambda d \geq x \geq 0$
  - If $\lambda \to \infty$, then $c^T(x + \lambda d) \to \infty$

$\Rightarrow$ This contradicts the assumption that there exists an optimal solution. □

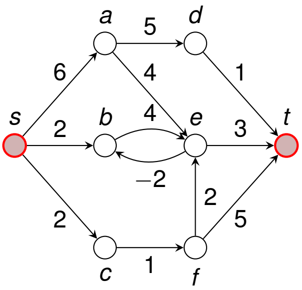


Outline

Shortest Paths

Single-Pair Shortest Path Problem

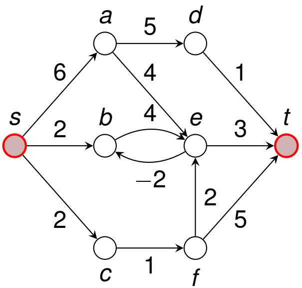
- Given: directed graph $G = (V, E)$ with edge weights $w : E \to \mathbb{R}$, pair of vertices $s, t \in V$



Shortest Paths

Single-Pair Shortest Path Problem

- Given: directed graph $G = (V, E)$ with edge weights $w : E \to \mathbb{R}$, pair of vertices $s, t \in V$
- Goal: Find a path of minimum weight from s to t in G

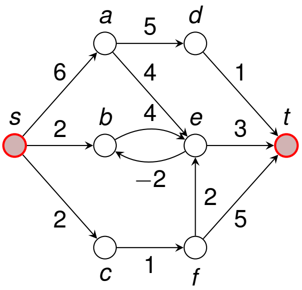


Shortest Paths

Single-Pair Shortest Path Problem

- Given: directed graph $G = (V, E)$ with edge weights $w : E \to \mathbb{R}$, pair of vertices $s, t \in V$
- Goal: Find a path of minimum weight from s to t in G

$p = (v_0 = s, v_1, \ldots, v_k = t)$ such that $w(p) = \sum_{i=1}^{k} w(v_{k-1}, v_k)$ is minimized.

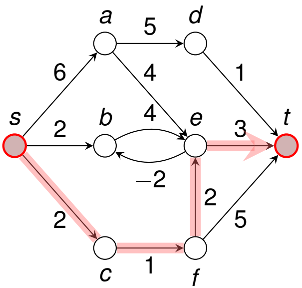


Shortest Paths

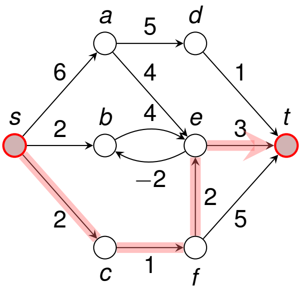
Single-Pair Shortest Path Problem

- Given: directed graph $G = (V, E)$ with edge weights $w : E \to \mathbb{R}$, pair of vertices $s, t \in V$
- Goal: Find a path of minimum weight from s to t in G

$p = (v_0 = s, v_1, \ldots, v_k = t)$ such that $w(p) = \sum_{i=1}^{k} w(v_{k-1}, v_k)$ is minimized.



Shortest Paths



Single-Pair Shortest Path Problem

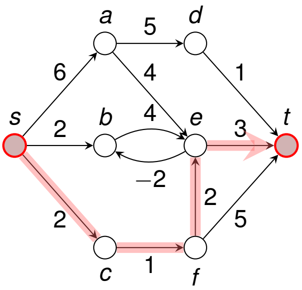
- Given: directed graph $G = (V, E)$ with edge weights $w : E \to \mathbb{R}$, pair of vertices $s, t \in V$
- Goal: Find a path of minimum weight from s to t in G

$p = (v_0 = s, v_1, \ldots, v_k = t)$ such that $w(p) = \sum_{i=1}^{k} w(v_{k-1}, v_k)$ is minimized.

Shortest Paths as LP

subject to

Shortest Paths



Single-Pair Shortest Path Problem

- Given: directed graph $G = (V, E)$ with edge weights $w : E \to \mathbb{R}$, pair of vertices $s, t \in V$
- Goal: Find a path of minimum weight from s to t in G

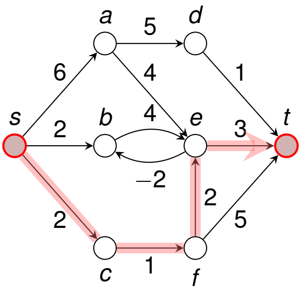
$p = (v_0 = s, v_1, \ldots, v_k = t)$ such that $w(p) = \sum_{i=1}^{k} w(v_{k-1}, v_k)$ is minimized.

Shortest Paths as LP

subject to

$$\begin{aligned} d_v &\leq d_u + w(u,v) \quad \text{for each edge } (u,v) \in E, \\ d_s &= 0. \end{aligned}$$

Shortest Paths



Single-Pair Shortest Path Problem

- Given: directed graph $G = (V, E)$ with edge weights $w : E \to \mathbb{R}$, pair of vertices $s, t \in V$
- Goal: Find a path of minimum weight from s to t in G

$p = (v_0 = s, v_1, \ldots, v_k = t)$ such that $w(p) = \sum_{i=1}^{k} w(v_{k-1}, v_k)$ is minimized.

Shortest Paths as LP

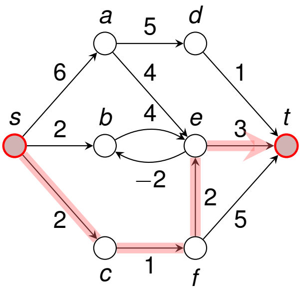
$$\begin{array}{llllll} \text{maximize} & d_t & & & & \\ \text{subject to} & & & & & \\ & d_v & \leq & d_u & + \; w(u,v) & \text{for each edge } (u,v) \in E, \\ & d_s & = & 0. & & \end{array}$$

Shortest Paths

Single-Pair Shortest Path Problem

- Given: directed graph $G = (V, E)$ with edge weights $w : E \to \mathbb{R}$, pair of vertices $s, t \in V$
- Goal: Find a path of minimum weight from s to t in G

$p = (v_0 = s, v_1, \ldots, v_k = t)$ such that $w(p) = \sum_{i=1}^{k} w(v_{k-1}, v_k)$ is minimized.



Shortest Paths as LP

$$
\begin{array}{llll}
\text{maximize} & d_t & & \\
\text{subject to} & & & \\
& d_v \leq d_u + w(u,v) & & \text{for each edge } (u,v) \in E, \\
& d_s = 0. & &
\end{array}
$$

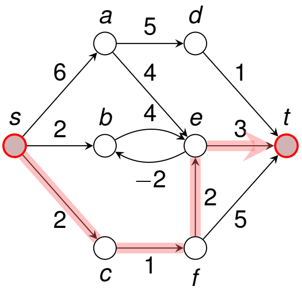
this is a maximization problem!

Shortest Paths

Single-Pair Shortest Path Problem

- Given: directed graph $G = (V, E)$ with edge weights $w : E \to \mathbb{R}$, pair of vertices $s, t \in V$
- Goal: Find a path of minimum weight from s to t in G

$p = (v_0 = s, v_1, \ldots, v_k = t)$ such that $w(p) = \sum_{i=1}^{k} w(v_{k-1}, v_k)$ is minimized.



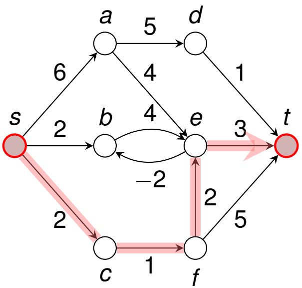
Shortest Paths as LP

$$\begin{array}{llllll} \text{maximize} & d_t & & & & \\ \text{subject to} & & & & & \\ & d_v & \leq & d_u & + & w(u,v) \quad \text{for each edge } (u,v) \in E, \\ & d_s & = & 0. & & \end{array}$$

Recall: When BELLMAN-FORD terminates, all these inequalities are satisfied.

this is a maximization problem!

Shortest Paths



Single-Pair Shortest Path Problem

- Given: directed graph $G = (V, E)$ with edge weights $w : E \to \mathbb{R}$, pair of vertices $s, t \in V$
- Goal: Find a path of minimum weight from s to t in G

$p = (v_0 = s, v_1, \ldots, v_k = t)$ such that $w(p) = \sum_{i=1}^{k} w(v_{k-1}, v_k)$ is minimized.

Shortest Paths as LP

maximize $\quad d_t$

subject to

$$d_v \leq d_u + w(u, v) \quad \text{for each edge } (u, v) \in E,$$
$$d_s = 0.$$

this is a maximization problem!

Recall: When BELLMAN-FORD terminates, all these inequalities are satisfied.

Solution $\overline{d}$ satisfies $\overline{d}_v = \min_{u:\,(u,v)\in E} \left\{ \overline{d}_u + w(u, v) \right\}$

Maximum Flow

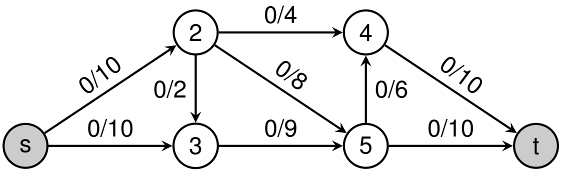
Maximum Flow Problem

- Given: directed graph $G = (V, E)$ with edge capacities $c : E \to \mathbb{R}^+$, pair of vertices $s, t \in V$

Maximum Flow

Maximum Flow Problem

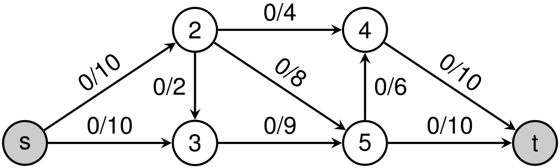
- Given: directed graph $G = (V, E)$ with edge capacities $c : E \to \mathbb{R}^+$, pair of vertices $s, t \in V$



Maximum Flow

Maximum Flow Problem

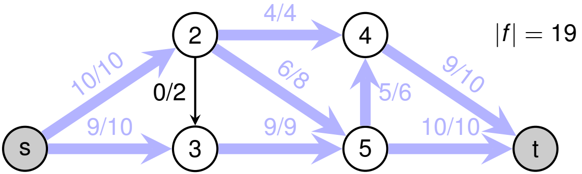
- Given: directed graph $G = (V, E)$ with edge capacities $c : E \to \mathbb{R}^+$, pair of vertices $s, t \in V$
- Goal: Find a maximum flow $f : V \times V \to \mathbb{R}$ from s to t which satisfies the capacity constraints and flow conservation



Maximum Flow

Maximum Flow Problem

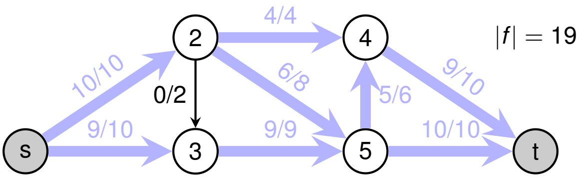
- Given: directed graph $G = (V, E)$ with edge capacities $c : E \to \mathbb{R}^+$, pair of vertices $s, t \in V$
- Goal: Find a maximum flow $f : V \times V \to \mathbb{R}$ from s to t which satisfies the capacity constraints and flow conservation



Maximum Flow

Maximum Flow Problem

- Given: directed graph $G = (V, E)$ with edge capacities $c : E \to \mathbb{R}^+$, pair of vertices $s, t \in V$
- Goal: Find a maximum flow $f : V \times V \to \mathbb{R}$ from s to t which satisfies the capacity constraints and flow conservation



Maximum Flow as LP

$$\begin{array}{lrcll}
\text{maximize} & \sum_{v \in V} f_{sv} & - & \sum_{v \in V} f_{vs} & \\
\text{subject to} & & & & \\
& f_{uv} & \leq & c(u, v) & \text{for each } u, v \in V, \\
& \sum_{v \in V} f_{vu} & = & \sum_{v \in V} f_{uv} & \text{for each } u \in V \setminus \{s, t\}, \\
& f_{uv} & \geq & 0 & \text{for each } u, v \in V.
\end{array}$$

Minimum-Cost Flow

Extension of the Maximum Flow Problem

Minimum-Cost-Flow Problem

Minimum-Cost Flow

Extension of the Maximum Flow Problem

Minimum-Cost-Flow Problem

- Given: directed graph $G = (V, E)$ with capacities $c : E \to \mathbb{R}^+$, pair of vertices $s, t \in V$, cost function $a : E \to \mathbb{R}^+$, flow demand of d units

Minimum-Cost Flow

Extension of the Maximum Flow Problem

Minimum-Cost-Flow Problem

- Given: directed graph $G = (V, E)$ with capacities $c : E \to \mathbb{R}^+$, pair of vertices $s, t \in V$, cost function $a : E \to \mathbb{R}^+$, flow demand of d units
- Goal: Find a flow $f : V \times V \to \mathbb{R}$ from s to t with $|f| = d$ while minimising the total cost $\sum_{(u,v)\in E} a(u, v) f_{uv}$ incurrred by the flow.

Minimum-Cost Flow

Extension of the Maximum Flow Problem

Minimum-Cost-Flow Problem

- Given: directed graph $G = (V, E)$ with capacities $c : E \to \mathbb{R}^+$, pair of vertices $s, t \in V$, cost function $a : E \to \mathbb{R}^+$, flow demand of d units
- Goal: Find a flow $f : V \times V \to \mathbb{R}$ from s to t with $|f| = d$ while minimising the total cost $\sum_{(u,v)\in E} a(u,v) f_{uv}$ incurrred by the flow.

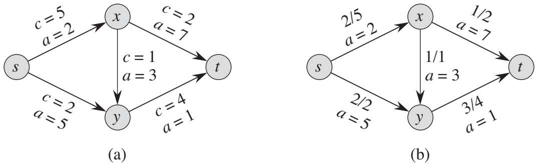


Figure 29.3 **(a)** An example of a minimum-cost-flow problem. We denote the capacities by c and the costs by a. Vertex s is the source and vertex t is the sink, and we wish to send 4 units of flow from s to t. **(b)** A solution to the minimum-cost flow problem in which 4 units of flow are sent from s to t. For each edge, the flow and capacity are written as flow/capacity.

Minimum-Cost Flow

Extension of the Maximum Flow Problem

Minimum-Cost-Flow Problem

- Given: directed graph $G = (V, E)$ with capacities $c : E \to \mathbb{R}^+$, pair of vertices $s, t \in V$, cost function $a : E \to \mathbb{R}^+$, flow demand of d units
- Goal: Find a flow $f : V \times V \to \mathbb{R}$ from s to t with $|f| = d$ while minimising the total cost $\sum_{(u,v)\in E} a(u,v) f_{uv}$ incurrred by the flow.

Optimal Solution with total cost:
$\sum_{(u,v)\in E} a(u,v) f_{uv} = (2\cdot 2)+(5\cdot 2)+(3\cdot 1)+(7\cdot 1)+(1\cdot 3) = 27$

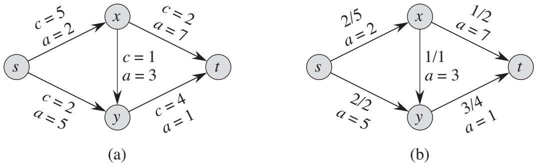


Figure 29.3 **(a)** An example of a minimum-cost-flow problem. We denote the capacities by c and the costs by a. Vertex s is the source and vertex t is the sink, and we wish to send 4 units of flow from s to t. **(b)** A solution to the minimum-cost flow problem in which 4 units of flow are sent from s to t. For each edge, the flow and capacity are written as flow/capacity.

Minimum-Cost Flow as a LP

Minimum Cost Flow as LP

minimize $\sum_{(u,v)\in E} a(u,v) f_{uv}$

subject to

$$
\begin{aligned}
f_{uv} &\leq c(u,v) && \text{for each } u, v \in V, \\
\sum_{v\in V} f_{vu} - \sum_{v\in V} f_{uv} &= 0 && \text{for each } u \in V \setminus \{s,t\}, \\
\sum_{v\in V} f_{sv} - \sum_{v\in V} f_{vs} &= d\ , && \\
f_{uv} &\geq 0 && \text{for each } u, v \in V.
\end{aligned}
$$

Minimum-Cost Flow as a LP

Minimum Cost Flow as LP

minimize $\sum_{(u,v)\in E} a(u,v) f_{uv}$

subject to

$$\begin{aligned} f_{uv} &\leq c(u,v) && \text{for each } u,v \in V, \\ \sum_{v\in V} f_{vu} - \sum_{v\in V} f_{uv} &= 0 && \text{for each } u \in V \setminus \{s,t\}, \\ \sum_{v\in V} f_{sv} - \sum_{v\in V} f_{vs} &= d\,, && \\ f_{uv} &\geq 0 && \text{for each } u,v \in V. \end{aligned}$$

Real power of Linear Programming comes from the ability to solve **new problems**!

Outline

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: 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

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

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

Extended Example: Conversion into Slack Form

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

Conversion into slack form

Extended Example: Conversion into Slack Form

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

Conversion into slack form

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

Extended Example: Iteration 1

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

Extended Example: Iteration 1

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

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

Extended Example: Iteration 1

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

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

This basic solution is **feasible**

Extended Example: Iteration 1

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

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

This basic solution is **feasible**

Objective value is 0.

Extended Example: Iteration 1

Increasing the value of x_1 would increase the objective value.

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

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

This basic solution is **feasible**

Objective value is 0.

Extended Example: Iteration 1

Increasing the value of x_1 would increase the objective value.

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

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.

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

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.

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

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.

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

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

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

Extended Example: Iteration 2

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

Basic solution: $(\overline{x_1}, \overline{x_2}, \dots, \overline{x_6}) = (9, 0, 0, 21, 6, 0)$ with objective value 27

Extended Example: Iteration 2

Increasing the value of x_3 would increase the objective value.

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

Basic solution: $(\overline{x_1}, \overline{x_2}, \ldots, \overline{x_6}) = (9, 0, 0, 21, 6, 0)$ with objective value 27

Extended Example: Iteration 2

Increasing the value of x_3 would increase the objective value.

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

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.

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

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.

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

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.

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

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: $(\overline{x_1}, \overline{x_2}, \ldots, \overline{x_6}) = (\frac{33}{4}, 0, \frac{3}{2}, \frac{69}{4}, 0, 0)$ 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: $(\overline{x_1}, \overline{x_2}, \ldots, \overline{x_6}) = (\frac{33}{4}, 0, \frac{3}{2}, \frac{69}{4}, 0, 0)$ 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.

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

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

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

Extended Example: Iteration 4

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

Basic solution: $(\overline{x_1}, \overline{x_2}, \ldots, \overline{x_6}) = (8, 4, 0, 18, 0, 0)$ with objective value 28

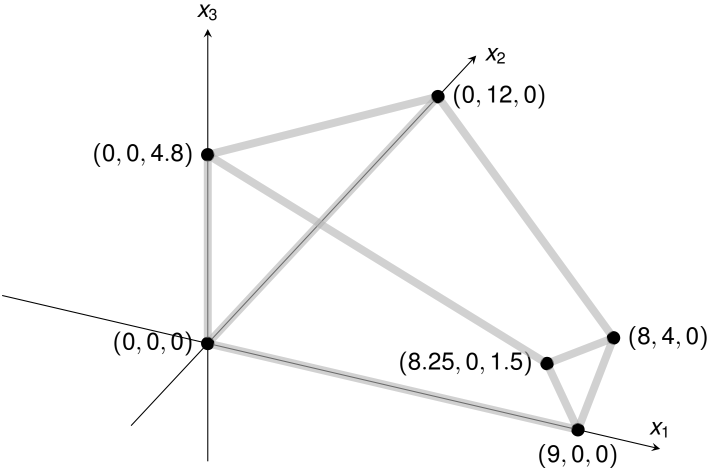
Extended Example: Iteration 4

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

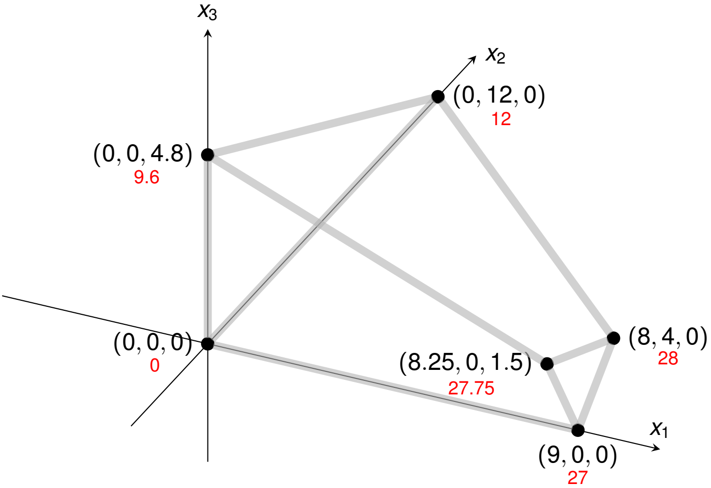
$$
\begin{aligned}
z &= 28 - \frac{x_3}{6} - \frac{x_5}{6} - \frac{2x_6}{3} \\
x_1 &= 8 + \frac{x_3}{6} + \frac{x_5}{6} - \frac{x_6}{3} \\
x_2 &= 4 - \frac{8x_3}{3} - \frac{2x_5}{3} + \frac{x_6}{3} \\
x_4 &= 18 - \frac{x_3}{2} + \frac{x_5}{2}
\end{aligned}
$$

Basic solution: $(\overline{x_1}, \overline{x_2}, \ldots, \overline{x_6}) = (8, 4, 0, 18, 0, 0)$ with objective value 28

Extended Example: Visualization of SIMPLEX

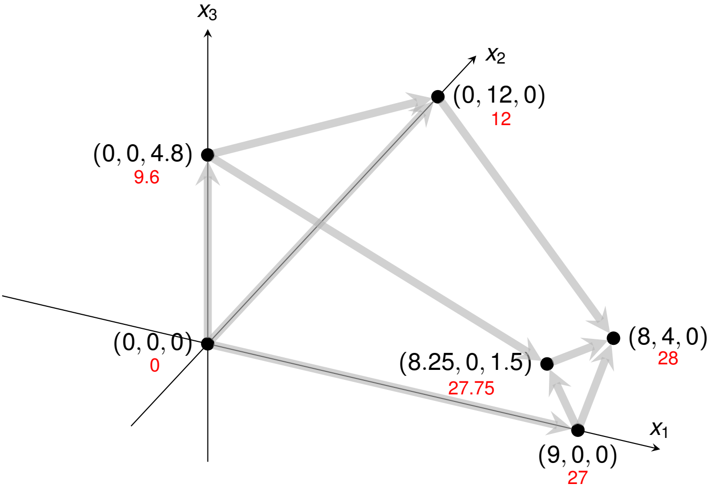


Extended Example: Visualization of SIMPLEX



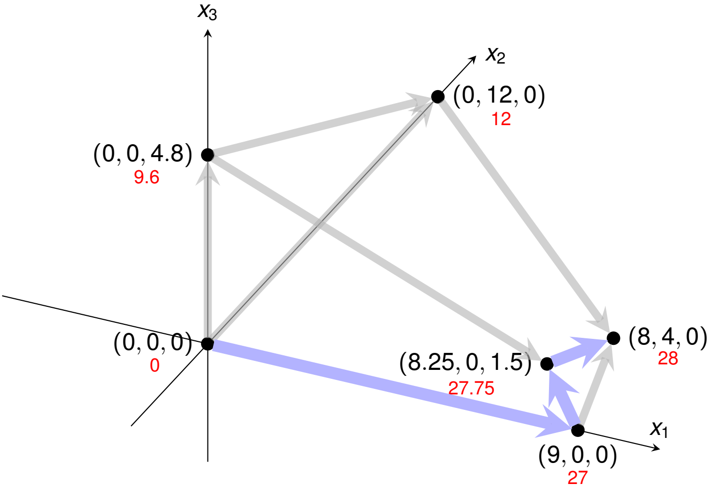
Exercise: How many basic solutions (including non-feasible ones) are there?

Extended Example: Visualization of SIMPLEX



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

Extended Example: Visualization of SIMPLEX



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

Extended Example: Alternative Runs (1/2)

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

Extended Example: Alternative Runs (1/2)

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

Switch roles of x_2 and x_5

Extended Example: Alternative Runs (1/2)

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

Switch roles of x_2 and x_5

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

Extended Example: Alternative Runs (1/2)

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

↓ Switch roles of x_2 and x_5

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

↓ Switch roles of x_1 and x_6

Extended Example: Alternative Runs (1/2)

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

Switch roles of x_2 and x_5

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

Switch roles of x_1 and x_6

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

Extended Example: Alternative Runs (2/2)

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

Extended Example: Alternative Runs (2/2)

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

↓ Switch roles of x_3 and x_5

Extended Example: Alternative Runs (2/2)

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

Switch roles of x_3 and x_5

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

Extended Example: Alternative Runs (2/2)

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

Switch roles of x_3 and x_5

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

Switch roles of x_1 and x_6

Extended Example: Alternative Runs (2/2)

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

Switch roles of x_3 and x_5

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

Switch roles of x_1 and x_6

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

Extended Example: Alternative Runs (2/2)

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

Switch roles of x_3 and x_5

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

Switch roles of x_1 and x_6

Switch roles of x_2 and x_3

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

Extended Example: Alternative Runs (2/2)

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

Switch roles of x_3 and x_5

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

Switch roles of x_1 and x_6

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

Switch roles of x_2 and x_3

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

The Pivot Step Formally

```
PIVOT(N, B, A, b, c, v, l, e)
// Compute the coefficients of the equation for new basic variable x_e.
let Â be a new m × n matrix
b̂_e = b_l / a_le
for each j ∈ N − {e}
    â_ej = a_lj / a_le
â_el = 1 / a_le
// Compute the coefficients of the remaining constraints.
for each i ∈ B − {l}
    b̂_i = b_i − a_ie b̂_e
    for each j ∈ N − {e}
        â_ij = a_ij − a_ie â_ej
    â_il = −a_ie â_el
// Compute the objective function.
v̂ = v + c_e b̂_e
for each j ∈ N − {e}
    ĉ_j = c_j − c_e â_ej
ĉ_l = −c_e â_el
// Compute new sets of basic and nonbasic variables.
N̂ = N − {e} ∪ {l}
B̂ = B − {l} ∪ {e}
return (N̂, B̂, Â, b̂, ĉ, v̂)
```

The Pivot Step Formally

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

// Compute the coefficients of the equation for new basic variable x_e.
let $\widehat{A}$ be a new $m \times n$ matrix
$\widehat{b}_e = b_l / a_{le}$
for each $j \in N - \{e\}$
    $\widehat{a}_{ej} = a_{lj} / a_{le}$
$\widehat{a}_{el} = 1 / a_{le}$
// Compute the coefficients of the remaining constraints.
for each $i \in B - \{l\}$
    $\widehat{b}_i = b_i - a_{ie} \widehat{b}_e$
    for each $j \in N - \{e\}$
        $\widehat{a}_{ij} = a_{ij} - a_{ie} \widehat{a}_{ej}$
    $\widehat{a}_{il} = -a_{ie} \widehat{a}_{el}$
// Compute the objective function.
$\widehat{v} = v + c_e \widehat{b}_e$
for each $j \in N - \{e\}$
    $\widehat{c}_j = c_j - c_e \widehat{a}_{ej}$
$\widehat{c}_l = -c_e \widehat{a}_{el}$
// Compute new sets of basic and nonbasic variables.
$\widehat{N} = N - \{e\} \cup \{l\}$
$\widehat{B} = B - \{l\} \cup \{e\}$
return $(\widehat{N}, \widehat{B}, \widehat{A}, \widehat{b}, \widehat{c}, \widehat{v})$

Rewrite "tight" equation for enterring variable x_e.

The Pivot Step Formally

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

// Compute the coefficients of the equation for new basic variable x_e.
let $\widehat{A}$ be a new $m \times n$ matrix
$\widehat{b}_e = b_l / a_{le}$
for each $j \in N - \{e\}$
    $\widehat{a}_{ej} = a_{lj} / a_{le}$
$\widehat{a}_{el} = 1/a_{le}$
// Compute the coefficients of the remaining constraints.
for each $i \in B - \{l\}$
    $\widehat{b}_i = b_i - a_{ie}\widehat{b}_e$
    for each $j \in N - \{e\}$
        $\widehat{a}_{ij} = a_{ij} - a_{ie}\widehat{a}_{ej}$
    $\widehat{a}_{il} = -a_{ie}\widehat{a}_{el}$
// Compute the objective function.
$\widehat{v} = v + c_e \widehat{b}_e$
for each $j \in N - \{e\}$
    $\widehat{c}_j = c_j - c_e \widehat{a}_{ej}$
$\widehat{c}_l = -c_e \widehat{a}_{el}$
// Compute new sets of basic and nonbasic variables.
$\widehat{N} = N - \{e\} \cup \{l\}$
$\widehat{B} = B - \{l\} \cup \{e\}$
return $(\widehat{N}, \widehat{B}, \widehat{A}, \widehat{b}, \widehat{c}, \widehat{v})$

Rewrite “tight” equation for enterring variable x_e.

Substituting x_e into other equations.

The Pivot Step Formally

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

// Compute the coefficients of the equation for new basic variable x_e.
let $\widehat{A}$ be a new $m \times n$ matrix
$\widehat{b}_e = b_l / a_{le}$
for each $j \in N - \{e\}$
    $\widehat{a}_{ej} = a_{lj} / a_{le}$
$\widehat{a}_{el} = 1 / a_{le}$
// Compute the coefficients of the remaining constraints.
for each $i \in B - \{l\}$
    $\widehat{b}_i = b_i - a_{ie}\widehat{b}_e$
    for each $j \in N - \{e\}$
        $\widehat{a}_{ij} = a_{ij} - a_{ie}\widehat{a}_{ej}$
    $\widehat{a}_{il} = -a_{ie}\widehat{a}_{el}$
// Compute the objective function.
$\widehat{v} = v + c_e\widehat{b}_e$
for each $j \in N - \{e\}$
    $\widehat{c}_j = c_j - c_e\widehat{a}_{ej}$
$\widehat{c}_l = -c_e\widehat{a}_{el}$
// Compute new sets of basic and nonbasic variables.
$\widehat{N} = N - \{e\} \cup \{l\}$
$\widehat{B} = B - \{l\} \cup \{e\}$
return $(\widehat{N}, \widehat{B}, \widehat{A}, \widehat{b}, \widehat{c}, \widehat{v})$

Rewrite "tight" equation for enterring 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)

// Compute the coefficients of the equation for new basic variable x_e.
let $\widehat{A}$ be a new $m \times n$ matrix
$\widehat{b}_e = b_l / a_{le}$
for each $j \in N - \{e\}$
    $\widehat{a}_{ej} = a_{lj} / a_{le}$
$\widehat{a}_{el} = 1/a_{le}$
// Compute the coefficients of the remaining constraints.
for each $i \in B - \{l\}$
    $\widehat{b}_i = b_i - a_{ie}\widehat{b}_e$
    for each $j \in N - \{e\}$
        $\widehat{a}_{ij} = a_{ij} - a_{ie}\widehat{a}_{ej}$
    $\widehat{a}_{il} = -a_{ie}\widehat{a}_{el}$
// Compute the objective function.
$\widehat{v} = v + c_e\widehat{b}_e$
for each $j \in N - \{e\}$
    $\widehat{c}_j = c_j - c_e\widehat{a}_{ej}$
$\widehat{c}_l = -c_e\widehat{a}_{el}$
// Compute new sets of basic and nonbasic variables.
$\widehat{N} = N - \{e\} \cup \{l\}$
$\widehat{B} = B - \{l\} \cup \{e\}$
return $(\widehat{N}, \widehat{B}, \widehat{A}, \widehat{b}, \widehat{c}, \widehat{v})$

Rewrite "tight" equation for enterring 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)

// Compute the coefficients of the equation for new basic variable x_e.
let $\widehat{A}$ be a new $m \times n$ matrix
$\widehat{b}_e = b_l / a_{le}$
for each $j \in N - \{e\}$
    $\widehat{a}_{ej} = a_{lj} / a_{le}$
$\widehat{a}_{el} = 1 / a_{le}$
// Compute the coefficients of the remaining constraints.
for each $i \in B - \{l\}$
    $\widehat{b}_i = b_i - a_{ie} \widehat{b}_e$
    for each $j \in N - \{e\}$
        $\widehat{a}_{ij} = a_{ij} - a_{ie} \widehat{a}_{ej}$
    $\widehat{a}_{il} = -a_{ie} \widehat{a}_{el}$
// Compute the objective function.
$\widehat{v} = v + c_e \widehat{b}_e$
for each $j \in N - \{e\}$
    $\widehat{c}_j = c_j - c_e \widehat{a}_{ej}$
$\widehat{c}_l = -c_e \widehat{a}_{el}$
// Compute new sets of basic and nonbasic variables.
$\widehat{N} = N - \{e\} \cup \{l\}$
$\widehat{B} = B - \{l\} \cup \{e\}$
return $(\widehat{N}, \widehat{B}, \widehat{A}, \widehat{b}, \widehat{c}, \widehat{v})$

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

Rewrite "tight" equation for enterring variable x_e.

Substituting x_e into other equations.

Substituting x_e into objective function.

Update non-basic and basic variables

Effect of the Pivot Step

Lemma 29.1

Consider a call to PIVOT(N, B, A, b, c, v, l, e) in which $a_{le} \neq 0$. Let the values returned from the call be $(\widehat{N}, \widehat{B}, \widehat{A}, \widehat{b}, \widehat{c}, \widehat{v})$, and let $\overline{x}$ denote the basic solution after the call. Then

Effect of the Pivot Step

Lemma 29.1

Consider a call to PIVOT(N, B, A, b, c, v, l, e) in which $a_{le} \neq 0$. Let the values returned from the call be $(\widehat{N}, \widehat{B}, \widehat{A}, \widehat{b}, \widehat{c}, \widehat{v})$, and let $\overline{x}$ denote the basic solution after the call. Then

1. $\overline{x}_j = 0$ for each $j \in \widehat{N}$.
2. $\overline{x}_e = b_l / a_{le}$.
3. $\overline{x}_i = b_i - a_{ie}\widehat{b}_e$ for each $i \in \widehat{B} \setminus \{e\}$.

Effect of the Pivot Step

Lemma 29.1

Consider a call to PIVOT(N, B, A, b, c, v, l, e) in which $a_{le} \neq 0$. Let the values returned from the call be $(\widehat{N}, \widehat{B}, \widehat{A}, \widehat{b}, \widehat{c}, \widehat{v})$, and let $\overline{x}$ denote the basic solution after the call. Then

1. $\overline{x}_j = 0$ for each $j \in \widehat{N}$.
2. $\overline{x}_e = b_l / a_{le}$.
3. $\overline{x}_i = b_i - a_{ie}\widehat{b}_e$ for each $i \in \widehat{B} \setminus \{e\}$.

Proof:

Effect of the Pivot Step

Lemma 29.1

Consider a call to PIVOT(N, B, A, b, c, v, l, e) in which $a_{le} \neq 0$. Let the values returned from the call be $(\widehat{N}, \widehat{B}, \widehat{A}, \widehat{b}, \widehat{c}, \widehat{v})$, and let $\overline{x}$ denote the basic solution after the call. Then

1. $\overline{x}_j = 0$ for each $j \in \widehat{N}$.
2. $\overline{x}_e = b_l / a_{le}$.
3. $\overline{x}_i = b_i - a_{ie}\widehat{b}_e$ for each $i \in \widehat{B} \setminus \{e\}$.

Proof:

1. holds since the basic solution always sets all non-basic variables to zero.
2. When we set each non-basic variable to 0 in a constraint
$$x_i = \widehat{b}_i - \sum_{j \in \widehat{N}} \widehat{a}_{ij} x_j,$$
we have $\overline{x}_i = \widehat{b}_i$ for each $i \in \widehat{B}$. Hence $\overline{x}_e = \widehat{b}_e = b_l / a_{le}$.
3. After the substituting in the other constraints, we have
$$\overline{x}_i = \widehat{b}_i = b_i - a_{ie}\widehat{b}_e.$$

Effect of the Pivot Step

Lemma 29.1

Consider a call to PIVOT(N, B, A, b, c, v, l, e) in which $a_{le} \neq 0$. Let the values returned from the call be $(\widehat{N}, \widehat{B}, \widehat{A}, \widehat{b}, \widehat{c}, \widehat{v})$, and let $\overline{x}$ denote the basic solution after the call. Then

1. $\overline{x}_j = 0$ for each $j \in \widehat{N}$.
2. $\overline{x}_e = b_l / a_{le}$.
3. $\overline{x}_i = b_i - a_{ie}\widehat{b}_e$ for each $i \in \widehat{B} \setminus \{e\}$.

Proof:

1. holds since the basic solution always sets all non-basic variables to zero.
2. When we set each non-basic variable to 0 in a constraint
$$x_i = \widehat{b}_i - \sum_{j \in \widehat{N}} \widehat{a}_{ij} x_j,$$
we have $\overline{x}_i = \widehat{b}_i$ for each $i \in \widehat{B}$. Hence $\overline{x}_e = \widehat{b}_e = b_l / a_{le}$.
3. After the substituting in the other constraints, we have
$$\overline{x}_i = \widehat{b}_i = b_i - a_{ie}\widehat{b}_e. \qquad \square$$

Formalizing the Simplex Algorithm: Questions

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?

Formalizing the Simplex Algorithm: Questions

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)
(N, B, A, b, c, v) = INITIALIZE-SIMPLEX(A, b, c)
let Δ be a new vector of length m
while some index j ∈ N has c_j > 0
    choose an index e ∈ N for which c_e > 0
    for each index i ∈ B
        if a_ie > 0
            Δ_i = b_i / a_ie
        else Δ_i = ∞
    choose an index l ∈ B that minimizes Δ_i
    if Δ_l == ∞
        return "unbounded"
    else (N, B, A, b, c, v) = PIVOT(N, B, A, b, c, v, l, e)
for i = 1 to n
    if i ∈ B
        x̄_i = b_i
    else x̄_i = 0
return (x̄_1, x̄_2, ..., x̄_n)
```

The formal procedure SIMPLEX

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

```
SIMPLEX(A, b, c)
(N, B, A, b, c, ν) = INITIALIZE-SIMPLEX(A, b, c)
let Δ be a new vector of length m
while some index j ∈ N has c_j > 0
    choose an index e ∈ N for which c_e > 0
    for each index i ∈ B
        if a_ie > 0
            Δ_i = b_i/a_ie
        else Δ_i = ∞
    choose an index l ∈ B that minimizes Δ_i
    if Δ_l == ∞
        return "unbounded"
    else (N, B, A, b, c, ν) = PIVOT(N, B, A, b, c, ν, l, e)
for i = 1 to n
    if i ∈ B
        x̄_i = b_i
    else x̄_i = 0
return (x̄_1, x̄_2, ..., x̄_n)
```

The formal procedure SIMPLEX

SIMPLEX(A, b, c)

$(N, B, A, b, c, \nu) = $ INITIALIZE-SIMPLEX(A, b, c)

let Δ be a new vector of length m

while some index $j \in N$ has $c_j > 0$

  choose an index $e \in N$ for which $c_e > 0$

  for each index $i \in B$

    if $a_{ie} > 0$

      $\Delta_i = b_i / a_{ie}$

    else $\Delta_i = \infty$

  choose an index $l \in B$ that minimizes Δ_i

  if $\Delta_l == \infty$

    return "unbounded"

  else $(N, B, A, b, c, \nu) = $ PIVOT$(N, B, A, b, c, \nu, l, e)$

for $i = 1$ **to** n

  if $i \in B$

    $\bar{x}_i = b_i$

  else $\bar{x}_i = 0$

return $(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$

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

The formal procedure SIMPLEX

```
SIMPLEX(A, b, c)
(N, B, A, b, c, v) = INITIALIZE-SIMPLEX(A, b, c)
let Δ be a new vector of length m
while some index j ∈ N has c_j > 0
    choose an index e ∈ N for which c_e > 0
    for each index i ∈ B
        if a_ie > 0
            Δ_i = b_i / a_ie
        else Δ_i = ∞
    choose an index l ∈ B that minimizes Δ_i
    if Δ_l == ∞
        return "unbounded"
    else (N, B, A, b, c, v) = PIVOT(N, B, A, b, c, v, l, e)
for i = 1 to n
    if i ∈ B
        x̄_i = b_i
    else x̄_i = 0
return (x̄_1, x̄_2, ..., x̄_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 Δ 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 Δ_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** $(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$

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

Main Loop:

- terminates if all coefficients in objective function are negative
- Line 4 picks enterring variable x_e with negative 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, ν) = INITIALIZE-SIMPLEX(A, b, c)
2   let Δ be a new vector of length m
3   while some index j ∈ N has c_j > 0
4       choose an index e ∈ N for which c_e > 0
5       for each index i ∈ B
6           if a_ie > 0
7               Δ_i = b_i / a_ie
8           else Δ_i = ∞
9       choose an index l ∈ B that minimizes Δ_i
10      if Δ_l == ∞
11          return "unbounded"
12      else (N, B, A, b, c, ν) = PIVOT(N, B, A, b, c, ν, l, e)
13  for i = 1 to n
14      if i ∈ B
15          x̄_i = b_i
16      else x̄_i = 0
17  return (x̄_1, x̄_2, ..., x̄_n)
```

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

Main Loop:

- terminates if all coefficients in objective function are negative
- Line 4 picks enterring variable x_e with negative 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

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

```
SIMPLEX(A, b, c)
1   (N, B, A, b, c, v) = INITIALIZE-SIMPLEX(A, b, c)
2   let Δ be a new vector of length m
3   while some index j ∈ N has c_j > 0
4       choose an index e ∈ N for which c_e > 0
5       for each index i ∈ B
6           if a_ie > 0
7               Δ_i = b_i/a_ie
8           else Δ_i = ∞
9       choose an index l ∈ B that minimizes Δ_i
10      if Δ_l == ∞
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 ∈ B
15          x̄_i = b_i
16      else x̄_i = 0
17  return (x̄_1, x̄_2, ..., x̄_n)
```

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 $(N, B, A, b, c, \nu) =$ INITIALIZE-SIMPLEX(A, b, c)
2 let Δ 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 Δ_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, ν) = INITIALIZE-SIMPLEX(A, b, c)
2   let Δ be a new vector of length m
3   while some index j ∈ N has c_j > 0
4       choose an index e ∈ N for which c_e > 0
5       for each index i ∈ B
6           if a_ie > 0
7               Δ_i = b_i/a_ie
8           else Δ_i = ∞
9       choose an index l ∈ B that minimizes Δ_i
10      if Δ_l == ∞
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.

The formal procedure SIMPLEX

```
SIMPLEX(A, b, c)
1  (N, B, A, b, c, v) = INITIALIZE-SIMPLEX(A, b, c)
2  let Δ be a new vector of length m
3  while some index j ∈ N has c_j > 0
4      choose an index e ∈ N for which c_e > 0
5      for each index i ∈ B
6          if a_ie > 0
7              Δ_i = b_i / a_ie
8          else Δ_i = ∞
9      choose an index l ∈ B that minimizes Δ_i
10     if Δ_l == ∞
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.

The formal procedure SIMPLEX

```
SIMPLEX(A, b, c)
1   (N, B, A, b, c, v) = INITIALIZE-SIMPLEX(A, b, c)
2   let Δ be a new vector of length m
3   while some index j ∈ N has c_j > 0
4       choose an index e ∈ N for which c_e > 0
5       for each index i ∈ B
6           if a_ie > 0
7               Δ_i = b_i / a_ie
8           else Δ_i = ∞
9       choose an index l ∈ B that minimizes Δ_i
10      if Δ_l == ∞
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.

Termination

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

Termination

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

$$\begin{aligned} z &= x_1 + x_2 + x_3 \\ x_4 &= 8 - x_1 - x_2 \\ x_5 &= x_2 - x_3 \end{aligned}$$

Termination

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

$$\begin{aligned} z &= && x_1 + x_2 + x_3 \\ x_4 &= 8 &-& x_1 - x_2 \\ x_5 &= && \phantom{x_1 +{}} x_2 - x_3 \end{aligned}$$

Pivot with x_1 entering and x_4 leaving

Termination

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

$$\begin{aligned} z &= x_1 + x_2 + x_3 \\ x_4 &= 8 - x_1 - x_2 \\ x_5 &= x_2 - x_3 \end{aligned}$$

Pivot with x_1 entering and x_4 leaving

$$\begin{aligned} z &= 8 + x_3 - x_4 \\ x_1 &= 8 - x_2 - x_4 \\ x_5 &= x_2 - x_3 \end{aligned}$$

Termination

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

$$\begin{aligned} z &= x_1 + x_2 + x_3 \\ x_4 &= 8 - x_1 - x_2 \\ x_5 &= x_2 - x_3 \end{aligned}$$

Pivot with x_1 entering and x_4 leaving

$$\begin{aligned} z &= 8 + x_3 - x_4 \\ x_1 &= 8 - x_2 - x_4 \\ x_5 &= x_2 - x_3 \end{aligned}$$

Pivot with x_3 entering and x_5 leaving

Termination

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

$$\begin{aligned} z &= x_1 + x_2 + x_3 \\ x_4 &= 8 - x_1 - x_2 \\ x_5 &= x_2 - x_3 \end{aligned}$$

Pivot with x_1 entering and x_4 leaving

$$\begin{aligned} z &= 8 + x_3 - x_4 \\ x_1 &= 8 - x_2 - x_4 \\ x_5 &= x_2 - x_3 \end{aligned}$$

Pivot with x_3 entering and x_5 leaving

$$\begin{aligned} z &= 8 + x_2 - x_4 - x_5 \\ x_1 &= 8 - x_2 - x_4 \\ x_3 &= x_2 - x_5 \end{aligned}$$

Termination

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

$$
\begin{aligned}
z &= x_1 + x_2 + x_3 \\
x_4 &= 8 - x_1 - x_2 \\
x_5 &= x_2 - x_3
\end{aligned}
$$

Pivot with x_1 entering and x_4 leaving

$$
\begin{aligned}
z &= 8 + x_3 - x_4 \\
x_1 &= 8 - x_2 - x_4 \\
x_5 &= x_2 - x_3
\end{aligned}
$$

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

Pivot with x_3 entering and x_5 leaving

$$
\begin{aligned}
z &= 8 + x_2 - x_4 - x_5 \\
x_1 &= 8 - x_2 - x_4 \\
x_3 &= x_2 - x_5
\end{aligned}
$$

Termination and Running Time

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 $\widehat{b_i} = b_i + \epsilon_i$, 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 $\widehat{b_i} = b_i + \epsilon_i$, 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 $\widehat{b}_i = b_i + \epsilon_i$, 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.

Outline

Finding an Initial Solution

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

Finding an Initial Solution

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

↓ Conversion into slack form

Finding an Initial Solution

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

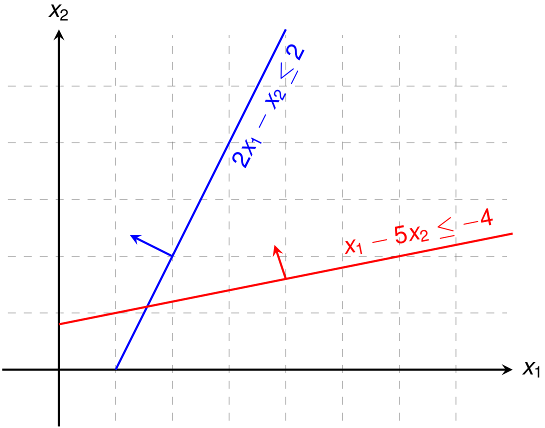
Conversion into slack form

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

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

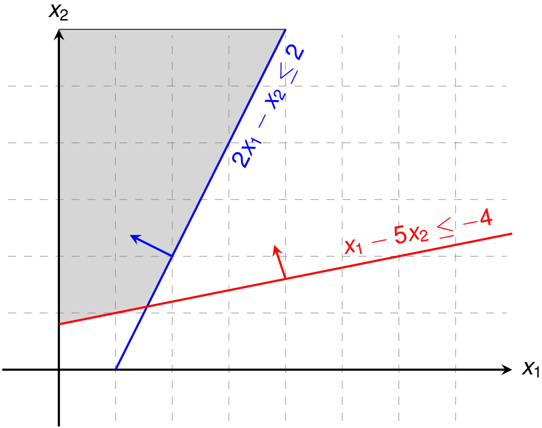
Geometric Illustration

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



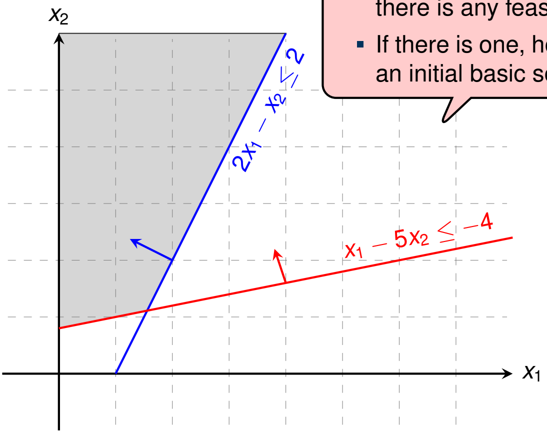
Geometric Illustration

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



Geometric Illustration

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



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

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

subject to

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

Formulating an Auxiliary Linear Program

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

subject to

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

↓ Formulating an Auxiliary Linear Program

Formulating an Auxiliary Linear Program

$$\begin{array}{ll} \text{maximize} & \sum_{j=1}^{n} c_j x_j \\ \text{subject to} & \end{array}$$

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

↓ Formulating an Auxiliary Linear Program

$$\begin{array}{ll} \text{maximize} & -x_0 \\ \text{subject to} & \end{array}$$

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

Formulating an Auxiliary Linear Program

$$\begin{array}{ll} \text{maximize} & \sum_{j=1}^{n} c_j x_j \\ \text{subject to} & \end{array}$$

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

↓ Formulating an Auxiliary Linear Program

$$\begin{array}{ll} \text{maximize} & -x_0 \\ \text{subject to} & \end{array}$$

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

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

$$\begin{array}{ll} \text{maximize} & \sum_{j=1}^{n} c_j x_j \\ \text{subject to} & \end{array}$$

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

↓ Formulating an Auxiliary Linear Program

$$\begin{array}{ll} \text{maximize} & -x_0 \\ \text{subject to} & \end{array}$$

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

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.

Formulating an Auxiliary Linear Program

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

subject to

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

↓ Formulating an Auxiliary Linear Program

maximize $-x_0$

subject to

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

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.

- "$\Rightarrow$": Suppose L has a feasible solution $\overline{x} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)$

Formulating an Auxiliary Linear Program

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

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

↓ Formulating an Auxiliary Linear Program

maximize $-x_0$
subject to

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

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.

- "$\Rightarrow$": Suppose L has a feasible solution $\overline{x} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)$
  - $\overline{x}_0 = 0$ combined with $\overline{x}$ is a feasible solution to L_{aux} with objective value 0.

Formulating an Auxiliary Linear Program

$$
\begin{array}{ll}
\text{maximize} & \sum_{j=1}^{n} c_j x_j \\
\text{subject to} & \\
& \begin{array}{rcll}
\sum_{j=1}^{n} a_{ij} x_j & \leq & b_i & \text{for } i = 1, 2, \ldots, m, \\
x_j & \geq & 0 & \text{for } j = 1, 2, \ldots, n
\end{array}
\end{array}
$$

↓ Formulating an Auxiliary Linear Program

$$
\begin{array}{ll}
\text{maximize} & -x_0 \\
\text{subject to} & \\
& \begin{array}{rcll}
\sum_{j=1}^{n} a_{ij} x_j - x_0 & \leq & b_i & \text{for } i = 1, 2, \ldots, m, \\
x_j & \geq & 0 & \text{for } j = 0, 1, \ldots, n
\end{array}
\end{array}
$$

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.

- "$\Rightarrow$": Suppose L has a feasible solution $\overline{x} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)$
  - $\overline{x}_0 = 0$ combined with $\overline{x}$ is a feasible solution to L_{aux} with objective value 0.
  - Since $\overline{x}_0 \geq 0$ and the objective is to maximize $-x_0$, this is optimal for L_{aux}

Formulating an Auxiliary Linear Program

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

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

↓ Formulating an Auxiliary Linear Program

maximize $-x_0$
subject to

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

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.

- "$\Rightarrow$": Suppose L has a feasible solution $\overline{x} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)$
  - $\overline{x}_0 = 0$ combined with $\overline{x}$ is a feasible solution to L_{aux} with objective value 0.
  - Since $\overline{x}_0 \geq 0$ and the objective is to maximize $-x_0$, this is optimal for L_{aux}
- "$\Leftarrow$": Suppose that the optimal objective value of L_{aux} is 0

Formulating an Auxiliary Linear Program

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

subject to

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

↓ Formulating an Auxiliary Linear Program

maximize $-x_0$

subject to

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

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.

- "$\Rightarrow$": Suppose L has a feasible solution $\overline{x} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)$
  - $\overline{x}_0 = 0$ combined with $\overline{x}$ is a feasible solution to L_{aux} with objective value 0.
  - Since $\overline{x}_0 \geq 0$ and the objective is to maximize $-x_0$, this is optimal for L_{aux}
- "$\Leftarrow$": Suppose that the optimal objective value of L_{aux} is 0
  - Then $\overline{x}_0 = 0$, and the remaining solution values $(\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)$ satisfy L.

Formulating an Auxiliary Linear Program

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

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

↓ Formulating an Auxiliary Linear Program

maximize $-x_0$
subject to

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

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.

- "$\Rightarrow$": Suppose L has a feasible solution $\overline{x} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)$
  - $\overline{x}_0 = 0$ combined with $\overline{x}$ is a feasible solution to L_{aux} with objective value 0.
  - Since $\overline{x}_0 \geq 0$ and the objective is to maximize $-x_0$, this is optimal for L_{aux}
- "$\Leftarrow$": Suppose that the optimal objective value of L_{aux} is 0
  - Then $\overline{x}_0 = 0$, and the remaining solution values $(\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)$ satisfy L. □

INITIALIZE-SIMPLEX

```
INITIALIZE-SIMPLEX(A, b, c)
let k be the index of the minimum b_i
if b_k ≥ 0                      // is the initial basic solution feasible?
    return ({1, 2, ..., n}, {n + 1, n + 2, ..., n + m}, A, b, c, 0)
form L_aux by adding −x_0 to the left-hand side of each constraint
    and setting the objective function to −x_0
let (N, B, A, b, c, ν) be the resulting slack form for L_aux
l = n + k
// L_aux has n + 1 nonbasic variables and m basic variables.
(N, B, A, b, c, ν) = PIVOT(N, B, A, b, c, ν, l, 0)
// The basic solution is now feasible for L_aux.
iterate the while loop of lines 3–12 of SIMPLEX until an optimal solution
    to L_aux is found
if the optimal solution to L_aux sets x̄_0 to 0
    if x̄_0 is basic
        perform one (degenerate) pivot to make it nonbasic
    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
    return the modified final slack form
else return "infeasible"
```

Initialize-Simplex

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

```
INITIALIZE-SIMPLEX(A, b, c)
let k be the index of the minimum b_i
if b_k ≥ 0                    // is the initial basic solution feasible?
    return ({1, 2, ..., n}, {n + 1, n + 2, ..., n + m}, A, b, c, 0)
form L_aux by adding −x_0 to the left-hand side of each constraint
    and setting the objective function to −x_0
let (N, B, A, b, c, v) be the resulting slack form for L_aux
l = n + k
// L_aux has n + 1 nonbasic variables and m basic variables.
(N, B, A, b, c, v) = PIVOT(N, B, A, b, c, v, l, 0)
// The basic solution is now feasible for L_aux.
iterate the while loop of lines 3–12 of SIMPLEX until an optimal solution
    to L_aux is found
if the optimal solution to L_aux sets x̄_0 to 0
    if x̄_0 is basic
        perform one (degenerate) pivot to make it nonbasic
    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
    return the modified final slack form
else return "infeasible"
```

INITIALIZE-SIMPLEX

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

```
INITIALIZE-SIMPLEX(A, b, c)
let k be the index of the minimum b_i
if b_k ≥ 0                    // is the initial basic solution feasible?
    return ({1, 2, ..., n}, {n + 1, n + 2, ..., n + m}, A, b, c, 0)
form L_aux by adding −x_0 to the left-hand side of each constraint
    and setting the objective function to −x_0
let (N, B, A, b, c, ν) be the resulting slack form for L_aux
l = n + k
// L_aux has n + 1 nonbasic variables and m basic variables.
(N, B, A, b, c, ν) = PIVOT(N, B, A, b, c, ν, l, 0)
// The basic solution is now feasible for L_aux.
iterate the while loop of lines 3–12 of SIMPLEX until an optimal solution
    to L_aux is found
if the optimal solution to L_aux sets x̄_0 to 0
    if x̄_0 is basic
        perform one (degenerate) pivot to make it nonbasic
    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
    return the modified final slack form
else return "infeasible"
```

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

INITIALIZE-SIMPLEX

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

```
INITIALIZE-SIMPLEX(A, b, c)
let k be the index of the minimum b_i
if b_k ≥ 0                    // is the initial basic solution feasible?
    return ({1, 2, ..., n}, {n + 1, n + 2, ..., n + m}, A, b, c, 0)
form L_aux by adding −x_0 to the left-hand side of each constraint
    and setting the objective function to −x_0
let (N, B, A, b, c, v) be the resulting slack form for L_aux
l = n + k
// L_aux has n + 1 nonbasic variables and m basic variables.
(N, B, A, b, c, v) = PIVOT(N, B, A, b, c, v, l, 0)
// The basic solution is now feasible for L_aux.
iterate the while loop of lines 3–12 of SIMPLEX until an optimal solution
    to L_aux is found
if the optimal solution to L_aux sets x̄_0 to 0
    if x̄_0 is basic
        perform one (degenerate) pivot to make it nonbasic
    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
    return the modified final slack form
else return "infeasible"
```

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

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

INITIALIZE-SIMPLEX(A, b, c)

let k be the index of the minimum b_i
if $b_k \geq 0$ // is the initial basic solution feasible?
  return $(\{1, 2, \ldots, n\}, \{n+1, n+2, \ldots, n+m\}, A, b, c, 0)$
form L_{aux} by adding $-x_0$ to the left-hand side of each constraint and setting the objective function to $-x_0$
let (N, B, A, b, c, ν) be the resulting slack form for L_{aux}
$l = n + k$
// L_{aux} has $n+1$ nonbasic variables and m basic variables.
$(N, B, A, b, c, \nu) = \text{PIVOT}(N, B, A, b, c, \nu, l, 0)$
// The basic solution is now feasible for L_{aux}.
iterate the **while** loop of lines 3–12 of SIMPLEX until an optimal solution to L_{aux} is found
if the optimal solution to L_{aux} sets $\bar{x}_0$ to 0
  if $\bar{x}_0$ is basic
    perform one (degenerate) pivot to make it nonbasic
  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
  return the modified final slack form
else return "infeasible"

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)

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

Example of INITIALIZE-SIMPLEX (1/3)

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

↓ Formulating the auxiliary linear program

Example of INITIALIZE-SIMPLEX (1/3)

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

Formulating the auxiliary linear program

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

Example of INITIALIZE-SIMPLEX (1/3)

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

↓ Formulating the auxiliary linear program

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

↓ Converting into slack form

Example of INITIALIZE-SIMPLEX (1/3)

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

Formulating the auxiliary linear program

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

Converting into slack form

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

Example of INITIALIZE-SIMPLEX (1/3)

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

Formulating the auxiliary linear program

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

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

Converting into slack form

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

Example of INITIALIZE-SIMPLEX (2/3)

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

Example of INITIALIZE-SIMPLEX (2/3)

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

Pivot with x_0 entering and x_4 leaving

Example of INITIALIZE-SIMPLEX (2/3)

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

Pivot with x_0 entering and x_4 leaving

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

Example of Initialize-Simplex (2/3)

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

Pivot with x_0 entering and x_4 leaving

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

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

Example of Initialize-Simplex (2/3)

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

Pivot with x_0 entering and x_4 leaving

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

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

Pivot with x_2 entering and x_0 leaving

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

Example of INITIALIZE-SIMPLEX (2/3)

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

Pivot with x_0 entering and x_4 leaving

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

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

Pivot with x_2 entering and x_0 leaving

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

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

Example of INITIALIZE-SIMPLEX (3/3)

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

Example of INITIALIZE-SIMPLEX (3/3)

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

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

Example of INITIALIZE-SIMPLEX (3/3)

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

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

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

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

Example of INITIALIZE-SIMPLEX (3/3)

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

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

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

$$\begin{aligned} z &= -\frac{4}{5} + \frac{9x_1}{5} - \frac{x_4}{5} \\ x_2 &= \frac{4}{5} + \frac{x_1}{5} + \frac{x_4}{5} \\ x_3 &= \frac{14}{5} - \frac{9x_1}{5} + \frac{x_4}{5} \end{aligned}$$

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

Example of INITIALIZE-SIMPLEX (3/3)

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

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

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

$$
\begin{aligned}
z &= -\frac{4}{5} + \frac{9x_1}{5} - \frac{x_4}{5} \\
x_2 &= \frac{4}{5} + \frac{x_1}{5} + \frac{x_4}{5} \\
x_3 &= \frac{14}{5} - \frac{9x_1}{5} + \frac{x_4}{5}
\end{aligned}
$$

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)

Any linear program L, given in standard form, either

1. has an optimal solution with a finite objective value,
2. is infeasible, or
3. is unbounded.

If L is infeasible, SIMPLEX returns "infeasible". If L is unbounded, SIMPLEX returns "unbounded". Otherwise, SIMPLEX returns an optimal solution with a finite objective value.

Fundamental Theorem of Linear Programming

Theorem 29.13 (Fundamental Theorem of Linear Programming)

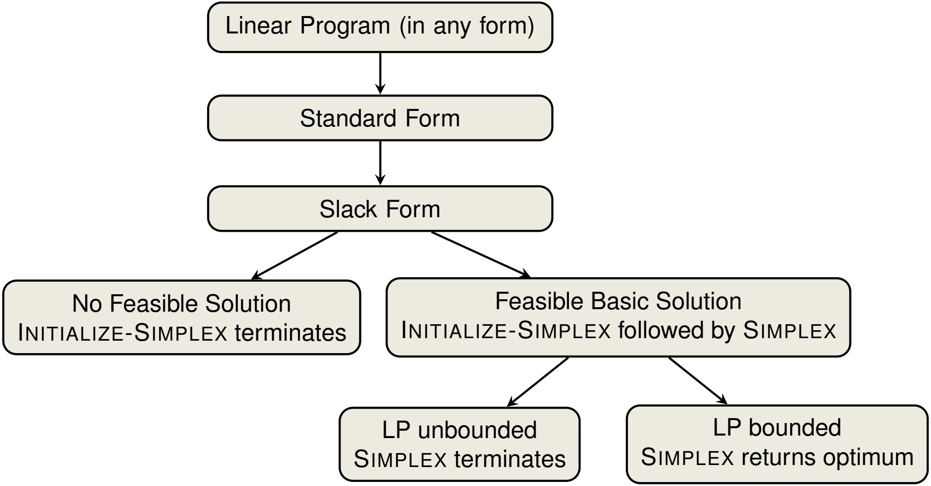
Any linear program L, given in standard form, either

1. has an optimal solution with a finite objective value,
2. is infeasible, or
3. is unbounded.

If L is infeasible, SIMPLEX returns "infeasible". If L is unbounded, SIMPLEX returns "unbounded". Otherwise, SIMPLEX returns an optimal solution with a finite objective value.

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

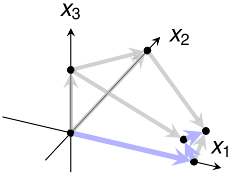
Linear Programming and Simplex: Summary and Outlook

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., $O(m+n)$



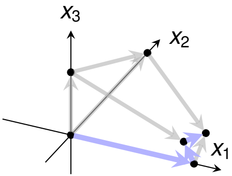
Linear Programming and Simplex: Summary and Outlook

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



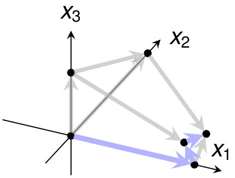
Linear Programming and Simplex: Summary and Outlook

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?

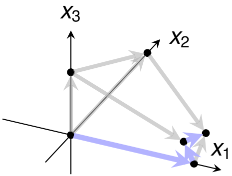
Linear Programming and Simplex: Summary and Outlook

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

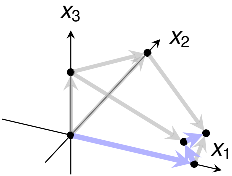
Linear Programming

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

Simplex Algorithm

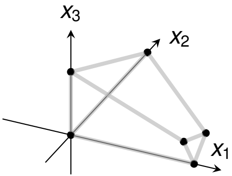
- In practice: usually terminates in polynomial time, i.e., $O(m+n)$
- In theory: even with anti-cycling may need exponential time

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

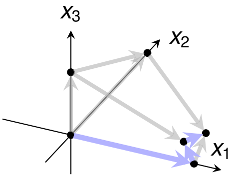
Linear Programming

- 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!)