Module 1: Formulations (IP Models)

Recap: WaterTech

$$
\begin{aligned}
\max \quad & 300x_1 + 260x_2 + 220x_3 + 180x_4 - 8y_s - 6y_u \\
\text{s.t.} \quad & 11x_1 + 7x_2 + 6x_3 + 5x_4 \leq 700 \\
& 4x_1 + 6x_2 + 5x_3 + 4x_4 \leq 500 \\
& 8x_1 + 5x_2 + 5x_3 + 6x_4 \leq y_s \\
& 7x_1 + 8x_2 + 7x_3 + 4x_4 \leq y_u \\
& y_s \leq 600 \\
& y_u \leq 650 \\
& x_1, x_2, x_3, x_4, y_u, y_s \geq 0.
\end{aligned}
$$

Recap: WaterTech

$$
\begin{aligned}
\max \quad & 300x_1 + 260x_2 + 220x_3 + 180x_4 - 8y_s - 6y_u \\
\text{s.t.} \quad & 11x_1 + 7x_2 + 6x_3 + 5x_4 \leq 700 \\
& 4x_1 + 6x_2 + 5x_3 + 4x_4 \leq 500 \\
& 8x_1 + 5x_2 + 5x_3 + 6x_4 \leq y_s \\
& 7x_1 + 8x_2 + 7x_3 + 4x_4 \leq y_u \\
& y_s \leq 600 \\
& y_u \leq 650 \\
& x_1, x_2, x_3, x_4, y_u, y_s \geq 0.
\end{aligned}
$$

Optimal solution: $x = (16 + \frac{2}{3}, 50, 0, 33 + \frac{1}{3})^T$, $y_s = 583 + \frac{1}{3}$, $y_u = 650$

Recap: WaterTech

$$\begin{aligned}
\max \quad & 300x_1 + 260x_2 + 220x_3 + 180x_4 - 8y_s - 6y_u \\
\text{s.t.} \quad & 11x_1 + 7x_2 + 6x_3 + 5x_4 \leq 700 \\
& 4x_1 + 6x_2 + 5x_3 + 4x_4 \leq 500 \\
& 8x_1 + 5x_2 + 5x_3 + 6x_4 \leq y_s \\
& 7x_1 + 8x_2 + 7x_3 + 4x_4 \leq y_u \\
& y_s \leq 600 \\
& y_u \leq 650 \\
& x_1, x_2, x_3, x_4, y_u, y_s \geq 0.
\end{aligned}$$

Optimal solution: $x = (16 + \frac{2}{3}, 50, 0, 33 + \frac{1}{3})^T$, $y_s = 583 + \frac{1}{3}$, $y_u = 650$

Fractional solutions are often not desirable! Can we force solution to take on only integer values?

- Yes!

  An integer program is a linear program with added integrality constraints for some/all variables.

$$\begin{aligned} \max \quad & x_1 + x_2 + 2x_4 \\ \text{s.t.} \quad & x_1 + x_2 \leq 1 \\ & -x_2 - x_3 \geq -1 \\ & x_1 + x_3 = 1 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

- Yes!

  An integer program is a linear program with added integrality constraints for some/all variables.

$$
\begin{aligned}
\max \quad & x_1 + x_2 + 2x_4 \\
\text{s.t.} \quad & x_1 + x_2 \le 1 \\
& -x_2 - x_3 \ge -1 \\
& x_1 + x_3 = 1 \\
& x_1, x_2, x_3 \ge 0 \\
& x_1, x_3 \text{ integer.}
\end{aligned}
$$

- Yes!

  An integer program is a linear program with added integrality constraints for some/all variables.

- We call an IP mixed if there are integer and fractional variables, and pure otherwise.

$$\begin{aligned}
\max \quad & x_1 + x_2 + 2x_4 \\
\text{s.t.} \quad & x_1 + x_2 \leq 1 \\
& -x_2 - x_3 \geq -1 \\
& x_1 + x_3 = 1 \\
& x_1, x_2, x_3 \geq 0 \\
& x_1, x_3 \text{ integer.}
\end{aligned}$$

- Yes!

  An integer program is a linear program with added integrality constraints for some/all variables.

- We call an IP mixed if there are integer and fractional variables, and pure otherwise.

- Difference between LPs and IPs is subtle. Yet: LPs are easy to solve, IPs are not!

$$
\begin{aligned}
\max \quad & x_1 + x_2 + 2x_4 \\
\text{s.t.} \quad & x_1 + x_2 \leq 1 \\
& -x_2 - x_3 \geq -1 \\
& x_1 + x_3 = 1 \\
& x_1, x_2, x_3 \geq 0 \\
& x_1, x_3 \text{ integer.}
\end{aligned}
$$

Can we solve IPs?

- Integer programs are provably difficult to solve!

Can we solve IPs?

- Integer programs are provably difficult to solve!
- Every problem instance has a size which we normally denote by n.

.

Can we solve IPs?

- Integer programs are provably difficult to solve!
- Every problem instance has a size which we normally denote by n.
  Think: $n \sim$ number of variables/constraints of IP.

Can we solve IPs?

- Integer programs are provably difficult to solve!
- Every problem instance has a size which we normally denote by n.
  Think: $n \sim$ number of variables/constraints of IP.
- The running time of an algorithm is then the number of steps that an algorithm takes.

Can we solve IPs?

- Integer programs are provably difficult to solve!
- Every problem instance has a size which we normally denote by n.
  Think: $n \sim$ number of variables/constraints of IP.
- The running time of an algorithm is then the number of steps that an algorithm takes.
- It is stated as a function of n: $f(n)$ measures the largest number of steps an algorithm takes on an instance of size n.

Can we solve IPs?

- An algorithm is efficient if its running time $f(n)$ is a polynomial of n.

Can we solve IPs?

- An algorithm is efficient if its running time $f(n)$ is a polynomial of n.
- LPs can be solved efficiently.

The New York Times U.S.

WORLD U.S. N.Y. / REGION BUSINESS TECHNOLOGY SCIENCE HEALTH SPORTS OPINION

POLITICS EDUCATION TEXAS

BREAKTHROUGH IN PROBLEM SOLVING

By JAMES GLEICK
Published: November 19, 1984

A 28-year-old mathematician at A.T.&T. Bell Laboratories has made a startling theoretical breakthrough in the solving of systems of equations that often grow too vast and complex for the most powerful computers.

The discovery, which is to be formally published next month, is already circulating rapidly through the mathematical world. It has also set off a deluge of inquiries from brokerage houses, oil companies and airlines, industries with millions of dollars at stake in problems known as linear programming.

These problems are fiendishly complicated systems, often with

FACEBOOK TWITTER GOOGLE+ EMAIL SHARE PRINT SINGLE PAGE REPRINTS

Can we solve IPs?

- An algorithm is efficient if its running time $f(n)$ is a polynomial of n.
- LPs can be solved efficiently.
- IPs are very unlikely to admit efficient algorithms!


The New York Times

U.S.

WORLD | U.S. | N.Y. / REGION | BUSINESS | TECHNOLOGY | SCIENCE | HEALTH | SPORTS | OPINION

POLITICS EDUCATION TEXAS

BREAKTHROUGH IN PROBLEM SOLVING

By JAMES GLEICK
Published: November 19, 1984

A 28-year-old mathematician at A.T.&T. Bell Laboratories has made a startling theoretical breakthrough in the solving of systems of equations that often grow too vast and complex for the most powerful computers.

The discovery, which is to be formally published next month, is already circulating rapidly through the mathematical world. It has also set off a deluge of inquiries from brokerage houses, oil companies and airlines, industries with millions of dollars at stake in problems known as linear programming.

These problems are fiendishly complicated systems, often with

FACEBOOK
TWITTER
GOOGLE+
EMAIL
SHARE
PRINT
SINGLE PAGE
REPRINTS

Can we solve IPs?

- An algorithm is efficient if its running time $f(n)$ is a polynomial of n.
- LPs can be solved efficiently.
- IPs are very unlikely to admit efficient algorithms!
- It is very important to look for an efficient algorithm for a problem. The table states actual running times of a computer that can execute 1 million operations per second on an instance of size $n = 100$:


The New York Times

U.S.

WORLD | U.S. | N.Y. / REGION | BUSINESS | TECHNOLOGY | SCIENCE | HEALTH | SPORTS | OPINION

POLITICS EDUCATION TEXAS

BREAKTHROUGH IN PROBLEM SOLVING

By JAMES GLEICK
Published: November 19, 1984

A 28-year-old mathematician at A.T.&T. Bell Laboratories has made a startling theoretical breakthrough in the solving of systems of equations that often grow too vast and complex for the most powerful computers.

The discovery, which is to be formally published next month, is already circulating rapidly through the mathematical world. It has also set off a deluge of inquiries from brokerage houses, oil companies and airlines, industries with millions of dollars at stake in problems known as linear programming.

These problems are fiendishly complicated systems, often with

FACEBOOK
TWITTER
GOOGLE+
EMAIL
SHARE
PRINT
SINGLE PAGE
REPRINTS

$f(n)$	n	$n\log_2(n)$	n^3	1.5^n	2^n

IP Models: Knapsack

KitchTech Shipping

- Company wishes to ship crates from Toronto to Kitchener.

KitchTech Shipping

- Company wishes to ship crates from Toronto to Kitchener.
- Each crate type has weight and value:

Type	1	2	3	4	5	6
weight (lbs)	30	20	30	90	30	70
value ($)	60	70	40	70	20	90

KitchTech Shipping

- Company wishes to ship crates from Toronto to Kitchener.
- Each crate type has weight and value:

Type	1	2	3	4	5	6
weight (lbs)	30	20	30	90	30	70
value ($)	60	70	40	70	20	90

- Total weight of crates shipped must not exceed 10,000 lbs.

KitchTech Shipping

- Company wishes to ship crates from Toronto to Kitchener.
- Each crate type has weight and value:

Type	1	2	3	4	5	6
weight (lbs)	30	20	30	90	30	70
value ($)	60	70	40	70	20	90

- Total weight of crates shipped must not exceed 10,000 lbs.
- Goal: Maximize total value of shipped goods.

IP Model

- Variables.

IP Model

- Variables. One variable x_i for number of crates of type i to pack.

IP Model

- Variables. One variable x_i for number of crates of type i to pack.
- Constraints.

IP Model

- Variables. One variable x_i for number of crates of type i to pack.
- Constraints. The total weight of a crates picked must not exceed 10000 lbs.

IP Model

- Variables. One variable x_i for number of crates of type i to pack.
- Constraints. The total weight of a crates picked must not exceed 10000 lbs.

$$30x_1 + 20x_2 + 30x_3 + 90x_4 + 30x_5 + 70x_6 \leq 10000$$

IP Model

- Variables. One variable x_i for number of crates of type i to pack.
- Constraints. The total weight of a crates picked must not exceed 10000 lbs.

$$30x_1 + 20x_2 + 30x_3 + 90x_4 + 30x_5 + 70x_6 \leq 10000$$

- Objective function: Maximize total value.

IP Model

- Variables. One variable x_i for number of crates of type i to pack.
- Constraints. The total weight of a crates picked must not exceed 10000 lbs.

$$30x_1 + 20x_2 + 30x_3 + 90x_4 + 30x_5 + 70x_6 \leq 10000$$

- Objective function: Maximize total value.

$$\max \quad 60x_1 + 70x_2 + 40x_3 + 70x_4 + 20x_5 + 90x_6$$

IP Model

$$
\begin{aligned}
\max \quad & 60x_1 + 70x_2 + 40x_3 + 70x_4 + 20x_5 + 90x_6 \\
\text{s.t.} \quad & 30x_1 + 20x_2 + 30x_3 + 90x_4 + 30x_5 + 70x_6 \leq 10000 \\
& x_i \geq 0 \quad (i \in [6]) \\
& x_i \text{ integer} \quad (i \in [6])
\end{aligned}
$$

IP Model

$$
\begin{aligned}
\max \quad & 60x_1 + 70x_2 + 40x_3 + 70x_4 + 20x_5 + 90x_6 \\
\text{s.t.} \quad & 30x_1 + 20x_2 + 30x_3 + 90x_4 + 30x_5 + 70x_6 \leq 10000 \\
& x_i \geq 0 \quad (i \in [6]) \\
& {\color{red} x_i \text{ integer} \quad (i \in [6])}
\end{aligned}
$$

Let's make this model a bit more interesting...

KitchTech: Added Conditions

Suppose that ...

- We must not send more than 10 crates of the same type.

$$
\begin{aligned}
\max \quad & 60x_1 + 70x_2 + 40x_3 + \\
& 70x_4 + 20x_5 + 90x_6 \\
\text{s.t.} \quad & 30x_1 + 20x_2 + 30x_3 + \\
& 90x_4 + 30x_5 + 70x_6 \leq 10000 \\
& x_i \geq 0 \quad (i \in [6]) \\
& x_i \text{ integer} \quad (i \in [6])
\end{aligned}
$$

KitchTech: Added Conditions

Suppose that ...

- We must not send more than 10 crates of the same type.

$$
\begin{aligned}
\max \quad & 60x_1 + 70x_2 + 40x_3 + \\
& 70x_4 + 20x_5 + 90x_6 \\
\text{s.t.} \quad & 30x_1 + 20x_2 + 30x_3 + \\
& 90x_4 + 30x_5 + 70x_6 \leq 10000 \\
& 0 \leq x_i \leq 10 \quad (i \in [6]) \\
& x_i \text{ integer} \quad (i \in [6])
\end{aligned}
$$

KitchTech: Added Conditions

Suppose that ...

- We must not send more than 10 crates of the same type.
- Can only send crates of type 3, if we send at least 1 crate of type 4.

$$
\begin{aligned}
\max \quad & 60x_1 + 70x_2 + 40x_3 + \\
& 70x_4 + 20x_5 + 90x_6 \\
\text{s.t.} \quad & 30x_1 + 20x_2 + 30x_3 + \\
& 90x_4 + 30x_5 + 70x_6 \leq 10000 \\
& 0 \leq x_i \leq 10 \quad (i \in [6]) \\
& x_i \text{ integer} \quad (i \in [6])
\end{aligned}
$$

KitchTech: Added Conditions

Suppose that ...

- We must not send more than 10 crates of the same type.
- Can only send crates of type 3, if we send at least 1 crate of type 4.

  Note: Can send at most 10 crates of type 3 by previous constraint!

$$
\begin{aligned}
\max \quad & 60x_1 + 70x_2 + 40x_3 + \\
& 70x_4 + 20x_5 + 90x_6 \\
\text{s.t.} \quad & 30x_1 + 20x_2 + 30x_3 + \\
& 90x_4 + 30x_5 + 70x_6 \leq 10000 \\
& 0 \leq x_i \leq 10 \quad (i \in [6]) \\
& x_i \text{ integer} \quad (i \in [6])
\end{aligned}
$$

KitchTech: Added Conditions

Suppose that ...

- We must not send more than 10 crates of the same type.
- Can only send crates of type 3, if we send at least 1 crate of type 4.

  Note: Can send at most 10 crates of type 3 by previous constraint!

$$
\begin{aligned}
\max \quad & 60x_1 + 70x_2 + 40x_3 + \\
& 70x_4 + 20x_5 + 90x_6 \\
\text{s.t.} \quad & 30x_1 + 20x_2 + 30x_3 + \\
& 90x_4 + 30x_5 + 70x_6 \le 10000 \\
& x_3 \le 10x_4 \\
& 0 \le x_i \le 10 \quad (i \in [6]) \\
& x_i \text{ integer} \quad (i \in [6])
\end{aligned}
$$

KitchTech: Added Conditions

Correctness:

- $x_4 \geq 1 \longrightarrow$ new constraint is redundant!

$$
\begin{aligned}
\max \quad & 60x_1 + 70x_2 + 40x_3 + \\
& 70x_4 + 20x_5 + 90x_6 \\
\text{s.t.} \quad & 30x_1 + 20x_2 + 30x_3 + \\
& 90x_4 + 30x_5 + 70x_6 \leq 10000 \\
& x_3 \leq 10x_4 \\
& 0 \leq x_i \leq 10 \quad (i \in [6]) \\
& x_i \text{ integer} \quad (i \in [6])
\end{aligned}
$$

KitchTech: Added Conditions

Correctness:

- $x_4 \geq 1 \longrightarrow$ new constraint is redundant!
- $x_4 = 0 \longrightarrow$ new constraint becomes

$$x_3 \leq 0.$$

$$\begin{aligned} \max \quad & 60x_1 + 70x_2 + 40x_3 + \\ & 70x_4 + 20x_5 + 90x_6 \\ \text{s.t.} \quad & 30x_1 + 20x_2 + 30x_3 + \\ & 90x_4 + 30x_5 + 70x_6 \leq 10000 \\ & x_3 \leq 10x_4 \\ & 0 \leq x_i \leq 10 \quad (i \in [6]) \\ & x_i \text{ integer} \quad (i \in [6]) \end{aligned}$$

KitchTech: One More Tricky Case

Suppose that we must

1. take a total of at least 4 crates of type 1 or 2, or
2. take at least 4 crates of type 5 or 6.

$$
\begin{aligned}
\max \quad & 60x_1 + 70x_2 + 40x_3 + \\
& 70x_4 + 20x_5 + 90x_6 \\
\text{s.t.} \quad & 30x_1 + 20x_2 + 30x_3 + \\
& 90x_4 + 30x_5 + 70x_6 \leq 10000 \\
& x_3 \leq 10x_4 \\
& 0 \leq x_i \leq 10 \quad (i \in [6]) \\
& x_i \text{ integer} \quad (i \in [6])
\end{aligned}
$$

KitchTech: One More Tricky Case

Suppose that we must

1. take a total of at least 4 crates of type 1 or 2, or
2. take at least 4 crates of type 5 or 6.

Ideas?

$$
\begin{aligned}
\max \quad & 60x_1 + 70x_2 + 40x_3 + \\
& 70x_4 + 20x_5 + 90x_6 \\
\text{s.t.} \quad & 30x_1 + 20x_2 + 30x_3 + \\
& 90x_4 + 30x_5 + 70x_6 \leq 10000 \\
& x_3 \leq 10x_4 \\
& 0 \leq x_i \leq 10 \quad (i \in [6]) \\
& x_i \text{ integer} \quad (i \in [6])
\end{aligned}
$$

KitchTech: One More Tricky Case

Suppose that we must

1. take a total of at least 4 crates of type 1 or 2, or
2. take at least 4 crates of type 5 or 6.

Ideas?

Create a new variable y s.t.

1. $y = 1 \longrightarrow x_1 + x_2 \geq 4$,
2. $y = 0 \longrightarrow$

$$\begin{aligned} \max \quad & 60x_1 + 70x_2 + 40x_3 + \\ & 70x_4 + 20x_5 + 90x_6 \\ \text{s.t.} \quad & 30x_1 + 20x_2 + 30x_3 + \\ & 90x_4 + 30x_5 + 70x_6 \leq 10000 \\ & x_3 \leq 10x_4 \\ & 0 \leq x_i \leq 10 \quad (i \in [6]) \\ & x_i \text{ integer} \quad (i \in [6]) \end{aligned}$$

KitchTech: One More Tricky Case

Create a new variable y s.t.

1. $y = 1 \longrightarrow x_1 + x_2 \geq 4$,
2. $y = 0 \longrightarrow x_5 + x_6 \geq 4$.

Force y to take on values 0 or 1.

Add constraints:

$$
\begin{aligned}
\max \quad & 60x_1 + 70x_2 + 40x_3 + \\
& 70x_4 + 20x_5 + 90x_6 \\
\text{s.t.} \quad & 30x_1 + 20x_2 + 30x_3 + \\
& 90x_4 + 30x_5 + 70x_6 \leq 10000 \\
& x_3 \leq 10x_4 \\
& 0 \leq x_i \leq 10 \quad (i \in [6]) \\
& x_i \text{ integer} \quad (i \in [6])
\end{aligned}
$$

KitchTech: One More Tricky Case

Create a new variable y s.t.

1. $y = 1 \longrightarrow x_1 + x_2 \geq 4$,
2. $y = 0 \longrightarrow x_5 + x_6 \geq 4$.

Force y to take on values 0 or 1.

Add constraints:

1. $x_1 + x_2 \geq 4y$

$$
\begin{aligned}
\max \quad & 60x_1 + 70x_2 + 40x_3 + \\
& 70x_4 + 20x_5 + 90x_6 \\
\text{s.t.} \quad & 30x_1 + 20x_2 + 30x_3 + \\
& 90x_4 + 30x_5 + 70x_6 \leq 10000 \\
& x_3 \leq 10x_4 \\
& 0 \leq x_i \leq 10 \quad (i \in [6]) \\
& x_i \text{ integer} \quad (i \in [6])
\end{aligned}
$$

KitchTech: One More Tricky Case

Create a new variable y s.t.

1. $y = 1 \longrightarrow x_1 + x_2 \geq 4$,
2. $y = 0 \longrightarrow x_5 + x_6 \geq 4$.

Force y to take on values 0 or 1.

Add constraints:

1. $x_1 + x_2 \geq 4y$
2. $x_5 + x_6 \geq 4(1 - y)$

$$
\begin{array}{ll}
\max & 60x_1 + 70x_2 + 40x_3 + \\
& 70x_4 + 20x_5 + 90x_6 \\
\text{s.t.} & 30x_1 + 20x_2 + 30x_3 + \\
& 90x_4 + 30x_5 + 70x_6 \leq 10000 \\
& x_3 \leq 10x_4 \\
& 0 \leq x_i \leq 10 \quad (i \in [6]) \\
& x_i \text{ integer} \quad (i \in [6])
\end{array}
$$

KitchTech: One More Tricky Case

Create a new variable y s.t.

1. $y = 1 \longrightarrow x_1 + x_2 \geq 4$,
2. $y = 0 \longrightarrow x_5 + x_6 \geq 4$.

Force y to take on values 0 or 1.

Add constraints:

1. $x_1 + x_2 \geq 4y$
2. $x_5 + x_6 \geq 4(1 - y)$
3. $0 \leq y \leq 1$

$$
\begin{aligned}
\max \quad & 60x_1 + 70x_2 + 40x_3 + \\
& 70x_4 + 20x_5 + 90x_6 \\
\text{s.t.} \quad & 30x_1 + 20x_2 + 30x_3 + \\
& 90x_4 + 30x_5 + 70x_6 \leq 10000 \\
& x_3 \leq 10x_4 \\
& 0 \leq x_i \leq 10 \quad (i \in [6]) \\
& x_i \text{ integer} \quad (i \in [6])
\end{aligned}
$$

KitchTech: One More Tricky Case

Create a new variable y s.t.

1. $y = 1 \longrightarrow$ $x_1 + x_2 \geq 4$,
2. $y = 0 \longrightarrow$ $x_5 + x_6 \geq 4$.

Force y to take on values 0 or 1.

Add constraints:

1. $x_1 + x_2 \geq 4y$
2. $x_5 + x_6 \geq 4(1 - y)$
3. $0 < y < 1$

$$
\begin{array}{rl}
\max & 60x_1 + 70x_2 + 40x_3 + \\
 & 70x_4 + 20x_5 + 90x_6 \\
\text{s.t.} & 30x_1 + 20x_2 + 30x_3 + \\
 & 90x_4 + 30x_5 + 70x_6 \leq 10000 \\
 & x_3 \leq 10x_4 \\
 & 0 \leq x_i \leq 10 \quad (i \in [6]) \\
 & x_i \text{ integer} \quad (i \in [6])
\end{array}
$$

KitchTech: One More Tricky Case

Create a new variable y s.t.

1. $y = 1 \longrightarrow x_1 + x_2 \geq 4$,
2. $y = 0 \longrightarrow x_5 + x_6 \geq 4$.

Force y to take on values 0 or 1.

Add constraints:

1. $x_1 + x_2 \geq 4y$
2. $x_5 + x_6 \geq 4(1 - y)$
3. $0 \leq y \leq 1$

$$
\begin{aligned}
\max \quad & 60x_1 + 70x_2 + 40x_3 + \\
& 70x_4 + 20x_5 + 90x_6 \\
\text{s.t.} \quad & 30x_1 + 20x_2 + 30x_3 + \\
& 90x_4 + 30x_5 + 70x_6 \leq 10000 \\
& x_3 \leq 10x_4 \\
& x_1 + x_2 \geq 4y \\
& x_5 + x_6 \geq 4(1 - y) \\
& 0 \leq y \leq 1 \\
& 0 \leq x_i \leq 10 \quad (i \in [6]) \\
& y \text{ integer} \\
& x_i \text{ integer} \quad (i \in [6])
\end{aligned}
$$

Binary Variables

Variable y is called a binary variable.

These are very useful for modeling logical constraints of the form:

[Condition (A or B) and C] $\longrightarrow$ D

Will see more examples ...

$$
\begin{array}{rl}
\max & 60x_1 + 70x_2 + 40x_3 + \\
 & 70x_4 + 20x_5 + 90x_6 \\
\text{s.t.} & 30x_1 + 20x_2 + 30x_3 + \\
 & 90x_4 + 30x_5 + 70x_6 \le 10000 \\
 & x_3 \le 10x_4 \\
 & x_1 + x_2 \ge 4y \\
 & x_5 + x_6 \ge 4(1-y) \\
 & 0 \le y \le 1 \\
 & 0 \le x_i \le 10 \quad (i \in [6]) \\
 & y \text{ integer} \\
 & x_i \text{ integer} \quad (i \in [6])
\end{array}
$$

IP Models: Scheduling

- The neighbourhood coffee shop is open on workdays.

- The neighbourhood coffee shop is open on workdays.
- Daily demand for workers:

Mon	Tues	Wed	Thurs	Fri
3	5	9	2	7

- The neighbourhood coffee shop is open on workdays.
- Daily demand for workers:

Mon	Tues	Wed	Thurs	Fri
3	5	9	2	7

- Each worker works for 4 consecutive days and has one day off.

- The neighbourhood coffee shop is open on workdays.
- Daily demand for workers:

Mon	Tues	Wed	Thurs	Fri
3	5	9	2	7

- Each worker works for 4 consecutive days and has one day off.
  e.g.: work: Mon, Tue, Wed, Thu; off: Fri
  or work: Wed, Thu, Fri, Mon; off: Tue

- The neighbourhood coffee shop is open on workdays.
- Daily demand for workers:

Mon	Tues	Wed	Thurs	Fri
3	5	9	2	7

- Each worker works for 4 consecutive days and has one day off.
  e.g.: work: Mon, Tue, Wed, Thu; off: Fri
  or work: Wed, Thu, Fri, Mon; off: Tue
- Goal: hire the smallest number of workers so that the demand can be met!

- The neighbourhood coffee shop is open on workdays.
- Daily demand for workers:

Mon	Tues	Wed	Thurs	Fri
3	5	9	2	7

- Each worker works for 4 consecutive days and has one day off.
  e.g.: work: Mon, Tue, Wed, Thu; off: Fri
  or work: Wed, Thu, Fri, Mon; off: Tue
- Goal: hire the smallest number of workers so that the demand can be met!

Can we solve this using IP?

1. Variables. What do we need to decide on?

1. Variables. What do we need to decide on?
$\rightarrow$ introduce variable x_d for every $d \in \{M, T, W, Th, F\}$ counting the number of people to hire with starting day d.

1. Variables. What do we need to decide on?
   $\rightarrow$ introduce variable x_d for every $d \in \{M, T, W, Th, F\}$ counting the number of people to hire with starting day d.
2. Objective function. What do we want to minimize?

1. Variables. What do we need to decide on?
   $\rightarrow$ introduce variable x_d for every $d \in \{M, T, W, Th, F\}$ counting the number of people to hire with starting day d.

2. Objective function. What do we want to minimize?
   $\rightarrow$ the total number of people hired:

$$\min x_M + x_T + x_W + x_{Th} + x_F.$$

1. Variables. What do we need to decide on?
   $\rightarrow$ introduce variable x_d for every $d \in \{M, T, W, Th, F\}$ counting the number of people to hire with starting day d.

2. Objective function. What do we want to minimize?
   $\rightarrow$ the total number of people hired:

$$\min x_M + x_T + x_W + x_{Th} + x_F.$$

3. Constraints. Need to ensure that enough people work on each of the days.

1. Variables. What do we need to decide on?
   $\rightarrow$ introduce variable x_d for every $d \in \{M, T, W, Th, F\}$ counting the number of people to hire with starting day d.

2. Objective function. What do we want to minimize?
   $\rightarrow$ the total number of people hired:

$$\min x_M + x_T + x_W + x_{Th} + x_F .$$

3. Constraints. Need to ensure that enough people work on each of the days.

   Question: Given a solution $(x_M, x_T, x_W, x_{Th}, x_F)$, how many people work on Monday?

1. Variables. What do we need to decide on?
   $\rightarrow$ introduce variable x_d for every $d \in \{M, T, W, Th, F\}$ counting the number of people to hire with starting day d.

2. Objective function. What do we want to minimize?
   $\rightarrow$ the total number of people hired:

$$\min x_M + x_T + x_W + x_{Th} + x_F.$$

3. Constraints. Need to ensure that enough people work on each of the days.

   Question: Given a solution $(x_M, x_T, x_W, x_{Th}, x_F)$, how many people work on Monday?
   All but those that start on Tuesday; i.e.,

$$x_M + x_W + x_{Th} + x_F.$$

Constraints

[Daily Demand]

Mon	Tues	Wed	Thurs	Fri
3	5	9	2	7

Monday:

$$x_M + x_W + x_{Th} + x_F \geq 3$$

Constraints

[Daily Demand]

Mon	Tues	Wed	Thurs	Fri
3	5	9	2	7

Monday: $x_M + x_W + x_{Th} + x_F \geq 3$

Tuesday: $x_M + x_T + x_{Th} + x_F \geq 5$

Constraints

[Daily Demand]

Mon	Tues	Wed	Thurs	Fri
3	5	9	2	7

Monday: $x_M + x_W + x_{Th} + x_F \geq 3$

Tuesday: $x_M + x_T + x_{Th} + x_F \geq 5$

Wednesday: $x_M + x_T + x_W + x_F \geq 9$

Constraints

[Daily Demand]

Mon	Tues	Wed	Thurs	Fri
3	5	9	2	7

Monday: $x_M + x_W + x_{Th} + x_F \geq 3$

Tuesday: $x_M + x_T + x_{Th} + x_F \geq 5$

Wednesday: $x_M + x_T + x_W + x_F \geq 9$

Thursday: $x_M + x_T + x_W + x_T \geq 2$

Constraints

[Daily Demand]

Mon	Tues	Wed	Thurs	Fri
3	5	9	2	7

Monday: $x_M + x_W + x_{Th} + x_F \geq 3$

Tuesday: $x_M + x_T + x_{Th} + x_F \geq 5$

Wednesday: $x_M + x_T + x_W + x_F \geq 9$

Thursday: $x_M + x_T + x_W + x_T \geq 2$

Friday: $x_T + x_W + x_{Th} + x_F \geq 7$

Scheduling LP

$$
\begin{array}{rl}
\text{min} & x_M + x_T + x_W + x_{Th} + x_F \\
\text{s.t.} & x_M + x_W + x_{Th} + x_F \geq 3 \\
 & x_M + x_T + x_{Th} + x_F \geq 5 \\
 & x_M + x_T + x_W + x_F \geq 9 \\
 & x_M + x_T + x_W + x_T \geq 2 \\
 & x_T + x_W + x_{Th} + x_F \geq 7 \\
 & x \geq \nvdash, x \text{ integer}
\end{array}
$$

Quiz

Given an integer program with integer variables $x_1, \ldots, x_6$. Let

$$\mathcal{S} := \{127, 289, 1310, 2754\}.$$

We want to add constraints and/or variables to the IP that enforce that the $x_1 + \ldots + x_6$ is in $\mathcal{S}$. How?

Quiz

Given an integer program with integer variables $x_1, \ldots, x_6$. Let

$$\mathcal{S} := \{127, 289, 1310, 2754\}.$$

We want to add constraints and/or variables to the IP that enforce that the $x_1 + \ldots + x_6$ is in $\mathcal{S}$. How?

- Add binary variables $y_{127}, y_{289}, y_{1310}, y_{2754}$, one for each $i \in \mathcal{S}$.

Quiz

Given an integer program with integer variables $x_1, \ldots, x_6$. Let

$$\mathcal{S} := \{127, 289, 1310, 2754\}.$$

We want to add constraints and/or variables to the IP that enforce that the $x_1 + \ldots + x_6$ is in $\mathcal{S}$. How?

- Add binary variables $y_{127}, y_{289}, y_{1310}, y_{2754}$, one for each $i \in \mathcal{S}$.
- Want: Exactly one of these variables is 1 in a feasible solution.

Quiz

Given an integer program with integer variables $x_1, \ldots, x_6$. Let

$$\mathcal{S} := \{127, 289, 1310, 2754\}.$$

We want to add constraints and/or variables to the IP that enforce that the $x_1 + \ldots + x_6$ is in $\mathcal{S}$. How?

- Add binary variables $y_{127}, y_{289}, y_{1310}, y_{2754}$, one for each $i \in \mathcal{S}$.
- Want: Exactly one of these variables is 1 in a feasible solution.
- If $y_n = 1$ for $n \in \mathcal{S}$ then $\sum_{i=1}^{6} x_i = n$

Quiz

Add the following constraints:

$$y_{127} + y_{289} + y_{1310} + y_{2754} = 1$$
$$\sum_{i=1}^{6} x_i = \sum_{i \in \mathcal{S}} i y_i$$
$$0 \leq y_i \leq 1, \quad y_i \text{ integer} \quad \forall i \in \mathcal{S}$$

Quiz

Add the following constraints:

$$
\begin{aligned}
& y_{127} + y_{289} + y_{1310} + y_{2754} = 1 \\
& \sum_{i=1}^{6} x_i = \sum_{i \in \mathcal{S}} i y_i \\
& 0 \leq y_i \leq 1, \quad y_i \text{ integer} \quad \forall i \in \mathcal{S}
\end{aligned}
$$

Why is the resulting IP correct?

Recap:

- An integer program is obtained by adding integrality constraints for some/all of the variables to an LP.

Recap:

- An integer program is obtained by adding integrality constraints for some/all of the variables to an LP.
- An algorithm is efficient if its running time can be bounded by a polynomial of the input size of the instance.

Recap:

- An integer program is obtained by adding integrality constraints for some/all of the variables to an LP.
- An algorithm is efficient if its running time can be bounded by a polynomial of the input size of the instance.
- While LPs admit efficient algorithms, IPs are unlikely to have efficient algorithms. Thus, whenever possible, formulate a problem as an LP!

Recap:

- An integer program is obtained by adding integrality constraints for some/all of the variables to an LP.
- An algorithm is efficient if its running time can be bounded by a polynomial of the input size of the instance.
- While LPs admit efficient algorithms, IPs are unlikely to have efficient algorithms. Thus, whenever possible, formulate a problem as an LP!
- Variables that can take value 0 or 1 only are called binary.

Recap:

- An integer program is obtained by adding integrality constraints for some/all of the variables to an LP.
- An algorithm is efficient if its running time can be bounded by a polynomial of the input size of the instance.
- While LPs admit efficient algorithms, IPs are unlikely to have efficient algorithms. Thus, whenever possible, formulate a problem as an LP!
- Variables that can take value 0 or 1 only are called binary.
- Binary variables are useful for expressing logical conditions.