# ISE 426
# Optimization models and applications

Lecture 3 — September 3, 2015

- Upper & lower bounds
- Relaxations: an example
- Linear programming

# What relaxations are for

- If $\mathbf{P}'$ is a relaxation of a problem $\mathbf{P}$, then the global optimum of $\mathbf{P}'$ is $\leq$ the global optimum of $\mathbf{P}$.
- Hence, any relaxation $\mathbf{P}'$ of $\mathbf{P}$ provides a lower bound on $\mathbf{P}$.

$\Rightarrow$ If a problem $\mathbf{P}$ is difficult but a relaxation $\mathbf{P}'$ of $\mathbf{P}$ is easier to solve than $\mathbf{P}$ itself, we can still try and solve $\mathbf{P}'$: (i) we get a lower bound and (ii) the solution of $\mathbf{P}'$ may help solve $\mathbf{P}$.

# The Knapsack problem

At a flea market in Rome, you spot $n$ objects (old pictures, a vessel, rusty medals...) that you could re-sell in your antique shop for about double the price.

- You want these objects to pay for your flight ticket to Rome, which cost $C$.
- Also, your backpack can carry all of them, but you don't want it heavy, so you want to buy the objects that will load your backpack as little as possible.

How do you solve this problem?

# The Knapsack problem

Each object $i = 1, 2 \ldots, n$ has a price $p_i > 0$ and a weight $w_i > 0$.

- Variables: one variable $x_i$ for each $i = 1, 2 \ldots, n$.

$\Rightarrow$ $x_i$ is a "yes/no" variable: either you take the $i$-th object ($x_i = 1$) or you do not ($x_i = 0$).

- Constraint: total revenue must be at least $C$

  (As you'll double the price when selling them at your store, the revenue for each object is exactly $p_i$)

- Objective function: the total weight

# Your first (non-trivial) optimization model

$$
\begin{aligned}
\mathbf{P}:\min \quad & \textstyle\sum_{i=1}^{n} w_i x_i \\
& \textstyle\sum_{i=1}^{n} p_i x_i \geq C \\
& x_i \in \{0,1\} \quad \forall i = 1,2,\ldots,n
\end{aligned}
$$

Nonconvex!

# Your first (non-trivial) optimization model

$$
\begin{aligned}
\mathbf{P}: \min \quad & \textstyle\sum_{i=1}^{n} w_i x_i \\
& \textstyle\sum_{i=1}^{n} p_i x_i \geq C \\
& x_i \in \{0,1\} \quad \forall i = 1,2,\ldots,n
\end{aligned}
$$

Nonconvex! Relaxation #1:

$$
\begin{aligned}
\mathbf{R1}: \min \quad & \textstyle\sum_{i=1}^{n} w_i x_i \\
& x_i \in \{0,1\} \quad \forall i = 1,2,\ldots,n
\end{aligned}
$$

This relaxation gives us $x_i = 0$ for all $i = 1, 2, \ldots, n$, and a lower bound of $\sum_{i=1}^{n} w_i x_i = 0$. Not so great...

# Your first (non-trivial) optimization model

$$\begin{aligned}\mathbf{P}: \min \quad & \textstyle\sum_{i=1}^n w_i x_i \\ & \textstyle\sum_{i=1}^n p_i x_i \geq C \\ & x_i \in \{0,1\} \quad \forall i = 1,2,\ldots,n\end{aligned}$$

Nonconvex! Relaxation #1:

$$\begin{aligned}\mathbf{R1}: \min \quad & \textstyle\sum_{i=1}^n w_i x_i \\ & x_i \in \{0,1\} \quad \forall i = 1,2,\ldots,n\end{aligned}$$

This relaxation gives us $x_i = 0$ for all $i = 1,2,\ldots,n$, and a lower bound of $\sum_{i=1}^n w_i x_i = 0$. Not so great... Relaxation #2:

$$\begin{aligned}\mathbf{R2}: \min \quad & \textstyle\sum_{i=1}^n w_i x_i \\ & \textstyle\sum_{i=1}^n p_i x_i \geq C \\ & 0 \leq x_i \leq 1 \quad \forall i = 1,2,\ldots,n\end{aligned}$$

By relaxing integrality we admit fractions of objects.
It is as if we pulverized object and took some spoonful of each.
Nonsense? It's a relaxation, and it gives a lower bound.

# Example

Suppose there are $n = 9$ objects and $C = 70$.

| $i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| $p_i$ | 27 | 24 | 8 | 35 | 29 | 8 | 31 | 18 | 12 |
| $w_i$ | 3 | 2 | 2 | 4 | 5 | 4 | 3 | 1 | 4 |

# Example

Suppose there are $n = 9$ objects and $C = 70$.

| $i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| $p_i$ | 27 | 24 | 8 | 35 | 29 | 8 | 31 | 18 | 12 |
| $w_i$ | 3 | 2 | 2 | 4 | 5 | 4 | 3 | 1 | 4 |

A local optimum is 8, solution is $(1, 1, 1, 0, 0, 0, 0, 1, 0)$.

# Example

Suppose there are $n = 9$ objects and $C = 70$.

| $i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| $p_i$ | 27 | 24 | 8 | 35 | 29 | 8 | 31 | 18 | 12 |
| $w_i$ | 3 | 2 | 2 | 4 | 5 | 4 | 3 | 1 | 4 |

A local optimum is 8, solution is $(1, 1, 1, 0, 0, 0, 0, 1, 0)$.

R#1: lower bound is 0, solution is $(0, 0, 0, 0, 0, 0, 0, 0, 0)$.

# Example

Suppose there are $n = 9$ objects and $C = 70$.

| $i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| $p_i$ | 27 | 24 | 8 | 35 | 29 | 8 | 31 | 18 | 12 |
| $w_i$ | 3 | 2 | 2 | 4 | 5 | 4 | 3 | 1 | 4 |

A local optimum is 8, solution is $(1, 1, 1, 0, 0, 0, 0, 1, 0)$.

R#1: lower bound is 0, solution is $(0, 0, 0, 0, 0, 0, 0, 0, 0)$.

R#2: lower bound is 5.71, solution is $(0, 1, 0, 0, 0, 0.903, 1, 0)$.

# Example

Suppose there are $n = 9$ objects and $C = 70$.

| $i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| $p_i$ | 27 | 24 | 8 | 35 | 29 | 8 | 31 | 18 | 12 |
| $w_i$ | 3 | 2 | 2 | 4 | 5 | 4 | 3 | 1 | 4 |

A local optimum is 8, solution is $(1, 1, 1, 0, 0, 0, 0, 1, 0)$.

R#1: lower bound is 0, solution is $(0, 0, 0, 0, 0, 0, 0, 0, 0)$.

R#2: lower bound is 5.71, solution is $(0, 1, 0, 0, 0, 0.903, 1, 0)$.

Global optimum is 6, solution is $(0, 1, 0, 0, 0, 0, 1, 1, 0)$.

# To recap

- convex problems are good
- if model is nonconvex, look for a (possibly convex) relaxation
- use it to get a lower bound!

# Linear Programming

# Linear programming

Consider the optimization problem:

$$
\begin{array}{lll}
\mathbf{P}: & \min & \sum_{i=1}^{n} c_i x_i \\
& & \sum_{i=1}^{n} a_{ji} x_i \geq b_j \quad \forall j = 1, 2, \ldots, m \\
& & l_i \leq x_i \leq u_i \qquad \forall i = 1, 2, \ldots, n,
\end{array}
$$

with $n$ variables and $m + n$ constraints. Problems like $\mathbf{P}$ are called **Linear Programming** (LP) problems.
They are often written in matricial form:

$$
\begin{array}{lll}
\mathbf{P}: & \min & c^T x \\
& & Ax \geq b \\
& & l \leq x \leq u
\end{array}
$$

$A$ is the *coefficient matrix*,

# Linear programming

Consider the optimization problem:

$$
\mathbf{P}: \quad \min \quad \begin{array}{ll} \sum_{i=1}^{n} c_i x_i & \\ \sum_{i=1}^{n} a_{ji} x_i \geq b_j & \forall j = 1, 2, \ldots, m \\ l_i \leq x_i \leq u_i & \forall i = 1, 2, \ldots, n, \end{array}
$$

with $n$ variables and $m + n$ constraints. Problems like $\mathbf{P}$ are called **Linear Programming** (LP) problems.
They are often written in matricial form:

$$
\mathbf{P}: \quad \min \quad \begin{array}{l} c^T x \\ Ax \geq b \\ l \leq x \leq u \end{array}
$$

$A$ is the *coefficient matrix*, $b$ is the *right-hand side vector*, and $c$ is the *objective coefficient vector*. We call $l_i$ and $u_i$ *lower* and *upper* bound on variable $x_i$. They don't need to be finite.

LP problems are convex, therefore they are "easy".

# Example

You are cast on an island in the middle of the Pacific, and the only source of food is a well-known restaurant.

# Example

You are cast on an island in the middle of the Pacific, and the only source of food is a well-known restaurant. Here's the menu:

| | | | |
|---|---|---|---|
| QP: | Quarter Pounder | FR: | Fries (small) |
| MD: | McLean Deluxe | SM: | Sausage McMuffin |
| BM: | Big Mac | 1M: | 1% Lowfat Milk |
| FF: | Filet-O-Fish | OJ: | Orange Juice |
| MC: | McGrilled Chicken | | |

Each food has a different combination of nutrient (proteins, Vitamin A, Iron, etc.) and a cost. You want to

- get the necessary nutrients every day (constraint!)
- minimize the total cost of the foods (objective function)
- what are the variables?

# Nutrients

| | QP | MD | BM | FF | MC | FR | SM | 1M | OJ | |
|---|---|---|---|---|---|---|---|---|---|---|
| Cost | 1.84 | 2.19 | 1.84 | 1.44 | 2.29 | 0.77 | 1.29 | 0.60 | 0.72 | Req'd |
| Prot | 28 | 24 | 25 | 14 | 31 | 3 | 15 | 9 | 1 | 55 |
| VitA | 15% | 15% | 6% | 2% | 8% | 0% | 4% | 10% | 2% | 100% |
| VitC | 6% | 10% | 2% | 0% | 15% | 15% | 0% | 4% | 120% | 100% |
| Calc | 30% | 20% | 25% | 15% | 15% | 0% | 20% | 30% | 2% | 100% |
| Iron | 20% | 20% | 20% | 10% | 8% | 2% | 15% | 0% | 2% | 100% |
| Cals | 510 | 370 | 500 | 370 | 400 | 220 | 345 | 110 | 80 | 2000 |
| Carb | 34 | 35 | 42 | 38 | 42 | 26 | 27 | 12 | 20 | 350 |

# Model

Define $F = \{QP, MD, BM, FF, MC, FR, SM, 1M, OJ\}$

# Model

Define $F = \{QP, MD, BM, FF, MC, FR, SM, 1M, OJ\}$ and
$N = \{\text{Prot}, \text{VitA}, \text{VitC}, \text{Calc}, \text{Iron}, \text{Cals}, \text{Carb}\}$.

- define variable $x_i$ as the amount of food $i$ you will buy every day ($i \in F$)

# Model

Define $F = \{QP, MD, BM, FF, MC, FR, SM, 1M, OJ\}$ and $N = \{\text{Prot}, \text{VitA}, \text{VitC}, \text{Calc}, \text{Iron}, \text{Cals}, \text{Carb}\}$.

- define variable $x_i$ as the amount of food $i$ you will buy every day ($i \in F$)
- define parameters:
  - $c_i$ is the cost per unit of food $i$
  - $a_{ij}$ is the amount of nutrient $j \in N$ per unit of food $i \in F$
  - $b_j$ is the amount of nutrient $j \in N$ required every day

Then the optimization model is an LP model:

$$
\begin{aligned}
\min \quad & c^T x \\
& Ax \geq b \\
& l \leq x \leq u
\end{aligned}
$$

# Overall model

$$
\begin{array}{lrrrrrrrrrl}
\min & 1.84x_{qp} & +2.19x_{md} & +1.84x_{bm} & +1.44x_{ff} & +2.29x_{mc} & +0.77x_{fr} & +1.29x_{sm} & +0.60x_{1m} & +0.72x_{oj} & \\
\\
(Prot) & 28x_{qp} & +24x_{md} & +25x_{bm} & +14x_{ff} & +31x_{mc} & +3x_{fr} & +15x_{sm} & +9x_{1m} & +1x_{oj} & \geq 55 \\
(VitA) & 15x_{qp} & +15x_{md} & +6x_{bm} & +2x_{ff} & +8x_{mc} & & +4x_{sm} & +10x_{1m} & +2x_{oj} & \geq 100 \\
(VitC) & 6x_{qp} & +10x_{md} & +2x_{bm} & & +15x_{mc} & +15x_{fr} & & +4x_{1m} & +120x_{oj} & \geq 100 \\
(Calc) & 30x_{qp} & +20x_{md} & +25x_{bm} & +15x_{ff} & +15x_{mc} & & +20x_{sm} & +30x_{1m} & +2x_{oj} & \geq 100 \\
(Iron) & 20x_{qp} & +20x_{md} & +20x_{bm} & +10x_{ff} & +8x_{mc} & +2x_{fr} & +15x_{sm} & & +2x_{oj} & \geq 100 \\
(Cals) & 510x_{qp} & +370x_{md} & +500x_{bm} & +370x_{ff} & +400x_{mc} & +220x_{fr} & +345x_{sm} & +110x_{1m} & +80x_{oj} & \geq 2000 \\
(Carb) & 34x_{qp} & +35x_{md} & +42x_{bm} & +38x_{ff} & +42x_{mc} & +26x_{fr} & +27x_{sm} & +12x_{1m} & +20x_{oj} & \geq 350
\end{array}
$$

Nonnegativity constraint: $x_i \geq 0, i \in F$.

# Another example

The manager of a post office is hiring new employees

They can be full or part time. The part time can be 1% to 99% — this is only to simplify the problem. Rules:

- at least these many employees each day of the week:

| day | S | M | T | W | Th | F | Sa |
|---|---|---|---|---|---|---|---|
| # empl. | 11 | 17 | 13 | 15 | 19 | 14 | 16 |

- (state regulations impose) that an employee works five days in a row and then receives two days off
- that the number of employees is minimum

What are the **variables** of the problem?

# Another example

The manager of a post office is hiring new employees

They can be full or part time. The part time can be 1% to 99% — this is only to simplify the problem. Rules:

- at least these many employees each day of the week:

| day | S | M | T | W | Th | F | Sa |
|---|---|---|---|---|---|---|---|
| # empl. | 11 | 17 | 13 | 15 | 19 | 14 | 16 |

- (state regulations impose) that an employee works five days in a row and then receives two days off
- that the number of employees is minimum

What are the **variables** of the problem?

- the number of employees working each day?
- the total number of employees to hire?

What do I (as a boss) want to know at the end?

# What to we want to know?

- If an employee works on Thu, his/her work days can be
  - Thu, Fri, Sat, Sun, Mon, or
  - Wed, Thu, Fri, Sat, Sun, or
  - Tue, Wed, Thu, Fri, Sat, or
  - Mon, Tue, Wed, Thu, Fri, or
  - Sun, Mon, Tue, Wed, Thu.

$\Rightarrow$ We don't know when he/she started his working shift.

- It is the variable we are looking for!
- Actually, we are only interested in…

## What to we want to know?

- If an employee works on Thu, his/her work days can be
  - Thu, Fri, Sat, Sun, Mon, or
  - Wed, Thu, Fri, Sat, Sun, or
  - Tue, Wed, Thu, Fri, Sat, or
  - Mon, Tue, Wed, Thu, Fri, or
  - Sun, Mon, Tue, Wed, Thu.

$\Rightarrow$ We don't know when he/she started his working shift.

- It is the variable we are looking for!
- Actually, we are only interested in...

  **the number of employees starting on a certain day**
- Define it as variable $x_i$, with
  $i \in \{\text{Sun}, \text{Mon}, \text{Tue}, \text{Wed}, \text{Thu}, \text{Fri}, \text{Sat}\}$.

# Now that we know what we are looking for…

We have variables. We can write constraints & objective f.

- constraint #1: there must be 19 employees on Thursdays.

# Now that we know what we are looking for…

We have variables. We can write constraints & objective f.

- constraint #1: there must be 19 employees on Thursdays.

$$x_{Thu} + x_{Wed} + x_{Tue} + x_{Mon} + x_{Sun} \geq 19$$

- constraint #2: an employee works five consecutive days and then receives two days off.

# Now that we know what we are looking for…

We have variables. We can write constraints & objective f.

- constraint #1: there must be 19 employees on Thursdays.

$$x_{Thu} + x_{Wed} + x_{Tue} + x_{Mon} + x_{Sun} \geq 19$$

- constraint #2: an employee works five consecutive days and then receives two days off. This is already included in the definition of our variables **and** in the above constraint.

# Now that we know what we are looking for…

We have variables. We can write constraints & objective f.

- constraint #1: there must be 19 employees on Thursdays.

$$x_{Thu} + x_{Wed} + x_{Tue} + x_{Mon} + x_{Sun} \geq 19$$

- constraint #2: an employee works five consecutive days and then receives two days off. This is already included in the definition of our variables **and** in the above constraint.
- objective function: the total number of employees (to be minimized).

# Now that we know what we are looking for…

We have variables. We can write constraints & objective f.

- constraint #1: there must be 19 employees on Thursdays.

$$x_{Thu} + x_{Wed} + x_{Tue} + x_{Mon} + x_{Sun} \geq 19$$

- constraint #2: an employee works five consecutive days and then receives two days off. This is already included in the definition of our variables **and** in the above constraint.
- objective function: the total number of employees (to be minimized).

⇒ number of employees starting on Monday, plus those starting on Tuesday, etc.

- we can sum them up because they define disjoint sets of employees: if one starts working on Thursday, he doesn't start on Friday…

# The model

$$
\begin{array}{rccccccccc}
\min & x_{Sun} & +x_{Mon} & +x_{Tue} & +x_{Wed} & +x_{Thu} & +x_{Fri} & +x_{Sat} & \\
\\
(Sun) & x_{Sun} & & & +x_{Wed} & +x_{Thu} & +x_{Fri} & +x_{Sat} & \geq 11 \\
(Mon) & x_{Sun} & +x_{Mon} & & & +x_{Thu} & +x_{Fri} & +x_{Sat} & \geq 17 \\
(Tue) & x_{Sun} & +x_{Mon} & +x_{Tue} & & & +x_{Fri} & +x_{Sat} & \geq 13 \\
(Wed) & x_{Sun} & +x_{Mon} & +x_{Tue} & +x_{Wed} & & & +x_{Sat} & \geq 15 \\
(Thu) & x_{Sun} & +x_{Mon} & +x_{Tue} & +x_{Wed} & +x_{Thu} & & & \geq 19 \\
(Fri) & & x_{Mon} & +x_{Tue} & +x_{Wed} & +x_{Thu} & +x_{Fri} & & \geq 14 \\
(Sat) & & & x_{Tue} & +x_{Wed} & +x_{Thu} & +x_{Fri} & +x_{Sat} & \geq 16 \\
\\
& x_{Sun}, & x_{Mon}, & x_{Tue}, & x_{Wed}, & x_{Thu}, & x_{Fri}, & x_{Sat} & \geq 0
\end{array}
$$

# The solution

LP: with part-time contracts (here $\frac{1}{3}$-time contracts used).

IP: solution with only full-time contracts.

| | Sun | Mon | Tue | Wed | Thu | Fri | Sat | Total |
|---|---|---|---|---|---|---|---|---|
| LP | 5 | $1+\frac{1}{3}$ | $5+\frac{1}{3}$ | 0 | $7+\frac{1}{3}$ | 0 | $3+\frac{1}{3}$ | $22+\frac{1}{3}$ |
| IP | 4 | 1 | 6 | 0 | 8 | 0 | 4 | 23 |