ISyE/Math/CS/Stat 525
Linear Optimization

Spring 2021 Course Introduction

Prof. Alberto Del Pia
University of Wisconsin-Madison

Based on the book *Introduction to Linear Optimization* by D. Bertsimas and J.N. Tsitsiklis

Linear Optimization

Optimization

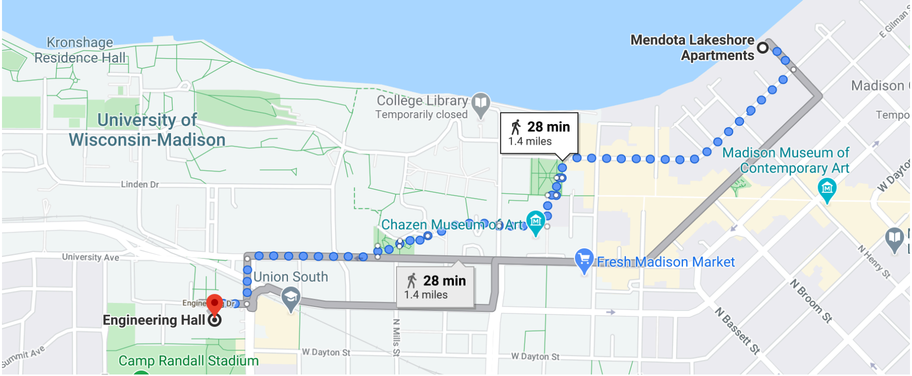
Optimization

This course is about taking   decisions.

Optimization

This course is about taking OPTIMAL decisions.

Common sense vs Optimization



Common sense is often sufficient in everyday life.

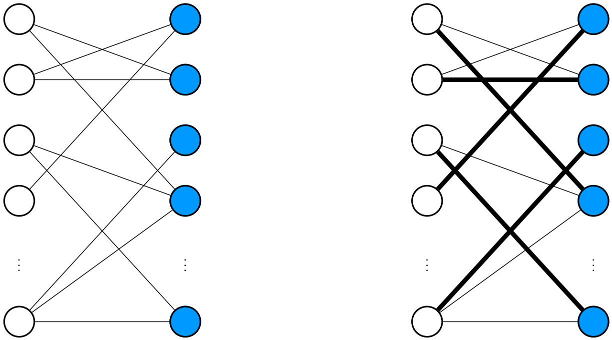
Common sense vs Optimization

Common sense is not sufficient to make decisions in complex systems.

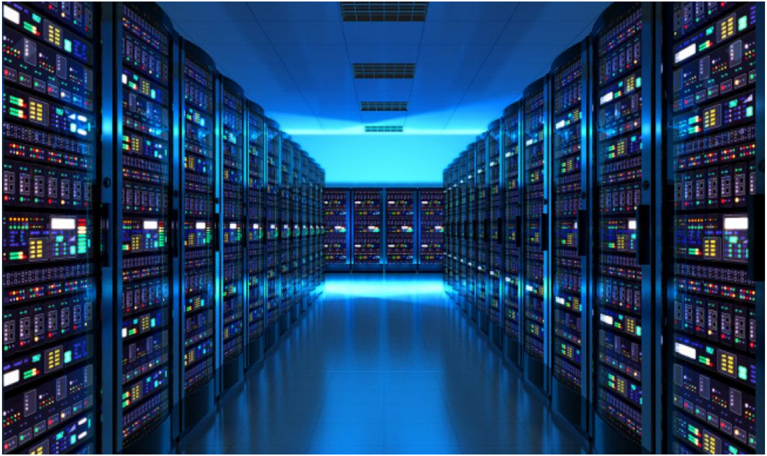
Common sense vs Optimization

A simple example

- Find the best assignment of 70 people to 70 jobs.



Can growing computing power help?



Can growing computing power help?

- Simple: check each possible assignment and select the best.

Can growing computing power help?

- Simple: check each possible assignment and select the best.
- But how long will it take? Let's use an IBM BlueGene/L:

Can growing computing power help?

- Simple: check each possible assignment and select the best.
- But how long will it take? Let's use an IBM BlueGene/L:

$\#people$	Solutions to check	Time
3	6	0
10	3.6×10^6	0
15	1.3×10^{12}	1 sec
30		
70		

Can growing computing power help?

- Simple: check each possible assignment and select the best.
- But how long will it take? Let's use an IBM BlueGene/L:

#*people*	Solutions to check	Time
3	6	0
10	3.6×10^6	0
15	1.3×10^{12}	1 sec
30	2.7×10^{32}	
70		

Can growing computing power help?

- Simple: check each possible assignment and select the best.
- But how long will it take? Let's use an IBM BlueGene/L:

#*people*	Solutions to check	Time
3	6	0
10	3.6×10^6	0
15	1.3×10^{12}	1 sec
30	2.7×10^{32}	14 billion years[(a)]
70		

[(a)] That's four times the age of the universe!

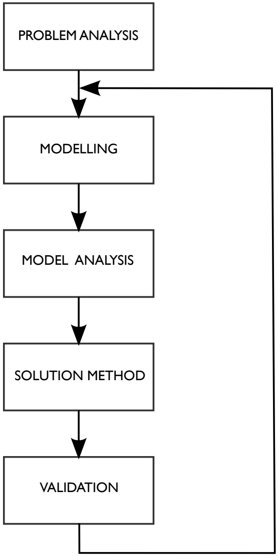
Can growing computing power help?

- Simple: check each possible assignment and select the best.
- But how long will it take? Let's use an IBM BlueGene/L:

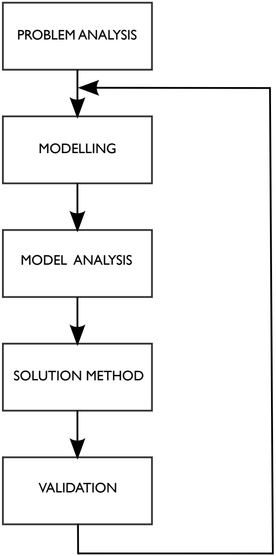
#*people*	Solutions to check	Time
3	6	0
10	3.6×10^6	0
15	1.3×10^{12}	1 sec
30	2.7×10^{32}	14 billion years[a]
70	1.2×10^{100} [b]	

[a] That's four times the age of the universe!
[b] That's more than the number of particles in the observable universe!

Modelling Approach

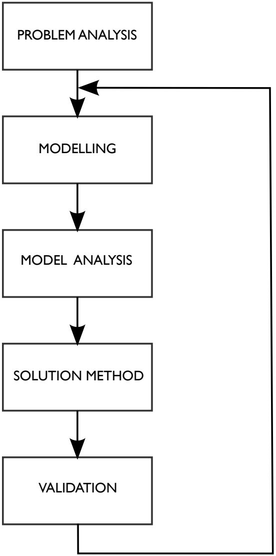


Modelling Approach



real-world relationships $\Leftrightarrow$ math relationships

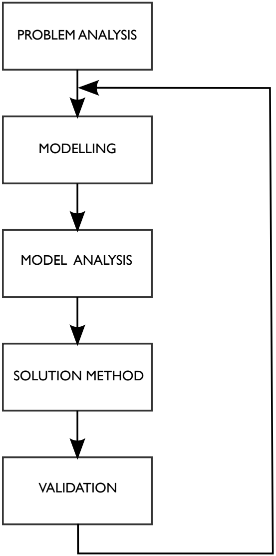
Modelling Approach



real-world relationships $\Leftrightarrow$ math relationships

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in S \end{array}$$

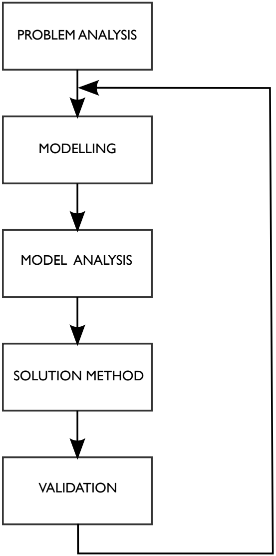
Modelling Approach



real-world relationships $\Leftrightarrow$ math relationships

$$
\begin{array}{lll}
\min & f(x) & \text{objective function} \\
\text{s.t.} & x \in S & \text{feasible set}
\end{array}
$$

Modelling Approach

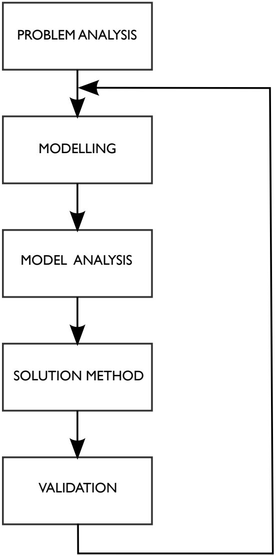


real-world relationships $\Leftrightarrow$ math relationships

$$\begin{array}{lll} \min & f(x) & \text{objective function} \\ \text{s.t.} & x \in S & \text{feasible set} \end{array}$$

study the mathematical properties of the model

Modelling Approach



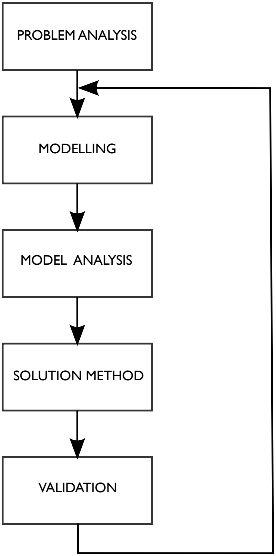
real-world relationships $\Leftrightarrow$ math relationships

$$\begin{array}{lll} \min & f(x) & \text{objective function} \\ \text{s.t.} & x \in S & \text{feasible set} \end{array}$$

study the mathematical properties of the model

apply an algorithm to compute an optimal solution x^*

Modelling Approach



real-world relationships $\Leftrightarrow$ math relationships

min	$f(x)$	objective function
s.t.	$x \in S$	feasible set

study the mathematical properties of the model

apply an algorithm to compute an optimal solution x^*

verification and simulation

Optimization and Programming

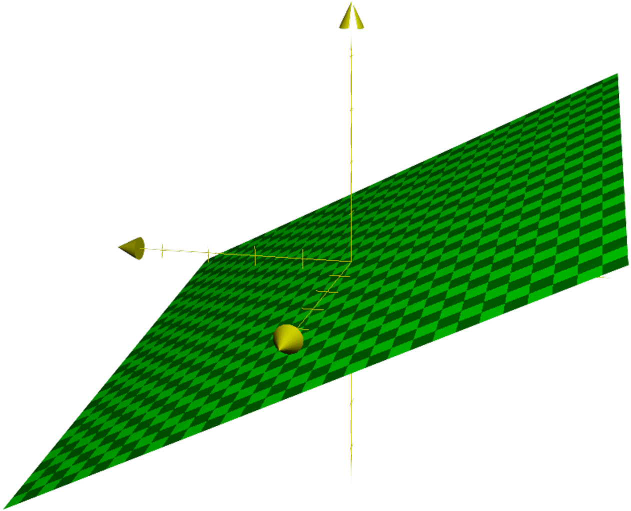
For us Optimization $\equiv$ Programming.

- "Programming" was used in the 1930s, before computers.
- It means "Planning", based on the military definition of "Program".
- Lately "Programming" is getting replaced by "Optimization".

Linear Optimization

Linear

Linear functions



Linear functions

A function $f: \mathbb{R}^n \to \mathbb{R}$ is linear if:

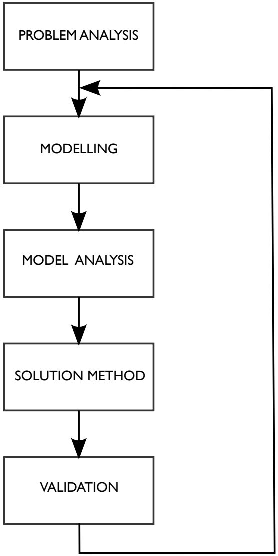
1. $f(x+y) = f(x) + f(y) \quad \forall x, y \in \mathbb{R}^n$
2. $f(\lambda x) = \lambda f(x) \quad \forall x \in \mathbb{R}^n, \forall \lambda \in \mathbb{R}$

Equivalently, f is linear if it can be written as:

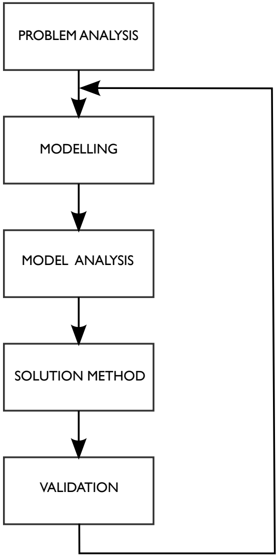
$$c'x = \sum_{i=1}^{n} c_i x_i = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n,$$

where $c_1, c_2, \ldots, c_n$ are real numbers.

Linear Optimization

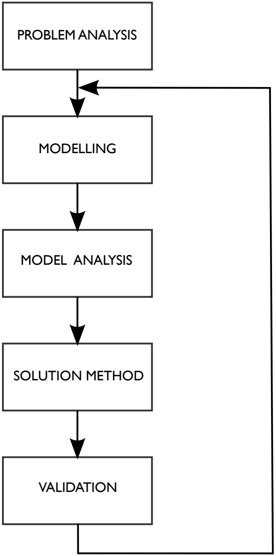


Linear Optimization



real-world relationships $\Leftrightarrow$ math relationships

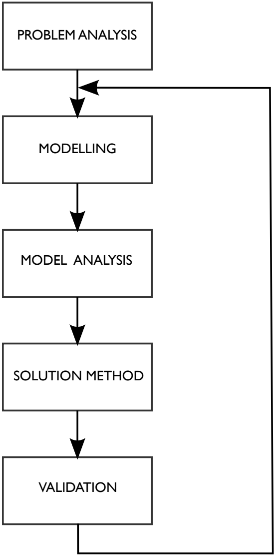
Linear Optimization



real-world relationships $\Leftrightarrow$ math relationships

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in S \end{array}$$

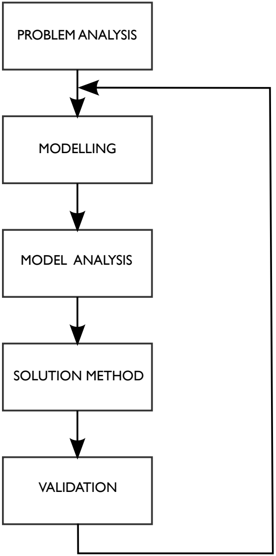
Linear Optimization



real-world relationships $\Leftrightarrow$ math relationships

min	$f(x)$	LINEAR objective function
s.t.	$x \in S$	LINEAR inequalities

Linear Optimization



real-world relationships $\Leftrightarrow$ math relationships

min	$c'x$	LINEAR objective function
s.t.	$Ax \geq b$	LINEAR inequalities

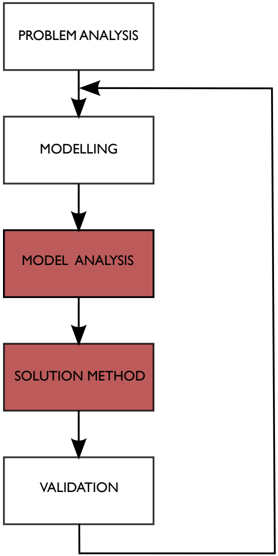
A Linear Programming (LP) problem

$$
\begin{array}{rl}
\text{minimize} & 2x_1 - x_2 + 4x_3 \\
\text{subject to} & x_1 + x_2 + x_4 \leq 2 \\
& 3x_2 - x_3 = 5 \\
& x_3 + x_4 \geq 3 \\
& x_1 \geq 0 \\
& x_3 \leq 0.
\end{array}
$$

- Strict inequalities like $x_3 + x_4 > 3$ are not allowed!

Algorithms for Linear Programming

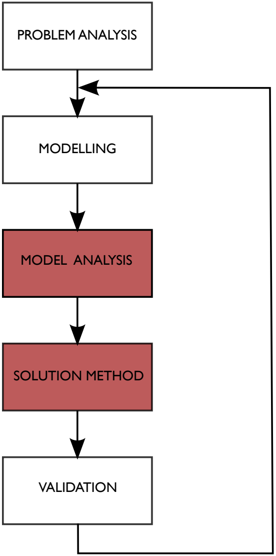
Algorithms for LP



real-world relationships $\Leftrightarrow$ math relationships

$$\begin{array}{lll} \min & c'x & \text{LINEAR objective function} \\ \text{s.t.} & Ax \geq b & \text{LINEAR inequalities} \end{array}$$

Algorithms for LP

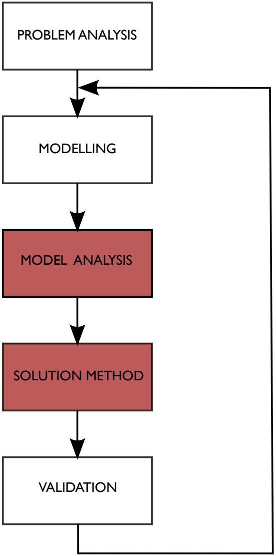


real-world relationships $\Leftrightarrow$ math relationships

min	$c'x$	LINEAR objective function
s.t.	$Ax \geq b$	LINEAR inequalities

- optimality conditions
- duality

Algorithms for LP



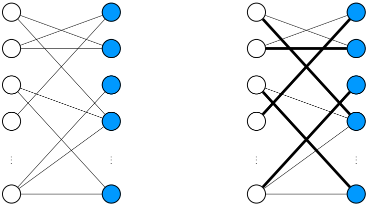
real-world relationships $\Leftrightarrow$ math relationships

$$\begin{array}{lll} \min & c'x & \text{LINEAR objective function} \\ \text{s.t.} & Ax \geq b & \text{LINEAR inequalities} \end{array}$$

- optimality conditions
- duality

SIMPLEX ALGORITHM

Can growing computing power help?



#*people*	Solutions to check	Time
3	6	0
10	3.6×10^6	0
15	1.3×10^{12}	1 sec
30	2.7×10^{32}	14 billion years[a]
70	1.2×10^{100} [b]	

[a] That's four times the age of the universe!
[b] That's more than the number of particles in the observable universe!

Can growing computing power help?

- It takes only a moment to find the optimum solution by posing the problem as a LP problem and applying the simplex algorithm.
- The theory behind LP drastically reduces the number of possible solutions that must be checked.
- LP is now a technology: It can be solved efficiently in theory and in practice.

LP is everywhere!

- Many practical problems in Operations Research can be expressed as LP problems.
  - Application areas include engineering, computer science, physics, biology, finance, and economics.
  - Some examples: Network problems, scheduling, microeconomics, company management, planning, production, transportation, technology, military operations.
- A number of algorithms for other types of optimization problems work by solving LP problems as sub-problems.

Course details

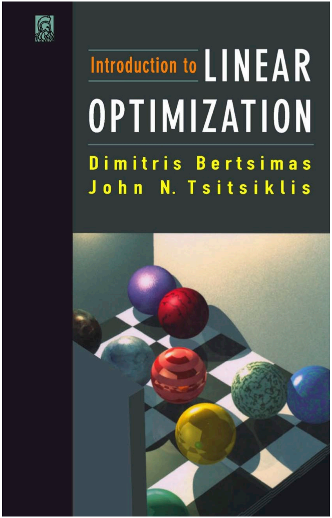
Purpose of this course

- Solid understanding of Linear Optimization.
- Special attention to theory.
- We will study the available solution methods:
  - HOW they work.
  - WHY they work.
- Our focus is on the interplay between algebra and geometry.
  - Ideas require a geometric view.
  - Problems are solved by algorithms, which can only be described algebraically.

What this course is NOT about:

- We won't spend time on software or on implementation.
- We won't spend much time on how to model problems as LP problems. (See ISyE 323: Operations Research - Deterministic Modeling.)

Recommended textbook



Introduction to Linear Optimization, by Dimitris Bertsimas and John N. Tsitsiklis (1997).

- Developed for a course at M.I.T.
- Rent $\sim$ \$40.
- Buy used $\sim$ \$70.
- Buy new $\sim$ \$85.

What we will cover

Ch. 1: Introduction §1.1–1.4

- The LP problem and examples.

Ch. 2: The geometry of linear programming §2.1–2.8

- Polyhedra and their geometric and algebraic properties.

Ch. 3: The simplex method §3.1–3.7 and 1.6

- The main method used to solve LP problems.

Ch. 4: Duality theory §4.1–4.6

- A new LP problem and its relation with the original one.

Ch. 7: Network flow problems §7.1–7.3

- Solving network flow problems with the simplex algorithm.

Ch. 8: Complexity of linear programming and the ellipsoid method §8.1–8.5

- The first polynomial time algorithm for LP problems.

Warning on course difficulty

This is a challenging course $\Rightarrow$ Much of the material in this course is deep.

- Requires serious math.
- Proofs will be done by me and by you.

Warning on course difficulty

- “Hard but rewarding”

Prerequisites

Essential background that will be assumed:

- Working knowledge of linear algebra.
  See Section 1.5 Linear algebra background and notation of the textbook (6 pages).
  - E.g., set theoretic notation, vectors and matrices, matrix inversion, subspaces and bases, affine subspaces, linear independence, and the rank of a matrix.
- Basic proof techniques.
  See Introduction to mathematical arguments by Michael Hutchings in Canvas.
- Mathematical notation.
  See Math symbols cheat sheet in Canvas.

Prerequisites

Essential background that will be assumed:

- Working knowledge of linear algebra.
  See Section 1.5 Linear algebra background and notation of the textbook (6 pages).
  - E.g., set theoretic notation, vectors and matrices, matrix inversion, subspaces and bases, affine subspaces, linear independence, and the rank of a matrix.
- Basic proof techniques.
  See Introduction to mathematical arguments by Michael Hutchings in Canvas.
- Mathematical notation.
  See Math symbols cheat sheet in Canvas.

Now in Canvas:

- Assignment 0, to test your Math skills and Linear Algebra.
- You do not need to submit it since it will not be graded.

Other Optimization courses

- ISyE/Math/CS 425: Introduction to Combinatorial Optimization.
- CS/ECE/ISyE 524: Introduction to Optimization.
- CS/ISyE 526: Advanced Linear Programming.
- ISyE/Math/CS/Stat 726: Nonlinear Optimization I.
- ISyE/Math/CS 728: Integer Optimization.

If interested in research in Optimization, consider applying to our Graduate programs.

Technical details

Weekly schedule

Monday	*Tuesday*	*Wednesday*	*Thursday*	*Friday*
new videos online	Prof office hours	TA office hours	Prof office hours	TA office hours
12am	9–10am	10–11am	4–5pm	4–5pm

- Homework deadline on Saturday night every 2/3 weeks.

Monday	*Tuesday*	*Wednesday*	*Thursday*	*Friday*
new videos online	Prof office hours	TA office hours	Prof office hours	TA office hours
12am	9–10am	10–11am	4–5pm	4–5pm

Lectures

- Pre-recorded video lectures.
- Videos are topic based, of different lengths.
- Available in YouTube in the following playlist.
- Useful to subscribe and turn on notifications.

- Videos are concise (1:2 scale). Use pause!
- For live discussions come to office hours!

Monday	*Tuesday*	*Wednesday*	*Thursday*	*Friday*
new videos online	Prof office hours	TA office hours	Prof office hours	TA office hours
12am	9–10am	10–11am	4–5pm	4–5pm

Lectures

Topic	*Study on*	*# videos*	*total videos length*
Ch. 1	weeks 1, 2	4	2:09:25
Ch. 2	weeks 3, 4	8	2:53:42
Ch. 3 part 1	weeks 5, 6, 7	4	3:24:26
Ch. 3 part 2	weeks 8, 9	4	2:05:22
Ch. 4	weeks 10, 11	6	2:22:05
Ch. 7	weeks 12, 13	3	2:09:26
Ch. 8	week 14	5	1:21:00

Monday	*Tuesday*	*Wednesday*	*Thursday*	*Friday*
new videos online	Prof office hours	TA office hours	Prof office hours	TA office hours
12am	9–10am	10–11am	4–5pm	4–5pm

Prof office hours

- For discussions on the material presented in the lectures and related theoretical questions.
- Office hours take place in Zoom.
- Meeting ID: 946 4024 8032. See syllabus for Passcode.
- You can just use this link.

Monday	*Tuesday*	*Wednesday*	*Thursday*	*Friday*
new videos online	Prof office hours	TA office hours	Prof office hours	TA office hours
12am	9–10am	10–11am	4–5pm	4–5pm

TA office hours

- The TA is Bainian Hao <bhao8@wisc.edu>.
- For questions regarding exercises and assignments.
- Office hours take place in Zoom.
- Same Meeting ID, passcode, and link.

Homework

- There are 6 Homework assignments. One every 2/3 weeks.
- Due date is always Saturday at 11:59pm.

Homework #	*Topic*	*Available on*	*Due on*
1	Ch. 1	week 2	end of week 3
2	Ch. 2	week 4	end of week 5
3	Ch. 3 part 1	week 6	end of week 8
4	Ch. 3 part 2	week 9	end of week 10
5	Ch. 4	week 11	end of week 12
6	Ch. 7	week 13	end of week 14

Homework

- Homework assignments are posted in Canvas.
- 5 compulsory exercises per assignment.
- Some exercises in the assignments are specific to undergraduate and graduate students.
- Optional exercises for keen students.

- Strongly encouraged to work in groups of 2 (same type).
- Submit pdf in Canvas (one submission per group).

- Assignments will not be completely graded. 3/5 randomly selected.
- Complete solutions will be published in Canvas.

Grading

Components of grade:

- 40% Homework.
- 30% Midterm.
- 30% Final.

- Both exams will be uploaded in Canvas like an assignment.
- 2 hours to submit your solution in Canvas.
- You cannot work in groups.

- We need to find a 2-hour slot for the Midterm that is good for everyone. Please fill out the following poll by Jan. 30: http://whenisgood.net/cr3rgm5
- The final exam will take place in the course slot in final week, which is **May 6, 5:05pm–7:05pm.**

Canvas web page

https://canvas.wisc.edu/courses/230426

- Syllabus, lecture slides, introductory material, homework assignments and solutions, grades.
- Use “Discussions” section to post questions about the course, and to find a teammate for the assignments.
- I cannot give you access to the Canvas webpage. You are given access when enrolled.

- See syllabus in Canvas for more info: /files/525syllabus

About me...

- BS and MS, University of Padova, Italy, 2005.
- PhD, University of Padova, Italy, 2009.

About me...

- BS and MS, University of Padova, Italy, 2005.
- PhD, University of Padova, Italy, 2009.

- 2009 – 2010: Otto-von-Guericke University, Magdeburg, Germany.
- 2010 – 2013: ETH Zürich, Zürich, Switzerland.
- 2013 – 2014: IBM, Watson Research Center, New York.

About me...

- BS and MS, University of Padova, Italy, 2005.
- PhD, University of Padova, Italy, 2009.
- 2009 – 2010: Otto-von-Guericke University, Magdeburg, Germany.
- 2010 – 2013: ETH Zürich, Zürich, Switzerland.
- 2013 – 2014: IBM, Watson Research Center, New York.
- Since 2014: University of Wisconsin-Madison.
  - Wisconsin Institute for Discovery.
  - Dept. Industrial and Systems Engineering.
  - Dept. Mathematics.
  - Dept. Computer Sciences.

About me...

- BS and MS, University of Padova, Italy, 2005.
- PhD, University of Padova, Italy, 2009.
- 2009 – 2010: Otto-von-Guericke University, Magdeburg, Germany.
- 2010 – 2013: ETH Zürich, Zürich, Switzerland.
- 2013 – 2014: IBM, Watson Research Center, New York.
- Since 2014: University of Wisconsin-Madison.
  - Wisconsin Institute for Discovery.
  - Dept. Industrial and Systems Engineering.
  - Dept. Mathematics.
  - Dept. Computer Sciences.
- Research areas: Integer optimization, Combinatorial optimization, Machine learning.

About you...

Please complete the Student Survey quiz in Canvas by Jan. 30!

Questions about the course?

- Please post your questions in Canvas' "Discussions" section.