CO327 Deterministic OR Models (2021-Spring)

Course mechanics

Prerequisites, Syllabus, Instructor, Grading

Instructor: Andersen Ang

Combinatorics and Optimization, U.Waterloo, Canada

ms**x**ang@uwaterloo.ca, where $\mathbf{x} = \lfloor \pi \rfloor$

Homepage: angms.science

First draft: December 27, 2020 Last update: May 26, 2021

CO327 Deterministic OR Models (Non-Specialist Level)

- Period: 2021 spring (May 10 - August 5)

- Instructor: Andersen Ang (email: ms**x**ang@uwaterloo.ca, $\mathbf{x} = \lfloor \pi \rfloor$)
  - Office: MC 6032 (but I work from home!)
  - Office hours: flexible, email for an appointment

- Course pages
  - Waterloo LEARN: administrative stuff / announcement
  - Teams: online meeting / video
  - `https://angms.science/CO327.html`: course notes / assignments

- Course prerequisites
  - Check the department page
  - "Confident" with linear algebra
  - Not necessary:
    - "Confident" with multi-variable calculus
    - "Confident" with numerical computation programming
    - Basic knowledge of graph theory

Look at the course title

Deterministic OR Models

- Deterministic: no randomness
- OR: mathematical programming
- Models: this course focuses on **many** models

The what of this course

- Motivation: useful
  Learn how to formulate (a limited subset of) some real-life problems.
- Mathematics: theory
  Learn some theorems that 1) sounds cool, 2) are useful and 3) make you a better human.
- Methods: algorithms
  Learn computational tools to solve (a limited subset of) problems.
  - Computational tools: write code to call computer to solve problems.

The why of this course

Ideally speaking:

- (Possibility to) get Nobel Prize.
- (Possibility to) save the world.
- (Possibility to) be rich.
- ~~I teach this course this year.~~
- ~~Prepare for data science coding competition. e.g. Google HashCode~~
- ~~Sell your code to tech company.~~
- Google it yourself.

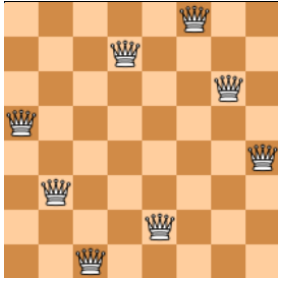
Reality:

- You don't know what course to pick / you pick this course by random.
- You are forced to pick this course.

Is this course easy?

- No answer, because "it depends".
- A sample of assignment question:
  Given an $N \times N$ chess board, your goal is to place N queens on the board such that no two queens "see each other":
  - They cannot be on the same row
  - They cannot be on the same column
  - They cannot be on the same diagonal

  Formulate this problem as a binary integer program.



Another example: formulate tower defence game as a *set covering problem*.

Syllabus

- In human language: How to formulate a problem as OR model and how to solve it.
- In fancy language
  - Linear programming (LP)
  - Linear Integer programming (LIP)
  - Geometry of LP: simplex and polytope
  - Fundamental theorem of linear programming
  - Lagrangian duality, dual problem
  - Optimality, sensitivity
  - Simplex algorithm, dual simplex algorithm
  - LP relaxation, integrality gap, cutting plane, branch-and-bound
  - Ellipsoid method, interior point method, Newton's method
  - Applications: Dieting, Transportation problem, Network flows, travelling salesman, decision making, set partition, work crew scheduling, fire station location problem, assignment problem, knapsack problem, multi-period investment, 2-player zero-Sum game, optimal control, optimal transport, Sudoku, N-Queen, urban planning, blablabla
  - A little bit of
    - quadratic programming, polynomial programming, nonliner programming
    - conic programming, semi-definite programming

Gray = not the main focus

Grading and course policy

- No "attendance policy".
  You are free to skip classes (but you need to do self-study).
- No final "sit-in" exam.
  Because it only tests your short-term memory but not understanding.
- Take-home assignments + take-home mid-term + take-home final.
- Scribing lectures.
  A training on making academic material.
  ~~Not because I am lazy to prepare it.~~
- Programming assignments: solve real-life problems.
  ~~Prepare for data science coding competition.~~
- Bonus: get bonus grade points if you catch me an error.
  Need to be non-typo error, describe the error, why it is wrong, and how to correct it.

Instructor: about me

- **Education**
  Undergrad in Electronic and Communication Engineering in HKU, Hong Kong
  M.Phil in Biomedical Engineering in HKU, Hong Kong
  Ph.D in Applied mathematics in UMONS, Mons, Belgium

- **Research**
  Topics in Nonnegative matrix-tensor factorizations, such as **continuous optimization**, and numerical linear algebra.

- CO327 is partly **discrete optimization** $\Longrightarrow$ slightly out of my expertise!

My teaching style

- Micro-lecture.
  Instead of a 400+ page notes, I teach piece-by-piece $\rightarrow$ easier to learn, easier to handle mentally.
- "Lecture" : high-level concepts and theory.
- "Tutorial": low-level know-how, step-by-step computations.
- Both lecture and tutorial are equally important!
- Solving the problems in assignments: it is also a learning process.
  Contents not covered in lecture $\rightarrow$ guided questions in assignments.
- I don't follow traditional syllabus because that is boring and outdated.

My assignment and assessment style

- Concept check / explanation-type questions.
- Question that requires computation by hand.
- Challenging questions for bonus points.
- Project-like assignment for computer assignment on real-life data.
- Mid-term and final all take-home.
- Discuss with your classmates are allowed and encouraged, but no direct copying.
- Also I will made up some questions that there is no way you can find the solution online.
- Or I will make questions from research papers so even you find them online you have a high chance can't understand them. But if you are managed to understand them: you get bonus points!
- Continuous assessment: there will be *many assignments*, but this is far better than *just 1 final*.

Formula on points

- Course grade points G (0-100) = a weighted sum of: {assignment points a_i, course scribing points s, midterm points m, final points f, project points p }, plus bonus points b.
- Late submission policy:

$$\text{final adjusted assignment points} = \text{assignment points} \times 0.8^D,$$

  D=number of days after due date.

- Tentative formula (may change!)

$$G = 0.4\mathbb{E}a_i + 0.1s + 0.15m + 0.15f + 0.2p + \min\{b, 10\}.$$

2021-July-9 Update: p (project), assignments 6,7 cancel
0.1s becomes 0.2s
0.15f becomes 0.25f

Scribing (10% course grade)

- What is scribing: taking careful notes during class and write them up.
- Usually in advanced course (postgrad level+), the original idea of scribing is to prepare notes for the course that do not have a good textbook.
- CO327 is very fundamental, then why scribing: a form of training on making academic material.
- What to scribe: theory part of the course. (Lecture 5 & 6 on Theory of LP and IP)
  So as to familiarize yourself with the theories!
  * You do not need to scribe other parts of the course.

How to scribe

- Take notes during class.
- Prepare your notes in LaTeX, written in complete sentences that would be *understandable by a student who might have missed class*, **not merely a transcription of the notes that you jotted down during class**.
- Learn how to "select information":
  - Do not put down administrative info (such as due dates for assignment) or my stupid jokes (if any) in your scribe notes.
  - Hide details that are less important.
  - Identify what is "important" or what is "difficult to understand", and then elaborate by including figure, example, tables (if appropriate) for better explanation.
- Installing LaTeX is not required: you can use Overleaf.
- Contact me if you have question(s).

How your scribe notes will be graded

- Grade points: 0 - 10.
  Base point = 1 if you scribe.
  Scale to 100 finally.

- Three grading criteria with 1-3 points: 1 poor, 2 average, 3 good.
  - Academic correctness: anything wrong in the notes?
    3: no error, showing the scriber has mastered the content.
    2: a few errors, showing the scriber does understand the content.
    1: many errors, showing the scriber doesn't understand the content.
  - Organization: how well is the notes organized
    3: good logical flow, easy to understand
    2: have some structure but incomplete
    1: just simply a collection of equations and theories, no organization
  - Language: Use of English
    3: looks like a textbook
    2: imperfect writing style but ok
    1: incorrect style and unreadable

Programming language

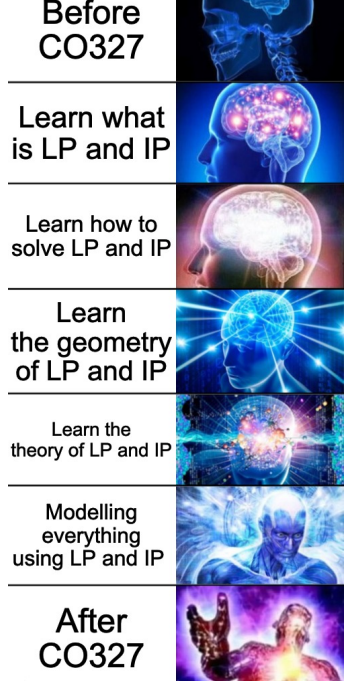
- MATLAB
  - Free (Waterloo license).
  - Good documentation and maintenance.
  - You can run it on cloud.

- Julia
  - Free.
  - (Claimed to be) Fastest
  - Argumentum ab auctoritate: because MIT, Stanford, etc, use it!

... more in computer programming lecture.

Reference material for the course

- No reference: notes come from my digested knowledge from various sources.
- If you really want some references
  - google or Youtube search "linear programming", "integer programming"!
  - search the library any book with names "linear programming", "integer programming", "optimization model", + 'first course of", "introduction to", "understanding", "decision making"
  - A random google search shows a FREE (and should be good? I didn't read it) textbook: `http://www-personal.umich.edu/~murty/books/opti_model/`
- You find a book, not sure is it useful: contact me for discussion.


Before CO327
Learn what is LP and IP
Learn how to solve LP and IP
Learn the geometry of LP and IP
Learn the theory of LP and IP
Modelling everything using LP and IP
After CO327