CLASS NOTES STARTING
SEPT 5

WELCOME! First Class

Math 441

Rough idea:

<u>September</u>: Applications of LP (linear programming) and related "mathematical programming" algorithms.
[We treat LP, IP (integer programming), QP (quadratic), etc. as "black boxes", with some general remarks, such as LP algorithms than IP algorithms.]
Also: some remarks on scientific writing.

<u>October</u>: More of September topics + investigation into IP algorithms

<u>November</u>: More of above topics + presentations to class.

=

Homework: 15%, roughly each week*

Project: 85%:
- 5% Proposal
- 10% Progress Report
- 65% Final Report
- 5% Class presentation

=

Deadlines:
- Proposal: Sept 28
- Progress: Oct 22
- Final: Nov 21

Presentations (Based on Progress Report): during November

[At this point we discussed the material on the course website; I've attached these pages and the annotations as the next 2 pages of this notebook.]

MATH 441 Homepage, Fall 2018

This page concerns MATH 441--Math Modeling: Discrete Optimization Problems--Section 101.

This page has the most up-to-date information on the course(s) and supersedes any other document. Some of the most basic course information on this website is replicated in this course overview (PDF).

News	Some of the most basic course information on this website is replicated in this course overview (PDF). Most of September will be spent discussing applications of LP's (IP's, QP's, etc.) that may lead to projects; you may look at the board scans (below) and this list of some applications discussed in class.
Overview	This course involves applications of linear programming and related optimization such as quadratic programming and integer programming; you are assumed to have seen linear programming. The focus of the course is a research project. For B.A. students, this course counts as a "research intensive approved course." Your grade is computed as follows: research proposal, due Sept 28 (5%); progress report, due Oct 22 (10%); class presentation, during November (5%); final project, due Nov 21 (65%); homework on class material, assigned throughout the course (15%). (There are no exams.) As in previous years, the grading and grade distribution will be reminiscent of an Arts course; historically the average has been in the low 70's, and top marks are only given for projects with clear and interesting writeups of difficult research. Historically projects usually obtain good results, but only occasionally have a stellar writeup.
Projects	Most of your grade is based on a project, to be done in groups of 3-4 students; depending on enrolment and demand, smaller groups may be possible. All reports must be submitted in LaTeX; overleaf.com lets you share projects, written in LaTeX. Here are the project requirements and guidelines; these applications of LP's and ILP's are possible project topics. We will spend the first few weeks discussing applications suitable for Math 441 projects; here is a list of some applications discussed in class.
Textbooks and Resources	All required textbooks and resources are available for free to anyone with a UBC CWL. They include: • "Linear Programming" by Vanderbei, 4th edition. • "Integer Programming" by Conforti, Cornuejols, and Zambelli. • my article (from 1998) on Linear Complimentarity and Mathematical (Non-linear) Programming. • my article (last revised 2014) on Matrix Games. • my article (from 2017, which I may revise) with additional details (and homework) on The Markowitz Model . cover in Math 441 Other resources include: • Prof. Anstee's recent Math 441 course; • Prof. van Willigenburg's recent Math 444 course; • The textbook "Linear Programming" by Chvatal is the clearest and simplest exposition I know of linear programming, and uses dictionary form. Chapters 1-5 of Vanderbei's textbook are based (chapter by chapter) on Chvatal's textbook.
Office Hours	By appointment for now--email me and let me know when you are free over the next few days. When things get too busy (e.g., near project deadlines), appointments will be scheduled more rigidly.
Boards Scans, Etc.	Scan of boards for:
- Run LP, IP, .. - Plot graph **Software**	We will use Gurobi in class examples; it offers free licenses for univesity students and is competitive with the best commercial products (e.g., CPLEX and Mosek); see Last year's webpage for examples in Gurobi. I recommend you use Gurobi for the homework and your projects; some homework may require another language, such as Maple or R; **you may use other software, but you need to make sure that your software works and can do what is required in the homework (e.g., solve linear programs and integer linear programs of moderate size, generate sequences of random numbers and plot data points).**
Prerequisite	The prerequisite for the course is Math 340. In class we will use the **dictionary form** of the simplex method, rather than tableau form. If you have not taken Math 340 at UBC, but know the material or bring other strengths to the course, please speak to me. Here is my last Math 340 homepage, from Fall 2015. You might look at the exam materials there to know what we covered.
Homework	Late homework will not be accepted. Your two lowest scores will be dropped in the overall homework computation.
New Material	New material includes a discussing of optimization related to linear programming such as integer programming (and "branch and bound"), convex programming, and quadratic programming. We will cover applications such as: when linear programming can handle non-linear optimization (e.g., progressive taxation); scheduling class presentations; time restricted task scheduling; bin packing and related problems (using integer linear programming); the Markovitz model (using quadratic programming); L-1 and L-infinity (Chebyshev) curve fitting; matching. We will also discuss some principles of linear programming that you may have seen in Math 340, especially those related to modeling; the most important is that the simplex method guarantees that there are optimal solutions to LP's that have many zero-valued decision variables if you have many more decision variables than constraints. We will also briefly discuss common mistakes in scientific writing.

MATH 441 Projects, Fall 2018

Group sizes: 3-4 people per group

Requirements	• Projects must use a model(s) involving linear programming, integer programming, convex programming, quadratic programming. • Projects involve original software written only by the group members. • Projects must test the software with "toy data" (simple data with predictable solutions) as a correctness check. • Projects must compute at least 5 variants of a similar optimization problem and compare the results. • Projects must be written in LaTeX (overleaf.com is one easy way to write group projects in LaTeX); • Projects must indicate what each group member contributed to the project.
Research Projects	Research projects can be done in groups of 3-4 students; depending on enrolment and demand, smaller group sizes may be possible. Projects generally model something in the real world and involving linear programming (e.g., simplex method for solving LP's, branch and bound method for solving integer LP's, etc.). Most projects involve some data, real or (partially) synthetic; finding the real data or generating realistic synthetic data can be a significant part of the problem. For project ideas, you can look at my list of project ideas and the applications we covered in last year's Math 441 course. I may cover some new applications this year. Please tell me your project idea before you present your proposal; you are encouraged to consult me during the term whenever you have questions or difficulties arise.
Originality and Difficulty	Part of the grade for projects (under the "Modeling terminology and content" row of the Writing Rubric) regards the originality and difficulty of the project.
Tell Me Something We Don't Already Know	Ideal research projects should **tell me something that we don't already know.** For example, say that you are working with data that tries to schedule UBC exams to minimize the number of student exam conflicts (or hardships). Here are **some things we know**: • The longer the alloted exam period, the fewer the number of conflicts and hardships in an optimal schedule. • If the number of exam time slots times the number of chairs in the univeristy is less than the sum of the number of students in each course, there will be at least one conflict. Here are **some things we don't know**: • Fix a set of classes, of students, and of class lists (of which students are taking which classes). As the alloted exam period shrinks, does the number of conflicts increase gradually, or is there a sharp threshold? (I'm guessing that there is a pretty sharp threshold.) • Say that UBC takes certain large classes with many sections, and requires each section to have its own exam; now different sections of the same class can be scheduled at different times. Does this policy change significantly affect the number of exam conflicts? Can you draw some useful principles based on your results?
Final Report	The final report should be a PDF file, written in LaTeX, of roughly 3-7 pages (i.e., 600-1400 words), excluding references and appendices. You will have to be concise and focused! You may put additional material in appendices to the paper, but this should be used only for data/tables/figures/software, not for exposition on the project. You should print out your report, as well as submit the report plus your software online. I should be able to test your software and duplicate your experiments. The same Writing Rubric will be used to grade your progress reports and final reports, with different weightings to the rows; for the final report, "Modeling terminology and content" given a weight of 5, and all other rows each given a weight of 1.
Progress Report	The progress report should be roughly 3-5 pages (i.e. 600-1000 words). Here is sample progress report from last year and its LaTeX source (which I have given a .txt extension). Here is a Writing Rubric as a guide to the grading of projects. The progress reports will be graded on your overview, motivation, specific questions, and models (which correspond to Sections 1 and 2 of the above sample).
Presentations	Each member of your group must present some material/slides, and must be prepared to answer questions on this material. Your group needs to email a PDF file of your slides by 7am on the day of your presentation (it's OK to make small corrections to the slides and to send me a revised set of slides later in the day, after your talk). Here is a guide to the grading of presentations.

Sept 7! Review LP (Linear Programming) and Software

FIRST GOALS

1. Review LP terminology
2. Solve a "toy" LP
3. Illustrate Gurobi command line & shell
4. Generalize our "toy" LP (from 2) to a parametric LP with "phase transitions"
5. Use LP's to solve a non-linear program (a concave, piece-wise linear objective)
6. "Linear Programming without Linear Programming"

Linear Program: (standard form)

max $\vec{c}\cdot\vec{x}$ s.t. (subject to) $A\vec{x} \le \vec{b}$, $\vec{x} \ge \vec{0}$

e.g. You can watch up to 12 hours TV per day.

Options (decision variables)

x_1 = # hours watching The Expanse

x_2 = " " The X-files

x_3 = " " The Walking Dead

x_4 = " " other documentaries

Utility:

$$U(\vec{x}) = U(x_1, x_2, x_3, x_4)$$
$$= 10x_1 + 9x_2 + 5x_3 + 2x_4$$

want to maximize $z = 10x_1 + 9x_2 + 5x_3 + 2x_4$

s.t. $x_1 + x_2 + x_3 + x_4 \le 12$

$x_1, x_2, x_3, x_4 \ge 0$

this is

max: $\underbrace{\begin{bmatrix}10\\9\\5\\2\end{bmatrix}\cdot\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix}}_{\vec{c}\cdot\vec{x}}$, s.t. $A\vec{x} \le \vec{b}$: $[1\ 1\ 1\ 1]\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix} \le 12$, $\vec{x} \ge \vec{0}$

Utility:

$$U(\vec{x}) = U(x_1, x_2, x_3, x_4)$$
$$= 10x_1 + 9x_2 + 5x_3 + 2x_4$$

want to maximize $z = 10x_1 + 9x_2 + 5x_3 + 2x_4$

s.t. $x_1 + x_2 + x_3 + x_4 \le 12$

$x_1, x_2, x_3, x_4 \ge 0$

this is

max: $\underbrace{\begin{bmatrix}10\\9\\5\\2\end{bmatrix}\cdot\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix}}_{\vec{c}\cdot\vec{x}}$, s.t. $A\vec{x} \le \vec{b}$: $[1\ 1\ 1\ 1]\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix} \le 12$, $\vec{x} \ge \vec{0}$

Given max $\vec{c}\cdot\vec{x}$

s.t. $\underbrace{A\vec{x} \le \vec{b},\ \vec{x} \ge 0}$ } feasible region (for $\vec{x}$)

$\vec{x}$ called feasible if it statisfies (feasible solution)

The "objective" is $\vec{c}\cdot\vec{x}$

An LP is feasible if there is some feasible $\vec{x}$

" " " bounded if $\vec{c}\,\vec{x} \le$ something for all feasible $\vec{x}$

An "optimal solution" is an $\vec{x}^*$ that attaints the largest value of $\vec{c}\cdot\vec{x}$ over all feasible $\vec{x}$.

For now Simplex Method is some algorithm, really a family of algorithms, that we will use; for now, we won't say much about it.

AFTER REST OF STUFF
Now imagine
$5x_3 \to Ax_3$
$A \in \mathbb{R}$

want to maximize $z = 10x_1 + 9x_2 + 5x_3 + 2x_4$

s.t. $x_1 + x_2 + x_3 + x_4 \le 12$

$x_1, x_2, x_3, x_4 \ge 0$

Clear: if The Expanse has most utility per hour, then you watch 12 hour of The Expanse:

Optimal solution $\vec{x}^* = \begin{bmatrix} 12 \\ 0 \\ 0 \\ 0 \end{bmatrix}$, $x_1 = 12$, $x_2 = x_3 = x_4 = 0$

Max objective $z^* = 12 \cdot 10 + 9 \cdot 0 + 5 \cdot 0 + 2 \cdot 0 = 120$

=

Variant 1: Say add constraint $x_2 \le 4$.
What is optimal solution now?

"inactive constraint"

since at x^*:

"has slack" $\rightarrow$ $x_2^* < 4$

Variant 2: $x_1 \le 3$, $x_2 \le 4$. What is the new $\vec{x}^*$: new $x_1^* = 3$, $x_2^* = 4$, $x_3^* = 5$.

=

At this point we started taking a look at how the "Gurobi shell" works:

how to write an LP in the form that Gurobi understands, and how to use its features.

Sept. 10

LP (linear program) "standard form"

max $\vec{c}\cdot\vec{x}$ s.t. → objective

$$\left.\begin{array}{l} A\vec{x} \le \vec{b}, \\ \vec{x} \ge \vec{0} \end{array}\right\} \text{"feasible region"}$$

LP: $\vec{x}$ has real values

$\vec{x} \in \mathbb{R}^n$, A $m \times n$, $\vec{b}$ m-dim vect

n "decision" variables $\vec{x}$, m constaints

We write $\vec{x} \in \mathbb{R}^n$, $\mathbb{R}$ = real numbers, $\mathbb{R}^n$ - vector of n reals

LP: $\vec{x} \in \mathbb{R}^n$

Integer Program, (IP), $\vec{x} \in \mathbb{Z}^n$

$\mathbb{Z}$ = integers = $\{0, \pm 1, \pm 2, \ldots\}$, $\mathbb{Z}^n$ - vector of n integers

IP generally much harder to solve

IP often used in application

Some IP's can be solved with LP's...

"Toy" LP:

$$\max \quad 10x_1 + 9x_2 + 5x_3 + 2x_4$$

s.t.

$$x_1 + x_2 + x_3 + x_4 \le 12$$

$$x_1, \ldots, x_4 \ge 0 \qquad (\vec{x} \in \mathbb{R}^n)$$

Claim: This LP has optimal solution

$$\vec{x}^* = \begin{bmatrix} 12 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \text{ i.e. } x_1^* = 12, \; x_2^* = x_3^* = x_4^* = 0$$

Another easy LP: $x_1 \le 4.7$, $x_2 \le 1.3$,

What if $x_3 + x_4 \le 2.9$

What about $7x_1 + 9x_2 + 13.217x_3 - 4x_4 \le 20$ ☹

⋮

LP without LP:

If you n decision variables

m constraints

Standard form: $A\vec{x}_{dec} \le \vec{b}$ original $\vec{x}$ ⇝ $\vec{x}_{decision}$

Simplex method:

$\vec{x}_{slack} = \vec{b} - A\vec{x}_{dec}$ AND $\vec{x}_{slack} \ge 0$

Initial dictionary [if $\vec{b} \ge \vec{0}$],

All dictionaries (tableaux) have

m basic variables n non-basic variables

Upshot: With n decision vars, m contraints, any optimal solution that the simplex method produces has at most m non-zero values (among $\vec{x}_{dec}$ and $\vec{x}_{slack}$)

example $\max \; 10x_1 + 9x_2 + 5x_3 + 2x_4$ s.t.

$$x_1 + x_2 + x_3 + x_4 \le 12$$

$$x_1, \ldots, x_4 \ge 0$$

Optimal solution $x_1 = 12$, $x_2 = x_3 = x_4 = 0$

Not an accident...

If you are solving diet problem with 200 types of food, and 7 constraints on nutrients

Simplex method gives optimal solution with 193 foods not used...

Matrix games

$$\begin{array}{c|cc} & \text{Betty call} & \text{Betty fold} \\ \hline & 3 & 4 \\ & 5 & 6 \\ & 9.1 & 0.5 \\ & \vdots & \vdots \end{array}$$

Alice 3000 strategies

Betty has 2 pure strategies

Alice needs only 2 stragies for an optimal solution to matrix game