MIE 1603 / 1653: Integer Programming

Winter 2018
Syllabus

Instructor: Prof. Merve Bodur
Office: BA8106
Office hours: Friday 1-3 p.m. (or by appointment)
E-mail: `bodur@mie.utoronto.ca`
P.S. Please include course code (MIE 1603 or 1653) in your email subject

TA: Margarita Castro
Office hours: Friday 9-9:30 a.m, 10:30-11 a.m. (BA3116)
(or by appointment)
E-mail: `mpcastro@mie.utoronto.ca`
P.S. Please include course code (MIE 1603 or 1653) in your email subject

Lecture: Monday 2-5 p.m. (AB114)

Tutorial: Friday 9:30-10:30 a.m. (BA3116)

Supplementary texts: ***Integer Programming*, L.A. Wolsey. Wiley, 1998.**
Applied Integer Programming: Modeling and Solution, D.-S. Chen, R.G. Batson, and Y. Dang. John Wiley & Sons, 2010.
Integer and Combinatorial Optimization, G.L. Nemhauser and L.A. Wolsey. Wiley, 1999.
Integer Programming, M. Conforti, G. Cornuéjols, and G. Zambelli. Springer, 2014.
Theory of Linear and Integer Programming, A. Schrijver. John Wiley & Sons, 1998.

Course web page: Blackboard

Official course description: Formulation of integer programming problems and the characterization of optimization problems representable as integer and mixed-integer programs. The degree of difficulty of classes of integer programs and its relation to the structure of their feasible sets. Optimality conditions. Branch-and-bound, cutting plane, and decomposition methods for obtaining solutions or approximating solutions.

Prerequisite: MIE 262, APS 1005 or equivalent.

Overview

The aim of integer programming is to find optimal decisions in problems where the decisions are restricted to take on a finite set of possible values. In particular, integer programming problems can model "yes-no" decisions, a capability which makes the field of integer programming tremendously useful in an almost unlimited number of applications. (A very small list of examples includes: optimizing supply chain network configurations, designing radiation treatment plans, finding an optimal strategy for reducing the threat of smuggling of nuclear materials, and transportation routing and scheduling.)

The modeling power of integer programming is a double-edged sword; it is what makes it such an extremely useful tool, but it also means that there is always a possibility that a specific problem modeled as an integer programming problem may prove to be unsolvable even by state-of-the-art commercial software. However, in many such situations, a person with integer programming expertise may be able to use problem structure to devise a rigorous solution approach which can successfully solve the problem. It is my hope that each student in this class will become such an integer programming expert.

This course will cover both *theoretical* and *computational* aspects of integer programming. The theoretical component of this course will require graduate knowledge of linear programming *and strong mathematical skills* in general. For example, we will do formal proofs in the course, and students will be expected to prove things in assignments and on exams. As part of the computational component of this course, students will be required to implement some of the methods we learn, so familiarity with programming (or the ability and willingness to learn) will be necessary.

Objectives

- Understand how integer variables are used to model a variety of situations, and what properties are desirable in an integer programming formulation of a problem.
- Become familiar with a commercial integer programming solver. Be able to pass a problem to a solver and understand and set certain parameters of a solver that may help it solve a given problem.
- Know the basic concepts of complexity theory, in particular, understand the difference between "easy" problems and "hard" problems, and how to verify whether a given problem is "easy" or "hard."
- Learn the basic techniques for deriving valid inequalities for integer programming problems, and specifically learn some of the classes of valid inequalities that have been successfully used in commercial solvers.
- Gain exposure to basic polyhedral theory, and in particular understand the most desirable properties of a valid inequality or class of valid inequalities.
- Understand decomposition techniques for solving large-scale problems and have a good idea how to implement some of these techniques.

Course web sites

The primary course website can be found in `https://portal.utoronto.ca`. This is where lecture slides, homework assignments, and homework solutions will be posted. This site also contains a *Discussion Board*. When you have a question related to this course, you should first check this Discussion Board to see if it has already been answered, and if not, post your question there. This way, all students benefit from your question and the answer. You are also allowed, and encouraged, to answer questions posted by other students. We will use the website also for posting your grades. Please check here that your grades correctly match the grade given on your assignments/exams.

Grading

The course grade will be based on a weighted average of the following components:

	MIE1653	MIE1603
Class participation	5%	5%
Homeworks	20%	15%
Course project	-	15%
Midterm exam	35%	30%
Final exam	40%	35%

Letter grade cut-offs will be determined based on score distribution. However, if average homework score is less than 40 points out of 100, the final letter grade will be decreased by a full letter grade. In addition, for MIE1603 students, the similar letter grade adjustment will be applied according to the project score as well.

Tentative exam dates are as follows:

- Midterm exam: **March 2, 9 - 11 a.m.**
- Project deadline: **April 12, 5 p.m.**
- Final exam: **April 23, 9:00 - 11:30 a.m.**

Students who have a conflict with the final exam time should notify me by the second week of class.

Homework

Homework assignments will be given roughly bi-weekly, and are a required part of the course. I will provide examples of correct solutions. Students may work in groups of up to two on homework assignments (as long as both students are enrolled to the same course code), but groups must work independently of each other, and solutions must not be copied from the internet or other sources. If I detect improper collaboration on an assignment, *all groups* involved will automatically receive a 0 on the assignment in question. Some assignments will include computational exercises, so in forming groups, you may want to make sure you have at least one member who is proficient at programming. Please note that as the exams must be done independently, it is in each student's best interest to take an active role in all homework exercises.

Late assignments will be penalized 20% of the possible points for each day that it is late. So an assignment can receive no credit if it is five days late.

Classroom participation

Students are expected to attend lectures and encouraged to participate by asking and answering questions. Students are also expected to follow standard classroom etiquette, including but not limited to:

- Coming to class on time,
- Not causing classroom disruptions by talking during class, leaving your phone volume on, sleeping, leaving early, etc.

For the classroom participation score, all students who regularly attend lecture and follow classroom etiquette will automatically receive 8/10 points. Students who also actively participate in lectures or post (good) answers to students' questions in the Q&A forum on the course web site will receive a higher score; students who miss class more than a few times during the semester or fail to follow classroom etiquette will receive a lower score.

Course project for MIE1603

MIE1603 students are expected to do a course project, in groups of at most two people (individual projects are also fine). Project topics must be approved by me. I may be able to provide some suggestions on project topics, or you can choose a topic that is interesting to you. Students must submit a one-page project proposal by **March 12**, although it is advised to start on the project earlier than this deadline.

All groups should submit a project report (prepared in LATEX) by **April 12**. Depending on time availability, students may be asked to present their work to the class; otherwise, students may be asked to present an "oral defense" of their project to me.

Here are some project ideas:

- Conduct a comprehensive literature review on a selected topic. More specifically, read papers and become an "expert" in an area that we are unlikely to cover in detail in this class. In this case, you should provide a comprehensive overview of the subject, putting past and modern work into a context. Many nice, recent papers on integer programming can be found at `http://www.optimization-online.org` or `https://arxiv.org`.

- Pick a (relatively recent) paper published in a "good journal", which includes some computational work. You may implement existing ideas in the paper, propose extensions, and provide computational tests and observations.

- Read papers in a particular area of application, experiment with different integer programming formulations and solution methodologies for a model or the models.

- Incorporate integer programming into your current line of research (other than your current thesis work), or any other research area of your interest.

Academic integrity

Academic misconduct will not be tolerated in this course. Please make yourself familiar with the University academic conduct guidelines: `http://academicintegrity.utoronto.ca`.

In this course, this mainly means that you should not copy anyone else's exam or allow anyone to copy your exam, and that assignment solutions should be your own work. Assignments will be done in groups of up to two, **but groups may not share answers with each other.** In any assignment or exam, you must properly give credit to any outside resources you use (such as books, papers, web sites, other people, etc.). It is not allowed to use assignment solutions from previous semesters or other sources. Please don't hesitate to ask me if you have any uncertainty about what would be considered academic misconduct in this course.

Tentative schedule of topics

Following is a list of general topics which we will cover in this course. This is subject to change, and if time permits we may cover some topics not explicitly listed here.

Week	Topic	Dates
1	Introduction to IP and modeling	08/01
2	Branch-and-bound	15/01
3	Advanced modeling	22/01
4	IP formulations	29/01
5	IP software	05/02
6	Complexity theory	12/02
7	Family Day - No Lecture	19/02
	Midterm review session	23/02
8	Well-solved IPs	26/02
	Midterm exam	**02/03**
9	Benders decomposition	05/03
10	Valid inequalities	12/03
11	Lagrangian relaxation	19/03
12	Column generation	26/03
13	General purpose valid inequalities	02/04
14	Strong valid inequalities	09/04
	Project presentations	13/04
	Final exam	**23/04**