Due: Thursday, September 24th (4:30 p.m.)

Reading

Read the Introduction and Chapters 1 and 2 of the AMPL book: https://ampl.com/learn/ampl-book/. Note that, except for Chapter 20, the AMPL book covers *linear programming* (LP), rather than integer programs.

From Conforti, Cornuéjols and Zambelli, read Sections 1.1, 1.2.1, 2.1 and 1.4. The remainder of Chapter 1 is also worth a look. Section 1.2.2 covers many of the ideas we'll develop during the course, but will likely be hard to understand right away. Sections 1.3 briefly introduces computational complexity, while Section 1.5 explores connection between discrete optimization and number theory, and are best appreciated if you have some background on those topics.

You should understand the translation of combinatorial optimization problems such as knapsack, set covering and travelling salesman into integer programs. Indeed it is possible to model many natural constraints using mixed integer programs. The integrality constraint allow us to model constraints of the form "A or B", "satisfying k of n constraints" or "takes a value from a specified discrete set", which are not easily phrased as linear inequalities.

Problems for Math 408 and Math 708

Written solutions will submitted via Crowdmark. Crowdmark does not support non-standard file types, so code submission will be done in Canvas. The code (AMPL) submission should be two files named hw1.dat and hw1.mod. Include your name on the first page of hw1.pdf and in the comments of hw1.dat and hw1.mod.

0. (Not graded.) Download and install AMPL on your computer from the link given in the Miscellaneous section of the course Canvas page. You don't need to make an account, but you should use your SFU e-mail address to connect, allowing the use of the academic licence. Note anyone can use the "Community Edition" licence available at https://try.ampl.com/, but it may not be able to handle larger problems later in the course.

There are now several ways to install and use AMPL. I use the standalone executable available as a "time-limited bundle", building a package with solvers Minos, Cplex and Gurobi, which works through a simple command-line interface, but you may prefer something else.

1. Take your nine digit student id number and add 10 to each digit to get a sequence of nine numbers $a_1, a_2, \ldots, a_9$ between 10 and 19. Similarly, add 10 to each of the first nine digits of π to get a sequence $b_1, b_2, \ldots, b_9$. Your *personal knapsack problem* is:

$$\text{Maximize } \sum_{i=1}^{9} b_i x_i \quad \text{subject to } \sum_{i=1}^{9} a_i x_i \leq \frac{1}{2} \sum_{i=1}^{9} a_i \quad \text{and } x_i \in \{0,1\} \text{ for } i = 1, \ldots, 9.$$

a. Solve this integer program using AMPL, either via the command line interface or the AMPL IDE graphical interface. Please include a screen shot of the final solution in with your written solutions.

Warning: If you use the default solver (Minos) in AMPL, it will *not* take the integrality constraints into account: it uses methods of continuous optimization and is not built to handle them. Two other available solvers are Cplex and Gurobi, you change to them via:

`option solver cplex;` and

`option solver gurobi;` respectively.

b. Use AMPL to solve the linear programming relaxation of your knapsack problem, that is:

$$\text{Maximize } \sum_{i=1}^{9} b_i x_i \quad \text{subject to } \sum_{i=1}^{9} a_i x_i \leq \frac{1}{2} \sum_{i=1}^{9} a_i \quad \text{and } 0 \leq x_i \leq 1 \text{ for } i = 1, \ldots, 9.$$

Do not submit the AMPL files for this problem. Instead explain briefly what changes you need to make to the files from the previous files to solve this problem.

c. Now use AMPL to solve a mixed-integer programming version of your knapsack problem, where only variables 5 through 9 are required to be integer:

$$\text{Maximize } \sum_{i=1}^{9} b_i x_i \quad \text{subject to } \sum_{i=1}^{9} a_i x_i \leq \frac{1}{2}\sum_{i=1}^{9} a_i \quad \text{and } 0 \leq x_i \leq 1 \text{ for } i = 1, \ldots, 4; \text{ and } x_i \in \{0, 1\} \text{ for } i = 5, \ldots, 9.$$

Do not submit a the AMPL files for this problem. Instead explain briefly what changes you need to make to the files from the previous files to solve this problem.

2. Draw the convex hull of the following 2-variable set

a. $S = \{x \in \mathbb{Z}^2_+ \ : \ a_8 x_1 + a_9 x_2 \leq 60, \ x_1 - x_2 \leq 2, \ x_2 - x_1 \leq 2\}$, using a_8 and a_9 from question 1.

b. $S_2 = \{(x, y) \in \mathbb{Z}_+ \times \mathbb{R}_+ \ : \ 5x + 5y \geq 6 \ x \leq 3, \ y \leq 3\}$.

3. Textbook Exercise 2.19, but do only the even-numbered parts.

4. Question 6.15 from the Supplementary Exercises to *Optimization Modelling with Spreadsheets, 3rd edition* by K. R. Baker. Provide written integer programming models for parts (a) and (b). Solve these with AMPL or software that you prefer. You do not need to submit the code, but use the answers you get from the solver to provide written answers to parts (a) and (b).

5. Consider the four sets described in Conforti, Cornuéjols and Zambelli Exercise 1.15, with $b = \frac{4}{3}$ and $d = \frac{2}{3}$. Illustrate each set and its convex hull. Use these illustrations to describe the convex hull using linear inequalities.

Additional Problems for Math 708

6. Generalize the results of the previous problem to arbitrary b and d.

7. Consider the problem of colouring the vertices of a graph $G = (V, E)$ using the minimum possible number of colours such that no edge connects vertices of the same colour. Formulate this problem as an integer program. You can assume that you have an a priori upper bound k for the number of colours you will use, perhaps $k = |V|$ or something smaller based on knowledge of the graph. Comment on how the size of your formulation scales with $|V|$, $|E|$ and k.

8. Question 6.18 from the Supplementary Exercises to *Optimization Modelling with Spreadsheets, 3rd edition* by K. R. Baker. Provide written integer programming models for parts (a) and (b). Solve these with AMPL or software that you prefer. You do not need to submit the code, but use the answers you get from the solver to provide written answers to parts (a) and (b).