Due: Thursday, September 19th (4:30 p.m.)

Reading

Sections 1.1 through 1.4 of of Applegate, Bixby, Chvátal and Cook's *The Traveling Salesman Problem : A Computational Study*. This provides some cultural background on the Travelling Salesman Problem, a key example in this course. SFU library link.

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 the textbook, Chapter 1 and Sections 2.1 and 2.2. Sections 1.3 briefly introduces computational complexity, while Section 1.5 explores connection between discrete optimization and number theory. If you are not familiar with these subjects, then you may find these sections challenging. We will touch on complexity at various points in this course.

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 following the instructions given in class.

You can use either the command-line or graphical (Integrated Development Environment) version.

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. The two other available solvers are `Cplex` and `Gurobi`, you change to them via:

   `option solver cplex;` and

   `option solver gurobi;`

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. Call the results from questions 1. ip^*, lp^* and mp^* respectively. Which of these numbers is largest, and which is smallest? Should this be true for all students in the class? Can some of the numbers be equal? Which ones?

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

4. Question 6.14 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), (b) and (c).

5. Consider the four sets described in the textbook Exercise 1.15, with $b = \frac{3}{2}$ and $d = \frac{1}{3}$. 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 6.21 from the Supplementary Exercises to *Optimization Modelling with Spreadsheets, 3rd edition* by K. R. Baker. Describe how to solve the problem using an integer program. Solve this with AMPL or software that you prefer. You do not need to submit the code, but do note the optimal solution and the optimal objective value, which should answer the question.