IEORE4004: Optimization Models and Methods	12/03/2019

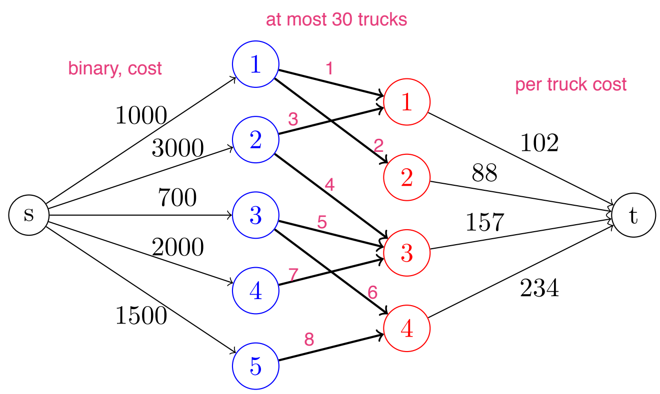
Assignment 6

Deadline: 12/12/2019, h: 11:40am | *Instructor: Christian Kroer*

- This assignment sheet has 4 exercises, one of which is a bonus exercise. No hint whatsoever will be given on the bonus exercise, so do not ask. You must submit your solution via Canvas before the deadline. The time constraint is strict. Late submissions will not be accepted. Each time you are requested to write an LP and solve it with a solver, report the formulation and the screenshot of the output of the solver.
- You are allowed to discuss the assignment with others but the write-up must be individual work. Please mention in your write-up all the people you have discussed the solution with.

Problem 1: A company operates trucks that sell food. There are 100 trucks in total. There are eight routes that can be used, each starting at s and ending at t. To each route, a certain per-truck revenue is associated, see the table below.

Route ID	Revenue	Nodes used
1	190	s, 1, 1, t
2	200	s, 1, 2, t
3	100	s, 2, 1, t
4	300	s, 2, 3, t
5	400	s, 3, 3, t
6	150	s, 3, 4, t
7	570	s, 4, 3, t
8	70	s, 5, 4, t



In addition, the company will incur a cost in operating the trucks, as indicate in the figure:

(a) The numbers on the left (on arcs from node s to the blue nodes) indicate a fixed cost for the arc. If we don't pay the cost the arc is not going to be available. If we pay the cost, the arc will be available and could carry all the trucks we want.

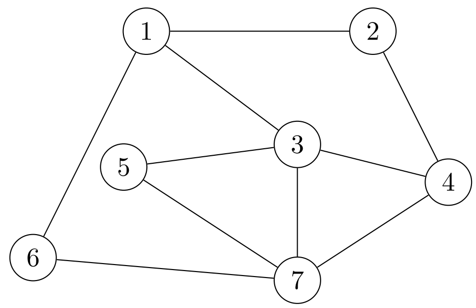
(b) The arcs shown in the middle (from the blue nodes to the red nodes), in bold, have no cost, but each can carry at most 30 trucks.

(c) The arcs on the right (from the red nodes to node t) do not have any fixed cost and have infinite capacity. The numbers shown next to the arcs are per-truck costs.

1. Formulate the problem of maximizing net revenue as an optimization problem using both binary and continuous variables.
2. Solve it using Gurobi or your favorite solver.

Problem 2: Recall the broadcasting problem seen in class. A TV station wants to broadcast in a set of cities. In order to do that, to each city a broadcasting frequence must be assigned. In the graph below, each node represents a city, and edges connect cities to which one cannot assign the same frequency. In order to have the right to transmit on a certain broadcasting frequency, the TV station must buy exclusive rights on that frequency, and that will cost $500,000. In class we saw how use Integer Programming to formulate the problem of broadcasting in all the cities at a minimum cost.

1. Suppose that, for each city v, we have a profit p_v associated to broadcasting in v. Formulate the problem of choosing in which cities to broadcast as to maximize the net profit.



2. Now, assume $p_1 = p_7 = p_2 = \$300,000$, $p_3 = \$450,000$, $p_4 = p_5 = \$100,000$, $p_6 = \$400,000$. Solve the problem using Gurobi or your favorite solver.

Problem 3[1]: The Lotus Point Condo Project will contain both homes and apartments. The site can accommodate up to 10,000 dwelling units. The project must contain a recreation project: either a swimming–tennis complex or a sailboat marina, but not both. If a marina is built, then the number of homes in the project must be at least triple the number of apartments in the project. A marina will cost \$1.2 million, and a swimming–tennis complex will cost \$2.8 million. The developers believe that each apartment can be sold at a price of \$48,000, and each home can be sold at a price of \$46,000. Each home (or apartment) costs \$40,000 to build. Formulate an IP to help Lotus Point maximize net profits.

Problem 4:

1. Assume you have three type of coins, \$1, \$2, and \$5, and a wallet that fits up to 5 coins. Use Dynamic Programming to find the smallest value (in whole dollars) that cannot be created and fit into the wallet. Here are a few examples how values can be created:
   - you can create \$7 by putting in 1 coin of \$5, and 1 coin of \$2.
   - you can create \$18 by putting in 3 coins of \$5, 1 coin of \$2, and 1 coin of \$1.
2. [**bonus**] Consider the generalization of the problem above: you have m type of coins, a wallet where you can fit at most k coins, and want to know the smallest value (in whole dollars) that cannot fit into the wallet. Write a program that takes as input a file containing the following sequence of numbers:

   m

   k

   the sequence of the m different types of coins

   and solves the problem above implementing the algorithm you developed in part (a). Test it on the file "coins.dat". Submit the code of the algorithm and the solution it outputs.

[1]Problem 16 from Section 9.2 of the book