	IEOR 4008: Computational Discrete Optimization	04/06/2020

IEOR 4008: Computational Discrete Optimization   04/06/2020

Assignment 4

Deadline: 04/22/2020, h: 1:10pm

Unless stated otherwise, in solving those exercises you are allowed to assume as known every statement that has been proved or stated in class, as well as basic linear algebra / graph theory facts. If you are uncertain whether you can use some result, please ask. You can also reuse (portion of) the code uploaded on the webpage of the course.

Exercise 1

Consider the following instance of the Vehicle Routing Problem: The depot is denoted by v_0 and customers by $v_1, \ldots, v_8$. Demand of customers are, respectively: $3, 3, 6, 9, 9, 9, 12, 12$. The capacity of the trucks is 16. Distances between nodes are given by the table below.

	v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_0	-								
v_1	5	-							
v_2	7	4	-						
v_3	7	4	4	-					
v_4	3	6	10	10	-				
v_5	4	7	4	7	7	-			
v_6	10	7	3	3	13	6	-		
v_7	9	4	8	4	6	11	7	-	
v_8	9	8	8	8	6	11	11	4	-

a) Compute the savings as in the Clarke-Wright algorithm.

b) Solve the instance above with the Clarke-Wright algorithm.

c) What is the minimum number of trucks needed to serve all customers? Motivate your answer.

Exercise 2

Recall that in class we saw that the Maximum Matching problem (MM) and the Maximum Weighted Matching problem (MWM) can be solved with efficient algorithms. Consider the following:

The Maximum Matching problem with a specific vertex (MMv)

Input: A graph $G(V, E)$ and a vertex $v \in V$.
Output: A matching M of G of maximum cardinality among those that contain v.

---

a) Show that, assuming v has at least a neighbor, the value of the optimal solution to (MM) is equal to the value of the optimal solution to (MMv) on the same input graph.

b) Give an algorithm that solves (MMv) (Hint: Reduce it to a problem you know how to solve)

c) In class we saw that we can formulate the problem of finding the maximum number of 2- and 3-way exchanges in a set of patients/donors as an Integer Program[1]. Now suppose that patients are ordered according to the time they spent waiting for a donation. This gives them priorities: patient v_1 has priority over v_2, which has priority over v_3, etc. We say that a solution S to the 2- and 3-way exchange problem *respects the priorities (RP)* if the following happens:

---

[1]See the formulation from page 7 of the lecture notes from the 16th and 17th lectures.
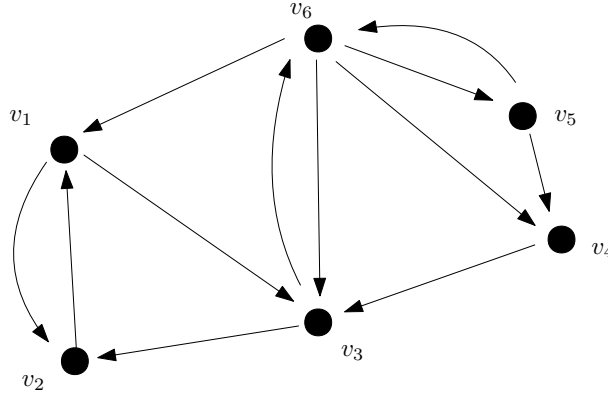
Figure 1: An instance of the Kidney exchange problem. Consider the following three solutions: A: The 3-way exchange v_1, v_2, v_3 plus the 2-way exchange v_5, v_6; B: The 3-way exchange v_3, v_6, v_4 plus the 2-way exchange v_1, v_2; B: The 3-way exchange v_1, v_3, v_6. Solution C does not achieve the maximum number of transplants, hence it is not RP. Both solutions A and B realize transplants for v_1, v_2, v_3, but A does not realize a transplant for v_4, while B does. Hence, A is not RP. One can check that indeed B is an RP solution.

  – it achieves the maximum number of possible transplants with 2- and 3-way exchanges.
  – take any other solution S' to the problem that achieves the same number of transplants with 2- and 3-way exchanges, and let v_i be the first (according to the priorities given above) vertex that receives a transplants only in one of S and S'. Then v_i receives a transplant in S and not in S'.

See Figure 1 for an example. Write an IP that finds a RP solution.

Exercise 3
Recall that, in the Kidney Exchange problem, deceased donors are voluntarily donating a kidney, without asking for one back. When a deceased donor v enters the market, it can sparkle a sequence of donations: v donates to v_1, whose donor donates to v_2, whose donor donates to v_3, etc. Finding the longest sequence of possible donation is called the **longest chain problem** (cfr. the notes and slides). Write an IP formulation for the longest chain problem.

Exercise 4
Consider the file kidney.py. It contains an instance of the Kidney exchange problem with 100 nodes stored as a list (called "data") of directed edges in the associated directed graph. Nodes with no incoming edges are deceased donors; all the others are patients, each with his/her own living donor.

a) Write an IP for the maximum possible number of 2-way exchanges in kidney.py, and solve it using Gurobi.

b) Write an IP for the maximum possible number 2- and 3-way exchanges in kidney.py, and solve it using Gurobi.

c) [**Optional, bonus points**] Now take into account the existence of deceased donors. Can you find a solution containing chains from deceased donors, 2- and 3-way exchanges that lead to more transplants than those found in part b)?