UBINET, Master 2 IFI Algorithms for telecommunications

Final Exam, November 2017

3 hours

Only 4 pages of manuscript notes are allowed. No computers, cellphones, books.

Instruction and comments: the points awarded for your answer will be based on the correctness of your answer as well as the clarity of the main steps in your reasoning. All proposed solutions must be proved. **All the exercises are independent.**

1 Graph Algorithms (15 points, 90 minutes)

Most of the questions below are independent. More difficult questions are indicated by a (∗).

The problem is dedicated to the design of a *Fixed Parameter Tractable* (FPT) algorithm to solve the k-Vertex Cover problem that asks if an input graph G admits a Vertex Cover of size at most k. For this purpose, we will use a powerful technique called *Iterative Compression.*

We recall that a Vertex Cover of a graph $G = (V, E)$ is a set $K \subseteq V$ of vertices such that, for every edge $xy \in E$, $K \cap \{x, y\} \neq \emptyset$ (K *"touches" every edge*).

Question 1 *Give a minimum Vertex Cover of the graph depicted in Figure 1.*

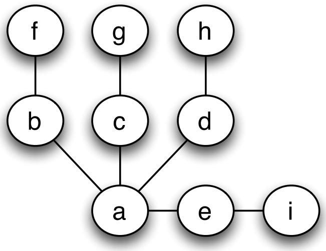

Figure 1: A graph.

Question 2 *Let k be a fixed parameter. Consider different algorithms that take an input of size n and with the following time-complexity. Which of these algorithms are FPT? Explain why.*

(a) $O(2^n)$*;*

(b) $2^{2^k} n^{100}$*;*

(c) $O(n^k)$*;*

(d) $O(n/k)$*.*

The main idea of the *Iterative Compression method* is to employ a so-called *compression Routine*. A Compression Routine is an algorithm that, given a problem instance and **an initial feasible solution** (not necessarily optimal), either calculates a smaller solution or proves that the given solution is of minimum size.

Using a Compression Routine, one finds an optimal solution to the problem by iteratively building up the structure of the instance and compressing intermediate solutions. (*This last sentence may be a bit obscur now, it should become clearer later*).

Algorithm 1 A 2-approximation algorithm for MINIMUM VERTEX COVER

Require: A graph $G = (V, E)$.

$H \leftarrow G$ *// initially, H is set to G*
$K \leftarrow \emptyset$ *// initially, K is the empty set*
While $E(H) \neq \emptyset$ *// while H has at least one edge*
  Let $\{u, v\} \in E(H)$ *// take any edge $\{u, v\}$ in H*
  $K \leftarrow K \cup \{u, v\}$ *// add u and v to K*
  $H \leftarrow H \setminus \{u, v\}$ *// remove u and v (and their incident edges) from H*
Return K

1.1 Compression Routine for Vertex Cover

Initial feasible solution. As said above, a Compression Routine needs an initial feasible solution. More precisely, we will need a “not too bad" initial feasible solution. To find such an initial solution, we will use an approximation algorithm.

Question 3 *Apply Algorithm 1 on the graph depicted in Figure 1.*

Question 4 *Let G be any graph. Prove that the set K returned by Algorithm 1 is a Vertex Cover of the input graph G.*

Question 5 *Prove that Algorithm 1 is a 2-approximation algorithm for the Minimum Vertex Cover problem.*

Question 6 *Let G be a graph and $k \in \mathbb{N}$. Assume that Algorithm 1 applied to G returns a set K such that $|K| > 2k$. What is the answer to the k-Vertex Cover Problem with G as input?*

Compression Routine. Let $K \subseteq V$ be any Vertex Cover of the graph $G = (V, E)$. The Compression Routine will use K to find an optimal solution. For this purpose, it will “check" all possible ways an optimal solution X can intersect K. More precisely, let $Y \subseteq K$ be any subset of K. We want to decide if there is a Vertex Cover X such that $X \cap K = Y$.

Let $N(K \setminus Y)$ be the set of vertices that have a neighbor in $K \setminus Y$.

Question 7 *Show that, if there is an edge between two vertices in $K \setminus Y$, then there is no Vertex Cover X such that $X \cap K = Y$.*

Question 8 *Let X be any Vertex Cover of G such that $X \cap K = Y$. Show that $N(K \setminus Y) \subseteq X$.*

Question 9 *(*) Assume there are no edges between vertices in $K \setminus Y$. Show that $X = Y \cup N(K \setminus Y)$ is a minimum Vertex Cover of G such that $X \cap K = Y$.*

Algorithm 2 First FPT algorithm for k-VERTEX COVER

Require: A graph $G = (V, E)$.

$K \leftarrow Algorithm1(G)$ *// K approx. Vertex Cover of G*
If $|K| > 2k$ **then Return** NO.
Else *// Next two lines correspond to the Compression Routine*
  For every $Y \subseteq K$ **do**
    If there are no edges in $G[K \setminus Y]$ **and** $|Y \cup N(K \setminus Y)| \leq k$ **Return** YES
Return NO

Question 10 *(*) Prove that Algorithm 2 solves the k-Vertex Cover Problem.*

Question 11 *What is the time-complexity of Algorithm 2 (as a function of $n = |V|$ and k)?*

1.2 Iterative Compression for Vertex Cover

Let $G = (V, E)$ be a graph and $V = \{v_1, v_2, \cdots, v_n\}$ be any arbitrary ordering of the vertices of G. For every $1 \leq i \leq n$, let $G_i = G[\{v_1, \cdots, v_i\}]$ be the subgraph induced by the vertices $v_1, \cdots, v_i$ (so the graph obtained from G by removing the vertices $v_{i+1}, \cdots, v_n$ and their incident edges).

Question 12 *For every $i \leq k$, give a Vertex Cover K_i with size $\leq k$ of G_i.*

Question 13 *Let $i < n$ and let K_i be a Vertex Cover of G_i. Show that $K \cup \{v_{i+1}\}$ is a Vertex Cover of G_{i+1}.*

Algorithm 3 Compression algorithm for k-Vertex Cover

Require: A graph $G = (V, E)$ and a Vertex Cover K of G such that $|K| = k + 1$.

For every $Y \subseteq K$ **do**

  If there are no edges in $G[K \setminus Y]$ **and** $|Y \cup N(K \setminus Y)| \leq k$**, Return** $Y \cup N(K \setminus Y)$

Return NO

Question 14 *Apply Algorithm 3 on the graph depicted in Figure 2 with the initial Vertex Cover depicted in red.*

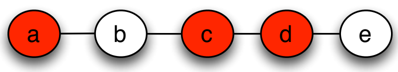

Figure 2: A graph G and an initial Vertex Cover $K = \{a, c, d\}$ depicted in red.

Question 15 *(*) Let G be any graph. Prove that Algorithm 3 returns a Vertex Cover of size $\leq k$ of G if it exists and returns NO otherwise.*

Question 16 *What is the time-complexity of Algorithm 3 (as a function of $n = |V|$ and k)?*

Algorithm 4 A better FPT algorithm for k-Vertex Cover

Require: A graph $G = (V, E)$ with $V = \{v_1, \cdots, v_n\}$.

$K_k \leftarrow \{v_1, \cdots, v_k\}$ *// K_k is a Vertex Cover of G_k with size k*

For $i = k + 1$ **to** n **do**

  Let $K = K_{i-1} \cup \{v_i\}$ *// K is a Vertex Cover of G_i with size $\leq k + 1$*

  If Algorithm3(G_i, K) $= NO$ **then Return** NO

  Else let $K_i =$ **Algorithm3**(G_i, K) *// K_i is a Vertex Cover of G_i with size $\leq k$*

Return YES

Question 17 *(*) Prove that Algorithm 4 solves the k-Vertex Cover Problem.*

Question 18 *What is the time-complexity of Algorithm 4 (as a function of $n = |V|$ and k)? Conclusion?*

2 Linear Programming

Most of the questions below are independent. More difficult questions are indicated by a $(*)$.

Exercise 1 (Vehicule Routing Problem (15 points, 90 minutes)) *We consider the following problem. A company has some goods to serve to a set of* n *customers* N. *The goods to be served to customer* i *has weight* Q_i. *To this end, the company possesses a fleet of* K *vehicles. A vehicle can transport a maximum weight of goods* P. *The company wants to minimize the cost to serve all its customers. This cost is given by the distance traveled by the vehicles. A vehicle is doing a single route which has to start from and return to the company depot denoted by* D. *Because of gas constraints, a route cannot be longer than* T. *A digraph* $G = (V = N \cup \{D\}, A, c)$ *is given in input, where* A *is the set of roads,* $c_{ij} \in \mathbb{R}$ *is the length of the road between client* $i \in N$ *and client* $j \in N$ *(or the depot).*

An example is given in Figure 3.

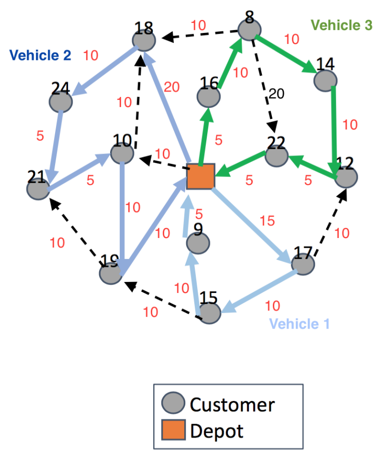

Figure 3: The customers are served by 3 vehicles. Vehicule 1 carries a weight of 17+15+9 = 41 when it leaves the depot. Its route is of length 40. The total cost for the company is 40 + 60 + 45 = 145.

The problem can be formulated as the integer linear program below. The main variables are the binary variables x_{ijk}, *for* $i, j \in V$ *and* $1 \leq k \leq K$, *defined as follows:* $x_{ijk} = 1$, *if the route of vehicle* k *uses the arc* ij *and 0 otherwise. Do not pay attention to the variables* y_i $i \in V$ *which are used in Constraint (9).*

$$
\begin{array}{lllr}
\textit{Min} & \sum_{ij \in A} \sum_k c_{ij} x_{ijk} & & (0) \\
\textit{Subject to:} & & & \\
& \sum_i \sum_k x_{ijk} = 1 & \forall j \in N & (1) \\
& \sum_j \sum_k x_{ijk} = 1 & \forall i \in N & (2) \\
& \sum_i \sum_k x_{ijk} - \sum_j \sum_k x_{ijk} = 0 & \forall i \in N & (3) \\
& \textit{Missing constraint on the maximum weight of a vehicle} & & (4) \\
& \sum_{ij \in A} c_{ij} x_{ijk} \leq T & \forall 1 \leq k \leq K & (5) \\
& \sum_j x_{Djk} = 1 & \forall 1 \leq k \leq K & (6) \\
& \sum_i x_{iDk} = 1 & \forall 1 \leq k \leq K & (7) \\
& y_i - y_j + n x_{iDk} \leq n - 1 & \forall 1 \leq i \neq j \leq n, 1 \leq k \leq V & (9) \\
& x_{ijk} \in \{0, 1\}, y_i \in \mathbb{R} & & (10)
\end{array}
$$

1. *Explain well all constraints, except Constraint (9).*

   For your information, Constraint (9) eliminates subtours, that are routes which are not starting and finishing in the depot. The formulation of this last constraint is a smart way to avoid an exponential number of constraints to prevent an exponential number of subtours.

2. *(*) Write the missing Constraint (4) corresponding to the maximum weight of a vehicle.*

3. (*) *Discuss the efficiency of the fractional relaxation of this program focusing on the maximum weight constraint.*

Another formulation of the problem is provided below. The set Ω of all possible routes is given as input, as well as the constants $v_{ip}, \forall i \in N, p \in \Omega$. $v_{ip} = 1$ if route p visits customer i and 0 otherwise. (Note that the v_{ip} are not variables of the following linear program, there are constants.)
The variables of the linear program are: $\theta_p, \forall p \in \Omega$, $\theta_p = 1$ if route p is used, 0 otherwise.

$$
\begin{array}{lll}
\textit{Min} & \sum_{p\in\Omega} c_p\theta_p & \\
\textit{Subject to:} & & \\
& \sum_{p\in\Omega} v_{ip}\theta_p = 1 \qquad \forall i \in N & (11)\\
& \theta_p \in \{0,1\} &
\end{array}
$$

4. *Explain the linear program, in particular how to define and compute c_p, the objective function, and Constraint (11).*

We consider the input given in Figure 4. We suppose than $P = 20$ (maximum weight of a vehicle) and $T = 32$ (maximum length of a route).

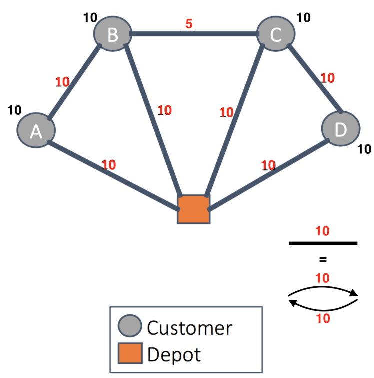

Figure 4: Example. An undirected edge of length l represents two directional arcs of length l.

5. *Provide the list of possible routes. (Two routes serving the same clients are considered equivalent and are counted only once).*

The linear program solving the example of Figure 4 can be written as follows. For a technical reason, we write the constraints of the program as inequalities, instead of equality.

$$
\begin{array}{lccccccccc}
\textit{Min} & 20\theta_1 & +20\theta_2 & +20\theta_3 & +20\theta_4 & +30\theta_5 & +30\theta_6 & +25\theta_7 & & \\
\textit{Subject to:} & & & & & & & & & \\
& \theta_1 & & & & +\theta_5 & & & \geq & 1\\
& & \theta_2 & & & +\theta_5 & & +\theta_7 & \geq & 1\\
& & & \theta_3 & & & +\theta_6 & +\theta_7 & \geq & 1\\
& & & & \theta_4 & & +\theta_6 & & \geq & 1
\end{array}
$$

6. *Provide a possible matching between the routes and the variables of the ILP.*

7. *We go back to the general problem (not the specific example). What is the maximum possible number of routes of a digraph with n customers? Provide a digraph for which the bound is reached and explain. Note that we consider that a route passing through a customer serves the customer.*

8. *Because of this large number of potential routes, we will solve the problem using column generation. We go back to the example given in Figure 4. We start with a trivial initial solution, which is to use the 4 direct routes to serve each client A, B, C, D. (e.g. To serve client A, we use the routes Depot->A->Depot which serves only A). What is the cost of the solution?*

When running the simplex (with θ_A, θ_B, θ_C and θ_D as slack variables), we obtain the following final table:

$$\begin{array}{rcrrrrr}
\theta_1 & = & 1 & -\theta_A & & & \\
\theta_2 & = & 1 & -\theta_A & -\theta_B & & \\
\theta_3 & = & 1 & -\theta_A & & -\theta_C & \\
\theta_4 & = & 1 & -\theta_A & & & -\theta_D \\
\hline
z & = & 80 & -20\theta_A & -20\theta_B & -20\theta_C & -20\theta_D
\end{array}$$

and the values of the optimal solution for the dual problem are $\pi_A = \pi_B = \pi_C = \pi_D = 20$.

10. *What is the solution corresponding to the table? What is its value? What are the variables in the basis? What are the reduced costs of the variables not in the basis?*

To find a new route improving the initial solution, we have to solve the pricing problem. The objective function is to find the column with the minimum reduced cost, $\bar{c} = c - \sum_i a_i \pi_i$, where $a_i = 1$ if customer i is visited and 0 otherwise. The variables are x_{ij}, $\forall ij \in A$, which is equal to 1 if the route uses arc ij and 0 otherwise.

$$\begin{array}{lrl}
\textit{Min} & \sum c_{ij}x_{ij} - \sum_i a_i\pi_i & \\
\textit{Subject to:} & & \\
& \sum_j x_{ij} - \sum_j x_{ji} = 0 & \forall i \in N \\
& \sum_j x_{dj} = 1 & \\
& \sum_i x_{id} = 1 & \\
& a_i \leq \sum_j x_i j & \forall i \\
& \textit{Maximum weight of a vehicle} & \\
& \sum_{ij \in A} x_{ijk} \leq T & \forall 1 \leq k \leq K \\
& \theta_p \in \{0,1\} &
\end{array}$$

11. *Provide the reduced cost of the column corresponding to the route Deport -> A - > B -> Depot.*
12. *Without formally solving the pricing LP, exhibit an optimal route (column) for the example.*

$$\begin{array}{rcrrrrr|c|}
\hline
\theta_1 & = & 1 & -\theta_A & & & & \\
\hline
\theta_2 & = & 1 & -\theta_A & -\theta_B & & & \\
\hline
\theta_3 & = & 1 & -\theta_A & & -\theta_C & & \\
\hline
\theta_4 & = & 1 & -\theta_A & & & -\theta_D & \\
\hline
z & = & 80 & -20\theta_A & -20\theta_B & -20\theta_C & -20\theta_D & \\
\hline
\end{array}$$

13. If you did Question 13, skip this question and go to Question 15. If you are blocked by Question 13, go to Exercise 2.
14. *Write the corresponding dual program. Using complementary slackness, determine the values of an optimal solution of the dual.*
15. *Again, without formally solving the pricing LP, exhibit a new column/route with minimum reduced cost.*
16. *Without writing too many details, continue the resolution till you cannot add an improving column.*
17. *Did you obtained an optimal solution of the problem? Discuss.*

Exercise 2 (Simplex and Duality (5 points, 30 minutes)) This exercise should be only done if you are blocked in Exercise 1 of the Linear Programming section.
We consider the following linear program.

$$
\begin{array}{lccccccccc}
\textit{Min} & 20\theta_1 & +20\theta_2 & +20\theta_3 & +20\theta_4 & +30\theta_5 & +30\theta_6 & +25\theta_7 & & \\
\multicolumn{2}{l}{\textit{Subject to:}} & & & & & & & & \\
 & \theta_1 & & & & +\theta_5 & & & \geq & 1 \\
 & & \theta_2 & & & +\theta_5 & & +\theta_7 & \geq & 1 \\
 & & & \theta_3 & & & +\theta_6 & +\theta_7 & \geq & 1 \\
 & & & & \theta_4 & & +\theta_6 & & \geq & 1
\end{array}
$$

1. *Solve the linear program using the simplex method.*
2. *Provide the optimal solution of the dual using complementary slackness.*