UBINET, Master 2 IFI Algorithms for telecommunications

Final Exam, November 2016

2 hours

Course and 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.** *The points are indicated so you may adapt your effort.*

1 Linear Programming

Exercise 1 (Modeling as a linear program (4 points, 20 minutes)) *On Nice Street cars can be parked on both sides of the street. Mr. Garibaldi, who lives at number 1, is organizing a party for around 30 people, who will arrive in 15 cars. The length of the i-th car is λ_i, expressed in meters as follows:*

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
λ_i	4	4.5	5	4.1	2.4	5.2	3.7	3.5	3.2	4.5	2.3	3.3	3.8	4.6	3

In order to avoid bothering the neighbours, Mr. Edmonds would like to arrange the parking on both sides of the street so that the length of the street (i.e the maximum length on both sides) occupied by his friends' cars should be minimum.

1. *Formulate the problem as an integer linear program. Explain well all constraints.*
2. *How does the program change if on the street left side the cars should not occupy more than 15 m?*
3. *How does the program change if the cars should not occupy more than 15 m on at least one of both street sides?*

Exercise 2 (Duality and Optimality certificates. (4 points, 20 minutes)) *Consider the following linear program.*

$$\begin{array}{lrcrcl} \textit{Maximize} & 4x_1 & + & x_2 & & \\ \textit{Subject to:} & & & & & \\ & 2x_1 & - & 3x_2 & \leq & 6 \\ & 4x_1 & - & 2x_2 & \leq & 8 \\ & x_1 & + & 2x_2 & \leq & 4 \\ & & & x_1, x_2 & \geq & 0 \end{array}$$

1. *Write the dual program of this linear program.*
2. *Is $x_1 = 2$, $x_2 = 1$ an optimal solution of the linear programme above? To answer, use the method to provide optimality certificates seen during the class. Detail all steps.*

2 Approximation algorithms

Exercise 3 (Multiway Cut **Problem (12 points)) All questions are independant. Do not spend too much time on a question.**

Recall that, for any $c \geq 1$, a c-approximation algorithm for a minimization problem is an algorithm that computes **in polynomial-time** a feasible solution such that

$$OPT \leq value(solution) \leq c \cdot OPT$$

where OPT is the optimal value.

Notations: In this section, n will always denote the number of vertices of a graph and m will denote the number of its edges.

Problème 1 (Multiwaycut) *Let $G = (V, E)$ a weighted graph with $w(e)$ the weight of edge $e \in E$. Given a set of terminals $S = \{s_1, s_2, ..., s_k\} \subseteq V$, a multiway cut is a set of edges, C, whose removal disconnects the terminals from each other, that is in $G \backslash C$, the terminals are in pairwise distinct connected components. The multiway cut problem asks for the minimum weight of such set (the weight of C is $w(C) = \sum_{e \in C} w(e)$).*

The problem of finding a minimum weight multiway cut is NP-hard for any fixed $k \geq 3$. Observe that the case $k = 2$ is precisely the minimum st-cut problem.

Question 1 *Give an exponential-time algorithm that computes a minimum multiway cut in arbitrary graphs. Provide (and prove) its time-complexity as a big-O of a function of m and n.*

Define an isolating cut for s_i to be a set of edges whose removal disconnects s_i from the rest of the terminals.

Algorithm 1 A 2-approximation algorithm for MULTIWAYCUT

Require: A weighted graph $G = (V, E)$ with $w(e)$ the weight of edge $e \in E$. A set of k terminals $S = s_1, s_2, ..., s_k \subseteq V$.

1: For each $i = 1, \ldots, k$, compute a minimum weight isolating cut for s_i, say C_i.

2: Discard the heaviest of these cuts, and output the union of the rest, say C.

We will prove the following result:

Theorem 1 *Algorithm 1 achieves an approximation guarantee of $2 - 2/k$.*

Question 2 *How can a minimum weight isolating cut for s_i be computing efficiently?*
Hint: The first step of this efficient computation is to identify the terminals in $S \setminus \{s_i\}$ into a single node.

Question 3 *Show that C, computed by Algorithm 1, is a multiway cut.*

Let A be an optimal multiway cut in G. We can view A as the union of k cuts as follows:

Question 4 *Show that the removal of A from G creates* exactly *k connected components, each having one terminal inside.*

Let A_i be the cut separating the component containing s_i from the rest of the graph, i.e., let K_i be the connected component of $G \setminus A$ that contains s_i and let A_i be the set of edges in A with one end in K_i. Then

$$A = \cup_{i=1}^{k} A_i.$$

Question 5 *Show that*

$$\sum_{i=1}^{k} w(A_i) = 2w(A).$$

Question 6 *Explain why we have $w(C_i) \leq w(A_i)$.*

Notice that this already gives a factor 2 algorithm, by taking the union of all k cuts C_i. The proposed algorithm does better.

Question 7 *Explain why we have*

$$w(C) \leq \left(1 - \frac{1}{k}\right) \sum_{i=1}^{k} w(C_i)$$

Question 8 *Use the results of the three last questions to finish the proof of Theorem 1.*

We now prove that the bound of Algorithm 1 is tight.

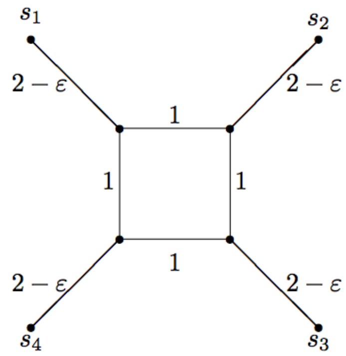

Figure 1: An example of graph which is tight for the bound of Algorithm 1.

Question 9 *Show that the instance in Figure 1 is tight for the bound of Algorithm 1.*

Question 10 *Write an integer linear program (LP1) to model the problem.*
Hint: *Use variables $d_{uv} \in \{0, 1\}$ for $uv \in E$, such that $d_{uv} = 1$ if uv is in the multiway cut. Write one constraint per path p such that p is a path between two terminals.*

Question 11 *Write the dual program of (LP1).*

Question 12 *Give an interpretation of the signification of the dual variables and of the dual problem.*

Question 13 *A natural greedy algorithm for computing a multiway cut is the following. Starting with G, compute minimum $s_i s_j$ cuts for all pairs s_i, s_j that are still connected and remove the lightest of these cuts; repeat this until all pairs s_i, s_j are disconnected. Prove that this algorithm also achieves a guarantee of $2 - 2/k$.*