Communications and Transport Systems
Department of Science and Technology **Fall 2018**
Linköping University

Christiane Schmidt

Design and Analysis of Algorithms Part 2 - Approximation and online algorithms homework 5, 28.11.2018

Problem 1 (Greedy Set Cover Algorithm):
Apply the Greedy Set Cover Algorithm (Algorithm 2.3 from the lecture) to the following Set Cover instance:
$c(S_i) = |S_i| + 1$, $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20\}$, and
$S_1 = \{1, 2, 3, 4\}$
$S_2 = \{5, 6, 7, 8\}$
$S_3 = \{9, 10, 11, 12\}$
$S_4 = \{13, 14, 15, 16\}$
$S_5 = \{17, 18, 19, 20\}$
$S_6 = \{1, 2, 3, 5, 6, 7, 9, 10, 11\}$
$S_7 = \{14, 15, 16, 17, 18, 19\}$
$S_8 = \{12, 13, 14, 15\}$
$S_9 = \{4, 5, 6\}$
$S_{10} = \{7, 8, 9\}$
$S_{11} = \{18, 19, 20\}$.

In case the maximum in step 2 is not uniquely defined, choose set S_i with minimum index.
What is the value of the computed set cover?
Can you give a better set cover?

Problem 2 (Maximum Coverage Problem (MCP)):
Given: $\mathcal{S} = \{S_j | j \in J\}, (S_j \subseteq U = \{1, \dots, n\})$, positive weights w_i of the elements, an integer K.
Task: Find $X \subseteq J$ with $|X| = k$, such that $\sum_{i \in \cup_{j \in X} S_j} w_i$ is maximal.
Let us denote $w(M) = \sum_{i \in M} w_i$.
Greedy algorithm for MCP:
(1) Let $X := \emptyset$
(2) Choose $j \in J$, such that $w(\cup_{i \in X \cup \{j\}} S_i) - w(\cup_{i \in X} S_i)$ is maximal.
(3) Set $X = X \cup \{j\}$, continue unitl $|X| = K$.
Show: The greedy algorithm for MCP is an $(1 - (1 - \frac{1}{k})^k$-approximation algorithm. (Because $(1 - (1 - \frac{1}{k})^k < 1 - \frac{1}{e}$, this implies a $1 - \frac{1}{e}$-approximaiton.)

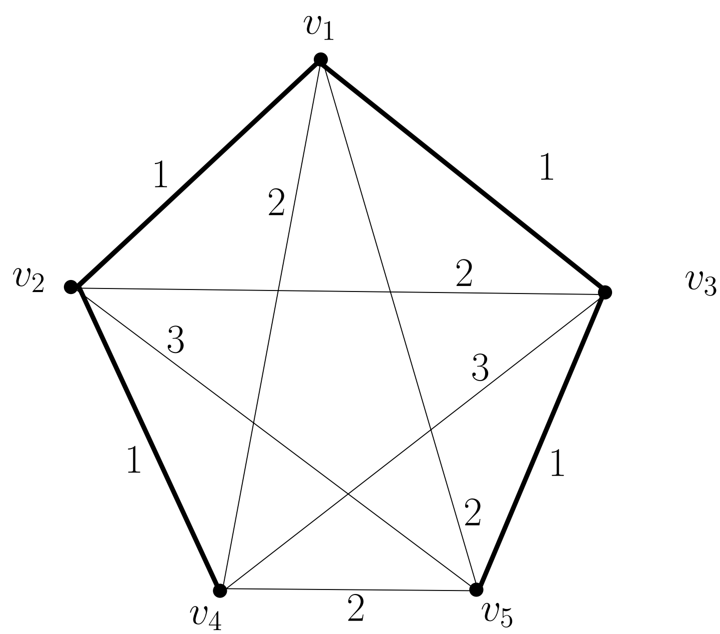

Figure 1: Graph H. An MST rooted in v_1 is shown in bold.

Problem 3 (4/3-approximation for $(1, 2)$-TSP):
Consider a complete undirected graph G in which all edges have length either 1 or 2 (G satisfies the triangle inequality!). Give a 4/3-approximation for this special TSP variant.
Hint: Start with a minimum 2-matching in G. A 2-matching is a subset M_2 of edges so that every vertex in G is incident to exactly two edges in M_2. Note: a 2-matching can be computed in polynomial time.

Problem 4 (Bottleneck TSP):
Take a graph G with edge costs that satisfy the triangle inequality. We want to find a Hamiltonian cycle C for which the maximum cost edge in C is minimized.

(a) Give a 3-approximation algorithm for this problem.
Hints: (i) Consider the MST of G. (ii) Think about "appropriate" shortcuts.

(b) Apply your algorithm to the graph H from Figure 1, using the given MST.