TD: Approximation Algorithms

Algorithmique et Programmation

Nicolas Schabanel
nicolas.schabanel@cnrs.fr

Pierre Senellart
pierre.senellart@ens.fr

December 1st, 2016

The goal of this exercise session is to get familiar with techniques to construct approximation algorithms and to analyze approximation ratios.

1 Randomized Approximation of Max k-Cut

A k-cut of an undirected graph $G = (V, E)$ is a partition of the vertices into k disjoint sets $V_1, \dots, V_k$ such that $V = V_1 \cup \dots \cup V_k$. For two disjoint sets of vertices $A, B \subseteq V$, we denote by $E(A, B) := \{\{u, v\} \in E \mid u \in A \wedge v \in B\}$ the set of edges between A and B. We define the *size* of the k-cut as $\text{size}(V_1, \dots, V_k) = \sum_{i<j} |E(V_i, V_j)|$. The Max k-Cut problem consists in finding a k-cut of *maximum* size.

1a. Propose a simple *randomized* linear-time $(1 - 1/k)$-approximation algorithm A for Max k-Cut, i.e., an algorithm that makes random choices and such that

$$\mathbb{E}[\text{size}(A(G))] \geqslant (1 - 1/k)\,\text{OPT}(G)$$

for every graph G, where $A(G)$ and $\text{OPT}(G)$ denote respectively the k-cut output by A and the size of an optimal k-cut of G.

Give a proof for the approximation ratio.

1b. Exhibit a family of graphs for which the ratio $\frac{\mathbb{E}[\text{size}(A(G))]}{\text{OPT}(G)}$ is exactly $1 - 1/k$.

2 Coloring 3-Colorable Graphs

Consider the following problem: Given an undirected graph $G = (V, E)$, color its vertices with the minimum number of colors so that the two endpoints of each edge receive distinct colors.

2a. Give a greedy algorithm for coloring G with $\Delta + 1$ colors, where Δ is the maximum degree of a vertex in G. What is the complexity of this algorithm?

2b. Give an algorithm for coloring a 3-colorable graph with $O(\sqrt{n})$ colors, where n is the number of vertices in the graph.

Hint: Prove that for any vertex v, the induced subgraph on its neighbors is bipartite. How many colors do you need to color a vertex of degree $> \sqrt{n}$ and its neighborhood? How many colors do you need to color all the vertices with degree $\leqslant \sqrt{n}$?

3 Set Cover

Consider a set X of n elements and a family $\mathcal{S} = \{S_1, \dots, S_m\}$ of m subsets of X such that $\bigcup_{S \in \mathcal{S}} S = X$. The Set Cover problem is to find a subset $\mathcal{S}'$ of $\mathcal{S}$ of minimal cardinality such that $X = \bigcup_{S_i \in \mathcal{S}'} S_i$.

3a. Propose a greedy algorithm for Set Cover. What is its complexity?

3b. Without loss of generality, we assume that the greedy algorithm chooses, in order, the sets $S_1, \dots, S_p$. For $x \in X$, let $c_x := \frac{1}{|S_i \setminus (S_1 \cup \dots \cup S_{i-1})|}$ where $i = \min\{1 \leqslant j \leqslant p \mid x \in S_j\}$. Intuitively, we consider that each time the greedy algorithm adds a set S_i to its cover, it adds a cost of 1 and distributes this cost uniformly as a cost c_x for every $x \in X$ that was not yet covered.

Fix a set $S \in \mathcal{S}$. Let:

$$\begin{aligned} u_0 &= |S| \\ u_1 &= |S \setminus S_1| \\ &\dots \\ u_i &= |S \setminus (S_1 \cup \dots \cup S_i)| \\ &\dots \end{aligned}$$

Let k be the minimum index such that $u_k = 0$. We denote the harmonic series as $H(i) := \sum_{j=1}^{i} \frac{1}{j}$. By using the u_i's, show that $\sum_{x \in S} c_x \leqslant H(|S|)$.

3c. Show that the greedy algorithm is a $H(\max_{S_i \in \mathcal{S}} |S_i|)$-approximation algorithm for Set Cover.

3d. Deduce the existence of a polynomial-time $O(\log n)$-approximation algorithm for Set Cover.

4 Metric k-Center

Consider the Metric k-Center problem: Let $G = (V, E)$ be a complete undirected graph with n vertices and $m = \frac{n(n-1)}{2}$ edges, and edge costs w satisfying the triangle inequality. Given k a positive integer, let us define for each vertex $v \in V$ and each vertex set $S \subseteq V$, $\text{connect}(v, S)$ to be the cost of the cheapest edge from v to a vertex in S ($\text{connect}(v, S) = 0$ if $v \in S$). The goal is to find a set $S \subseteq V$, with $|S| = k$, so as to minimize $\max_v\{\text{connect}(v, S)\}$, that is to say minimizing the maximum distance from any vertex to S. Let OPT denote the optimum cost of the k-center.

Note that if the cost of the optimal solution is t, then one need only consider edges with weight $\leqslant t$. To exploit this idea, we sort the edges in non-decreasing cost order: $w(e_1) \leqslant \cdots \leqslant w(e_m)$. We then introduce for $1 \leqslant i \leqslant m$ the graph $G_i = (V, E_i)$ where $E_i = \{e_1, \ldots, e_i\}$.

We say that a set of vertices $D \subseteq V$ is a *dominant* of a graph $H = (V, F)$ if for all $u \in V$, $u \notin D \Rightarrow (\exists v \in D)\ \{u, v\} \in F$. We denote by $\mathrm{dom}(H)$ the minimum size of a dominant set in H: $\mathrm{dom}(H) := \min\{|D| \text{ such that } D \text{ is a dominant set of } H\}$.

4a. Let $i^* := \min\{i \mid \mathrm{dom}(G_i) \leqslant k\}$. Show that $\mathrm{OPT} = w(e_{i^*})$.

4b. Given a graph $H = (V, F)$, the *square graph* H^2 of H is the graph that contains an edge $\{u, v\}$ for all pair of vertices u and v connected by a path of at most 2 edges in H. We say that a set of vertices $I \subseteq V$ is *independent* if for all $u, v \in I, \{u, v\} \notin F$. An independent set I is said *maximal* if for all $u \in V \setminus I$, $I \cup \{u\}$ is not independent.

Propose an efficient algorithm to compute a maximal independent set of a graph H. What is its complexity?

4c. Independent set will allow us to compute a lower bound on OPT as follows: Show that for all graph H and all independent set I in H^2, $|I| \leqslant \mathrm{dom}(H)$.

Hint: What is the square of a star graph?

4d. Let us now consider Algorithm 1.

Algorithm 1 A 2-approximation for Metric k-Center

```
Compute G_1^2, G_2^2, ..., G_m^2
Compute a maximal independent set I_i in each G_i^2.
j ← min{i : |I_i| ⩽ k}.
Output I_j.
```

What is its complexity?

4e. Show that Algorithm 1 is a 2-approximation for Metric k-Center.

4f. Exhibit a family of tight instances.