Graded Homework 1, exercise 3

Marco Milanta

October 28, 2021

Avid Traveler (15 points)

There are n cities $\{1, \cdots, n\}$ in Europe, and a set of m local train tickets $\{T_1, \cdots, T_m\}$, where T_i is a list of cities you can travel to using the ith ticket, and let $p_i \in \{1, \cdots, n^{100}\}$ to be the price of the ith ticket. We are guaranteed that each city c_i is in at least 1 local ticket, but is in at most 10 local tickets. There are also 10 super tickets $\{S_1, \cdots S_{10}\}$, each costs s_i. You want to buy a set of tickets such that you can travel to any city in Europe while minimizing the cost. Moreover, as a condition for buying super ticket, we are not allowed to have a city simultaneously in a local ticket and a super ticket we bought (i.e. $T_i \cap S_j = \emptyset$ for any T_i and S_j we buy). Devise a polynomial-time approximation algorithm for this problem with a constant approximation guarantee, and for the smallest approximation factor that you can achieve.

Solution

Fix the super tickets: Suppose we are taking some super tikets

$$S_{i_1}, \ldots, S_{i_k} \subseteq \{S_1, \ldots, S_{10}\}, \qquad k \leq 10.$$

Now we want to cover the leftover cities with local train tickets. The leftover cities will be

$$C_{i_1,\ldots,i_k} := \{1, \ldots, n\} \setminus \bigcup_{j=1}^{k} S_{i_j}$$

Furthermore, we consider the set of local tickets that we can still buy considering the condition that there can not be a city both in a local ticket and in a super ticket:

$$\mathcal{T}_{i_1,\ldots,i_k} := \left\{T \in \{T_1, \ldots, T_m\} \mid T \cap S_{i_j} = \emptyset, j = 1, \ldots, k\right\}$$

Now we can ask ourselves what is the smallest set covering of sets in $\mathcal{T}_{i_1,\ldots,i_k}$ to cover $C_{i_1,\ldots,i_k}$. To solve this problem, we use a greedy algorithm based on the assumption that each city in $C_{i_1,\ldots,i_k}$ will appear in at most 10 trains. We have theorem 4.4 from lecture notes that guarantees us a polynomial approximated algorithm that finds a solution $\mathcal{T}' \subseteq \mathcal{T}_{i_1,\ldots,i_k}$ that minimizes:

$$c(\mathcal{A}_{i_1,\ldots,i_k}) = \sum_{t \in \mathcal{T}'} p_t$$

where $\mathcal{T}'$ is a set cover of $C_{i_1,\ldots,i_k}$. The approximation will only have a factor of 10

For all possible super tickets combinations: The idea is to iterate the previous point for all possible subset of $\{S_1, \ldots, S_{10}\}$. This yields $2^{10} = 1024$ different iterations. Even though this number is big, it doesn't scale up: the number of super tickets is not a parameter, but it is fixed to be 10.

Finally, we have 1024 different solutions, and we pick the best one considering also the cost of the super tickets

$$c(\mathcal{A}) = \min_{\{i_1,\ldots,i_k\}\in 2^{1:k}} \left(\sum_{j=i_1,\ldots,i_k} s_j + c(\mathcal{A}_{i_1,\ldots,i_k}) \right).$$

Note that we cannot guarantee that each combination of $S_{i_1}, \ldots, S_{i_k}$ yields a valid solution, but we can guarantee that for at least one combination it happens: we don't take any super tickets. Therefore, the minimum is still well-defined (we just take $c(\mathcal{A}_{i_1,\ldots,i_k}) = \infty$ if $\{S_{i_1}, \ldots, S_{i_k}\}$ doesn't allow any solution). Assume that the optimal solution picks $\mathcal{S}_{OPT} \subseteq \{S_1, \ldots, S_m\}$ as super tickets. Then we know that

$$c(\mathcal{A}) \leq \sum_{j\in\mathcal{S}_{OPT}} s_j + c(\mathcal{S}_{OPT}) \leq 10c(OPT).$$

Finally, we can conclude that we have found a 10-approximation algorithm. 10 was the best we could do, and indeed, we suspect that it is impossible to do better than this.