Advanced Algorithms Deadline: October 24, 2018, 11:59 pm

Graded Homework 1

Lecturer: Mohsen Ghaffari

Note 1: *For all problems, besides providing the algorithm, you should present a complete analysis that proves the claimed performance, e.g., approximation guarantee or running time.*

Note 2: *Your solutions must by typeset in LaTeX, and you should email the pdf version to Manuela Fischer (manuela.fischer at inf.ethz.ch) by the indicated deadline.*

1 Team Formation (10 points)

We have a collection of n students, and a prescribed list L of m potential teams, each consisting of exactly 3 students. Each student might be in many potential teams. Devise a polynomial-time 3-approximation algorithm for the problem of selecting the maximum number of teams, subject to the constraint that each student can be in at most one selected team.

2 Patrolling (15 points)

We are given a list L of m buildings, and a collection of n patrolling teams that we can recruit, each of which can patrol some fixed subset of the buildings. Devise a polynomial-time approximation algorithm for the objective of maximizing the number of patrolled buildings, subject to the constraint that we have the budget to pay for only k patrolling teams. What is the best approximation that you can obtain?

3 Vertex Cover via DFS (15 points)

Let $G = (V, E)$ be a graph and let T be a Depth First Search tree of G. Define S to be the set of all non-leaf vertices in the tree T. Prove that S is a vertex cover of G and that it gives a 2 approximation for the minimum cardinality vertex cover problem.

4 Server Optimization (20 points)

We have a set $\mathcal{S}$ of n servers and a collection $\mathcal{C}$ of m clients. Each client should be served by one server but each server can serve many clients. However, for each $i \in \{1, \dots, m\}$, the i^{th} client can be served only by the servers in the subset $S_i \subseteq \mathcal{S}$. We are given the guarantee that $|S_i| \geq k, \forall \in i\{1, \dots, m\}$. Devise a polynomial-time algorithm that finds a collection of at most $n \cdot \frac{O(\log m)}{k}$ servers that can serve all the clients.

5 Scheduling on your Quadcore Computer (20 points)

The operating system of your quadcore computer has a list of n computational tasks, each of which it should assign to one of its four identical cores. Performing the i^{th} task takes $t(i)$ units of time, for each core. Devise a Fully Polynomial Time Approximation Scheme (FPTAS) which assigns the tasks to the cores, with the objective of minimizing the time-span in which all cores are done with their computational tasks.

6 Recruiting for Social Welfare (20 points)

We are given a list of n tasks, where performing the i^{th} task creates some payoff $p(i) \in [P_{\min}, P_{\max}]$, and where $\frac{P_{max}}{P_{min}}$ is some fixed constant c. We would like to recruit as many as possible workers so that each of them performs some of these tasks, subject to the constraint that the tasks performed by each worker should in total create at least P_{max} payoff for him/her. That is, the summation of the payoff of the tasks assigned to each worker should be at least P_{max}. Each task can be assigned to only one worker, and it can not be split among different workers. Devise an asymptotic Polynomial Time Approximation Scheme (PTAS) for the problem of maximizing the number of recruited workers. Here, asymptotic PTAS means we want a $1 + \varepsilon$ approximation, for any desirable constant $\varepsilon > 0$, when OPT— that is, the maximum number of workers that we can recruit, subject to the payoff constraint per worker—is assumed to be sufficiently larger (e.g., greater than any desirably larger fixed constant).