# CS264: Homework #9

Due by midnight on Thursday, March 16, 2017

**Instructions:**

(1) Form a group of 1-3 students. You should turn in only one write-up for your entire group. See the course site for submission instructions.

(2) Please type your solutions if possible and feel free to use the LaTeX template provided on the course home page.

(3) All students should complete all of the exercises. Students taking the course for a letter grade should also complete the problems.

(4) Write convincingly but not excessively. Exercise solutions rarely need to be more than 1-2 paragraphs. Problem solutions rarely need to be more than a half-page (per part), and can often be shorter.

(5) You may refer to your course notes, and to the textbooks and research papers listed on the course Web page *only*. You cannot refer to textbooks, handouts, or research papers that are not listed on the course home page. (Exception: feel free to use your undergraduate algorithms textbook.) Cite any sources that you use, and make sure that all your words are your own.

(6) If you discuss solution approaches with anyone outside of your team, you must list their names on the front page of your write-up.

(7) Exercises are worth 5 points each. Problem parts are labeled with point values.

(8) No late assignments will be accepted.

## Lecture 17 Exercises

### Exercise 45

The point of this exercise is to extend the analysis from lecture to the case of $n$ points in $d$ dimensions (in lecture, $d = 2$). Assume that every point's density function $f_i$ has support in $[0, 1]^d$, with $f_i(\mathbf{x}) \leq \frac{1}{\sigma}$ for every $\mathbf{x} \in [0, 1]^d$. The objective function is again to minimize the total $\ell_1$ distance of the tour, with the distance between two points $\mathbf{x}, \mathbf{y}$ defined as $\|\mathbf{x} - \mathbf{y}\|_1 = \sum_{i=1}^{d} |x_i - y_i|$.

Prove that for every fixed constant $d$, the expected running time of the 2-OPT local search algorithm (defined as in lecture) is $O(\sigma^{-1} n^6 \log n)$. Roughly what is the dependence of your bound on $d$ (polynomial, exponential, doubly exponential, etc.)?

## Lecture 18 Exercises

### Exercise 46

A *fully polynomial-time approximation scheme (FPTAS)* for a maximization problem takes as input a problem instance and a parameter $\epsilon$, and returns a feasible solution with objective function value at least $(1 -$

$\epsilon$) times the maximum possible, in time polynomial in the size of the instance and in $\frac{1}{\epsilon}$. Prove that if a binary optimization problem with a maximization objective admits an FPTAS, then it also admits a pseudopolynomial-time algorithm (i.e., an exact algorithm that, when the objective function coefficients $v_i$ are integers, runs in time polynomial in the instance size and $\max_{i=1}^{n} v_i$).

[Comment: there are also converses to this statement, but they are harder to prove.]

### Exercise 47

Look up the terms "strongly NP-hard" and "$ZPP$." Recall from lecture that if a binary optimization problem admits an algorithm with polynomial smoothed complexity, then it also admits an algorithm with pseudopolynomial expected running time. Explain why this result implies that strongly $NP$-hard binary optimization problems do not have polynomial smoothed complexity, unless $NP \subseteq ZPP$.

### Exercise 48

The point of this exercise is to investigate the independence assumptions we've been making in our smoothed analyses. Recall the following:

(i) In Lecture #17, our perturbation model assumed that each point in the TSP instance was drawn independently from a distribution with density function bounded above by $1/\sigma$.

(ii) In Lecture #18, our perturbation model assumed that each value in the instance of a binary optimization problem was drawn independently from a distribution with density function bounded above by $1/\sigma$.

Suppose we relax these independence assumptions to the following: for every point/value, conditioned arbitrarily on the other points/values, the conditional distribution has density function bounded above by $1/\sigma$. Do either of the smoothed analyses continue to hold, with the same proofs, under this relaxed assumption? Justify your answer.

## Problems

### Problem 32

(20 points) Recall the Max Cut problem, where the input is an undirected graph in which the edges have nonnegative weights, and the goal is to compute a cut that maximizes the total weight of the crossing edges. Given a cut, the allowable *local moves* are to pick one vertex and move it to the opposite side of the cut. (In addition, both sides of the cut must stay non-empty.) *Local search* means repeatedly making local moves that strictly increase the weight of the cut until a *local optimum* — a cut from which there are no improving local moves — is reached. It turns out that, in the worst case, local search can require an exponential number of iterations to reach a locally optimal cut.

Let's consider the smoothed complexity of local search for the Max Cut problem (analogous to that of the 2-OPT heuristic that we studied in Lecture #17). Suppose each edge weight is drawn independently from a probability distribution with a density function that is bounded everywhere by $1/\sigma$, where $\sigma$ is a parameter. Assume that all edge weights lie in $[0, 1]$ (with probability 1). Suppose also that the graph has maximum degree $O(\log n)$, where $n$ is the number of vertices. Prove that local search has polynomial smoothed complexity in such instances, meaning that the expected running time (over the random edge weights) to reach a locally optimal cut is polynomial in $n$ and $1/\sigma$.

[Hints: Consider first a fixed local move — what is the probability that it is an improving move that makes very little progress? Second, given that the maximum degree is small, how many "fundamentally distinct" local moves are there?]

## Problem 33

Recall the scheduling problem mentioned in lecture, which provides another example of a binary optimization problem that can be solved in pseudopolynomial (and hence, by Lecture #18, smoothed polynomial) time. The input consists of $n$ jobs, each with a known positive processing time $p_j$, deadline $d_j$, and cost $c_j$. You should assume that all costs are integral. The feasible solutions are orderings of these jobs on a single machine. The *finishing time* $F_j$ of a job $j$ in an ordering is the sum of $p_j$ and the processing times of all the jobs scheduled prior to $j$. A job is *late* in an ordering if its finishing time is strictly larger than its deadline. The goal of the problem is to compute the ordering of the jobs that minimizes the total cost of the late jobs.

(a) (3 points) Prove that if a subset $S$ of jobs can all be scheduled to finish by their deadlines, then scheduling them in order of increasing deadline accomplishes this.

(b) (7 points) Give a dynamic programming algorithm that solves the scheduling problem in time polynomial in $n$ and $C$, where $C = \max_{j=1}^{n} c_j$.

## Problem 34

(15 points) The point of this problem is to investigate analogs of the Isolation Lemma that accommodate random constraints rather than a random objective function value. Consider a binary optimization problem with a maximization objective. Let the objective function $\max \sum_{i=1}^{n} v_i x_i$ be fixed, with $v_i > 0$ for every $i$. Also fixed is a preliminary feasible set $F \subseteq \{0,1\}^n$. You can assume that for every $i = 1, 2, \ldots, n$, there are members $\mathbf{x}$ of $F$ with $x_i = 0$ and with $x_i = 1$. In addition, we consider a random linear constraint of the form $\sum_{i=1}^{n} w_i x_i \leq W$. Assume that $W$ is fixed and at least $t$, where $t \geq 0$ is the minimum number of 1s in a member of $F$. Assume that each $w_i$ is drawn independently from $[0,1]$ according to one of our usual smoothed distributions, with density function $f_i : [0,1] \to [0, \frac{1}{\sigma}]$ for a parameter $\sigma$. The final (random) feasible set is defined as the set of $\mathbf{x} \in F$ with $\sum_{i=1}^{n} w_i x_i \leq W$. Note that under our assumptions, the feasible set is non-empty with probability 1.

Define the *loser gap* $L$ as follows: let $V^*$ denote the maximum value of a feasible solution, let $\bar{\mathbf{x}}$ minimize $\sum_{i=1}^{n} w_i x_i$ over all $\mathbf{x} \in F$ with $\sum_{i=1}^{n} v_i x_i > V^*$ and $\sum_{i=1}^{n} w_i x_i > W$, and set $L = \sum_{i=1}^{n} w_i \bar{x}_i - W$. Note that $L$ is a random variable. Prove that the probability (over the $w_i$'s) that $L$ is less than $\epsilon$ is at most $n\epsilon/\sigma$.

[Hint: follow the same proof template as for analyzing the winner gap. Analyze the probability that a variable $x_i$ is "$\epsilon$-bad," in the sense that if $\bar{\mathbf{x}}^{(i)}$ denotes the maximum-value feasible solution with $x_i = 0$, then there is a solution $\mathbf{x}^{(i)} \in F$ with $x_i = 1$, $\sum_{j=1}^{n} v_j x_j^{(i)} > \sum_{j=1}^{n} v_j \bar{x}_j^{(i)}$, and $\sum_{j=1}^{n} w_j x_j^{(i)} \in (W, W + \epsilon)$.]