CS G713: Advanced Algorithms	Problem Set 5
Assigned: December 2, 2008	Due: December 16, 2008 before class

Guidelines If you write "I do not know how to solve this problem" then you get 1/4 of the score for the problem. If you write nonsense then you get 0. To hand in your solutions: Give it to me, slide it under my door West Village H (246), or email it to csg713-instructor@ccs.neu.edu.

For this problem set **you cannot collaborate: You must solve the problems on your own** (but as usual you can ask the instructor).

Problem 1. 2SAT. A k-*CNF* formula consists of a conjunction of clauses, where each clause contains **at most** k literals. A literal is either a Boolean variables x_i or its negation $\neg x_i$. For example $(x_1) \bigwedge (\neg x_1 \bigvee x_2)$ is a 2-CNF formula.

The k-SAT problem is the problem of determining, for a given k-CNF formula, whether there exists an assignment of the variables that makes the formula true, i.e., it satisfies all the clauses. The MAX-k-SAT problem is the problem of finding, for a given k-CNF formula, an assignment that satisfies as many clauses as possible.

(1) Given an efficient (poly(n) time) algorithm for 2-SAT. Hint: Construct a graph which has the literals as nodes (i.e., both variables and their negations). View each clause as implications; for example view $(x \bigvee y)$ as $(\neg x \rightarrow y)$ and $(\neg y \rightarrow x)$. Prove that a certain property of the graph holds if and only if the formula is satisfiable. For partial credit, prove just one direction of the "if and only if."

(2) Show that MAX-2-SAT is NP-hard. Hint: Show that an efficient algorithm for it would imply an efficient algorithm for 3-SAT, proved NP-complete in class. For this, use the following gadget: Replace a clause $(x \vee y \vee z)$ in a 3-SAT instance with the following ten: $x, y, z, w, (\neg x \vee \neg y), (\neg y \vee \neg z), (\neg z \vee \neg x), (x \vee \neg w), (y \vee \neg w), (z \vee \neg w)$.

(3) Give an efficient (poly(n) time) randomized algorithm for MAX-2-SAT that produces an assignment for the variables that in expectation satisfies at leat half the clauses of the input formula. (Therefore this algorithm has approximation factor 2.)

(4) Assuming (3), give an algorithm that takes as input a MAX-2-SAT formula and a parameter $\varepsilon > 0$, runs in times poly($n, 1/\varepsilon$), and with probability $(1 - \varepsilon)$ produces an assignment that satisfies at least a $(1 - \varepsilon)/2$ fraction of the clauses.

Hint: Use Markov's inequality to convert a result that holds in expectation into a result that holds with high probability; then amplify the probability.

Note: There are deterministic algorithms that always satisfy at least half the clauses, and thus in particular are a valid solution to (4). To solve (4) you can exhibit such an algorithm, but, as hinted above, that is not the only way.

(5) In this problem you will prove correct a formulation of MAX-2-SAT as a strict quadratic program. The formulation is as follows. For a MAX-2-SAT instance in vari-

ables $x_1, \ldots, x_n \in \{0,1\}$, let $y_0, y_1, \ldots, y_n$ be variables $\in \{-1,1\}$. We think of a variable $x_i, i \in \{1, \ldots, n\}$, in the MAX-2-SAT instance as being true if $y_i = y_0$ and false otherwise. Prove that the value of MAX-2-SAT can be written as

$$\max \sum_{0 \le i < j \le n} a_{i,j}(1 + y_i y_j) + b_{i,j}(1 - y_i y_j),$$

where the $a_{i,j}$ and $b_{i,j}$ are constants computable in polynomial time given the MAX-2-SAT instance.

For the next problem, you will need to consider the *vector relaxation* of the above. Recall this is the program where each variable y_i is replaced with a vector v_i (in $n+1$ dimensions) of length 1, and $y_i \cdot y_j$ is replaced with the inner product of the two vectors. You can assume that such programs can be solved exactly.

(6) Give a randomized algorithm that produces a better approximation than 2 (derived in (3)) for MAX-2-SAT. Use (a) the above relaxation to a vector program, (b) the fact that the inner product between two vectors of length 1 equals the cosine of the angle between them, and (c) the following two trigonometric inequalities that hold for an absolute constant $\alpha > 0.87$ and any angle $\theta \in [0, \pi]$:

$$\frac{\theta}{\pi} \ge \frac{\alpha}{2}(1 - \cos\theta), \quad 1 - \frac{\theta}{\pi} \ge \frac{\alpha}{2}(1 + \cos\theta).$$