Additional Practice Problems for the Final Exam

- The following set of practice problems has been provided to supplement the earlier set of practice problems that were released. Please see the instructions provided in that document.

Problem-1: T/F Questions. [3 points each] Answer *true* or *false* to each of the following. *No justification is required for your answer.*

(a) [T/F] Consider a function, $f(x)$, that is *strictly convex* on the interval $[-2017, 2017]$. Then $\frac{d^2}{dx^2}f(x) > 0$ for all $x \in [-2017, 2017]$.

Solution: False. If $f(x) = x^6 + x^5$ then $\frac{d^2}{dx^2}f(x) = 30x^4 + 20x^3$ and $\frac{d^2}{dx^2}f(-0.3) < 0$.

(b) [T/F] The gradient descent algorithm either diverges to $-\infty$ or converges to a local minimum or maximum.

Solution: False. Consider that the algorithm may converge to a saddle point.

(c) [T/F] Given a function f that is has α-strong convexity and β-smoothness. Suppose we can show that f has $\frac{\alpha}{2}$-strong convexity. Then we also expect for the gradient descent algorithm to converge faster.

Solution: True. Recall that the gradient descent algorithm will converge to within ϵ of the optimum after $O(\kappa \log \frac{f(x^{(0)})-f(x^*)}{\epsilon})$ steps, where $\kappa = \frac{\beta}{\alpha}$. Decreasing the value of α increases the value of κ, which means that the algorithm will converge on the optimum more quickly.

(d) [T/F] Suppose we would like to identify a local minima for a function f using the gradient descent algorithm. Picking the step size to equal the inverse Hessian will get to the local minima in a single step.

Solution: False. This is only true if the function is quadratic and can be written as just the sum of the first three terms in the taylor approximation.

Problem-2: Distributed Coloring [20 points]

Consider an undirected graph $G = (V, E)$ in which every vertex has degree at most Δ. Define a new graph $G' = (V', E')$, the ***Cartesian product*** of G with a clique of size $\Delta + 1$.

Specifically, V' is the set of pairs (v, i) for all vertices $v \in V$ and integers i with $0 \leq i \leq \Delta$, and E' consists of two types of edges:

(a) For each edge $\{u, v\} \in E$, there is an edge between (u, i) and (v, i) in E', for all $0 \leq i \leq \Delta$. (Thus, each index i forms a copy of G.)

(b) For each vertex $v \in V$, there is an edge between (v, i) and (v, j) in E', for all $i \neq j$ with $0 \leq i, j \leq \Delta$. (Thus each v forms a $(\Delta + 1)$-clique.)

Here is an example of this transformation with $\Delta = 3$:

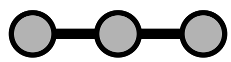

Figure 1: Graph G.

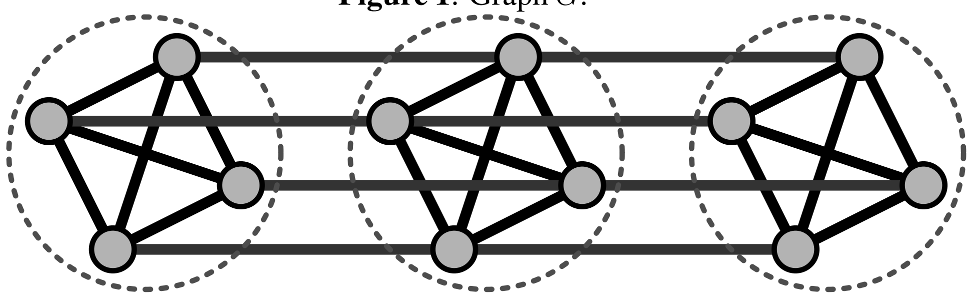

Figure 2: The Cartesian product G' of G and a clique of size 4.

(a) Let S be any *maximal* independent set of G' (i.e., adding any other vertex to S would violate independence). Prove that, for each vertex $v \in V$, S contains exactly one of the $\Delta + 1$ vertices in V' of the form (v, i).

Hint: Use the Pigeonhole Principle.

Solution: It cannot contain more than one, since all of these are connected in G' and that would violate independence.

Now suppose for contradiction that, for some particular u, S contains no vertices of the form (u, i). Then by maximality, every vertex of the form (u, i) must have some G'-neighbor in S. Since that neighbor is not of the form $(u, *)$, it must be of the form (v, i), for some v with $(u, v) \in E$.

Thus, each of the $\Delta + 1$ vertices of the form (u, i) has some neighbor of the form (v, i) in S, where $(u, v) \in E$. Since u has at most Δ neighbors in G, by the Pigeonhole Principle, there must be two different values of i, say i_1 and i_2, for which there is a single v such that (u, i_i) is a G'-neighbor of (v, i_1),

(u, i_2) is a G'-neighbor of (v, i_2), and both (v, i_1) and (v, i_2) are in S. That is a contradiction because S can contain at most one vertex of the form $(v, *)$.

(b) Now consider a synchronous network of processes based on the graph G, where every vertex knows an upper bound Δ on the degree. Give a distributed algorithm to find a vertex $(\Delta + 1)$-coloring of G, i.e., a mapping from vertices in V to colors in $\{0, 1, \ldots, \Delta\}$ such that adjacent vertices have distinct colors. The process associated with each vertex should output its color. Argue correctness.

Hint: Combine part (a) with Luby's algorithm.

Solution: The "colors" will be chosen from $\{0, 1, \ldots, \Delta\}$.

The nodes of G simulate an MIS algorithm for G'. Specifically, the node associated with vertex u of G simulates the $\Delta + 1$ nodes associated with vertices of the form (u, i) of G'. The algorithm produces an MIS S for G', where each node of G learns which of its simulated nodes correspond to vertices in S.

By Part (a), for each vertex u of G, there is a unique color i such that $(u, i) \in S$; the node associated with u chooses this color i.

Obviously, this strategy uses at most $\Delta + 1$ colors.

To see that no two neighbors in G are colored with the same color, suppose for contradiction that neighbors u and v are colored with the same color, say i. That means that both (u, i) and (v, i) are in S. But (u, i) and (v, i) are neighbors in G', contradicting the independence property for S.

An alternative solution that many students wrote involved executing $\Delta + 1$ instances of Luby's MIS directly on G, in succession. In each instance i, the winners are colored with color i. Then we remove just the winners before executing the next instance. This works, but leaves out some details w.r.t. synchronizing the starts of the successive instances. Also, its performance is quite a bit worse than the recommended solution above.

(c) Analyze the expected time and communication costs for solving the coloring problem in this way, including the cost of Luby's algorithm.

Solution: The costs are just those of solving MIS on G'; the final decisions are local and don't require any extra rounds.

Time (number of rounds): The expected time to solve MIS on G' is $O(\lg(n \cdot \Delta))$, because the number of nodes in G' is $n \cdot (\Delta + 1)$. The $O(\lg(n \cdot \Delta))$ bound can be simplified to $O(\lg n)$.

Communication (number of messages): The expected number of messages is $O(E \lg n)$, corresponding to $O(\lg n)$ rounds and messages on all edges (in both directions) at each round.