CSL 356: Analysis and Design of Algorithms **Date:** August 15, 2011
Instructor: Ragesh Jaiswal

3.1 Greedy Algorithms

1. There is a currency system that has coins of value $1, c, c^2, c^3, ..., c^k$ for some integers $c, k > 1$. You have to pay a person V units of money using this currency. Your goal is to minimize the total number of coins. Can you think of a greedy algorithm that minimizes the total number of coins. Prove the optimality of your algorithm.

3.2 Greedy Approximation Algorithms

Approximation factor : In the lectures, we looked at some *optimization problems* where we had to optimize (i.e., minimize or maximize) certain objective function while satisfying some constraints. We looked at how in certain cases a greedy algorithm, even though it did not give an optimal solution, gave a solution that was provably close to the optimal solution. The closeness was in the following sense: Suppose we are talking about an optimization problem where the goal is to maximize some objective while satisfying some constraints. If OPT is the cost of the optimal solution and G is the cost of the solution returned by some greedy algorithm, and $G \leq \alpha \cdot OPT$, then α is called the approximation factor of the greedy algorithm. Such a greedy algorithm is called an α-approximation algorithm for the problem. We looked at a $(\ln n)$-approximation algorithm for the Set-Cover problem and a 2-approximation algorithm for the Minimum-makespan problem.

Tightness of approximation : Consider a maximization problem P and let A be a greedy algorithm for the problem. Suppose through some clever thinking , we were able to show that if G is the cost of the solution returned by the greedy algorithm and OPT be the optimal value, then $G \leq \alpha \cdot OPT$. Does that mean that A cannot be a β-approximation algorithm for some $\beta < \alpha$? The answer is "no". This is simply because we may not have analyzed our greedy algorithm in the best possible manner. May be we were not clever enough while analyzing our algorithm. We say that the approximation factor α is "tight" if there is an example where $G = \alpha \cdot OPT$. Now, if we show a tightness example for our greedy algorithm, then we can say that A cannot be a β-approximation algorithm for $\beta < \alpha$.

1. Consider the Minimum-makespan problem with m machines and n jobs. Recall that we looked at a greedy algorithm and proved that it gave a 2-approximation. Try showing that the 2- approximation factor is tight? What goes wrong?

   If you are repeatedly failing in showing the tightness, then maybe the algorithm gives you a better approximation factor. In the next problem we will show that this is indeed true.

2. Show that our greedy algorithm actually gives a $(2 - 1/m)$-approximation. Show that $(2 - 1/m)$-approximation is tight.