CSC2420 - Algorithm Design, Analysis and Theory **Fall 2022**

Lecture 4: Priority Algorithms

Lecturer: Allan Borodin *Scribe: Kevin Gao*

Recall the template for the fixed order priority algorithm template:

$\mathcal{J}$ = set of all possible inputs
$\preceq$ = a total ordering on $\mathcal{J}$ (typically induced by a function f)
$\mathcal{I} \subset \mathcal{J}$ = actual input to the algorithm
$S = \emptyset$ // items already examined by the algorithm
$i = 0$
while $\mathcal{I} - S \neq \emptyset$
    $i = i + 1$
    $\mathcal{I} = \mathcal{I} - S$
    $I_i = \min_{\preceq}\{I \in \mathcal{I}\}$ // select min element based on the ordering $\preceq$
    make an irrevocable decision D_i concerning I_i
    $S = S \cup \{I_i\}$

and the adaptive priority algorithm:

$\mathcal{J}$ = set of all possible inputs
$\mathcal{I} \subset \mathcal{J}$ = actual input to the algorithm
$S = \emptyset$ // items already examined by the algorithm
$i = 0$
while $\mathcal{I} - S \neq \emptyset$
    $i = i + 1$
    $\preceq_i$ = a total ordering on $\mathcal{J}$ (typically induced by a function f_i)
    $\mathcal{I} = \mathcal{I} - S$
    $I_i = \min_{\preceq_i}\{I \in \mathcal{I}\}$ // select min element based on the ordering $\preceq_i$
    make an irrevocable decision D_i concerning I_i
    $S = S \cup \{I_i\}$
    $\mathcal{J} = \mathcal{J} - \{I \in \mathcal{I} \mid I \preceq_i I_i\}$

4.1 Inapproximations for Deterministic Priority Algorithms

Once we have a precise model, we can then argue that certain approximation bounds are impossible within this model. We first consider the weighted interval selection problem.

For the interval selection problem, we have a set of intervals and each interval has a weight w_j. No priority algorithm can achieve a constant approximation. In an undergraduate course, we have shown that the unweighted interval selection problem can be optimally solved using a greedy algorithm by ordering the inputs by the earliest finishing time. We will prove a weaker result.

Theorem 4.1. *There is no priority approximation algorithm can achieve an approximation ratio better than 3.*

Proof: We use a charging argument. ∎

4.2 Set Cover

In the set cover problem, we are given a collection $\mathcal{S} = \{S_1, \ldots, S_n\}$ of sets with $S_i \subseteq U$ for some universe U. In the weighted set cover problem, each set S_i has a cost or weight $c(S_i)$. The objective is to find a minimum cost subcollection $\mathcal{S}'$ such that $\bigcup_{S \in \mathcal{S}} S = U$.

For the set cover problem, the "natural adaptive greedy algorithm" is essentially the best priority algorithm.

Greedy-Set-Cover($\mathcal{S}$)

$\mathcal{S}' = \emptyset$
while U is uncovered
    $j = \arg\min_i \{w(S_i)/|S_i \cap U|\}$
    $\mathcal{S}' = \mathcal{S}' \cup \{S_j\}$
    $U = U \setminus \{S_j\}$