Online Algorithms, an introduction and important Results

Outline

- Mathematical preliminaries and Introduction
- Important Results
- Applications to Databases
- Applications to Compilers

Motivation

- Traditional Design:
  - Algorithm has complete knowledge of the entire input
- Often an unrealistic assumption in practice
- Relevant data will arrive in the future
- Need a new paradigm

Examples

- Resource Management in Operating systems
  - Paging: To maintain actively reference pages in fast memory not knowing page requests in future
- Data Structures
  - Need to dynamically maintain data structures like trees, link lists for cheap access cost
- Scheduling:
  - Scheduling jobs on P processors without knowing future jobs.
  - Optimize objective function

Formally…

- Presented with a *request sequence* $\sigma = \sigma(1), \sigma(2), \sigma(3) \ldots \sigma(m)$.
- Request $\sigma(t)$, $1 \leq t \leq m$ must be served in order of occurrence.
- When serving $\sigma(t)$, Algorithm does not know any $\sigma(t`) \geq t$.
- Serving each request has cost
- Goal : Minimize total cost paid on entire sequence.

Online Algorithm in Paging

- $\sigma(t)$, $1 \leq t \leq m$ is a request for page from small fast memory
- If page not present, *page fault* occurs, and page must be moved from fast to slow memory
- Online algorithm specifies which page to request not knowing any $\sigma(t`) \geq t$
- Goal: Minimize number of page faults

Competitive Analysis

- Sleator and Tarjan:
  - Evaluate the performance of an online algorithm using *competitive analysis.*
- A deterministic online Algorithm, A, is compared to an *optimal offline Algorithm Aopt (Adversary)*
- *Aopt* knows all $\sigma(t),\ 1 \le t \le m.$
- Let $A(\sigma)$ and $OPT(\sigma)$ be costs
- A is *c-competitive* if there exists a constant *a such that*
  - $A(\sigma) \le c \bullet OPT(\sigma)$ + a for all σ

Randomization

- What if we use randomization?
- Competitive ratio of Randomized algorithm is defined with respect to an adversary.
- Three kinds of adversaries
  - Oblivious Adversary
  - Adaptive online Adversary
  - Adaptive offline Adversary

Self organizing data structures

- One of the first online problems studied.
- Three deterministic algorithms known
  - Move-to-Front
  - Transpose
  - Frequency-Count