Chapter 3

COMPETITIVE ANALYSIS - CONCEPTS AND SOLUTION TECHNIQUES

In this chapter, we address the fundamental concepts of Competitive Analysis. Competitive analysis serves as a useful analytical tool in quantifying the performance of on-line algorithms. In a way, for an algorithmic designer, this tool aids to fine-tune his/her design to get the solution as closer as possible to an optimal solution. Problems that can be considered for analysis using this analytical tool include K-Server Problem, Paging Problem, Distributed Object Management, Web-Caching, Load Balancing in computer networks, Disk Header Scheduling, to quote a few. We first give formal definitions of competitiveness for both deterministic on-line algorithms and randomized on-line algorithms. We then enumerate several techniques in competitive analysis such as Potential Function method, Phase Partition Technique, and so forth. Several illustrative examples are provided for the ease of understanding to elicit important competitive properties of an on-line algorithm.

1. Competitiveness

1.1 On-line Problems

Suppose you are a share-holder of a company. As we know, depending on the response from the consumers, the prices of the stocks may fluctuate (magnitudes are unpredictable). Thus, when the prices rise you may attempt to sell some or all of your shares to profit. Further, you may also wait for further rise in price of the stocks. On the other hand, when the prices fall, you may attempt to buy some more shares in the view of minimizing the average price of each share or simply sell

1Refer to Chapter 2 on a brief discussion on some of these topics.

the existing shares to minimize your net loss. Thus, while stocks prices are fluctuating quantities, depending on the current pricing values of the shares, you may need to decide whether or not to buy or sell the shares. In such a case, you are confronted with an on-line problem.

Over the past two decades, on-line problems are very common in computer science and engineering and have received considerable research interest. To a system designer or a computer scientist, it is imperative to understand on the type of systems that deserve on-line solutions. Predominantly, systems that are expected to service/process a currently submitted request without any knowledge of the future requests to be submitted to the system qualify to look for on-line solutions. Thus, on-line problems are completely unaware of behavioral pattern of the incoming requests. The response of on-line algorithms designed to handle such on-line requests would only consider historical data for making any decisions. A completely complementary approach is to design an off-line algorithm, which is designed on the basis of a complete knowledge of all the current and the future requests to be processed, such as a VOR system discussed in Chapter 2. However, this attempt is often unrealistic in real-life applications. Many of the algorithms designed to tackle real-life problems are of on-line in nature. In these problems, input information is only partially available because some relevant inputs can only arrive in the future and may not be accessible at present. Therefore, an on-line algorithm must generate an output without the knowledge of the entire input. Some common on-line problems that are applicable in a wide variety of real-life situations are as follows.

- **K-Server Problem**
- **Paging Problem**
- **Distributed Object Management**
- **Disk Header Scheduling**

These problems clearly provide the motivation on why competitive analysis serves as a natural choice in the design and analysis of on-line algorithms.

**K-Server Problem:**

The K-Server problem, introduced by Manasse et al. [MMS88], may be defined as follows. We are given a metric space \( \mathcal{M} \), which is a set of points with a non-negative symmetric distance function \( d \) that satisfies the triangle inequality, i.e., for distinct points \( x, y, \) and \( z \) we