IT584 Approximation Algorithms January 2, 2023

Lecture 1

Lecturer: Rachit Chhaya Scribe/s: Meet Kothari (202311030)

1 Introduction

We discussed a fundamental overview of approximation methods in this lecture. We begin by defining approximation algorithms and outlining the necessity for their study . Then, using a classic optimization problem as an example, we define some algorithms addressing the problem.

Decision Problem: A computational problem known as a decision problem maybe expressed as a yes or no question of the input values.

Optimization Problem: A computational problem is known as an optimization problem in which the object is to find the b est of all possible solutions.

Polynomial Time Reduction: Make A and B two decision-making problems. Let's say the initial algorithm solves B. In other words, depending on the input for problem B, the first algorithm will provide a yes or no response. Finding a transformation from A to B so that the first algorithm (that solves B) ca

$$A \leq B \; ; \forall \; A \; in \; N \; P$$

1.1 Polynomial Time Problems Class

The complexity class P contains the set of problems that can be resolved in polynomial time. This describes the class of problems that can be effectively solved through a theoretical point of view.

1.2 Non-deterministic Polynomial Time Problems Class

The class of problems that can be solved in non-deterministic polynomial time, or additionally , problems for which a solution can be verified in polynomial time, defines the class NP .

$$P \subseteq N \; P$$

1.2.1 NP-Complete problem set

Problem B is called NP-complete if and only if
• problem B is complete in Non-deterministic polynomial time,
• Every problem in NP is reducible to problem B in polynomial time.
F or every problem A in NP;

$$A \leq B \; ; \forall \; A \; in \; N \; P$$

1.2.2 NP-Hard Problem set

Problem B is called NP-hard if
• Every problem in NP is reducible to problem B in polynomial time. As an outcome, the NP-Hard set contains the NP-Complete set as a subset.

$$N \; P - \; Complete \subseteq N \; P - \; H \; ar \; d$$

2 Approximation (Why?)

Let us say we are given an optimization problem to solve. The ideal algorithm would b e one that always finds an optimal solution. Ho w ev er, implementing such an algorithm is unrealistic or impossible for many problems. NP-hard problems are a classic example of this t yp e of problem since they can never be solved effectively , despite the exceptional case where P = N P . It is also possible that we need to be given more details at the beginning of the proble m to handle it effectively .

2.1 α -Approximation Algorithm

Approximation Ratio (α): The approximation ratio (factor) of an algorithm is the ratio between the solution obtain e d b y the algorithm and the optimal solution.

α - *Approximation for maximization problem:*
An algorithm is considered approximation for a maximization problem if, for all instances,

$$S\ O\ L \geq \alpha \cdot O\ P\ T\ (\alpha < 1)$$

where SOL and OPT stand f or the costs of the solution and the b est solution, respectively .

α - *Approximation for minimization problem:*
An algorithm is considered approximation for a minimization problem if, for all instances,

$$S\ O\ L \leq \alpha \cdot O\ P\ T\ (\alpha > 1)$$