A Problem in Optimization

Summer Project

Approximation Ratio

- The large time complexity of prior algorithms makes obtaining the maximum possible reward difficult.
- It may be necessary to sacrifice attaining the maximum reward for reducing the time complexity
- Algorithms that work in polynomial time can be developed to obtain sequences that don't achieve the maximum possible reward, but come close.
- The performance of such an algorithm is measured using the Approximation Ratio:

  $$\text{Approximation Ratio} = \frac{\text{Reward given by algorithm}}{\text{Maximum Reward}}$$

Feasibility Algorithm: Introduction

This is an approximation algorithm that uses the divide and conquer approach.

Notion of Feasibility:

- In a given function, if a large reward occurs at a small value of tau, we would want to obtain through our sequence.
- If a large reward occurs at a large value of tau, the reward may not be worth the delay in reaching the value. We would be sacrificing a large number of time slots in which we cannot play this function. We may be better off aiming to get many smaller rewards.
- Thus, there is a trade-off between the reward obtained and the Delay in getting it.
- Define a parameter called feasibility to represent this trade-off:

  $$\text{Feasibility} = \frac{\text{Potential Reward}}{\text{Delay}}$$

Feasibility Algorithm: Strategy

We have K functions and we need to find a sequence of length T.
Each of the functions needs to be defined upto $\tau = T - 1$.
(If they are not, it suffices to assume zero reward for the values of τ for which it is not defined)
Thus, we have KT ordered triplets of the form: (function,τ,reward)

Strategy:

- The basic strategy of the algorithm is to attain as many entries with the highest feasibility values as possible.

Notion of state:

- Define *state* to be a K-dimensional vector containing the current τ values of each of the K functions.

Feasibility Algorithm: Procedure

- Let S be the sequence of length T which needs to be filled. Let Φ represent the initial state, and Λ represent the set of functions which can be used to fill the sequence.
- T,Φ and Λ need to be given as inputs to fill the sequence.
- Find the feasibility value of each reward that can be potentially obtained in this sequence. The feasibility values depend upon the initial state Φ.
- This creates a list of length of KT, which we call the *feasibility list* of this sequence. For convenience, arrange the entries in descending order of their feasibility.

Feasibility Algorithm: Procedure(cont.)

- Let the entry with the highest feasibility value in the list correspond to the function f_i. Let d_a represent the associated delay value.
- Play f_i in the $(d_a + 1)^{th}$ time slot. Call this the pivot element.
- There are d_a empty slots in between which need to be filled by using the all functions in Λ except f_i. Call this S_1.
- The empty slots from the $(\tau_a + 2)^{th}$ to the T^{th} time slot can be filled with using all functions in Λ. Call this S_2.
- S_1 and S_2 are filled recursively using the same procedure, by providing the appropriate inputs.
- S $= S_1 \cup f_i \cup S_2$
- Return the sequence S, the reward obtained and the final state after applying the sequence.