. 0pt 0pt

AM5801: Computational Lab

Assignment 5

Date: September 11, 2025

Deadline: September 17, 2025 Max mark: 50

1. Design and implement a randomized load balancing algorithm for a large-scale cloud data center consisting of n servers and m incoming computational tasks. Each task has a unique ID, a computational weight randomly generated between 1 and 1000 CPU units, and a priority level categorized as High, Medium, or Low. The data center has a fixed total CPU capacity, and the goal of the algorithm is to assign tasks to servers such that the load is balanced as evenly as possible, the number of overloaded servers is minimized, and the overall system throughput is maximized.

   The algorithm should follow a randomized strategy where, for each incoming task, two servers are selected uniformly at random and the task is assigned to the server with the lesser current load. You should implement the algorithm efficiently, using appropriate data structures to track server loads and avoid recomputation, while ensuring that it can handle large-scale inputs, where n can be up to 10^4 and m up to 10^7.

   Your implementation should produce clear outputs showing the following key performance metrics: the maximum server load, the average load per server, the standard deviation of the server load distribution, a histogram of the load distribution, and the number of servers whose load exceeds a threshold (e.g., 1.5 times the average expected load). The task dataset should be generated randomly with a balanced mix of High, Medium, and Low priority tasks. **(25)**

2. An urban traffic control authority is tasked with minimizing the total average vehicle delay at a busy network of traffic intersections in a city. Each intersection has multiple signal phases, and the green light duration for each phase directly affects the average vehicle delay at that intersection. The total delay depends nonlinearly on the green times across all intersections due to complex traffic interactions and time-dependent vehicle arrivals. Analytical optimization is infeasible due to the nonlinear nature of the delay function and the high dimensionality of the problem. Therefore, the authority decides to model the system as an optimization problem where the objective is to minimize the total average delay by adjusting the green time durations of all signal phases using an iterative Gradient Descent algorithm. Your task is to implement a Gradient Descent-based optimization algorithm where the

decision variables are the green times of all signal phases across intersections. The total average vehicle delay is provided by a nonlinear, differentiable cost function (given as a black-box function or a formula).

The program should start from an initial feasible green time configuration and iteratively update the green times by computing the gradient of the delay function with respect to each green time, moving in the direction of steepest descent. The implementation must allow specifying the learning rate and number of iterations, and output the optimized green time configuration along with the total delay before and after optimization. Include a convergence graph showing how the total delay decreases over iterations. Explain why Gradient Descent is appropriate for this problem, especially considering the difficulty of deriving analytical solutions and the high dimensionality of the variable space. Also, discuss how the learning rate affects convergence speed and stability. **(25)**