CMSC 27200 Project Presentation

[noname]

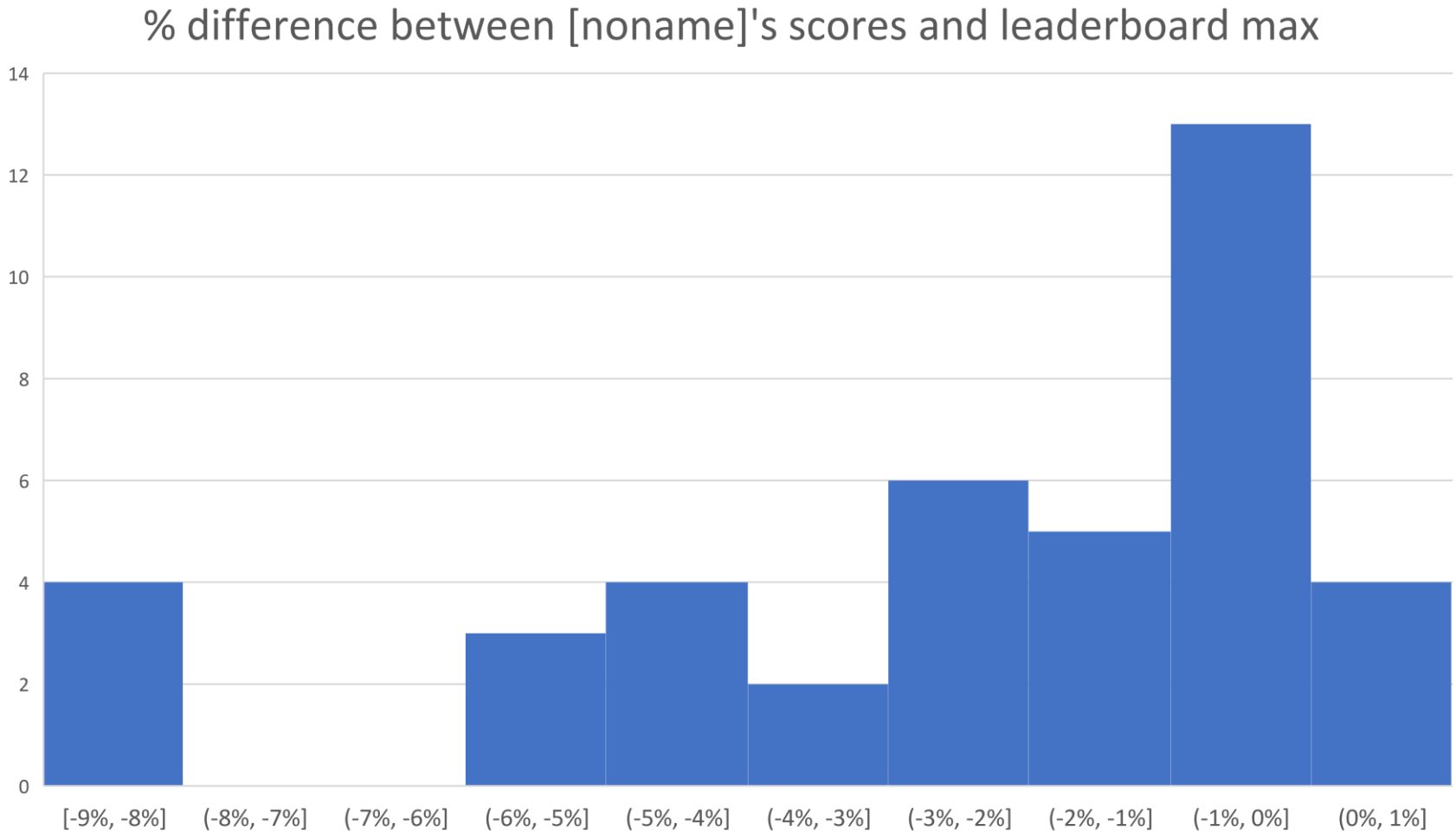
What your code does

- Greedy + 2 forms of brute force: Greedier(n, α, β, m)
  - Part 1: Feasible brute force for the first n nodes
  - Part 2: Feasible greedy, select top m nodes each time
    - Heuristic is based on utility/time
      - We define time $T_{i,j,t} = (1+\alpha)d_{i,j} + w_{j,t} + t_j + \beta d_{j,O}$.
      - Heuristic: Given that Jonathan is at point i at time t, the score for choosing node j as the next feasible node is: $S(i,j,t) = \alpha\frac{u_j}{T_{i,j,t}} + \frac{1}{|H|}\sum_{h\in H}\frac{u_h}{T_{j,h,(t+T_{i,j,t})}}$, where H is the set of all feasible nodes given that Jonathan is at node j at time $t + T_{i,j,t}$, and has already visited a certain sequence of nodes.

How well you think it works

- Greedier(n, α, β, m)
- n: Good with caveats
  - Worked for inputs with lots of infeasible nodes. Could easily do n = 1, 2, 3, 4.
  - Way too slow for inputs with few start/end time restrictions, especially large inputs. Could only do n = 1 or 2 max. Partial results were bad as well.
  - Greedy(n) alone beat the baseline.
- α, β: Marginal improvement, not that significant.
  - Didn't have time to try out many different combinations anyway
- m: Good in general
  - Worked much better for the inputs that couldn't run under large n.
  - The final improvement that moved us up the leaderboards to beat admin's score

How well you think it works



Why you settled on the approach you did

- Pull factors:
  - Relatively simple
  - Able to quickly modify heuristic / algorithm in response to challenging inputs
  - Able to get better results with time, via brute force
- Push factors:
  - Genetic algorithm failed
  - No idea how to do DP or LP

What you would do if you had more time

- Focus on cutting the time required for Greedier(n, α, β, m) :
  - Actual implementation had n at least 1, but it should be 0. On hindsight, implementing m was more than sufficient to mitigate the benefits of n.
  - New parameter k: Only get top m nodes in greedy for up to the first k nodes in any sequence
  - Reduce inefficiencies in the code to speed up runtime, e.g. if β = 0, then the extra calculation should be skipped, not executed then thrown away.
- Try out many more values of α, β.