Project Idea for the COS 126/226/217 Lab Queue Staffing

Grace Guan (`gg17`)

Every semester, over 500 students enroll in Princeton's introductory computer science classes COS 126 (Computer Science: An Interdisciplinary Approach), COS 226 (Algorithms and Data Structures), and COS 217 (Introduction to Programming Systems). To assist students currently in the courses with debugging code, the COS department hires Lab TAs, who are undergraduate students who have successfully completed these three courses. To manage demand for these Lab TAs, students place their name on a Lab Queue, and Lab TAs address student requests in a first-come-first-served fashion. The Lab Queue is currently hosted at `https://www.labqueue.io/`*. The queue faces high demand at certain peak hours, specifically the day before major assignments (e.g. N-Body, Othello) are due, and the current wait time predictions of the queue are not accurate. The lack of predictability in wait times of the Lab Queue both drives students away from and lowers ratings of COS classes at Princeton.*

I suggest a potential project that could be undertaken as a COS IW project or an independent paid project that would tackle the issue of wait times from a different perspective: that of allocating staffing to adjust for student demand ahead of time.

1 Queuing Theory Project

Prerequisites: Understanding of probability theory and stochastic systems at the level of ORF 309 or COS 445 (Markov chains; Poisson and Exponential distributions). Willingness to learn stochastic (discrete-time) simulations and modeling.

Goal: 1. Determine how many TAs are needed to keep wait times under x minutes or the queue under length n. 2. Build a tool to inform the Head Lab TA how many TAs will be needed on a given day and hour to achieve aforementioned objectives.

Details: We model the Lab Queue as an $M/M/c$ Queue.[1] The c of $M/M/c$ indicates how many staff are currently servicing the queue, and the M/M means that we have exponentially distributed

[1] `https://en.wikipedia.org/wiki/M/M/c_queue`

inter-arrival and service times. Thus, what $M/M/C$ means concretely is: we construct a Markov Chain with states $\{0, 1, 2, ...\}$ with each state representing how many students are waiting in the queue. A student enters the queue according to a Poisson process with rate λ, and each arrival moves the process from state i to $i + 1$. The time differences between arrivals (inter-arrival times) thus correspond to an exponential distribution with mean $1/\lambda$. Service times follow an exponential distribution with mean $1/\mu$ (and thus mean service rate is μ), and each service moves the process from state i to $i - 1$ if $i \geq 1$. The proportion of the time the lab queue has more students queued than TAs is $\lambda/c\mu$.

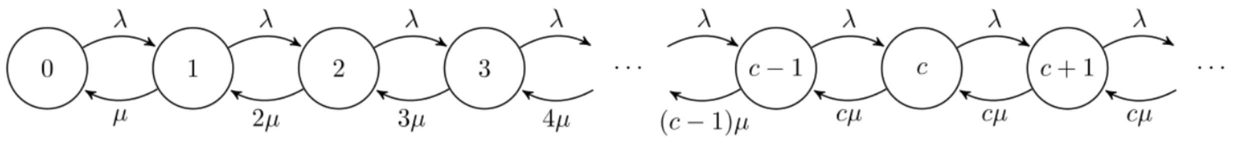

Figure 1: State space diagram for the continuous Markov chain created by a $M/M/c$ Queue. Note this is not a discrete Markov chain which was taught in ORF 309. Source: M/M/c Queue, Wikipedia

Stationary analysis can be performed on this Markov chain to find a closed formula for expected wait times and the probability that a student arrives and is able to see a TA immediately (long-run probability we are in states $\leq c-1$) in terms of μ, λ, and c. Then, given four semesters of historical data of the Lab Queue arrival and service times, we can extract estimations of λ and μ from the data to solve for the necessary c to keep wait times under x minutes or the queue length under n. (The Erlang C formula represents the probability that a student arriving to the Lab Queue will have to queue, but it's good to derive this formula by hand to understand it).

This project makes the assumption that inter-service and inter-arrival times are exponentially distributed, but this may not be the case in the data. However, there will most likely exist features in the data that influence the service and arrival rate parameters such that this model can be imposed to estimate how many staff are needed. For example, stratifying data by historical date in semester (i.e. Saturday before N-Body is due) as well as hour within the day should give adequate estimates.

Additional Resources: There is code for a $M/M/C/C$ queue (a slight variation of this $M/M/C = M/M/C/\infty$ queue where we can have only C jobs waiting and in service) in the GitHub repository `guanzgrace/edX-queueing-theory` in the file `Week5_Lab.ipynb`[2], and code for

[2]https://github.com/guanzgrace/edX-queueing-theory/blob/master/Week5_Lab/Week5_Lab.ipynb

a $M/M/C$ queue in the file Week5_Lab_part2.ipynb[3].

To visualize and download wait time data for the Lab Queue, see the GitHub repository guanzgrace/princeton-labqueue-projects[4]. The file data_visualization.ipynb gives an example of reading and plotting the data. You can use the data files labqueue_anon.csv (has no outliers removed) and labqueue_anon_1.csv (has some outliers removed). The dataset documentation is in Projects/This_Project.pdf.

2 Acknowledgements

The author thanks Dr. Jeremie Lumbroso for supporting these high-impact, methodologically-interesting projects and Professor Matt Weinberg for offering to supervise students doing these projects.

[3]https://github.com/guanzgrace/edX-queueing-theory/blob/master/Week5_Lab_part2.ipynb

[4]https://github.com/guanzgrace/princeton-labqueue-projects/