Adaptive Nested Algorithms for Balanced Scheduling

Stetson Bost
Department of Mathematics
Harvey Mudd College
Claremont, CA 91711
sbost@hmc.edu

Abstract

Scheduling, adaptive algorithm, nested algorithm, machine learning, big data

1 Scheduling Problem

Scheduling is a very large component of modern life. In a fast-paced world, it can be important to effectively plan one's schedule in a way that is conducive to productivity while prioritizing the importance of a healthy balance between different types of activities.

We aimed to develop algorithm that create balanced schedules that can be adapted to person to prioritize his or her needs.

2 Background and Motivation

The primary motivation for pursuing this work was the perpetually active and hectic environment of a college campus, primarily with Harvey Mudd College in mind. Students are faced with many academic obligations, from doing homework to attending class to studying for exams. While focusing on academics is important and necessary, it is also valuable to have a healthy balance of different activities to support a student's personal wellbeing. There are many ways for students to arrange their schedules, but rarely does a "one size fits all" approach work for creating students' schedules. Each person has a unique style of working and individual needs, so it makes sense that a person's schedule should be tailored to fit the person.

Additionally, schedules are inherently dynamic. Events can be scheduled and canceled. One may occasionally have to deal with an emergency that preclude all other scheduled activities. Priorities can change.

3 General Approach

For the remainder of this paper, we will use the terms *events* and *tasks* interchangeably to refer to an activity of some duration that should be added to a schedule. Their roles in the scheduling algorithms will be identical, but their use will depend on how they are used in typical English.

To address scheduling in a way that took into account balance and personal preferences, we tried to mimic human intuition when possible. Humans seem to intuitively prioritize tasks, but the ways that they do so can vary between different contexts and from person to person. Therefore, we wanted to create algorithms that were *adaptive*, in two senses of the word. First, we wanted something that could take in a stream of tasks to schedule and *adapt* according to when tasks are introduced. Second, we needed an algorithm that could *adapt* to an individual, taking into account the person's preferences and way of working.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

We decided to use *nested* algorithms that assigned events to a schedule in stages, where earlier stages schedule events more generally (i.e. the week it will be scheduled), while the later stages arranged events more specifically (i.e. the exact time slot when the event should take place).

4 Algorithm for Personal Scheduling

5 Algorithms for Event Scheduling

5.1 Probabilistic Approach

Suppose we know the probabilities for all potential attendees that they will attend all possible events. This means we can know how likely they are to attend a new event if it is scheduled for a particular time slot. Let A be the list of attendees, t be a possible start time, $p_a(t)$ be the probability that an attendee a will attend the event if scheduled at time t, and $\mathbb{E}[t]$ be the expected value of attendees at time t. Notice that for any particular start time t, if we sum all potential attendees' probability of attending, we will get the expected value of attendees at t. For each potential attendee a, go through all possible start times t, add the probability $p_a(t)$ of a attending at t to the value in the index corresponding to t. That is,

$$\mathbb{E}[t] = \sum_{a \in A} p_a(t).$$

If there are m possible start times, then we can create a $1 \times m$ matrix $\mathbb{E}$ of the expected number of attendees at each time. This matrix is calculated as

$$\begin{bmatrix} \sum_{a \in A} p_a(t_1) & \sum_{a \in A} p_a(t_2) & \cdots & \sum_{a \in A} p_a(t_n) \end{bmatrix}$$

for all $1 \leq i \leq m$.

6 Data

We have not yet used a real data set on which to test our algorithms. Historic data could be collected from the Harvey Mudd community through a combination of techniques such as online surveys and in-person interviews.

Acknowledgments

References