



Adaptive Nested Algorithms for Balanced Scheduling

Stetson Bost



Introduction

Scheduling is very important for staying organized and managing stress for both individuals and larger organizations. With this project, we wanted to develop algorithms capable of creating effective, adaptable, and balanced schedules.

Algorithm for Personal Scheduling

Our algorithm for creating personal schedules aims to accomplish all necessary tasks while maintaining a balanced schedule. A schedule is a collection of relatively short *time slots*, and contiguous time slots can form *blocks*. A group of consecutive blocks can form a *section*, identified by it starting and ending blocks. The algorithm has three nested steps.

1. Assign tasks to sections.
2. Assign tasks to blocks within sections.
3. Assign tasks to time slots within blocks.

Each step takes into account various aspects of the tasks, including deadlines, estimated duration, and intensity. Additionally, each block has a minimum permissible amount of time reserved for non-work activities, which may or may not be structured, depending on user preference. By maintaining near equal intensity levels among blocks, the algorithm prioritizes a balanced schedule.

Task	Due Date	Duration	Intensity
Task 1	Monday	6 hours	0.3
Task 2	Friday	3 hours	0.6
Task 3	Thursday	4 hours	0.5
...

(a) Start with a list of tasks that need to be completed.

Section: Tuesday 8 AM - Midnight			
Block 1	8 AM - 12 noon	Non-Work Time	4 hours
Block 2	12 noon - 6 PM	Task V	2 hours
Block 3	6 PM - 11 PM	Task W	1 hour
		Task X	2 hours
		Task Y	5 hour
		Task Z	2 hour

(b) Step 1: Create a section made up of time blocks, and assign tasks to the section. Non-work time is treated as a task.

Block 2: Tuesday 12 noon - 6 PM	
Non-Work Time	2 hours
Task X	2 hours
Task Y	1 hour
Task Z	1 hour

(c) Step 2: Assign tasks to each block in the section.

Task Schedule: Tuesday	
12 noon	...
1 PM	Non-Work Time
2 PM	Task Y
3 PM	Task X
4 PM	Task Z
5 PM	Non-Work Time
6 PM	...

(d) Step 3: Schedule tasks within the block.

Figure 1: Example for personal scheduling algorithm.

For Further Information

For more information about this project, please contact sbost@hmc.edu or gu@hmc.edu.

Advisor: Weiqing Gu

Algorithm for Event Scheduling

Members of a group often have to schedule events where a number of people must attend. We wanted to create an algorithm capable of scheduling events with multiple attendees. An event can be either recurring (“regular”) or non-recurring (“special”), will have restrictions on available times, and may have a list of desired or required attendees.

Probabilistic Approach

One approach is to use historic data and a probabilistic version of the *k*-nearest neighbors algorithm to determine the likelihoods that all potential attendees will attend a specific event scheduled at potential times (Holmes). By summing these likelihoods across all attendees, we are able to get the expected value for the number of people at the event.

Nested Approach

An alternate approach makes use of the nested personal scheduling algorithm detailed to the left with some slight modifications. The basic structure of time slots, blocks, and sections is still the same, but it may be less likely to have adjacent blocks or contiguous sections, depending on the group’s scheduling practices. We treat events as tasks with possible time windows, and use those to create sections. In place of task intensity, we use a quantity that describes the priority of the event. We also create some conditions for balance, such as demands on the frequency of regular and special events.

Future Work

A good next step for this work would be to use code to implement the algorithms so that they can be tested on a group of people. To further validate the algorithms, it will be useful to test the algorithms on actual scheduling problems. Ideally this testing should be with people and groups that are willing to try the schedules produced by the algorithm. Additionally, for a group that uses a common calendar application, it might be possible to integrate the event scheduling algorithms with the calendar to automatically schedule events.

References

C. C. Holmes and N. M. Adams, 2002. *A probabilistic nearest neighbour method for statistical pattern recognition*. <http://hedibert.org/wp-content/uploads/2016/02/holmes-adams-2002.pdf>.

Acknowledgments

I would like to thank my advisor, Professor Weiqing Gu, for her guidance throughout this research. I am also very grateful for the kindness and generosity of the Rose Hills Foundation through their Science and Engineering Summer Undergraduate Research Fellowship, which has made this project possible.