Michigan Medicine Infectious Disease Department: Efficient Attending Scheduling

Honors Capstone Final Report

Submitted by:

Laura Hattar

Industrial & Operations Engineering, University of Michigan

1205 Beal Avenue

Ann Arbor, MI 48109

Faculty Advisor:

Professor Amy Cohn

Industrial & Operations Engineering, University of Michigan

1205 Beal Avenue

Ann Arbor, MI 48109

Date Submitted: April 30th, 2021

INTRODUCTION

Scheduling employees is crucial for many businesses and jobs. Efficient scheduling ensures that a business is able to operate to its standard or provide a service customers expect and depend on. It also allows employees to plan their day-to-day lives, but this can easily become cumbersome if an efficient scheduling process is not in place.

Many organizations construct their schedules by hand. This becomes taxing on the scheduler as they must accommodate all rules and requests to generate a feasible schedule. More often than not, if a mistake is found, the scheduling process must be restarted, becoming time-consuming for generating a final schedule.

This scenario becomes all too relevant in a medical field setting, specifically with attendings, who are trained physicians practicing in their specialties. Assigning shifts in a hospital setting can often be complicated. Attendings often want fair schedules that avoid requiring them to work long hours, during requested vacations dates, or back-to-back shifts that change hospital locations. Scheduling may be further complicated by needing to assign attendings to shifts they are qualified to work based on their experience. It often takes significant time to manually create a single draft schedule with these constraints in mind and only becomes more time consuming to adjust and redraft a given schedule. These factors all make scheduling for the Infectious Disease (ID) Department at Michigan Medicine highly challenging under normal circumstances. Additional complications due to the Covid-19 pandemic precipitated the ID leadership team to ask for assistance from our team at the Center for Healthcare

Engineering and Patient Safety (CHEPS) to automate and improve their scheduling processes.

The problem described above can be formulated as a linear programming model, incorporating not just the elements necessary to build a valid schedule but also the elements that make one possible schedule preferable to another. We translate this model into code which automatically generates candidate schedules. The CHEPS team has previously created a tool for a different scheduling project and the task at hand is to tweak the existing tool to meet the requirements for this specific project.

Therefore, the purpose of this project is to provide an automated system to generate the attending physician call schedule for the calendar year quickly and conveniently, even with complex constraints and rules. In addition, we aim to create a smart spreadsheet that will facilitate making small adjustments to the schedule over the course of the year without having to regenerate a completely new schedule.

The overall goal of this project is to provide the stated deliverables so that the quality of schedules and scheduling process for attendings is improved. This will save attendings time and stress, ultimately leading to better patient care. The project will also give the same potential benefits to institutions outside of Michigan Medicine as the tools will be accessible on a public form.

OUTLINE OF QUESTIONS/PROBLEMS ADDRESSED

The primary objective of the project is to provide an automated system to generate attending schedules quickly and conveniently. The research questions corresponding to this objective are as follows:

- How can we take an existing scheduling tool and modify it to match the needs of the Infectious Disease Department at Michigan Medicine?
- How can we make the scheduling tool intuitive and easy-to-use?
- How can we design a smart spreadsheet that allows minor swaps to be made to a generated schedule without altering it too extensively?
- How do we document the process of building the scheduling tool so that it can be easily altered/understandable by new students working on the project?

The baseline goal of this project is to improve the quality of schedules and the scheduling process for the attendings. This will save attendings time and stress which will in turn lead to better patient care. The stretch goal of this project is to improve patient care within the Infectious Disease Department at Michigan Medicine.

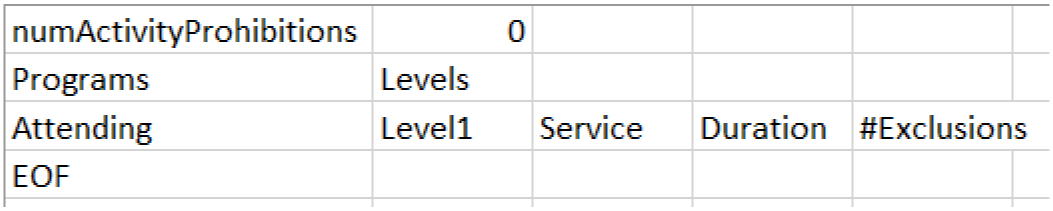
This project will be considered successful if an accurate schedule can be generated and customized with little alterations to the original schedule, so attendings are able to swap shifts if they desire.

METHODS

The major tasks of this project will be described below.

The first task will be to update an existing scheduling tool to match the requirements and needs of the Infectious Disease Department. The current tool can accommodate complicated constraints and rules to generate a schedule such as time off requests, educational requirements, spacing and sequencing rules, prohibited shift assignments, paring employees together for certain shifts, etc.. This existing functionality should encapsulate everything needed for the Infectious Disease Department, making the current tool a flexible baseline model.

To prepare the current tool for adjustments, the existing input files were emptied. An example of an empty file is shown below in Figure 1: there are column labels, but no data within the file.



numActivityProhibitions	0			
Programs	Levels			
Attending	Level1	Service	Duration	#Exclusions
EOF				

Figure 1: Empty Input Data File

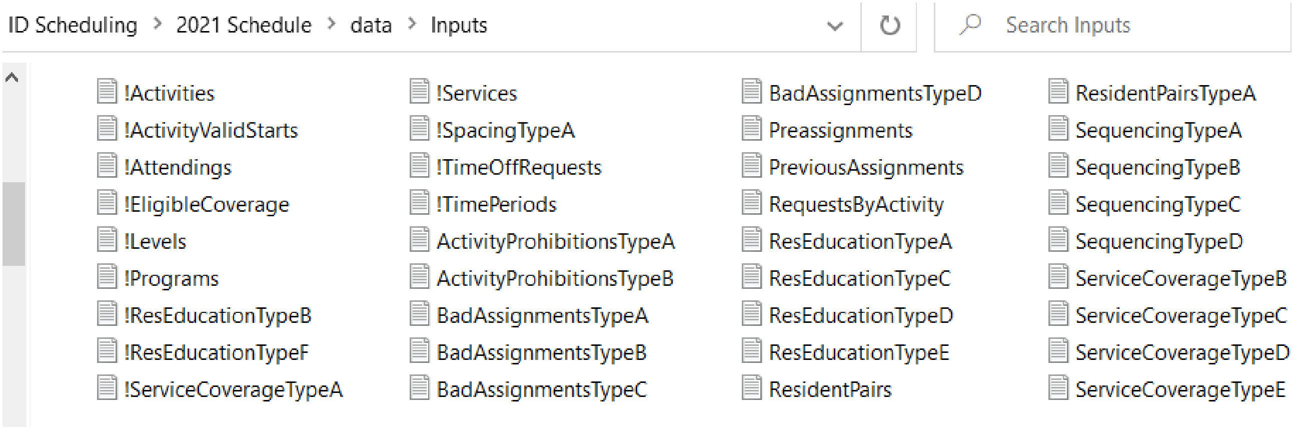
Emptying the input files removes all constraints except for fundamental rules like ensuring that each individual is only assigned to one activity at a time. However, it is important to note that no input files were deleted, even if they were deemed unnecessary. This was done for two reasons. The first being that keeping all the input files provides flexibility in the code if any constraints or rules change within the Infectious Disease Department in the future. The second reason is that the code will not run if any input files are missing. When an input file is empty, the code will quickly scan it and move on since its contents are empty, therefore, all input files remained in the model and only required input files were filled with information.

In order to adjust the current tool to one that will generate feasible schedules for the Infectious Disease Department, the key inputs must be identified. The core inputs to the tool include the set of attendings, set of time periods, and set of services. We use the tool to determine to which service each attending is assigned during each time period in the planning horizon. We also have inputs to specify shift eligibility, service coverage requirements, prohibitions, spacing requirements between shifts, and time off requests. The shift assignments are Michigan Medicine Team A, Michigan Medicine Team B, Veterans Affairs Hospital (VA), transplant shift, or not being assigned. The list of attendings contains all attending names with their program and level which were both kept the same so that all attendings fell into the same grouping. The time period of the schedule follows two week intervals. These are denoted as 1A, 1B,..., 12A, 12B which covers a full calendar year. Shift eligibility distinguishes between which attendings are able to work each service. For example, only some attendings are able to work a shift in the transplant service.This is the same for each service listed with the exception of not being assigned a shift. Shift requirements enforce a range of shift assignments each attending must work. For example, it may be required that a certain attending must work 1-3 shifts of the transplant service. Service prohibitions prevent a certain attending from working some service(s) in a certain time period. For example, Attending 1 may be prohibited from working the transplant and VA service in time period 3A. Spacing requirements between shifts allows there to be an assigned time period gap before an attending is able to work the same service twice. For example, if a spacing requirement for the VA service is two, then at least two time periods must pass after an attending works the VA service before they are able to be assigned the VA service again. Time off requests simply add a constraint that an attending should not be assigned a shift on a time period that they select.

Once these required inputs were identified, their corresponding input data files were marked and filled. To ensure the scheduling tool was altered properly, only basic rules of the schedule were initially added. The basic rules implemented consisted of the key inputs described above.

To test that the tool was functioning properly, we completed two tests building sample schedules. First, we tested for simple schedule feasibility. Second, we tested for optimality. Both tests were meant to check that the code would run without error and generate a schedule. The feasibility test proved we could generate a schedule without consideration for the model metrics. The optimality test proved that we could additionally build a schedule of high quality. The metrics that we can incorporate into the schedule include the number of fulfilled vacation requests and the number of bad shift assignments. Having passed both of these tests, it was deemed ready to receive official inputs from the Infectious Disease Department.

Once the functionality of the tool was verified, the next step was to make it intuitive and user-friendly. The clearest way to distinguish between necessary and unnecessary input files was to use an exclamation point on all input files that need to be updated for each new schedule generated. This helps alleviate some possibility of updating the wrong input files and allows anyone to quickly identify the files that need to be edited. The input file folder is shown below for reference:



ID Scheduling > 2021 Schedule > data > Inputs

!Activities
!ActivityValidStarts
!Attendings
!EligibleCoverage
!Levels
!Programs
!ResEducationTypeB
!ResEducationTypeF
!ServiceCoverageTypeA
!Services
!SpacingTypeA
!TimeOffRequests
!TimePeriods
ActivityProhibitionsTypeA
ActivityProhibitionsTypeB
BadAssignmentsTypeA
BadAssignmentsTypeB
BadAssignmentsTypeC
BadAssignmentsTypeD
Preassignments
PreviousAssignments
RequestsByActivity
ResEducationTypeA
ResEducationTypeC
ResEducationTypeD
ResEducationTypeE
ResidentPairs
ResidentPairsTypeA
SequencingTypeA
SequencingTypeB
SequencingTypeC
SequencingTypeD
ServiceCoverageTypeB
ServiceCoverageTypeC
ServiceCoverageTypeD
ServiceCoverageTypeE

Figure 2: Distinguished Input Data Files via Exclamation Point

Another simple way to make the tool more intuitive was to change the language within the inputs. The most common update was to change the word “resident” to “attending”.

In order for the project to be successful in the long-run, brainstorming was done to come up with a solution to clearly document all steps and troubleshooting methods when generating a schedule. It was determined that Zoom’s recording feature would be the best method for this as it will allow current and future students to collaborate and share their thought process out loud when using the scheduling tool to generate new schedules. These Zoom recordings will serve as a library of resources that new students can review to understand the various inputs and errors they may encounter.

RESULTS

The total run time of the tool to generate a schedule was 0.692 seconds. Out of the 216 requests made for time off, none of them were denied. In order to generate this feasible schedule, some changes were made in the implementation of the input files which is discussed below.

Adjustments

As the input files were updated to run the schedule, a few snags led to infeasible results. First is that in order to generate a feasible schedule, there must be an attending assigned to each service for every time period. This is a total of 144 assignments (6 services * 24 time periods). This criteria was met initially but two attendings had to be removed since they would not be working the upcoming year. When we reset the upper and lower bounds of all the remaining attendings for each service, we had a maximum capacity of 138 assignments. Since there were not enough attending assignments available to meet the required 144 assignments, the schedule was infeasible. After reaching out to ID, we learned that one attending needed to work additional assignments and that a soon-to-be-hired attending would need to be added to the schedule to work two assignments. With these changes, the total number of attending assignment capacity matched the number of required assignments.

Second, exemptions had to be added to any attending that worked during a holiday in the last three years. In our input files, attendings can be preassigned a service in a given time period. 30 time off service assignments were given to various attendings on Thanksgiving and/or Christmas if they had previously worked these dates in the last 3 years. Other preassignments were made to ensure attendings were assigned to certain services for certain time periods based on administrative preference.

Lastly, to ensure each attending only works the total number of assignments they request, the lower and upper bounds of their time off services were both equal to the total number of time periods minus the assignment expectation per attending.

Flexibility of Spacing Service Assignments

In order to study the flexibility of the tool, the spacing requirement between services was examined. The baseline input files had no time off request and flexible lower and upper bounds for attendings to work the services they are eligible for. The spacing requirement between services was increased until the schedule came back infeasible. This occurred when the spacing requirement was five for all services but four for the transplant service.

Since so few attendings are eligible to work transplant (only seven out of 30), this service becomes the constraining factor on whether a feasible schedule can be generated with large spacing rules. In general, it was observed that increasing the spacing rule leads to more time off requests denied.

DISCUSSIONS/CONCLUSIONS

As the project wrapped up, it was clear the majority of the questions we posed on page four of this report were answered.

A significant factor for this project is the flexibility of the tool. By adjusting all the input files of the old scheduling tool, a customizable and flexible tool was made specifically for the ID department. It is able to add and subtract the number of attendings, services, locations, vacation requests, prohibitions, etc.

In order to ensure that the scheduling tool is intuitive for those taking over the project next, a documentation was put in place to create a reference location for debugging the tool in the future. This includes how input files should be updated and why as well as common solutions to different error messages.

A method of documenting the scheduling process will be to record schedule building sessions so that new students can easily learn the ins and outs of the tool.

Throughout this process I learned the importance of communication and documentation. As the baseline tool was built it was crucial for me to understand the fundamentals of the tool. This included but is not limited to what each input file stores and what rules they implement, how to update each input file, how to change the settings of the tool to test for feasibility versus optimality, how to make minor adjustments to the tool to debug any infeasibility, how to read each output file, etc. This required me to frequently meet with my faculty advisors to go over the tool and update the input files. Another way in which communication was important to the success of the project was by having open communication with the clients at the ID department. This allowed us to receive clarifications from them on what they wanted specifically. It also allowed us to check in with them with draft schedules to see if what we implemented met their needs and clear up any miscommunication with the input files we received from them.

Another important lesson from the project is that documentation is key to future success. By having a clear documentation of the process of building and running the tool, future students will have references to help them continue generating schedules for the ID department in the future.