A SAT Based Scheduler for Interpreters and Student-Teacher Conferences

David Weinflash
Department of Computer Science
University of California, Santa Barbara
dweinflash@ucsb.edu

Abstract

Scheduling problems have often been expressed in the form of boolean satisfiability problems. Indeed, many classical problems such as job shop scheduling and the nurse scheduling problem have been solved with the help of a SAT solver. In this project, I propose to utilize propositional satisfiability and a SAT solver in the context of a practical scheduling problem, where volunteer language interpreters are to be matched with student-teacher conferences throughout the Goleta and Santa Barbara school districts. In order to most successfully solve the problem, the SAT solver utilizes a conjunctive normal form formula that takes into account availability, fairness, and practicality constraints to optimally distribute interpreters amongst student-teacher conferences. The quality of the proposed schedule is evaluated in light of these constraints, where the optimal schedule distributes interpreters fairly and practically so that a maximal number of student-teacher conferences may include an interpeter.

Keywords: Constraint Programming, SAT, Scheduling

1 Introduction

In an effort to overcome language barriers between teachers and parents, the *Interdisciplinary Humanities Center* at the University of California, Santa Barbara matches volunteer interpreters with student-teacher conferences throughout the Goleta and Santa Barbara school districts. Student-teacher conferences, which last 30 minutes and take place between 12 PM and 6 PM for one week during the Fall and Spring quarters, allow teachers the opportunity to share with parents the academic progress of each of their students. After a few successful iterations, the demand for interpreters for student-teacher conferences has increased enough to justify the application of a sophisticated system to solve the scheduling problem that arises when matching a limited number of interpreters with an increasing number of student-teacher conferences.

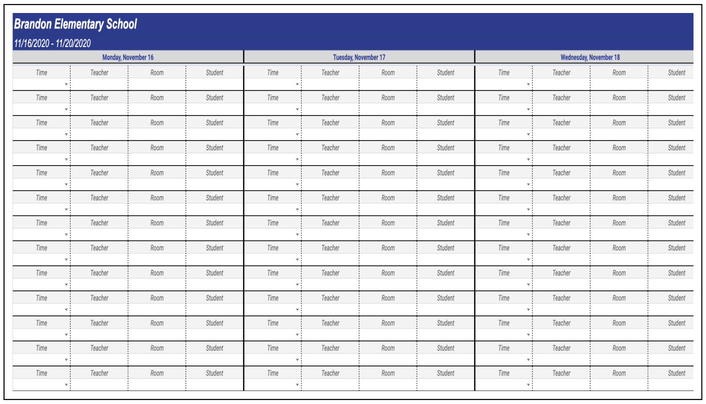
In order to model the scheduling problem, teachers from each elementary school are asked to submit their student-teacher conference requests for interpretation, indicating when each meeting will occur and with which student. Next, volunteer interpreters - composed of undergraduate and graduate students from the University of California, Santa Barbara - are then asked to submit their weekly availability for interpretation and indicate whether or not they have access to transportation. With this data collected, a schedule must then be organized in such a way to ensure that interpreters are equitably and practically matched with the requested student-teacher conferences.

To construct an optimal schedule, the following constraints would be taken into account when pairing interpreters with student-teacher conferences: *Availability* - Interpreters are matched with conferences according to each interpreter's availability calendar. *Fairness* - Interpreters are distributed fairly amongst schools, ensuring that no one school is unfairly assigned a majority of interpreters and no one interpreter is unfairly assigned a disproportional amount of meetings. *Transportation* - Interpreters with access to transportation are assigned to the most distant schools, limiting unnecessary transportation costs. *Convenience* - Interpreters ideally will not have to wait longer than 30 minutes between meetings.

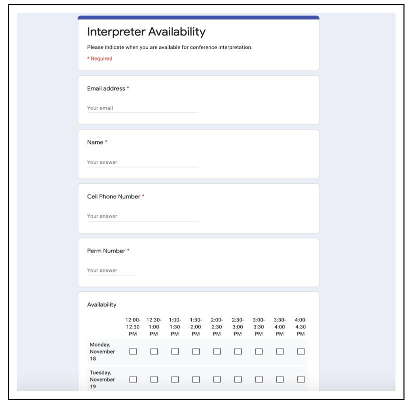
With all constraints accounted for, a master schedule is then organized, divided and ultimately distributed to all schools, teachers and interpreters. Each individual actor will thus receive their own personal schedule for the week; that is to say, each teacher will have their own calendar displaying their requested conferences and respective interpreters, and each interpreter will have their own calendar indicating when and where each student-teacher conference will take place. Scheduling is considered finished once all teachers and interpreters have confirmed the viability of their individual calendars.

CS292C: Computer-Aided Reasoning for Software, UC Santa Barbara

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

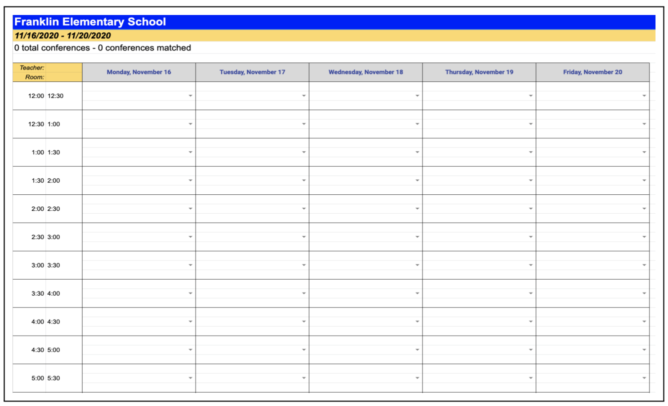


(a) Meeting Requests: Google Sheets

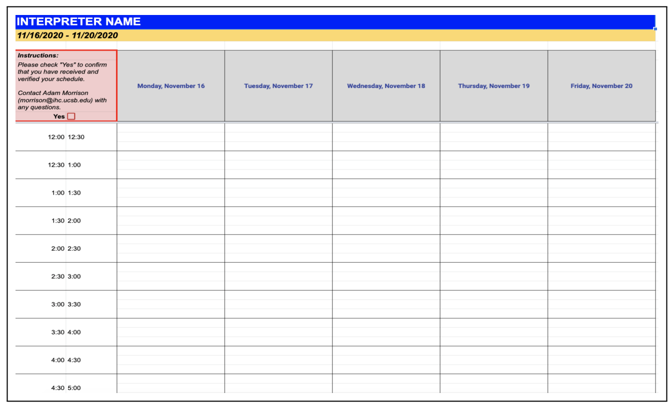


(b) Interpreter Availability: Google Forms

Figure 1. G Suite Applications for Data Collection



(a) Teacher Calendar: Conferences with assigned interpreters



(b) Interpreter Calendar: Conferences with assigned teachers

Figure 2. G Suite Applications for Data Visualization

2 Overview

2.1 Data Collection and Data Visualization

All data collection and data visualization is accomplished using Google's G Suite applications. Specifically, interpreter availability is collected using Google Forms and school meeting requests are collected using Google Sheets. Once all the necessary data is collected, schedules are organized and presented to teachers and interpreters alike via Google Sheets. Figure 1 illustrates the templates used for data collection, while Figure 2 illustrates the calendar templates used for data visualization. In both cases, Google's G Suite applications provide an ideal environment to organize and display data, allowing any modifications necessary to be made quickly and effortlessly.

2.2 Data Pipeline

Once interpreter availability and teacher meeting request data is collected, it is fed into a seperate Google Sheets application - or the *Interdisciplinary Humanities Center* Sheduler application - where it is organized and processed. In terms of data structure, for each interpreter and each teacher, a weekly bitmap is used to represent interpreter availability and teacher meeting requests, respectively. Weekly calendars are then arranged based on matching schedules, where the final output schedule matches a maximal number of interpreters with student-teacher conferences.

Before the integration of Google's CP-SAT solver in the scheduling problem, interpreters were paired with teachers following a round-robin, draft-pick scheduling algorithm. Essentially, each school was sequentially allowed their fair pick of an interpreter who had an availability that best matched the school's total meeting requests. Upon assigning all interpreters to schools, interpreters were then distributed to teachers within the school, maximizing the amount of meetings

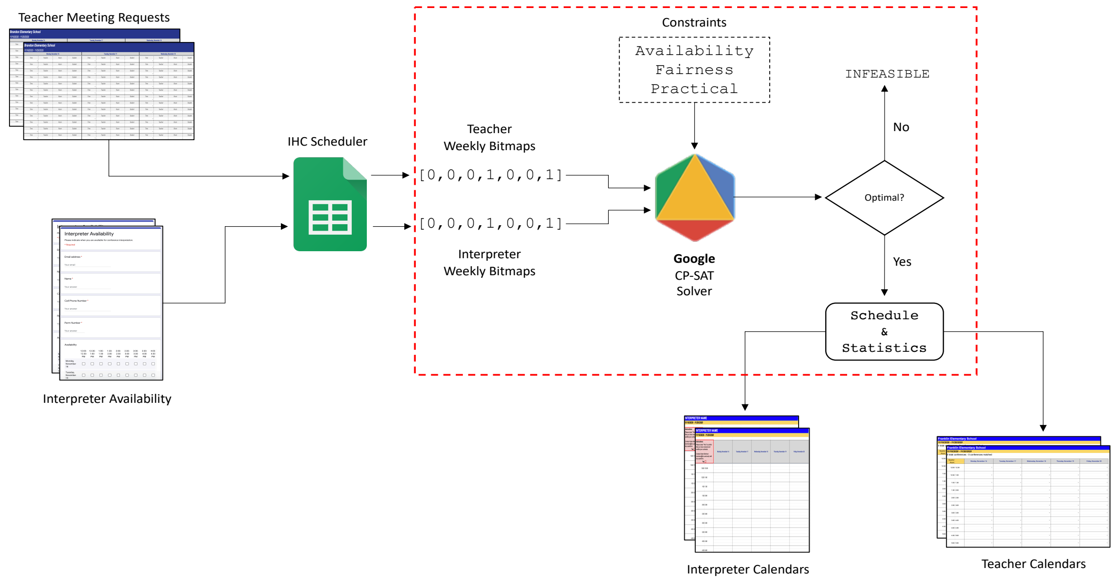


Figure 3. Teacher meeting requests and interpreter availability data is first collected and imported into the *Interdisciplinary Humanities Center* Scheduler application. The collected data is then transformed into weekly bitmaps and exported to the CP-SAT solver. The final schedule output from the solver is organized and distributed to individual teacher and interpreter calendars.

paired with interpreters. With the introduction of Google's CP-SAT solver in the pipeline, however, the round-robin algorithm was able to be replaced with a SAT solver and corresponding objective function. Figure 3 provides an illustration of the data pipeline and indicates where the SAT solver fits in the process.

2.3 CP-SAT Solver

Adopted from Google's open source software suite OR-Tools, the CP-SAT solver utilizes an objective function to maximize the number of student-teacher conferences assigned an interpreter. Figure 4 displays the objective function, where *interp_avails[i][d][s] * mtg_reqs[t][d][s] * shifts[(i, t, d, s)]* equals 1 when a shift s is assigned to the available interpreter i with teacher t on day d.

When optimizing the objective function, the CP-SAT solver is subject to a number of formalized constraints. Primarily, these constraints fall into the constraint categories *Availability*, *Fairness* and *Practicality*. These constraint categories restrict the solver and form the primary challenges of the scheduling problem.

- **Availability** - Scheduling is restricted to matching available interpreters with requested teacher meetings.
- **Fairness** - Interpreters are distributed fairly amongst schools, ensuring no one school is unfairly assigned a majority of interpreters and no one interpreter is unfairly assigned a disproportionate amount of meetings.
- **Practicality** - Based on predefined limits, each interpreter works at least X meetings per week and at most Y meetings per day.

3 Problem Formulation

3.1 Solution Space

The solution space of all combinations matching interpreters to teacher meeting requests is non-trivially large. Indeed, bounding the matching algorithm by an objective function and corresponding availability constraint significantly limits the amount of potential schedules. Consider, for instance, the following example where a school requests 3 meetings per day for 5 days, or 15 weekly meetings. In this example, let there be 3 interpreters, where an interpreter is available to work any shift during any day. In order to distribute shifts evenly, each interpreter is assigned 5 shifts for the week. For the first interpreter, there are $\binom{15}{5}$, or 3,003, ways to attend 5 weekly meetings. Continuing, there are $\binom{10}{5}$ (252) and $\binom{5}{5}$ (1) remaining scheduling combinations for the second and third interpreters, respectively. Combining all possible arrangements, there are $\binom{15}{5} * \binom{10}{5} * \binom{5}{5}$ or 756,756 total ways of scheduling just 3 interpreters to attend 15 weekly meetings.

3.2 Encoding

In order to find an optimal schedule, it is first necessary to define all shifts, days, interpreters and teachers. Afterwards, constraints are applied to the model to ensure that the final schedule is fair, practical and convenient.

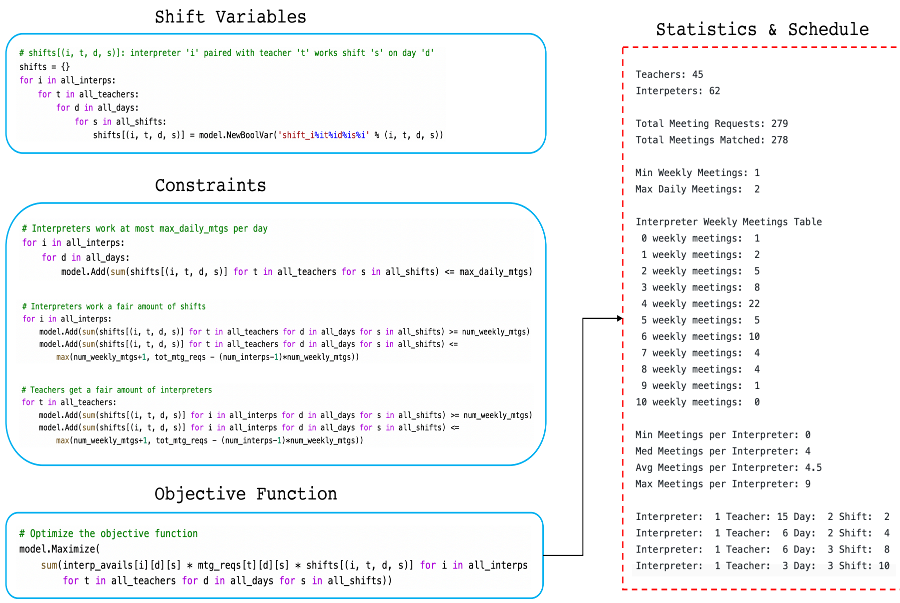


Figure 4. An implementation example of the variables, constraints and objective function the CP-SAT solver utilizes to create a schedule, matching volunteer interpreters with student-teacher conferences. The final output provides various high-level statistics to help describe the final schedule.

Variables. Assume the schedule's shifts are defined by the set S, the days defined by set D, the interpreters by set I and the teachers by set T. Shift variables are thus defined as being $s_{i,t,d}$, meaning that shift s is assigned to interpreter i with teacher t on day d.

Constraints. To narrow down the solution space and equitably assign interpreters to meetings, constraints are applied to the model. In addition to limiting the number of possible solutions, the constraints restrict and guide the objective function.

- Each shift is assigned at most one interpreter:

$$\forall_{t \in T} \, \forall_{d \in D} \, \forall_{s \in S} \, \textstyle\sum_{i \in I} s_{i,t,d} \leq 1$$

- Interpreters booked with at most one teacher at a time:

$$\forall_{i \in I} \, \forall_{d \in D} \, \forall_{s \in S} \, \textstyle\sum_{t \in T} s_{i,t,d} \leq 1$$

- Interpreters work at most M shifts per day:

$$\forall_{i \in I} \, \forall_{d \in D} \, \textstyle\sum_{t \in T} \sum_{s \in S} s_{i,t,d} \leq M$$

- Interpreters work at least F shifts per week:

$$\forall_{i \in I} \, \textstyle\sum_{t \in T} \sum_{d \in D} \sum_{s \in S} s_{i,t,d} \geq F$$

- Interpreters work at most F' shifts per week:

$$\forall_{i \in I} \, \textstyle\sum_{t \in T} \sum_{d \in D} \sum_{s \in S} s_{i,t,d} \leq F'$$

- Teachers are assigned at least F interpreters per week:

$$\forall_{t \in T} \, \textstyle\sum_{i \in I} \sum_{d \in D} \sum_{s \in S} s_{i,t,d} \geq F$$

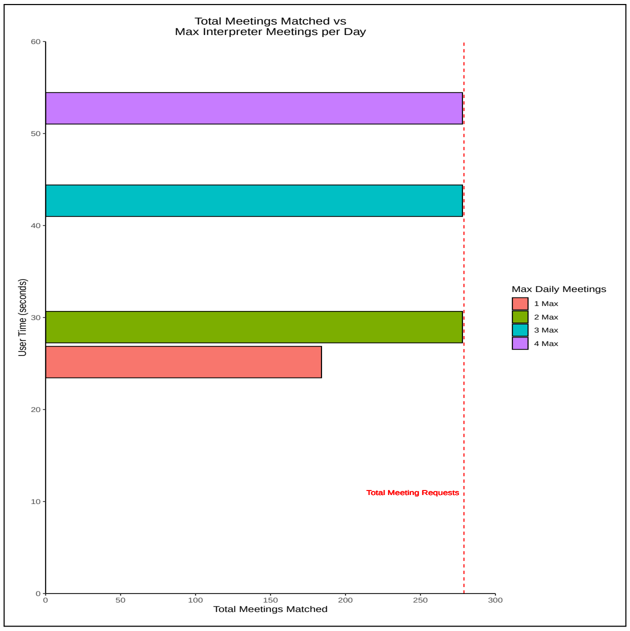
- Teachers are assigned at most F' interpreters per week:

$$\forall_{t \in T} \, \textstyle\sum_{i \in I} \sum_{d \in D} \sum_{s \in S} s_{i,t,d} \leq F'$$

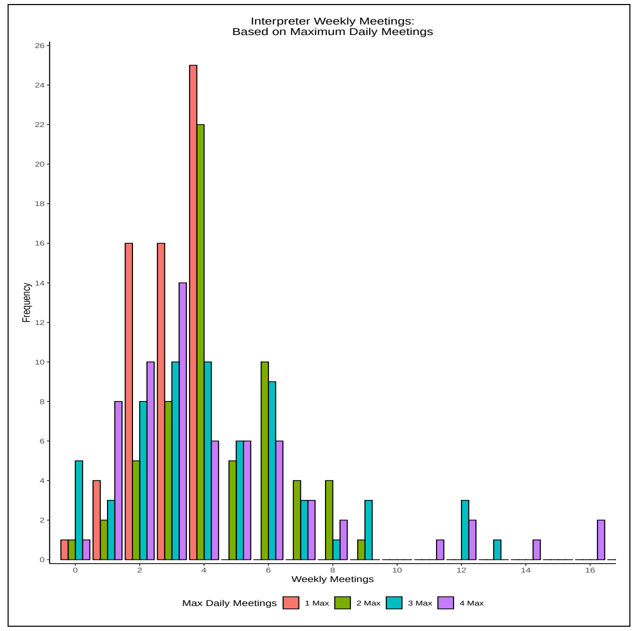
Fairness. As set forth in the constraints above, each interpreter works a fair amount of weekly shifts and each teacher is assigned a fair amount of interpreters. In order to evenly bound the schedules, a few restrictions and calculations are taken into account. In particular, the minimum fair weekly value (F) is bound by the *Practicality* constraint, where each interpreter is set to work at least X meetings per week. The minimum fair weekly value is also restricted by an upper bound, set as the average number of weekly shifts per interpreter (α). The following equation describes the minimum fair weekly value F, used to ensure that every interpreter attends at least F weekly student-teacher conferences:

$$X \leq F \leq \alpha$$

Next, a maximum fair weekly value (F') must be established so that no one interpreter is assigned a majority of shifts and no one school recieves a majority of interpreters. This maximum value is derived from the minimum fair weekly value, where it is set to either F+1 or the amount of



(a) Total meetings matched is largely dependent on the number of daily meetings each interpreter is allowed to attend. As the constraint *Maximum Daily Meetings* increases, so to does the time required to generate an optimal schedule.



(b) The number of weekly meetings assigned to each interpreter varies based on *Maximum Daily Meetings*. Interpreters are assigned a greater number of weekly meetings as the constraint *Maximum Daily Meetings* increases.

Figure 5. Schedules are heavily influenced by the constraint *Maximum Daily Meetings*.

meetings remaining after all interpreters have been assigned their fair share, whichever is larger. The following equation describes the maximum fair weekly value F', using β to represent Total Meeting Requests and λ to represent Number of Interpreters:

$$F' = \max(F+1,\ \beta - (\lambda - 1) * F)$$

4 Evaluation

4.1 Maximum Daily Meetings

The CP-SAT model produces a wide array of schedules when varying the constraint *Maximum Daily Meetings* for each interpreter. In each case, however, the model successfully maximizes the amount of interpreters paired with student-teacher conferences. Indeed, as Figure 5a demonstrates, the model satisfies less than 99% of meeting requests only when the *Maximum Daily Meetings* constraint is restricted to 1 meeting per day. When interpreters are permitted to work greater than 1 meeting per day, the model produces a diverse array of schedules, as illustrated in Figure 5b. Here, the number of weekly meetings per interpreter largely clusters around the mean (or average number of meetings per interpreter), where a greater amount of permitted daily meetings equates to a larger number of weekly meetings per interpreter.

4.2 Scaling: Search Space

4.2.1 Booleans.

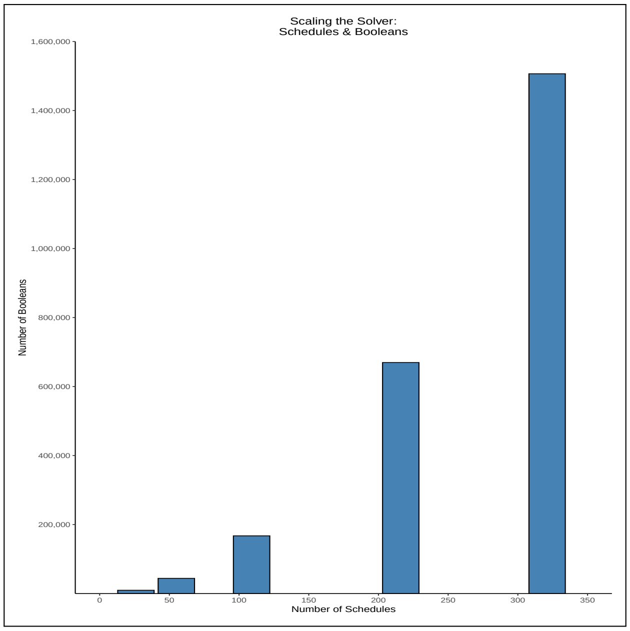
As expected, the SAT solver quickly grows unmanageable as input increases. Perhaps the best indicator of the model's rapid growth is the number of required *Shift Variables*, or booleans used to indicate that shift s is assigned to interpreter i with teacher t on day d. The number of shift variables is defined by the following formula, where each interpreter is thus able to consider an individual teacher schedule for every teacher. Here, $|S|$ represents the number of shift variables, $|D|$ the number of days, $|T|$ the number of teachers and $|I|$ the number of interpreters:

$$|S| = \textit{Maximum Daily Meetings} \cdot |D| \cdot |T| \cdot |I|$$

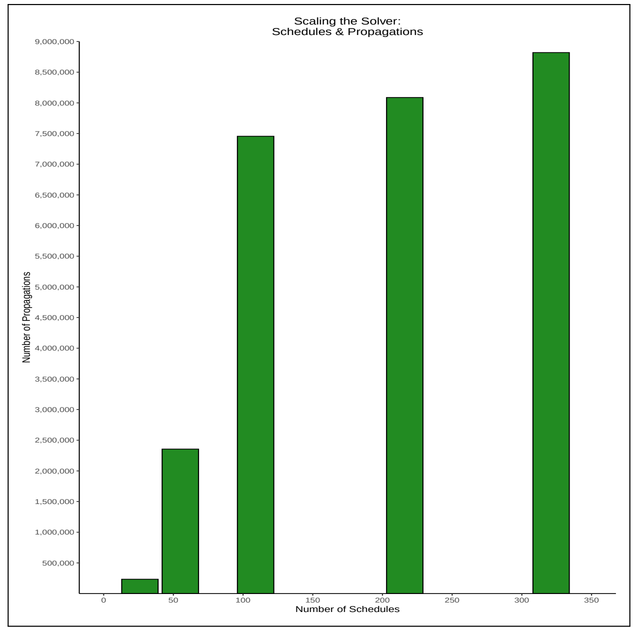
The rapid growth of required shift variables is illustrated in Figure 6a, where the *Number of Booleans* represents the number of shift variables required for each model. Each model receives as input a number of total schedules, made up of all teacher meeting requests and all interpreter availability.

4.2.2 Propagations.

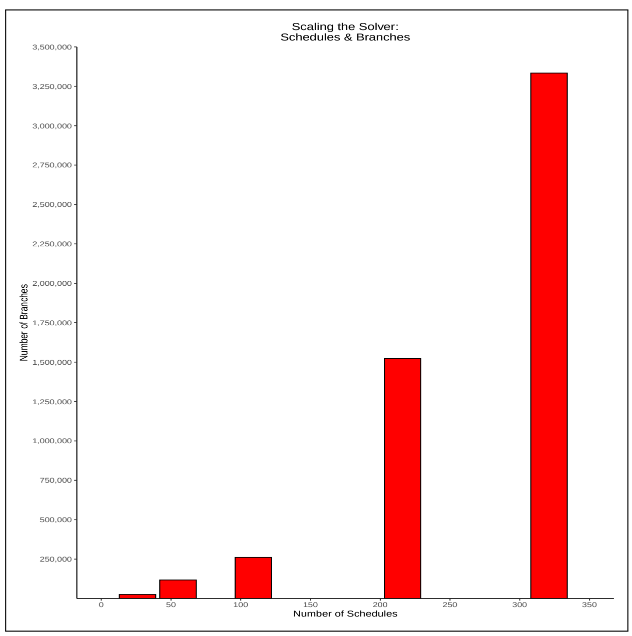
Derived from the number of booleans, the number of propagations also increases dramatically as the input size grows. Propagations, or the generation of clauses on boolean variables, quickly reaches 2.4 million for



(a) The number of required booleans - or *Shift Variables* - quickly increases as the number of considered schedules increases. When creating a model with 320 total schedules, the number of required *Shift Variables* is approximately 1.5 million.

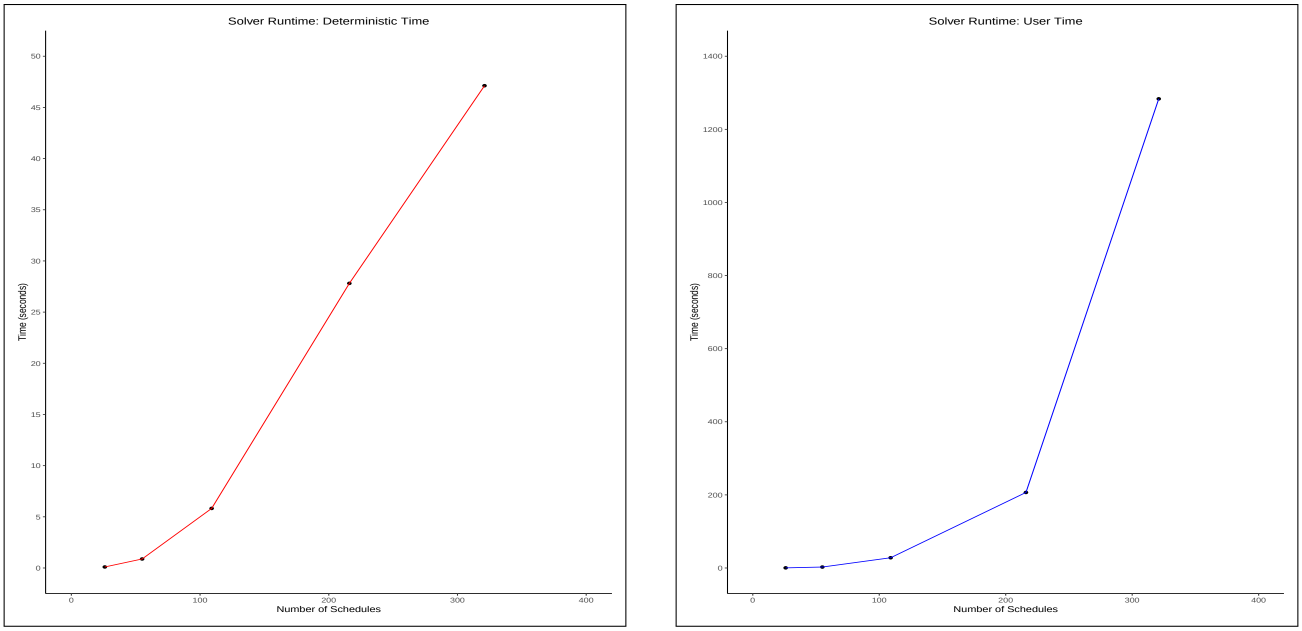


(b) The generation of clauses quickly accelerates with a growing number of input schedules. Such a large number of clauses contributes to the model's slow performance; indeed, creating an optimal arrangement of 320 schedules requires nearly 9 million propagations.



(c) The search space grows exponentially large as the number of considered schedules increases. As demonstrated above, a model considering 320 total schedules will traverse upwards of 3.2 million branches in the search tree before determining the optimal schedule.

Figure 6. The solver traverses an increasingly non-viable search space as the input grows in size.



(a) Deterministic Time measures model time without considering time used by the machine to perform typical operating system tasks. Total Deterministic Time used by the model is considerably less than total User Time.

(b) User Time measures the total time the user spent waiting while the model searched for an optimal schedule. User Time was as long as 20 minutes when considering the optimal arrangement of 320 total schedules.

Figure 7. Model time increases exponentially as the solver considers a growing number of schedules.

an input size as small as 55 total schedules. Indeed, when the input is scaled significantly - up to 320 total schedules - the number of propagations explodes to 8.8 million. Figure 6b demonstrates just how many clauses are generated with an increasing number of schedules.

4.2.3 Branches. The number of branches in the search tree quickly escalates as the model considers an increasing number of total schedules. As the solver explores all search branches - determining whether or not a branch is a dead end - the time required to produce an optimal schedule thus increases in tandem. Figure 6c illustrates the relationship between branches and schedules, where a model considering 320 total schedules will traverse up to 3.3 million branches.

4.3 Scaling: Time

The CP-SAT solver adopted from Google's open source software suite OR-Tools provides an array of performance statistics following successful execution. One statistic, or *Deterministic Time*, indicates the solver's total runtime when disregarding unrelated loads on the user's machine. According to Google, the majority of the solver's total runtime can be attributed to the standard side effects sustained by the operating system, such as context switches, core switches, cache misses and the like. In order to differentiate between the model's *Deterministic Time* and total runtime, Google provides an additional statistic, or *User Time*, to indicate the total time the model took to reach an optimal schedule on the user's machine. Figure 7 illustrates both *Deterministic Time* and *User Time* in relation to a scaled number of considered schedules.

5 Results

In order to best determine the feasibility of deploying the CP-SAT solver in production, the model was fed an anonymized version of the Spring 2020 dataset used by the *Interdisciplinary Humanities Center* at UC Santa Barbara to match volunteer interpreters with student-teacher conferences. The dataset includes 45 teacher request calendars and 62 interpreter availability calendars, or 107 total schedules. Interpreters were restricted to work at least one shift per week (*Min Weekly Meetings* = 1) and at most two shifts per day (*Max Daily Meetings* = 2). The following statistics were output upon deriving the optimal schedule:

```
Model User Time: 38.23 secs

Teachers: 45
Interpreters: 62

Total Meeting Requests: 279
Total Meetings Matched: 278

Min Weekly Meetings: 1
```

```
Interpreter:  1 Teacher: 15 Day:  2 Shift:  2
Interpreter:  1 Teacher:  6 Day:  2 Shift:  4
Interpreter:  1 Teacher:  6 Day:  3 Shift:  8
Interpreter:  1 Teacher:  3 Day:  3 Shift: 10

Interpreter:  2 Teacher: 11 Day:  1 Shift:  2
Interpreter:  2 Teacher:  5 Day:  1 Shift:  6
Interpreter:  2 Teacher: 17 Day:  2 Shift:  3
Interpreter:  2 Teacher: 16 Day:  2 Shift:  5
Interpreter:  2 Teacher: 10 Day:  3 Shift:  4
Interpreter:  2 Teacher:  7 Day:  3 Shift:  7

Interpreter:  3 Teacher: 13 Day:  1 Shift:  2
Interpreter:  3 Teacher: 10 Day:  1 Shift: 10
Interpreter:  3 Teacher:  6 Day:  2 Shift:  1
Interpreter:  3 Teacher: 12 Day:  2 Shift:  2
Interpreter:  3 Teacher:  2 Day:  3 Shift:  2
Interpreter:  3 Teacher:  1 Day:  3 Shift:  7

Interpreter:  4 Teacher:  2 Day:  1 Shift:  6
Interpreter:  4 Teacher:  1 Day:  1 Shift:  9
Interpreter:  4 Teacher:  2 Day:  2 Shift:  1
Interpreter:  4 Teacher: 16 Day:  2 Shift:  8
Interpreter:  4 Teacher:  8 Day:  3 Shift:  6
Interpreter:  4 Teacher: 12 Day:  4 Shift:  3

Interpreter:  5 Teacher: 12 Day:  2 Shift:  4
Interpreter:  5 Teacher: 10 Day:  2 Shift:  5
Interpreter:  5 Teacher: 28 Day:  3 Shift:  4
Interpreter:  5 Teacher: 11 Day:  3 Shift:  6
Interpreter:  5 Teacher:  5 Day:  4 Shift:  4

Interpreter:  6 Teacher: 11 Day:  2 Shift:  8
Interpreter:  6 Teacher:  3 Day:  2 Shift:  9
Interpreter:  6 Teacher: 16 Day:  3 Shift:  6
Interpreter:  6 Teacher:  5 Day:  4 Shift:  3

Interpreter:  7 Teacher:  6 Day:  1 Shift:  8
Interpreter:  7 Teacher: 15 Day:  3 Shift:  4
Interpreter:  7 Teacher: 11 Day:  3 Shift:  5

Interpreter:  8 Teacher: 12 Day:  1 Shift:  6
Interpreter:  8 Teacher:  1 Day:  1 Shift: 10

Interpreter:  9 Teacher: 12 Day:  1 Shift:  3
Interpreter:  9 Teacher:  3 Day:  1 Shift:  4
Interpreter:  9 Teacher: 15 Day:  3 Shift:  5
Interpreter:  9 Teacher: 25 Day:  3 Shift:  7

Interpreter: 10 Teacher:  4 Day:  1 Shift:  8
Interpreter: 10 Teacher: 15 Day:  1 Shift: 10

Interpreter: 11 Teacher: 12 Day:  4 Shift:  5

Interpreter: 12 Teacher: 16 Day:  2 Shift:  2
Interpreter: 12 Teacher: 15 Day:  2 Shift:  4
Interpreter: 12 Teacher:  6 Day:  3 Shift:  1
Interpreter: 12 Teacher: 18 Day:  3 Shift:  2

Interpreter: 13 Teacher: 10 Day:  2 Shift:  2
Interpreter: 13 Teacher: 15 Day:  2 Shift:  3
Interpreter: 13 Teacher: 19 Day:  3 Shift:  2
Interpreter: 13 Teacher:  5 Day:  3 Shift:  5
Interpreter: 13 Teacher:  2 Day:  4 Shift:  3

Interpreter: 14 Teacher: 14 Day:  1 Shift:  3
Interpreter: 14 Teacher: 17 Day:  1 Shift:  5
Interpreter: 14 Teacher:  4 Day:  2 Shift:  1
Interpreter: 14 Teacher: 11 Day:  2 Shift:  2
Interpreter: 14 Teacher:  5 Day:  3 Shift:  3
Interpreter: 14 Teacher:  5 Day:  4 Shift:  2

Interpreter: 15 Teacher:  5 Day:  4 Shift:  1

Interpreter: 16 Teacher: 15 Day:  1 Shift:  9
Interpreter: 16 Teacher: 27 Day:  2 Shift:  5
Interpreter: 16 Teacher: 10 Day:  2 Shift:  6
Interpreter: 16 Teacher:  5 Day:  3 Shift:  7
Interpreter: 16 Teacher: 15 Day:  4 Shift:  3

Interpreter: 17 Teacher:  4 Day:  1 Shift:  9
Interpreter: 17 Teacher:  3 Day:  1 Shift: 10
Interpreter: 17 Teacher:  4 Day:  2 Shift:  2
Interpreter: 17 Teacher:  7 Day:  2 Shift: 10
Interpreter: 17 Teacher: 21 Day:  3 Shift:  6
Interpreter: 17 Teacher: 21 Day:  3 Shift:  7
Interpreter: 17 Teacher:  7 Day:  4 Shift:  2

Interpreter: 18 Teacher: 20 Day:  1 Shift:  7
Interpreter: 18 Teacher: 11 Day:  1 Shift:  8
Interpreter: 18 Teacher:  3 Day:  2 Shift:  8
Interpreter: 18 Teacher: 18 Day:  2 Shift: 10
Interpreter: 18 Teacher: 15 Day:  4 Shift:  2

Interpreter: 19 Teacher: 20 Day:  1 Shift:  5
Interpreter: 19 Teacher: 18 Day:  1 Shift: 10
Interpreter: 19 Teacher:  3 Day:  2 Shift:  5
Interpreter: 19 Teacher: 16 Day:  2 Shift:  6
Interpreter: 19 Teacher:  2 Day:  3 Shift:  1
Interpreter: 19 Teacher: 36 Day:  3 Shift:  7

Interpreter: 20 Teacher: 17 Day:  1 Shift:  8
Interpreter: 20 Teacher: 28 Day:  2 Shift:  8
Interpreter: 20 Teacher:  1 Day:  2 Shift:  9
Interpreter: 20 Teacher:  5 Day:  4 Shift:  7

Interpreter: 21 Teacher: 26 Day:  3 Shift:  4
Interpreter: 21 Teacher: 26 Day:  3 Shift:  5
Interpreter: 21 Teacher: 10 Day:  4 Shift:  2

Interpreter: 22 Teacher: 22 Day:  1 Shift:  4
Interpreter: 22 Teacher: 11 Day:  1 Shift:  5

Interpreter: 23 Teacher:  8 Day:  1 Shift:  4
Interpreter: 23 Teacher: 24 Day:  1 Shift:  5
Interpreter: 23 Teacher: 33 Day:  2 Shift:  6
Interpreter: 23 Teacher:  2 Day:  2 Shift: 11
Interpreter: 23 Teacher: 10 Day:  3 Shift:  2
Interpreter: 23 Teacher: 32 Day:  3 Shift:  5
Interpreter: 23 Teacher: 12 Day:  4 Shift:  6

Interpreter: 24 Teacher: 14 Day:  2 Shift:  3
Interpreter: 24 Teacher:  8 Day:  2 Shift:  5
Interpreter: 24 Teacher: 12 Day:  3 Shift:  5
Interpreter: 24 Teacher:  2 Day:  4 Shift:  5

Interpreter: 25 Teacher:  2 Day:  2 Shift:  2
Interpreter: 25 Teacher: 18 Day:  2 Shift:  3
Interpreter: 25 Teacher: 11 Day:  4 Shift:  5
Interpreter: 25 Teacher: 11 Day:  4 Shift:  6

Interpreter: 26 Teacher:  2 Day:  1 Shift:  2
Interpreter: 26 Teacher: 20 Day:  1 Shift:  4
Interpreter: 26 Teacher: 19 Day:  3 Shift:  4
Interpreter: 26 Teacher: 33 Day:  3 Shift:  6

Interpreter: 27 Teacher:  4 Day:  1 Shift:  5
Interpreter: 27 Teacher: 15 Day:  1 Shift:  7
Interpreter: 27 Teacher: 27 Day:  3 Shift:  5
Interpreter: 27 Teacher: 37 Day:  3 Shift:  6

Interpreter: 28 Teacher: 12 Day:  1 Shift:  4
Interpreter: 28 Teacher: 35 Day:  2 Shift:  5
Interpreter: 28 Teacher: 17 Day:  2 Shift:  6
Interpreter: 28 Teacher:  4 Day:  3 Shift:  9
Interpreter: 28 Teacher:  2 Day:  4 Shift:  1
Interpreter: 28 Teacher: 14 Day:  4 Shift:  5

Interpreter: 29 Teacher: 17 Day:  2 Shift:  2
Interpreter: 29 Teacher:  6 Day:  2 Shift: 10
Interpreter: 29 Teacher: 14 Day:  3 Shift:  3
Interpreter: 29 Teacher: 35 Day:  3 Shift:  5

Interpreter: 30 Teacher: 15 Day:  2 Shift:  5
Interpreter: 30 Teacher:  4 Day:  2 Shift:  6
Interpreter: 30 Teacher:  1 Day:  3 Shift:  5
Interpreter: 30 Teacher: 27 Day:  3 Shift:  6

Interpreter: 31 Teacher: 24 Day:  1 Shift:  6
Interpreter: 31 Teacher:  9 Day:  2 Shift:  4
Interpreter: 31 Teacher:  9 Day:  2 Shift:  5
Interpreter: 31 Teacher: 30 Day:  3 Shift:  6
Interpreter: 31 Teacher: 10 Day:  3 Shift:  7
Interpreter: 31 Teacher:  9 Day:  4 Shift:  5
Interpreter: 31 Teacher:  5 Day:  4 Shift:  6

Interpreter: 32 Teacher: 11 Day:  2 Shift:  5
Interpreter: 32 Teacher:  5 Day:  2 Shift:  7
Interpreter: 32 Teacher:  5 Day:  3 Shift:  6
Interpreter: 32 Teacher: 11 Day:  3 Shift:  7

Interpreter: 33 Teacher:  1 Day:  1 Shift:  8
Interpreter: 33 Teacher: 10 Day:  1 Shift:  9
Interpreter: 33 Teacher:  2 Day:  2 Shift:  5
Interpreter: 33 Teacher: 27 Day:  2 Shift:  6

Interpreter: 34 Teacher: 16 Day:  1 Shift:  6
Interpreter: 34 Teacher: 18 Day:  1 Shift:  9
Interpreter: 34 Teacher:  4 Day:  3 Shift: 10
Interpreter: 34 Teacher:  2 Day:  3 Shift: 11

Interpreter: 35 Teacher: 10 Day:  1 Shift:  2
Interpreter: 35 Teacher: 15 Day:  1 Shift:  3
Interpreter: 35 Teacher: 14 Day:  2 Shift:  6
Interpreter: 35 Teacher: 33 Day:  2 Shift:  7
Interpreter: 35 Teacher: 15 Day:  3 Shift:  2
Interpreter: 35 Teacher:  8 Day:  4 Shift:  6
Interpreter: 35 Teacher: 15 Day:  4 Shift:  7

Interpreter: 36 Teacher:  6 Day:  1 Shift:  3
Interpreter: 36 Teacher: 13 Day:  3 Shift:  2
Interpreter: 36 Teacher: 11 Day:  3 Shift:  3
Interpreter: 36 Teacher:  9 Day:  4 Shift:  4

Interpreter: 37 Teacher:  2 Day:  1 Shift:  4
Interpreter: 37 Teacher: 10 Day:  2 Shift: 10
Interpreter: 37 Teacher:  1 Day:  2 Shift: 11
Interpreter: 37 Teacher: 11 Day:  3 Shift:  4
Interpreter: 37 Teacher: 39 Day:  3 Shift:  5
Interpreter: 37 Teacher:  2 Day:  4 Shift:  6

Interpreter: 38 Teacher:  2 Day:  1 Shift:  1
Interpreter: 38 Teacher: 14 Day:  1 Shift:  2
Interpreter: 38 Teacher: 13 Day:  2 Shift:  2
Interpreter: 38 Teacher: 11 Day:  2 Shift:  4
Interpreter: 38 Teacher:  5 Day:  3 Shift:  1
Interpreter: 38 Teacher: 19 Day:  3 Shift:  3
Interpreter: 38 Teacher: 12 Day:  4 Shift:  2
Interpreter: 38 Teacher: 15 Day:  4 Shift:  4

Interpreter: 39 Teacher:  7 Day:  1 Shift:  5
Interpreter: 39 Teacher: 15 Day:  1 Shift:  6
Interpreter: 39 Teacher: 10 Day:  3 Shift:  5

Interpreter: 40 Teacher: 13 Day:  1 Shift:  3
Interpreter: 40 Teacher:  9 Day:  1 Shift:  5
Interpreter: 40 Teacher: 14 Day:  2 Shift:  2

Interpreter: 41 Teacher: 12 Day:  1 Shift:  2
Interpreter: 41 Teacher:  8 Day:  2 Shift:  2
Interpreter: 41 Teacher: 16 Day:  3 Shift:  2
Interpreter: 41 Teacher: 30 Day:  3 Shift:  5

Interpreter: 42 Teacher: 12 Day:  2 Shift:  3
Interpreter: 42 Teacher: 18 Day:  3 Shift:  3
Interpreter: 42 Teacher:  8 Day:  4 Shift:  5

Interpreter: 43 Teacher: 15 Day:  1 Shift:  2
Interpreter: 43 Teacher:  9 Day:  1 Shift:  3
Interpreter: 43 Teacher: 16 Day:  3 Shift:  4
Interpreter: 43 Teacher: 25 Day:  3 Shift:  6

Interpreter: 44 Teacher:  2 Day:  1 Shift:  3
Interpreter: 44 Teacher: 16 Day:  1 Shift:  5
Interpreter: 44 Teacher:  2 Day:  2 Shift:  6
Interpreter: 44 Teacher:  8 Day:  3 Shift:  4

Interpreter: 45 Teacher:  9 Day:  1 Shift:  2
Interpreter: 45 Teacher: 13 Day:  1 Shift:  6
Interpreter: 45 Teacher: 11 Day:  4 Shift:  3

Interpreter: 46 Teacher:  9 Day:  1 Shift:  4
Interpreter: 46 Teacher:  2 Day:  1 Shift:  5
Interpreter: 46 Teacher:  5 Day:  2 Shift:  8
Interpreter: 46 Teacher:  7 Day:  3 Shift:  4

Interpreter: 47 Teacher: 21 Day:  1 Shift:  4
Interpreter: 47 Teacher: 17 Day:  1 Shift:  6
Interpreter: 47 Teacher:  5 Day:  2 Shift:  3

Interpreter: 48 Teacher: 16 Day:  1 Shift:  3
Interpreter: 48 Teacher: 23 Day:  1 Shift:  5
Interpreter: 48 Teacher: 19 Day:  2 Shift:  6
Interpreter: 48 Teacher: 11 Day:  3 Shift:  2

Interpreter: 49 Teacher:  9 Day:  3 Shift:  2
Interpreter: 49 Teacher:  5 Day:  3 Shift:  8

Interpreter: 50 Teacher:  5 Day:  1 Shift:  4
Interpreter: 50 Teacher: 11 Day:  1 Shift:  6
Interpreter: 50 Teacher: 29 Day:  2 Shift:  4
Interpreter: 50 Teacher:  4 Day:  3 Shift:  1
Interpreter: 50 Teacher:  8 Day:  3 Shift:  5

Interpreter: 51 Teacher:  5 Day:  1 Shift:  5
Interpreter: 51 Teacher:  8 Day:  1 Shift:  6

Interpreter: 52 Teacher: 25 Day:  1 Shift:  5
Interpreter: 52 Teacher: 26 Day:  1 Shift:  6
Interpreter: 52 Teacher: 35 Day:  2 Shift:  8
Interpreter: 52 Teacher: 35 Day:  2 Shift:  9
Interpreter: 52 Teacher: 37 Day:  3 Shift:  7
Interpreter: 52 Teacher: 15 Day:  3 Shift:  8
Interpreter: 52 Teacher: 16 Day:  4 Shift:  5
Interpreter: 52 Teacher: 16 Day:  4 Shift:  6

Interpreter: 53 Teacher: 17 Day:  1 Shift:  3
Interpreter: 53 Teacher: 24 Day:  1 Shift:  4
Interpreter: 53 Teacher: 19 Day:  2 Shift:  3
Interpreter: 53 Teacher: 33 Day:  2 Shift:  4
Interpreter: 53 Teacher: 34 Day:  3 Shift:  3
Interpreter: 53 Teacher: 30 Day:  3 Shift:  4
Interpreter: 53 Teacher: 16 Day:  4 Shift:  2
Interpreter: 53 Teacher: 16 Day:  4 Shift:  3

Interpreter: 55 Teacher: 34 Day:  1 Shift:  6
Interpreter: 55 Teacher: 34 Day:  1 Shift:  9
Interpreter: 55 Teacher: 38 Day:  2 Shift:  5
Interpreter: 55 Teacher: 36 Day:  2 Shift:  8
Interpreter: 55 Teacher: 17 Day:  4 Shift:  2
Interpreter: 55 Teacher: 17 Day:  4 Shift:  5

Interpreter: 56 Teacher: 15 Day:  3 Shift:  9
Interpreter: 56 Teacher: 15 Day:  3 Shift: 10
Interpreter: 56 Teacher: 17 Day:  4 Shift:  4
Interpreter: 56 Teacher: 21 Day:  4 Shift:  5

Interpreter: 57 Teacher: 35 Day:  1 Shift:  5
Interpreter: 57 Teacher: 35 Day:  1 Shift:  7
Interpreter: 57 Teacher: 38 Day:  2 Shift:  6
Interpreter: 57 Teacher: 39 Day:  2 Shift:  7
Interpreter: 57 Teacher: 41 Day:  3 Shift:  8
Interpreter: 57 Teacher: 41 Day:  3 Shift: 10
Interpreter: 57 Teacher: 27 Day:  4 Shift:  5
Interpreter: 57 Teacher: 26 Day:  4 Shift:  6

Interpreter: 58 Teacher: 36 Day:  1 Shift:  7
Interpreter: 58 Teacher: 35 Day:  1 Shift:  8
Interpreter: 58 Teacher: 39 Day:  2 Shift:  5
Interpreter: 58 Teacher: 40 Day:  2 Shift:  7
Interpreter: 58 Teacher: 42 Day:  3 Shift:  8
Interpreter: 58 Teacher: 41 Day:  3 Shift:  9
Interpreter: 58 Teacher: 31 Day:  4 Shift:  3
Interpreter: 58 Teacher: 37 Day:  4 Shift:  5
Interpreter: 58 Teacher: 15 Day:  5 Shift:  2

Interpreter: 59 Teacher: 40 Day:  2 Shift:  8
Interpreter: 59 Teacher: 41 Day:  2 Shift: 10
Interpreter: 59 Teacher: 42 Day:  3 Shift: 10
Interpreter: 59 Teacher: 42 Day:  3 Shift: 11
Interpreter: 59 Teacher: 43 Day:  4 Shift:  5
Interpreter: 59 Teacher: 43 Day:  4 Shift:  6

Interpreter: 60 Teacher: 38 Day:  1 Shift:  5
Interpreter: 60 Teacher: 37 Day:  1 Shift:  6
Interpreter: 60 Teacher: 43 Day:  3 Shift:  6
Interpreter: 60 Teacher: 42 Day:  3 Shift:  7

Interpreter: 61 Teacher: 43 Day:  4 Shift:  7
Interpreter: 61 Teacher: 43 Day:  4 Shift:  8
Interpreter: 61 Teacher: 18 Day:  5 Shift:  2

Interpreter: 62 Teacher: 43 Day:  1 Shift:  5
Interpreter: 62 Teacher: 38 Day:  1 Shift:  6
Interpreter: 62 Teacher: 41 Day:  2 Shift:  8
Interpreter: 62 Teacher: 41 Day:  2 Shift:  9
Interpreter: 62 Teacher: 45 Day:  3 Shift:  8
Interpreter: 62 Teacher: 42 Day:  3 Shift:  9
```

Figure 8. The final schedule output by the solver when considering annonymized interpreter and teacher schedules from the Spring 2020 *Interdisciplinary Humanities Center* dataset.

```
Max Daily Meetings: 2

Interpreter Weekly Meetings Table
 0 weekly meetings: 1
 1 weekly meetings: 2
 2 weekly meetings: 5
 3 weekly meetings: 8
 4 weekly meetings: 22
 5 weekly meetings: 5
 6 weekly meetings: 10
 7 weekly meetings: 4
 8 weekly meetings: 4
 9 weekly meetings: 1
10 weekly meetings: 0

Min Meetings per Interpreter: 0
Med Meetings per Interpreter: 4
Avg Meetings per Interpreter: 4.5
Max Meetings per Interpreter: 9
```

The final, most optimal schedule arranged by the solver is displayed in Figure 8. The output schedule is grouped in ascending order by interpreter, where each interpreter is presented alongside their assigned student-teacher conferences (as indicated by the corresponding teacher, day and shift number). The schedule thus provides an ideal solution for optimally organizing fair, pracical and convenient schedules for interpreters and teachers alike.

6 Future Work

As mentioned in the Introduction, the most optimal arrangement of interpreters and teacher meeting requests will take into account *Transportation* and *Convenience* constraints. Indeed, once student-teacher conferences are no longer virtual, it will be necessary to introduce new constraints. To limit transportation costs, for instance, interpreters with vehicles should be assigned to the most distant schools. Additionally, interpreters should have to wait no longer than 30 minutes between meetings so to eliminate the amount of down-time between shifts.

7 Conclusion

In this project, a SAT solver provided by Google's OR-Tools was utilized successfully to optimally arrange a schedule pairing volunteer interpreters with student-teacher conferences. Taking into account *Availability*, *Fairness* and *Practicality* constraints, the model outputs a schedule that maximizes the amount of student-teacher conferences assigned an interpreter. While unable to scale efficiently, the SAT solver nonetheless provides an optimal schedule, allowing the *Interdisciplinary Humanities Center* at UC Santa Barbara to effectively pair volunteer interpreters with student-teacher conferences across the Goleta and Santa Barbara school districts.

References

[1] Martins, Ruben and Justine Sherry. *Lisbon Wedding: Seating arrangements using MaxSAT.*

[2] Zhang, Hantao, Dapeng Li and Haiou Shen. *A Sat Based Scheduler for Tournament Schedules.*

[3] Google OR-Tools, `https://developers.google.com/optimization`