Scheduling B'nai Mitzvot Using Linear Optimization

Zachary Siegel

May 22, 2018

Abstract

Sinai Temple is the religious home of a thriving Jewish population in Los Angeles, CA. Among the congregants are 50-150 children between the ages of 12 and 13 in a given year. Through the Jewish practice of B'nai Mitzvah, each child "begins" Jewish life after their 13th Hebrew birthday.

With only 52 weeks in a year and three venues in which to hold a ceremony, Sinai Temple takes care to ensure that the B'nai Mitzvah schedule fits and meets the needs of the congregation.

Concerns in planning dates for each of the children celebrating B'nai Mitzvot has been carried out by a committee at Sinai Temple in recent years. Many hours of human effort are required to personally verify that each family's reported needs are being met.

Automation can both relieve the burden from community organizers planning B'nai Mitzvah dates, and simultaneously meet more and more complex requirements than those organizers on their own.

This project is to automate and optimize the planning of B'nai Mitzvah dates within a congregation.

1 Background

1.1 Hebrew Calendar and Religious Considerations

A child's B'nai Mitzvah celebration, according to Jewish tradition, should not precede a child's 13th birthday. The Jewish world in this case measures age with respect to the Hebrew/Jewish calendar. A Hebrew birthday in a given year may occur before or after a Gregorian birthday, but is usually not far off.

"The Hebrew lunar year is about eleven days shorter than the solar year and uses the 19-year Metonic cycle to bring it into line with the solar year,

with the addition of an intercalary month every two or three years, for a total of seven times per 19 years" (Wikipedia "Hebrew Calendar").

Conversion between Hebrew and Gregorian dates is easily achieved using online tools (using the open source Hebcal or a tool hosted by Chabad). For this project, date conversion was automated using the Python programming language package ConvertDate and the Javascript package Hebcal.

B'nai Mitzvot ceremonies are generally held on Saturday mornings. Certain dates are not appropriate for the celebrations, such as Jewish high holidays. For most families, common American holidays are not feasible dates.

1.2 Linear Optimization

Some real-world problems can be modeled with constraints that are *linear* in some quantity. This means that as a value changes, it affects the quality or feasibility of a solution by an amount proportional to the value itself.

For example, the quality of a route travelled increases as its total time or distance decreases; this is a linear relationship. In contrast, the value of a session with a personal trainer increases as the duration of the session increases, but only to a point, after which the value no longer increases (say, 1.5 hours).

Linear optimization is the process of finding a solution to a problem that is 1) *feasible*, meaning it satisfies constraints (which are themselves linear), and 2) *minimal/maximal*, meaning that it minimizes or maximizes a measurement of the "cost" or "quality" of the solution, expressed through linear relationships with the parts of the solution. The *objective function* of a *linear model* is a measurement of the desirability of the solution.

If a problem can be articulated through a linear model, its optimal solution can be found trivially unless it is extraordinarily complex. Algorithms to solve linear optimization models predate computers by a wide margin. The most complex linear optimization models are used in global logistics operations, such as fleet deployment and supply chain management. Simple linear models are used by GPS route guidance systems.

There is a wide range of problems to which linear optimization is applicable. Many scheduling problems are solved efficiently by linear optimization.

2 Motivating Example

A linear model may describe a scheduling problem in the following way.

Suppose Saturday, March 5th is a child's Hebrew birthday, and that date is 25 weeks from today. The variable x can represent the week chosen

for a Bar Mitzvah, from 0 to 52, where 0 represents *this* Saturday, and 52 represents Saturday *one year from today.*

The ideal value for x would be 25, the child's Hebrew birth date. The date must be after the birthday, so $x \geq 25$ is the first constraint.

Now suppose the child's family is unavailable March 5th and 11th. Then $x \neq 25$ and $x \neq 26$ are additional constraints.

For every week after Saturday, March 5th, a cost of 2 is applied to the objective function of the model.

So far, our model is:

$$\begin{array}{ll} \text{minimize} & 2x \\ \text{such that} & x \geq 25 \\ & x \neq 25 \\ & x \neq 26 \end{array}$$

There are no terms involving x^2, x^3, $\frac{1}{x}$, 2^x, $sin(x)$; only *linear* constraints such as $x \geq 25$ (or $10x \geq -39$, for another example) are permitted for a model to be *linear.*

It may seem clear that an optimal solution is 27. In this basic example, that would be a good solution. However, as articulated so far, this model would prefer a solution such as 26.1 or 26.00001. It turns out that $x \neq 26$ is *not* a linear constraint, and the optimal solution is undefined due to this constraint. By restricting x to be a *whole number* (an *integer*), however, we can include these constraints. Restricting variables to integers greatly increases the computational cost of solving a linear model, but allows whoever is articulating the model to use the same simple language, and the computer calculating the solution to employ many of the same efficient methods. This process is called *integer linear programming.*

Our model can now be fully realized as

$$\begin{array}{ll} \text{minimize} & 2x \\ \text{such that} & x \geq 25 \\ & x \neq 25 \\ & x \neq 26 \\ & x \text{ is an } \textit{integer} \end{array}$$

Any of the ubiquitous and efficient tools for solving linear optimization problems can be employed to solve this problem.

A scheduling problem involving many events and more complex requirements can become more difficult to articulate using a linear model, but many ideas can be expressed through a linear model not so different from this one.

3 Method: The Linear Model

The model solved to assign B'nai Mitzvah dates employs integer linear programming to choose for each student a *date* and a *venue* for their B'nai Mitzvah at Sinai Temple.

3.1 The Decision Variables

Every student is assigned an index from 1 to n, where there are n total students. Every venue is assigned an index from 1 to m, where there are m total venues (Sinai Temple has three venues). Every date under consideration, each a Saturday, is given an index denoting the number of weeks from *this* Saturday on which it falls.

A variable is created for every combination of student, venue, and date, and takes on the following values:

$$x_{ijk} = \begin{cases} 1 & \text{student } i \text{ has BM at venue } j \text{ on date } k \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The goal is for the solution to a linear model to assign values to the variables x_{ijk} such that the assignments honor constraints, and are optimal relative to some objective function. In order to achieve this, the constraints and objective function just need to be articulated as linear functions in the variables and a solver can produce a solution.

Additional decision variables $m_{j,k,l}$ are defined in the following way:

$$m_{j,k,l} = \begin{cases} 1 & \text{At least one student from school } l \text{ has a BM at venue } j \text{ on date } k \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

In this work, students are indexed by the order in which information was submitted. Venues are ordered arbitrarily from 1 to 3, as there are 3 venues at Sinai Temple suitable for a B'nai Mitzvah ceremony. Dates are all Saturdays, indexed by the number of weeks from the upcoming Saturday at time of optimization.

- i is the index of the i'th student
- j is the index of the j'th venue
- k is the index of the k'th date under consideration
- l is the index of the l'th school

3.2 The Constraints

Scheduling is subject to the following constraints:

1. A student cannot have their BM before their 13th Hebrew birthday. To avoid potentially violating this constraint for students born after sundown, one day is added to every 13th birthday.
2. Each venue can accommodate 0, 1, or 2 students on a given weekend, depending on the weekend.
3. Some venues are not available on some weekends.
4. Students who attend the same academic school cannot have BMs on the same day at different venues (lest the class be split)
5. Families can specify dates when they will be *unavailable*. Those requests are to be honored.
6. Twins who request a shared ceremony are to share a ceremony.

These constraints can all be articulated with the following linear constraints on $x_{i,j,k}$ and $m_{j,k,l}$ as defined in (2) and (1):

1. Let k_{b_i} be the index of the earliest Saturday on which student i can have their B'nai Mitzvah. Constraint:
$$x_{i,j,k} = 0 \ \forall \ k < k_{b_i}$$
2. Let $c_{j,k}$ be the capacity of venue j on day k. For all dates k, $c_{j,k}$ is less than the daily size capacity of venue j, which is one or two BMs. Constraint:
$$\sum_i x_{i,j,k} \leq c_{j,k} \ \forall \ j,k$$
3. If venue j is unavailable on weekend k, $c_{j,k} = 0$. The previous constraint is sufficient.

4. Let l_i be the index of the school attended by student i. The following two constraints ensures that no two schools can have students holding BMs at different venues on the same day:

$$\sum_j m_{j,k,l} \leq 1 \ \forall \ k, l$$
$$x_{i,j,k} \leq m_{j,k,l} \ \forall \ j, k, l \ \forall \ i \text{ where } i_l = l$$

5. Let $D_i = \{k_{i,1}, k_{i,2}, \ldots, k_{i,n_{d,i}}\}$ denote the dates on which student i *cannot* have their BM, as specified by their family. The following basic constraint ensures that no student is assigned a date they requested not to have:

$$x_{i,j,k} = 0 \ \forall \ j, \ \forall \ k \in D_i, \ \forall \ i$$

6. If students i_1 and i_2 are twins requesting a shared date, then the two students are merged before optimization into one student, here $i_{1',2'}$, such that

$$D_{i_{1',2'}} = D_{i_1} \cup D_{i_2}$$

Twins' venue preferences are averaged, in case a family specified different preferences for the children.

3.3 The Objective Function

For every student, the following affects the optimality of a solution:

- The number of weekend after their 13th Hebrew birthday. More weeks after the earliest possible date is less desirable.
- The preferences families report for each venue.

The following "soft constraints" were incorporated into the objective function. That means they were not "constraints", and could be violated, but imposed a penalty on doing so.

- Specifically requested dates were made more desirable.
- Requests for students to have their own ceremony (not a shared ceremony).