# ENGSCI 355 Project 1

Andrew Jackson
Logan Wu
Scott Sung

*Department of Engineering Science*

August 17, 2017 The University of Auckland

# 1 Formulation

Our model can be conceptualised as two, linked linear programs. The first is a roster of shifts, and the second one is a cyclical network representing ward occupancies during the 42 days of the roster.

## 1.a Parameters

$$S = \{\text{A, P, N, Z, X, O}\}; \quad \text{Set of shift types}$$
$$W = \{1, \dots, 6\}; \quad \text{Set of weeks in the roster cycle}$$
$$D = \{\text{Mon}, \dots, \text{Sun}\}; \quad \text{Set of days in a week}$$

Note that for continuity, a dummy week 0 and dummy day $\text{Sun}_{\text{dummy}}$ also exist but are not part of the sets $W$ and $D$.

## 1.b Decision Variables

$X$ is the array of binary variables determining if a type of shift belongs to a given week and day in the roster:

$$x_{s,w,d} \in \{0,1\} \quad \forall s \in S,\ w \in W \cup \{0\},\ d \in D \cup \{\text{Sun}_{\text{dummy}}\}$$

$Y$ determines if a given week in the roster is the night shift week. Note the night shift also includes the final three days of the preceding week:

$$y_w \in \{0,1\} \quad \forall w \in W \cup \{\text{Sun}_{\text{dummy}}\}$$

$V$ denotes whether registrars are forced to take a weekend off:

$$v_w \in \{0,1\} \quad \forall w \in W \cup \{\text{Sun}_{\text{dummy}}\}$$

## 1.c Constraints

Create a dummy week 0 that is equal to the final week.

$$x_{s,0,d} = x_{s,|W|,d} \quad \forall s \in S,\ d \in D \tag{1}$$

Create a dummy day 0 that is equal to Sunday of the previous week, allowing wrap-around from Sunday to Monday.

$$x_{s,w-1,|D|} = x_{s,w,0} \quad \forall s \in S,\ w \in W \tag{2}$$

Ensure every slot has a shift assigned by summing over all shift types, except for the night-shift week which must have two.

$$\sum_{s \in S} x_{s,w,d} - y_w = 1 \quad \forall w \in W,\ d \in D \tag{3}$$

Every day must have a single registrar assigned to each A, P and N shift.

$$\sum_{w \in W} x_{s,w,d} = 1 \quad \forall s \in \{\text{A, P, N}\},\ d \in D \tag{4}$$

Every P shift must follow an A shift, except for Sunday where an A must follow.

$$x_{\text{P},w,d} = x_{\text{A},w,d-1} \quad \forall w \in W,\ d \in D \tag{5}$$
$$x_{\text{A},w,d} = x_{\text{A},w,d-1} \quad \forall w \in W,\ d \in \{\text{Sun}\} \tag{6}$$

Setting up the night shift: only one week can be the full 'night shift' week.

$$\sum_{w \in W} y_w = 1 \tag{7}$$

The Friday to Sunday before the full night shift week are also night shifts.

$$x_{\text{N},w-1,d} - y_w = 0 \quad \forall w \in W,\ d \in \{\text{Fri, Sat, Sun}\} \tag{8}$$

The Monday to Thursday of the full night shift week are night shifts.

$$x_{\text{N},w,d} - y_w = 0 \quad \forall w \in W,\ d \in \{\text{Mon, Tue, Wed, Thu}\} \tag{9}$$

The Friday to Sunday of the full night shift week are sleep shifts.

$$x_{\text{Z},w,d} - y_w = 0 \quad \forall w \in W,\ d \in \{\text{Fri, Sat, Sun}\} \tag{10}$$

Ensure no one else has a sleep shift and is slacking off!

$$\sum_{w \in W,\ d \in D} x_{\text{Z},w,d} = \text{n.o. allowed rests (i.e., 3)} \tag{11}$$

To give people weekends off: no weekdays are allowed to be taken off:

$$x_{\text{X},w,d} = 0 \quad \forall w \in W,\ d \in \{\text{Mon}, \ldots, \text{Fri}\} \tag{12}$$

Weekends must be taken off if scheduled as a 'weekend off' (but may be taken off on other weekends):

$$x_{\text{X},w,d} \geq 0 \quad \forall w \in W,\ d \in \{\text{Sat, Sun}\} \tag{13}$$

No two consecutive weekends can pass without a weekend off being forced.

$$v_w + v_{w-1} \geq 1 \quad \forall w \in W \tag{14}$$

## 1.d Objective Parameters

There are several new definitions that are particular to this section of the formulation.

$$WA = \{\text{lime, navy, yellow}\}; \quad \text{Set of wards}$$
$$R = W \times D; \quad \text{Set of all days in the roster, to simplify } r+1 \text{ and } r-1 \text{ subscripts}$$
$$\rho = \text{Poisson constant for discharges}$$
$$s_{wa,re} \in \{1, \ldots, |W|\}, \quad re \in \{1, \ldots, n_{\text{registrars}}\}$$

S is the vector of starting weeks for each registrar, where $n_{\text{registrars}}$ is the number of registrars per ward. S is not a decision variable of the IP; rather, it is enumerated by a tree with 15 unique combinations.

## 1.e Objective-Related Variables

A is a unimodular matrix determining whether a ward is admitting on a certain day.

$$a_{wa,r} \in \{0,\ 1\} \quad \forall wa \in WA,\ r \in R$$

O is a matrix of each ward's mean expected occupancy for each day in the roster.

$$o_{wa,r} \geq 0 \quad \forall wa \in WA,\ r \in R$$

Ṗ is the mean daily admission rate for each day of the week.

$$\dot{p}_d \in Z^+ \quad \forall d \in D$$

Δ is the maximum pairwise difference between all wards for a given day of the roster.

$$\delta_r \geq 0 \quad \forall r \in R$$

## 1.f Objective Constraints and Objective Function

These constraints calculate the expected mean occupancy of each ward throughout the 42-day roster. % is the modulo function and $\lceil x \rceil$ rounds up to the nearest integer.

Control whether a ward is admitting on a given day of the 6-week cycle. The absolutely insane subscripts adjust for starting week offsets; i.e. on week one, day one, a registrar who started on week 6 will actually be looking at week six, day one of the roster.

$$a_{wa,(r+1)\%|W\times D|+1} = \sum_{re\in\{1\ldots n_{\text{registrars}}\}} x_{\mathrm{A},\lceil \frac{r+|D|\mathrm{s}_{wa,re}-1}{|D|} \rceil\%|W|+1,(r-1)\%|D|+1} \quad \forall wa \in WA,\ r \in R \tag{15}$$

Calculate the expected occupancy based on the discounted occupancy from the previous day, plus the previous day's admissions if the ward was admitting. We used a mean stay of 4.5 days to model discharges as a Poisson process with a rate of 0.22.

$$o_{wa,r\%|R|+1} = o_{wa,r} \times (1-\rho) + a_{wa,r} \times \dot{p}_{(r-1)\%|D|+1} \quad \forall wa \in WA,\ r \in R \tag{16}$$

For each day calculate the maximum pairwise difference, between the most and least occupied wards.

$$\delta_r \geq o_{wa,r} - o_{wb,r} \quad \forall r \in W\times D,\ wa \in WA,\ wb \in WA \tag{17}$$

Objective function: Minimize the maximum pairwise occupant difference, summing over all days in the roster.

$$\text{minimize} \sum_{r\in R} \delta_r \tag{18}$$

## 2 Introduction

Finally, after a lengthy formulation:

This linear program generates feasible rostering solutions with the aim of minimising disparity at any given time between the occupancy of the hospital's wards.