# An Automated Employee Timetabling System

Aaron Slobodin

University of Victoria

## Introduction

The Nurse Scheduling Problem seeks to assign shifts to nurses while accounting for their preference of when they work. The challenge is to balance the nurses preference while creating a feasible schedule. This NP-hard problem can be solved with the following basic Integer Linear Program (ILP).

## Basic Model

We define our variables:

$$X_{n,s} = \begin{cases} 1 & \text{if nurse } n \text{ is } \textbf{working} \text{ shift } s, \\ 0 & \text{if nurse } n \text{ is } \textbf{not working} \text{ shift } s. \end{cases}$$

We construct contraints to ensure the correct number of nurses work during each shift. Let $N$ be the set of all nurses. For each shift $s$ in the schedule period:

$$\sum_{n \in N} X_{n,s} = \text{ Required number of nurses during shift } s. \tag{1}$$

We define our preference coefficients:

$$P_{n,s} = \text{Preference of nurse } n \text{ to work shift } s.$$

Let $S$ be the set of all shifts. By maximizing

$$\sum_{n \in N} \sum_{s \in S} X_{n,s} \cdot P_{n,s}$$

subject to (1) we have solved the basic NSP.

## Extending the Model

In [HSB18], the authors provide a generalized extension of this model which adapts to larger business by allowing for:

1. Multiple roles
2. Flexible work hours
3. Fine grain employee preference in terms of working days and shifts

We present an implementation of this work at the retailer, Mountain Equipment Co-op Victoria (MEC). We extend the work in [HSB18] by

1. Creating a bonus system to assign employees consecutive days off
2. Extend their implementation from small business (< 30 employees) to large business (> 100 employees).

## Variables

Let $E$, $D$, $S$, and $R$ be the set of employees, days, shifts, and roles in the schedule period, respectively. We define our variables:

$$X_{e,d,s} = \begin{cases} 1 & \text{if employee } e \text{ is } \textbf{working} \text{ on day } d \text{ during shift } s, \\ 0 & \text{if employee } e \text{ is } \textbf{not working} \text{ on day } d \text{ during shift } s. \end{cases}$$

## Constraints

**Constraint 1.** (Maximum hours per schedule) Let $HR_e$ be the maximum number of hours that employee $e$ wishes to work. Let $hour(s)$ return the length of shift $s$ in hours. For each $e \in E$:

$$\sum_{d \in D} \sum_{s \in S} hour(s) \cdot X_{e,d,s} \leq HR_e. \tag{2}$$

**Constraint 2.** (Status employees are guaranteed hours) If employee $e$ has full-time status they are guaranteed 37.5 hours:

$$\sum_{d \in D} \sum_{s \in S} hour(s) \cdot X_{e,d,s} \geq 37.5 \tag{3}$$

If employee $e$ has part-time status they are guaranteed 15 hours:

$$\sum_{d \in D} \sum_{s \in S} hour(s) \cdot X_{e,d,s} \geq 15 \tag{4}$$

**Constraint 3.** (Employees can be scheduled at most 5 days a week) For each $e \in E$:

$$\sum_{d \in D} \sum_{s \in S} X_{e,d,s} \leq 5. \tag{5}$$

**Constraint 4.** (Labour laws) Employees are required to have 10 hours off between consecutive shifts. Let $start(s)$ and $end(s)$ return the start and end time of shift $s$, respectively. For all pairs of shifts $s_i$ and $s_j$, **if** $start(s_i) + (24 - end(s_j)) < 10$, for each $e \in E$ and $d \in D$:

$$X_{e,d,s_i} + X_{e,d+1,s_j} \leq 1. \tag{6}$$

**Constraint 5.** (Employees are only assigned shifts they can work) For each $r \in R$, if employee $e$ is not trained to work $r$, then for each $s \in S$ requiring $r$:

$$\sum_{d \in D} X_{e,d,s} = 0. \tag{7}$$

**Constraint 6.** (Employees can work up to one shift a day) For each $e \in E$ and $d \in D$:

$$\sum_{s \in S} X_{e,d,s} \leq 1. \tag{8}$$

**Constraint 7.** (Correct number of employees per shift) Let $num(s, d)$ return the number of employees needed to work during shift $s$ on day $d$. For each $s \in S$ and $d \in D$:

$$\sum_{e \in E} X_{e,d,s} = num(s, d). \tag{9}$$

## Bonus System for Consecutive Days Off

We define our variables:

$$Y_{e,d} = \begin{cases} 1 & \text{if employee } e \text{ has consecutive days off starting on day } d, \\ 0 & \text{otherwise.} \end{cases}$$

We create the following constraints to ensure that an employee $e$ is either working day $d$ or $d+1$, or has consecutive days off starting on day $d$. For each $e \in E$ and $d \in D$:

$$\sum_{s \in S} X_{e,d,s} + Y_{e,d} \leq 1, \quad \sum_{s \in S} X_{e,d+1,s} + Y_{e,d} \leq 1, \quad \sum_{s \in S} X_{e,d,s} + X_{e,d+1,s} + Y_{e,d} \geq 1. \tag{10}$$

## Coefficients

We define coefficients to take into account employees preference of when they work and differences in seniority between employees:

$$P_{e,d} = \text{Preference of employee } e \text{ to work on day } d.$$

$$S_{e,s} = \text{Seniority of employee } e \text{ to work the roles required for shift } s.$$

## Optimization

Our problem is solved by maximizing the following objective function subject to (2, 3, 4, 5, 6, 7, 8, 9, 10):

$$\sum_{e \in E} \sum_{d \in D} Y_{e,d} + \sum_{s \in S} X_{e,d,s} \cdot P_{e,d} \cdot S_{e,s}.$$

## Implementation

We implemented this ILP at MEC Victoria in August 2019. Our system creates optimal employee schedules based on employee preference while taking into account:

1. Vacation requests
3. Predesignated shifts
2. Multiple role types
4. Employees preference for consecutive days off

The ILP uses the open-source COIN-OR-CBC solver [CO17]. With

$$|E| = 110, \ |D| = 7, \ |S| = 140, \text{ and } |R| = 60$$

the ILP runs with +1,250,00 constraints and +125,000 variables. The solver typically runs in under 2 minutes on a 2012 4GB MacBook Pro with 2.3 GHz Intel Core i7 processor.

**Take Home:** The ILP has successfully produced weekly staff schedules at MEC Victoria since August 2019.

## References

[CO17] COIN-OR, *CBC: A COIN-OR integer programming solver*, `https://projects.coin-or.org/Cbc`, 2017, [Online; accessed 10-November-2019].

[HSB18] Richard Hoshino, Aaron Slobodin, and William Bernoudy, *An automated employee timetabling system for small businesses*, Proceedings of the 30th IAAI Conference on Artificial Intelligence (2018), 7673–7679.

## Acknowledgements

I would like to thank my co-workers at MEC for presenting this problem, Dr. Richard Hoshino and William Bernoudy for their initial work in [HSB18], and the University of Victoria and CMS for providing me funding and opportunity to attend this conference.

**Contact Author:** aslobodin@uvic.ca