Abstract

1 Introduction

On-call schedules for a fixed number of health-care providers are central to the efficient running of hospitals. Hospital departments provide services where patient needs, and thus the system's demands, often exceed the available supply. For example, it is important that a hospital department allocates its resources, such as the availability of a finite number of clinicians, optimally, to ensure the best possible service for its patients. Carefully allocated on-call schedules are meant to simultaneously ensure sufficient resources are provided to patients while not overworking clinicians to prevent costly mistakes [ref]. It is common practice for on-call schedules to be created manually. Yet manually-created schedules are prone to errors and potential for biases [ref]. First, when there is a large number of clinicians in a single department, or the constraints that need to be satisfied by the department are very complex, a manual method may not provide an optimal schedule. Second, such methods are likely to overlook certain constraints that must be maintained to have an operational department, such as XXXX. Third, manual scheduling is often time-consuming for the person developing the schedule. For these reasons, it is important to develop automated methods that can generate optimal schedules that satisfy the given constraints of the hospital department.

Automated methods to optimize schedules have been studied and applied in many industries, including transportation [??], manufacturing [??], [...]. Of special interest to a clinician on-call scheduling problem are the approaches to schedule nurses, who often work in shifts. In the nurse scheduling problem, the goal is to find an optimal assignment of nurses to shifts that satisfies all of the hard constraints, such as hospital regulations, and as many soft constraints as possible, which may include nurse preferences. A wide variety of approaches, including exact and heuristic approaches, have been used to solve the nurse scheduling problem: integer linear programming [??], network flows [??], genetic algorithms [??], simulated annealing [??], and artificial intelligence [??].

[...] Many of these approaches were designed to satisfy the requirements of a specific hospital department which causes a large number of variables and constraints to be incorporated into the problem formulation. While these department-specific approaches allow end-users to find precise schedules that satisfy the needs of the department and the preferences of the nurses and clinicians in that department, they are difficult to readily adapt to other departments in the same hospital or other hospitals. Moreover, the large number of variables and constraints also leads to computational complexity issues [ref], especially when using exact methods for finding the solution. In this paper, we tackle a version of the nurse scheduling problem arising from a case study of one clinical division, providing two different services simultaneously (general infectious

disease (ID) consults; and HIV consults service) at St. Michael's Hospital in Toronto, Canada. Our goal is to (1) present a simple formulation for the problem developed and tested at the hospital after switching from a manual approach to scheduling; and (2) analyze the performance of integer linear programming in solving difficult instances of the problem and compare the results with those of the manual approach; and (3) describe the adaptability of the formulation as a basic framework for solving similar problems in other departments.
We begin by describing the problem, then...

[...]

2 Problem

At St. Michael's Hospital, the division of infectious diseases (ID) offers general ID and HIV consultation on inpatients as two parallel services. Each service provides clinical care throughout the year, during regular work weeks as well as weekends and holidays. The clinicians in the division typically receive a schedule in advance, outlining their on-call service dates for the full year. In the yearly schedule each clinician is assigned to blocks of regular work weeks and weekends. Each block corresponds to two consecutive work weeks. Apart from long (holiday) weekends, a work week starts on Monday at 8 A.M. and ends on Friday at 5 P.M. Accordingly, weekend service starts on Friday at 5 P.M., and ends on Monday at 8 A.M. During the weekend, ID and HIV consultation services are combined and provided by one clinician. During the regular work week the ID and HIV services are led by one clinician each.

Several constraints are placed on the on-call assignments. First, each clinician has limits on the number of blocks they can and must work during the year, depending on the type of consultation. For instance, a clinician might have to provide 3-5 blocks of general ID consultation as well as 2-3 blocks of HIV consultation throughout the year. These limits may change from year to year as the number of clinicians in the department changes. Moreover, the schedule should not assign a clinician to work for two blocks or two weekends in a row. In addition, the schedule should distribute both regular and holiday weekends equally among all clinicians.

In addition to maintaining a balanced work load among clinicians, the schedule should also accommodate their preferences. Clinicians provide their requests for time off ahead of schedule generation so that the requests may be integrated into the schedule. Individuals may specify days, weeks or weekends off, with the understanding that any blocks overlapping with their request will be assigned to a different clinician where possible. For example, if a clinician only requests a given Monday and Tuesday off, the schedule will generally avoid assigning the entire block to that clinician. Clinicians also prefer to have their weekend and block assignments close together, so the schedule should account for this when distributing assignments. A summary of the outlined constraints is given in Table 1.

Constraint Name	Description
Block Coverage	each division needs to have exactly one clinician that covers any given block
Weekend Coverage	every weekend needs to have exactly one clinician that covers it
Min/Max	in a given division, each clinician can only work between the minimum and maximum number of allowed blocks
No Consecutive Blocks	any clinician should not work two consecutive blocks, across all divisions
No Consecutive Weekends	any clinician should not work two consecutive weekends
Equal Weekends	weekends should be equally distributed between clinicians
Equal Holidays	long weekends should be equally distributed between clinicians
Block Requests	each clinician can request to be off service during certain blocks throughout the year
Weekend Requests	each clinician can request to be off service during certain weekends throughout the year
Block-Weekend Adjacency	the block and weekend assignments of a given clinician should be adjacent

Table 1: Summary of the constraints for the clinician scheduling problem

In most scheduling problems, the constraints can be divided into hard and soft constraints. Hard constraints must be satisfied by any candidate solution, while soft constraints can be used to select a more favourable solution from all candidate solutions [ref]. Typically, soft constraints are encoded as an objective that needs to be maximized or minimized, rather than a constraint. In the case of the clinician scheduling problem, we chose Block Requests, Weekend Requests and Block-Weekend Adjacency as our soft constraints, while the rest of the constraints are hard. Though it is important to take clinician requests into account when constructing the schedule, it is crucial to that the work-load of the schedule is balanced among all clinicians, and the needs of the patients are fulfilled.

It is important to note that - irrespective of how the schedule is generated - clinicians may exchange certain weeks or days throughout the year after the schedule is implemented. The approach to solving the clinician scheduling problem does not account for these future exchanges, and only focuses on the full year time horizon.

3 Methods

In this section, we present an application of 0-1 Linear Programming to solve the clinician scheduling problem presented in Section ??. First, we describe the sets, indices and variables present in the formulation of the program. Later we convert the constraints given in Table 1 into mathematical terms, and outline the objective function of the linear program.

3.1 Sets and Indices

We denote the set of all services/divisions that clinicians in the department can provide as $\mathcal{D}$. It is important to note that in the following formulation, it is assumed that all clinicians are able to provide all services in this set, although this may not be true in practice. The set of all clinicians in the department is denoted as $\mathcal{C}$. The sets of blocks and weekends that clinicians will be assigned to are denoted as $\mathcal{B}$ and $\mathcal{W}$ respectively. Note that the size of a block is not constrained, and can be adapted to the needs of the given department. A subset of weekends are denoted $\mathcal{L}$, corresponding to the long/holiday weekends throughout the year. Lastly, we denote with $\mathcal{BR}_c$ and $\mathcal{WR}_c$ the block and weekend requests of clinicians as subsets of all blocks and weekends, respectively. For instance, if clinician c's requests intersect with blocks 1 and 2, and weekend 1, then $\mathcal{BR}_c = \{1, 2\}$ and $\mathcal{WR}_c = \{1\}$. Table 2 presents a summary of the sets and indices described.

Set	Index	Description
$\mathcal{D} = \{1, \ldots, D\}$	d	services/divisions
$\mathcal{C} = \{1, \ldots, C\}$	c	clinicians
$\mathcal{B} = \{1, \ldots, B\}$	b	blocks
$\mathcal{W} = \{1, \ldots, W\}$	w	weekends
$\mathcal{L} \subset \mathcal{W}$		long weekends
$\mathcal{BR}_c \subset \mathcal{B}$		block requests of clinician c
$\mathcal{WR}_c \subset \mathcal{W}$		weekend requests of clinician c

Table 2: Description of sets and indices in the problem

3.2 Variables

Since each clinician may be assigned to work for any service, during any block of the year, we denote such an assignment as $X_{c,b,d}$. A value of 1 indicates that the given clinician c is assigned to cover division d during block b. Similarly, we define weekend assignments as $Y_{c,w}$, although these are not indexed by service, as clinicians are expected to provide all services during the weekends. The optimization of the soft constraint Block-Weekend Adjacency is done by maximizing the product $X_{c,b,d} \cdot Y_{c,w}$ for adjacent blocks and weekends. In order to formulate such an objective as a linear function of variables, we introduce another set of variables, denoted by $Z_{c,b,d}$, with additional constraints on its range. Further details regarding this variable are described in Sections **??** and **??**. Lastly, we introduce a set of constants $m_{c,d}$ and $M_{c,d}$ to constrain the number of blocks each clinician is allowed to work during the year. Table 3 presents a summary of the constants and variables in the problem.

Name	**Description**
$X_{c,b,d} \in \{0,1\}$	assignment of clinician c for service/division d on block b
$Y_{c,w} \in \{0,1\}$	assignment of clinician c on weekend w
$Z_{c,b,d} \in \{0,1\}$	helper variable for optimizing Block-Weekend adjacency
$m_{c,d}$	minimum number of blocks clinician c should cover for division d
$M_{c,d}$	maximum number of blocks clinician c should cover for division d

Table 3: Description of variables and constants in the problem

3.3 Constraints

We now present the hard constraints in Table 1 in light of the mathematical setup given above.

$$\sum_{c=1}^{C} X_{c,b,d} = 1 \qquad \forall b \in \mathcal{B}, d \in \mathcal{D} \qquad \text{(Block Coverage)}$$

$$\sum_{c=1}^{C} Y_{c,w} = 1 \qquad \forall w \in \mathcal{W} \qquad \text{(Weekend Coverage)}$$

$$m_{c,d} \leq \sum_{b=1}^{B} X_{c,b,d} \leq M_{c,d} \qquad \forall\ c \in \mathcal{C}, d \in \mathcal{D} \qquad \text{(Min/Max)}$$

$$X_{c,b,d} + X_{c,b+1,d} \leq 1 \qquad \forall c \in \mathcal{C}, b \leq B-1, d \in \mathcal{D} \qquad \text{(No Consecutive Blocks)}$$

$$Y_{c,w} + Y_{c,w+1} \leq 1 \qquad \forall c \in \mathcal{C}, w \leq W-1 \qquad \text{(No Consecutive Weekends)}$$

$$\left\lfloor \frac{W}{C} \right\rfloor \leq \sum_{w=1}^{W} Y_{c,w} \leq \left\lceil \frac{W}{C} \right\rceil \qquad \forall c \in \mathcal{C} \qquad \text{(Equal Weekends)}$$

$$\left\lfloor \frac{|\mathcal{L}|}{C} \right\rfloor \leq \sum_{w \in \mathcal{L}} Y_{c,w} \leq \left\lceil \frac{|\mathcal{L}|}{C} \right\rceil \qquad \forall c \in \mathcal{C} \qquad \text{(Equal Holidays)}$$

3.4 Objectives

As mentioned briefly in Section **??**, the soft constraints of the clinician scheduling problem are: (1) satisfying clinician block off requests, (2) satisfying clinician weekend off requests, and (3) assigning weekends closer to blocks. We convert these constraints into linear objective functions of the binary variables defined in Section **??**. The first two objectives have a straight-forward formulation as linear functions:

$$Q_1(X) = \sum_{c=1}^{C}\sum_{b=1}^{B}\sum_{d=1}^{D}(-1)^{\mathbb{1}(b\,\in\,\mathcal{BR}_c)} \cdot X_{c,b,d} \qquad \text{(Block Requests)}$$

$$Q_2(Y) = \sum_{c=1}^{C}\sum_{w=1}^{W}(-1)^{\mathbb{1}(w\,\in\,\mathcal{WR}_c)} \cdot Y_{c,w} \qquad \text{(Weekend Requests)}$$

where $\mathbb{1}(\cdot)$ is the indicator function that has value 1 when the condition is met and 0 otherwise. In these two objectives, we penalize any assignments that conflict with a block or weekend request, and aim to maximize the non-conflicting assignments. The Block-Weekend Adjacency is optimized by considering the product $X_{c,b,d} \cdot Y_{c,w}$ for values of w "adjacent" to the value of b. For instance, clinicians might want to be assigned during a weekend that falls within an assigned block. So if clinician c is assigned to work during block 3, corresponding to weeks 5 and 6 assuming 2-week blocks, they might also want to be assigned to work during weekend 5. In that case, we would like $X_{c,b=3,d} \cdot Y_{c,w=5}$ to be 1, since that indicates both variables are assigned. If at least one of the two variables is not assigned, the product will be 0. This leads to the obvious maximization objective

$$Q_3(X,Y) = \sum_{c=1}^{C}\sum_{b=1}^{B}\sum_{d=1}^{D} X_{c,b,d} \cdot Y_{c,w=\varphi(b)} \qquad \text{(Block-Weekend Adjacency)}$$

where $\varphi(b)$ maps a block one-to-one to an adjacent weekend, by some appropriate definition of adjacency. For instance, in the above example we will have $\varphi(b) = 2b - 1$.

However, as it is, Q_3 is not a linear function of its variables and cannot be optimized in a linear programming framework. The general approach used to convert such functions into linear objectives is by introducing a helper variable and additional constraints [ref **??**]. In our case, introducing a variable $Z_{c,b,d}$ for every product $X_{c,b,d} \cdot Y_{c,w}$ with $w = \varphi(b)$, and constraining Z such that

$$Z_{c,b,d} \leq X_{c,b,d} \tag{1}$$

$$Z_{c,b,d} \leq Y_{c,w=\varphi(b)} \qquad \forall d \in \mathcal{D} \tag{2}$$

allows us to rewrite Q_3 as a linear function of Z,

$$Q_3(Z) = \sum_{c=1}^{C}\sum_{b=1}^{B}\sum_{d=1}^{D} Z_{c,b,d} \tag{3}$$

Indeed, whenever $X_{c,b,d} \cdot Y_{c,w} = 1$, $Z_{c,b,d}$ can attain a maximum value of 1, and whenever $X_{c,b,d} \cdot Y_{c,w} = 0$, at least one of $X_{c,b,d}$ or $Y_{c,w}$ must be 0, so $Z_{c,b,d}$ will be constrained to attain a maximum value of 0, giving us the correct adjacency maximization objective.

As we can see, these objectives turn our clinician scheduling problem into a multiple objective optimization problem. The most common approach to solving such problems is by optimizing a weighted sum of the normalized objective functions, as this guarantees the optimal solution to be Pareto optimal [ref **??**]. This is the approach we decided to use in our problem, to ensure all three objectives are considered when finding a solution. Under the assumption that each clinician in $\mathcal{C}$ provides all types of services in $\mathcal{D}$, the normalized objectives can be written as follows,

$$\bar{Q}_1(X) = \frac{Q_1(X)}{C \cdot B \cdot D} \tag{Block Requests}$$

$$\bar{Q}_2(Y) = \frac{Q_2(Y)}{C \cdot W} \tag{Weekend Requests}$$

$$\bar{Q}_3(Z) = \frac{Q_3(Z)}{C \cdot B \cdot D} \tag{Block-Weekend Adjacency}$$

where we divided each of the original objective functions by the sum of the absolute values of its coefficients [ref **??**]. The final weighted objective is given by

$$\alpha\bar{Q}_1(X) + \beta\bar{Q}_2(Y) + (1 - \alpha - \beta)\bar{Q}_3(Z) \tag{4}$$

with $0 \leq \alpha, \beta \leq 1$.

[...]

4 Results

5 Simulations

6 Discussion