Physician Scheduling in a Gynaecology Department

MSc Dissertation

S.T. Luen-English

Department of Mathematics
Cardiff University

Decemeber 2015

Outline

Outline

Introduction

Figure 1 : University of Twente, Enschede

(a)

(b)

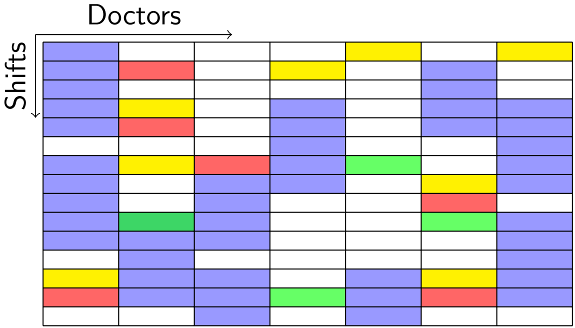
Figure 2 : Jeroen Bosch Ziekenhuis

Introduction

- Personnel scheduling problem found in the Gynaecology Department in Jeroen Bosch Hospital, 's-Hertogenbosch
- Problem of timetabling doctors: specifying both what shifts they work, but also what task they will be doing in that shift
- Currently done by hand, taking 2-3 days to schedule the next 6 weeks

Outline

Introduction



- Both essential and desired properties that a schedule must meet
  - Relating to individual doctors work patterns e.g. days off, contracted hours, ‘fair’ proportion of undesirable tasks etc...
  - Relating to overall scheduling of task e.g. adequate clinics scheduled each month, always two doctors on-call etc...

Similar Problems

- Nurse Rostering Problem
- (Master) Physician Scheduling Problem
- Employee Timetabling Problem

Outline

- General linear program formulation:

$$\begin{aligned} \text{Min} \quad & \mathbf{c}^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

- Decision variable:

$$x_{i,j,s,t} = \begin{cases} 1 & \text{if doctor } i \text{ does task } t \text{ on shift } s \text{ on day } j \\ 0 & \text{otherwise} \end{cases}$$

- Can easily express problem in terms 'hard' and 'soft' constraints; for example:

$$\sum_{t \in T} x_{i,j,s,t} \leq 1 \quad \forall\, i \in I, j \in J, s \in S$$

To ensure a doctor can only be assigned to one task per shift

Objective Function

Weighted sum of soft constraint violations, for example:

- Under-scheduling of tasks weekly
- Under and over scheduling of each doctor with respect to contract:
- Instances when a doctor has ‘holes’ in schedule
- Tasks being unfairly distributed among doctors

Solution Progress

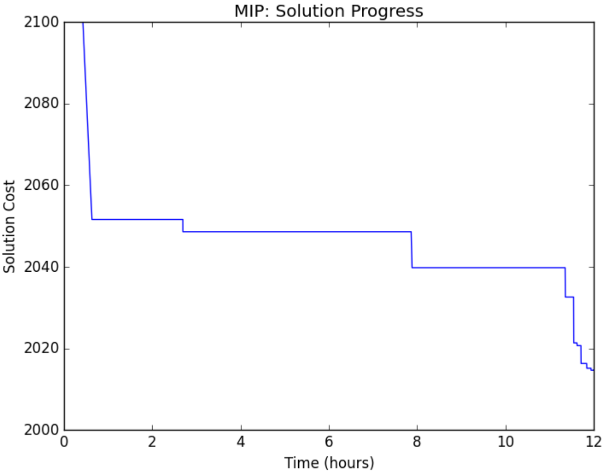


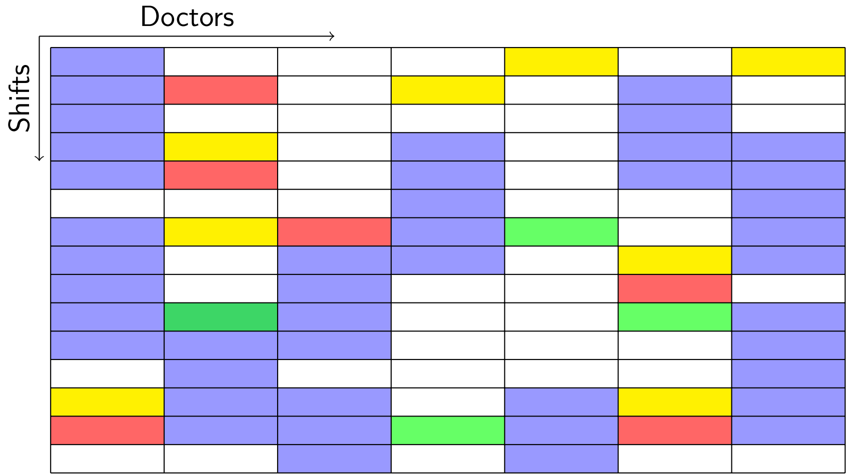
Figure 3 : 8 Week Planning Horizon: Solution Progress

Outline

Local Search Framework

- Starting from some initial solution, repeatedly attempt to improve solution through some small change
- Move operators:
  - Schedule task
  - Delete scheduled task
  - Swap 2 shift assignments in a week
  - Change doctor performing a on-call sequence
  - Swap between doctors performing on-call tasks

Move Operators



Simulated Annealing

- Local search likely to get stuck in ‘local optimum’
- Simulated annealing aims to overcome this by probabilistic acceptance of worsening moves
  - At beginning of search, high probability of accepting worsening moves
  - Probability decreases as search continues

Implementation

- Mixed Integer Linear Program implemented in AIMMS 4.6 and solved using CPLEX 12.6
- Repeated Local Search and Simulated Annealing where implemented in Python programming language
  - Repeated Local Search ran multiple search processes simultaneously over multiple CPU cores

Outline

Results

Table 1 : Results Summary

	4 Week			8 Week			12 Week		
Case	**MIP**	**SA**	**RLS**	**MIP**	**SA**	**RLS**	**MIP**	**SA**	**RLS**
0	1386	1476	1490	2203	2393	2380	114411	3393	3365
1	1336	1376	1441	2136	2249	2401	2938	3493	3273
2	1322	1483	1501	2123	2486	2454	110421	3351	3458
3	1358	1525	1489	2383	2502	2512	105894	3508	3411

Remarks

- In almost all cases the MIP was the best performer
- Local search algorithms could be improved through the use of ‘faster’ programming language
- Quantification of solutions is subjective; is beneficial to give decision maker a choice of ‘good solutions’: something local search algorithms can do well

The End

Thanks for listening!

Luen-EnglishST@cardiff.ac.uk

www.github.com/sluenenglish/Presentations