Healthcare Systems Process Improvement

Conference 2015

Sponsored by:

Society for Health Systems

Leading Healthcare Improvement

CENTER FOR
HEALTHCARE ENGINEERING & PATIENT SAFETY
UNIVERSITY OF MICHIGAN

Customization vs. Convenience When Developing Healthcare Scheduling Tools

Amy Cohn
University of Michigan
amycohn@med.umich.edu
CHEPS.engin.umich.edu

SHS Conference Orlando
February 2015

MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

Twitter

- **Driving Value in Healthcare through Leadership and Education**
- Key Learning: What waste do you see on a day to day basis that makes you wacky?
- #SHS2015 #SHSwackywaste
- @UofMCHEPS @ProfAmyCohn

Acknowledgments

- Several years of excellent students and stellar clinical collaborators
- Current Chief Resident: Ed O'Brien
- Current students: Young-Chae Hong, Billy Pozehl, Peter Mayoros, Ji Wang, Elizabeth Olin, Zak VerSchure, Brian Lemay, Yicong Zhang, Brittany Lopez
- Financial and moral support: The Bonder Foundation, The Doctors Company Foundation, CHEPS,UCH, UMHS (Pediatrics and Surgery), UM CoE SURE Program

Talk Overview

- What is healthcare provider scheduling and why is it challenging?
- Why are good schedules important?
- How are schedules typically built today?
- How can optimization tools help?
- “Customization vs. convenience”
- Conclusions, questions, and discussion

An Example:
Peds ED Shift Scheduling

- Assigning residents to shifts to cover the pediatric emergency department in Mott Children’s Hospital at UMHS
- Month-long schedule
- Approximately 15 residents per month, coming from four or five different residency programs

An Example: Peds ED Shift Scheduling

- Patient care requirements:
  - 7 overlapping shifts every day of the month
  - Every shift has to have exactly one resident assigned
  - Exceptions: 12p – 9p shift coverage is optional
    - Not *all* of these shifts can be left uncovered for the entire month
  - Certain shifts cannot be assigned to an intern
  - Certain overlapping pairs of shifts require a Peds resident on at least one of the two shifts
  - …

An Example:
Peds ED Shift Scheduling

- Resident availability
  - Senior residents switch on the first of the month
  - Interns switch on the 27th of the preceding month
  - Pre-assigned vacation time must be respected
  - Continuity clinics/post CC
  - Some shifts are pre-assigned to certain residents/programs
  - 10-hour rest rule
  - First and last shifts must recognize boundaries of other rotations
  - …

What is healthcare provider scheduling and why does it matter?

- Assigning providers (nurses, residents, attendings…) to:
  - Times
  - Places
  - Tasks
- So as to meet:
  - Patient needs (quality and continuity)
  - Provider needs (short term satisfaction, long-term pipeline)
  - Educational needs (for residents) – short- and long-term impact on patient care

Why is it difficult?

- Many complex and inter-dependent rules
- Tightly constrained resources
- Heterogeneity of providers, clinical environments
- Multiple competing goals
- Large-scale combinatorial optimization problem with ill-defined, multi-criteria objective function

How are schedules typically built today?

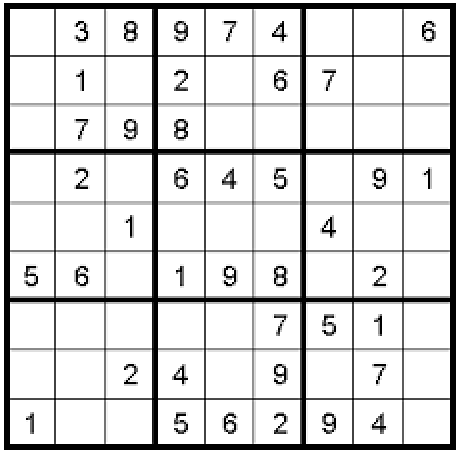
- “Computerized”
  - Mainly computerization of data entry
  - Some error checking
  - Good for disseminating, ties to payroll
  - Not so good for schedule creation except when very rigid (e.g. “every fourth night”)

How are schedules typically built today?

- Manually
  - Senior nurse schedules other nurses
  - Designated attending schedules other attendings
  - Chief Resident schedules other residents
- Yields power but also conflict
- Time consuming
- Poor use of skills and resources
- Reduced quality of solutions

How can optimization tools help?

- Core of scheduling is like a giant sudoko
- When solving manually, if you find an error, you have to re-start from scratch or allow rule violations
- But rules can inherently be modeled as sequences of simultaneous equations



	3	8	9	7	4			6
	1		2		6	7		
	7	9	8					
	2		6	4	5		9	1
		1				4		
5	6		1	9	8		2	
					7	5	1	
		2	4		9		7	
1			5	6	2	9	4	

General Mathematical Approach

- These are all combinatorial optimization problems
  - Set of decisions to make (assigning people to places and times)
  - Set of constraints to ensure acceptability
  - Objective function / metrics
- Binary assignment variables
- Linear constraints

General Mathematical Approach

- E.g. “x_{rsd} = 1” translates to “Resident *r* is assigned to work shift *s* on day *d*”
- How would we say “Resident *r* has to work exactly *n* shifts?

General Mathematical Approach

- E.g. “$x_{rsd} = 1$” translates to “Resident *r* is assigned to work shift *s* on day *d*”
- How would we say “Resident *r* has to work exactly *n* shifts?

$$\Sigma_{s \text{ in } S} \Sigma_{d \text{ in } D} x_{rsd} = n$$

Our Projects

- Scheduling
  - Call schedule for Psych residents at BUSM
  - Block schedule for Peds residents at CHM
  - Shift schedule for pediatric ED at UMHS
  - Block schedule for UMHS peds residents
  - Block schedule for UMHS surgical residents
  - OR/clinic time for UCH surgical attendings
- Optimization-based software in C++/Cplex

Key Issues

- Importance of true collaboration to define functionality
  - Surprisingly hard for people to articulate their requirements
  - Frustrating and time consuming to try to build the perfect system first time out
  - Iterative, incremental approach much more effective

Key Issues

- Importance of variable definition to ensure a viable model
  - Balance of information in the variables vs in the constraints
  - Shift assignments vs service pairs vs sequences vs templated sequences

Key Issues

- How to address multi-criteria objective function
  - Challenges with weights
    - Hard to quantify
    - Non-linearities
    - Impact on performance
  - What do people really want? Surprise – it's *not* optimality!
  - Interactive, incremental improvement approach

“Customization vs. Convenience”

- Importance of flexibility:
  - New rules and requirements arise all the time
  - Sometimes significant (7 shifts a day becomes 8; Christmas holidays treated completely differently from the rest of the year)
  - Sometimes small but critical (resident coming off maternity leave needs a specific set of shifts to complete her program)
  - Whenever possible, want to avoid new design, new code

“Customization vs. Convenience”

- Challenges of customization
  - The more general you are, the more complex the data entry and parameter setting
  - The more general you are, the less likely you are to have easy convergence
  - The more general you are, the harder it is to maintain the code

“Customization vs. Convenience”

- In moving from one project to the next we often find ourselves re-doing the same work
- We want to find the sweet spot between maintaining multiple separate programs and making a single program that is too complex/doesn’t meet any one program’s needs
- Generalization approach: an adaptable framework that uses input files and flags to turn on and off different functionality

"The Team, The Team, The Team"

- Importance of continuing collaborations:
  - We don't ever want to reach the point where we hand it off (or shrink wrap it) and walk away
  - We learn too much from working with the clinicians!
    - Continuously improving the tools
    - Identifying new problems to work on
    - Educating them in a more "engineering way" of thinking

CHEPS: Center for Healthcare Engineering and Patient Safety

Four-year old collaborative center between medicine, engineering, nursing, public health and more

Questions and Discussion