Implementing a Residency Scheduling Program at the University of Michigan Pediatric Emergency Department

Amy Cohn
University of Michigan

IFORS Barcelona
July 2014
Acknowledgements

• Ongoing collaboration with many talented contributors
  – Summer 2014 Team
    • UMHS Chief Resident Ed O’Brien
    • Students: Young-Chae Hong, William Pozehl, Zak VerShure, Peter Mayoros, Ji Wang, Jared Kott, Nina Scheinberg, Aaron Cohen, Hannah Schapiro, Daniel Hazlett
BRIEF BACKGROUND ON RESIDENCY
What is medical residency?

- Transition period between medical school and fully independent/unsupervised practice
  - Four years of med school
  - First year of residency – “Intern”
  - Two more years of residency
  - Possibly one or two additional years as “Chief Resident”
  - Possibly more years as a “Fellow”

- During all of this time, providing patient care (albeit with the oversight of a more senior “attending” physician – supervision decreases over time)
What is medical residency?

• A key issue: Dual role of residency
  – Learning experience: Residency (and Fellowship) are parts of the medical education training process
  – Patient care: Residents/Fellows provide a significant amount of the patient care in teaching hospitals and the associated clinical system

• A typical resident might engage in all of the following activities:
  – “Continuity clinics”
  – Shifts on service
  – Seminars, formal educational activities
  – Research
Inherent Time Conflicts

• How to schedule residents’ time
  – Need adequate patient coverage with a limited pool of residents
  – Need adequate training opportunities
  – Need adequate rest – fatigue increases risk of error
  – Need to address resident satisfaction, personal life

• Not just quantity of hours but pattern
  – Continuity of care
  – Sleep issues (especially associated with overnight shifts)
  – Opportunities for different medical experiences
  – Quality of life, fatigue, and stress issues
MONTHLY SHIFT SCHEDULING OF PEDIATRIC EMERGENCY DEPARTMENT RESIDENTS
What is the general problem?

- Given a set of residents assigned to a shift-based service, build a month-long schedule that satisfies all patient care, educational, and other requirements
What is our specific problem?

- Assigning residents to shifts to cover the pediatric emergency department in Mott Children’s Hospital at UMHS
- Eight overlapping shifts per day
- Month-long schedule (but conflicting switch dates depending on the resident)
- Approximately 15 residents per month, coming from four or five different residency programs
What are the rules?

• Patient care requirements:
  – 8 overlapping shifts every day of the month
  – Every shift has to have exactly one resident assigned
  – Exceptions: 10a – 7p and 12p – 9p shift coverage is optional
    • Not *all* of these shifts can be left uncovered for the entire month
    • Ideally one of the two “flex shifts” should be covered each day
  – 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
  – ...
What are the rules?

• 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 rule
  – First and last shifts must recognize boundaries of other rotations
  – …
What was the current state?

- Schedules typically built by Chief Residents
- Limited decision support
- No formal training
- Hard to satisfy all rules
- Unlikely to make everyone happy
Why is it hard to schedule manually?

- The more squares you fill in, the fewer choices you have left for what is valid
- Once you make a mistake, you might not know it for a long time
- Once you realize something is wrong, it can be very hard to back track and correct
- If a requirement changes, you have to start from scratch
Mixed integer programming approach
- \( x_{rsd} = 1 \) if resident \( r \) is assigned to shift \( s \) on day \( d \), else \( 0 \)

Feasibility constraints are straightforward to model

Run time using C++ and CPLEX on a standard PC is minimal (a few seconds at most)

Finding a schedule that satisfies the rules is already progress over what exists (especially given time required)

But not all feasible schedules are equally good
How to “optimize”?

• No one clear objective function, but many important metrics
  – Equity across residents
    • Number of shifts
    • Number of night shifts
    • General quality of schedule
  – “Bad sleep patterns”
  – Personal requests
  – Post-continuity clinic calls
  – Flex shift coverage
  – Transition shift coverage
How to “optimize”?

• We could treat this as a multi-criteria objective function, assign weights to normalize, and solve
  – Weights are hard to find
  – Convergence can slow dramatically

• Is “optimal” the right goal??
  – Is this an engineering construct that we’re imposing inappropriately?
How to “optimize”? 

- Our approach:
  - Set boundaries on the metrics
  - Define as hard constraints
  - Search for a feasible solution
    - If found, review and decide what to tighten next
    - If not found, loosen the boundaries
  - Repeat until satisfied
Current Work: Pareto Optimization

• No established rank order of metrics
• No weights can easily be established
  – “Game” for testing Chiefs’ ability to provide appropriate weights
  – Preferences change from one month to another, e.g. when launching EMR
• Attempting to generate Pareto-dominant set of solutions
  – Given a solution, can add appropriate cut and re-solve to find a Pareto-improvements
  – Need to then “jump” to new solutions
ANNUAL BLOCK SCHEDULING OF PEDIATRIC RESIDENTS
Some months are easy to solve the ED shift scheduling problem, others very hard – why?

Often depends on number of interns for the month (affect start/end periods and ability to fill shifts) and/or overlaps in continuity clinics

Why do some months have lots of interns, others few? Why might cc’s overlap (or not)?
• Over the course of the year, residents must work on many *services*, divided in monthly *blocks*
  – Emergency medicine
  – Hematology/Oncology
  – NICU
  – Etc.
• Residents span multiple levels of seniority and come from multiple programs
Block Scheduling
Requirements and Challenges

• Requirements
  – Coverage constraints (patient care)
  – Educational constraints

• Challenges
  – Vacation requests
  – Repeating hard services
  – Split-months (e.g. ½ ED, ½ vacation)
• Again, can formulate a straightforward IP
  – “Service pairs” as templates to recognize possible half-month combinations
  – $X_{rpm}=1$ if resident $r$ is assigned to service pair $p$ during month $m$

• Constraints are fairly straightforward
• Challenge is again with the objective function
  – Trading off vacation time requests vs “triple-H” limitations
  – Trading off vacations across residents, interpreting importance of requests
  – Lots of input from Chief (acceptable), lots of tweaking of the code (not so acceptable)
ISSUES OF GENERALIZATION
Higher Level Goals

- Shift scheduling occurs in many other contexts
- Block scheduling occurs in many other contexts
- Can we generalize models, computational frameworks, file i/o, etc. in such a way to generalize, encompass multiple environments with a single tool
  - Best to group by specialty, by time frame, or overall?
QUESTIONS, COMMENTS AND DISCUSSION