A Novel Approach To Speed Up The Identification Of A Feasible Solution To The Residency Block Scheduling Problem

Junhong Guo William Pozehl Amy Cohn

Center for Healthcare Engineering and Patient Safety (CHEPS)
Department of Industrial and Operations Engineering (IOE)

University of Michigan, Ann Arbor, MI

Nov. 13, 2020

M | CHEPS

Rx

A prescription to address system complexity in healthcare

INNOVATING HEALTHCARE DELIVERY

FOSTERING LEARNING

BUILDING COMMUNITY

POSITIVE IMPACT THROUGH...

Research
Education
Implementation
Outreach
Dissemination

Introduction

Overview

- Consider a personnel scheduling problem that assigns residents in a medical school to different service-specified rotations during the academic year.
- Due to the large-scale nature and problem complexity, **constructing a feasible schedule could be extremely time consuming** by conventional approaches, even using commercial solvers.
- The latency issue may delay the deployment of the finalized schedule, since an interactive, iterative framework is used to deal with the multi-criteria objectives nature. On the other hand, it could severely impact the solution quality, which consequently could impact the quality of care patients will receive.

Block

The planning horizon is evenly divided into a specific number of time periods. Each unit time period is also referred to as a *block*.

Block

The planning horizon is evenly divided into a specific number of time periods. Each unit time period is also referred to as a *block*.

Service

A *service* is a specialty in a specific unit in the hospital, e.g. AMB, ER, ...

Block

The planning horizon is evenly divided into a specific number of time periods. Each unit time period is also referred to as a *block*.

Service

A *service* is a specialty in a specific unit in the hospital, e.g. AMB, ER, ...

Rotation

A *rotation* is specified by a service and a duration in terms of blocks.

That is, assigned residents should do the specific service for the given number of blocks consecutively.

Resident Block Scheduling

Assign each resident to a sequence of rotations during a given planning horizon, where specific requirements are satisfied.

Resident Block Scheduling

Assign each resident to a sequence of rotations during a given planning horizon, where specific requirements are satisfied.

Example

CCU	AMB	AMB	ELEC	ER	CCMU
July	August	September	October	November	December

VACA	ELEC	NICU	NICU	CCU	ER
January	February	March	April	May	June

- **Resident Education Requirements**
  - Ensure residents receiving appropriate training for fulfilling their own educational goals
  - On individual resident; typically across the whole planning horizon
  - E.g. Resident A must have at least 2 but no more than 4 blocks of AMB service during the whole acedemic year
- **Service Coverage Requirements**
  - Ensure the units staffed appropriately for providing care to patients
  - Across a group of residents; typically on a single service and a single block
  - E.g. Service ER needs to be covered by exactly 3 residents from first-year IM program, during November

Side Constraints: Service Duration Requirements; Resident Pairing Rules; Spacing Rules; Sequencing Rules; Prohibitions; Pre-assignments

Model

- The problem is formulated as an integer program
  - All binary variables
  - E.g. $x_{r,s,t}$ - Whether resident r is assigned to service s at time period t

Model

- The problem is formulated as an integer program
  - All binary variables
  - E.g. $x_{r,s,t}$ - Whether resident r is assigned to service s at time period t

- Solving the integer program for real-world instances through the conventional Branch-and-Bound process, even without any objective, takes a huge amount of time

Model

The causes of the Latency

Model

The causes of the Latency

1. Large Size
   - Approximately 250 residents; 100 services; 24 (or 26) time periods
   - Result in about 1 million decision variables and 2 million constraints

Model

The causes of the Latency

1. Large Size
   - Approximately 250 residents; 100 services; 24 (or 26) time periods
   - Result in about 1 million decision variables and 2 million constraints
2. Complexity of the Constraints
   - Unlike nurse shift scheduling
   - Educational Requirements V.S. Coverage Requirements

Model

The causes of the Latency

1. Large Size
   - Approximately 250 residents; 100 services; 24 (or 26) time periods
   - Result in about 1 million decision variables and 2 million constraints
2. Complexity of the Constraints
   - Unlike nurse shift scheduling
   - Educational Requirements V.S. Coverage Requirements

	Time1	Time2	$\cdots$	Time24
Resident1			$\cdots$	
Resident2			$\cdots$	
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
ResidentN			$\cdots$	

Model

The causes of the Latency

1. Large Size
   - Approximately 250 residents; 100 services; 24 (or 26) time periods
   - Result in about 1 million decision variables and 2 million constraints
2. Complexity of the Constraints
   - Unlike nurse shift scheduling
   - Educational Requirements V.S. Coverage Requirements

	Time1	Time2	$\cdots$	Time24
Resident1			$\cdots$	
Resident2			$\cdots$	
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
ResidentN			$\cdots$	

3. The Great Amount of "Symmetries"

Proposed Approach

Motivation

Key Observations:

- If the coupling constraints that are imposed across residents, e.g. the ones corresponding to Coverage Requirements, are removed from the formulation, the integer program can be solved in a flash.
- Some services are tough to schedule. For example, a specific service requiring exactly 5 residents, where 3 of them must be from 3rd-year IM, one is from 4th-year MP, while the other one is from 3rd-year Peds, **introduce more difficulties**, in terms of coordinating with the educational requirements, than another service which is okay to be covered by 2 to 10 residents from any level and program.

Proposed Approach

Motivation

Key Observations:

- If the coupling constraints that are imposed across residents, e.g. the ones corresponding to Coverage Requirements, are removed from the formulation, the integer program can be solved in a flash.
- Some services are tough to schedule. For example, a specific service requiring exactly 5 residents, where 3 of them must be from 3rd-year IM, one is from 4th-year MP, while the other one is from 3rd-year Peds, **introduce more difficulties**, in terms of coordinating with the educational requirements, than another service which is okay to be covered by 2 to 10 residents from any level and program.

Try to fix the assignments of those tough services to their respective correct spots, through optimizing a "sub-problem" with a very limited coupling constraints. Then, try to arrange the assignments of the rest, easier services, to complete the scheduling.

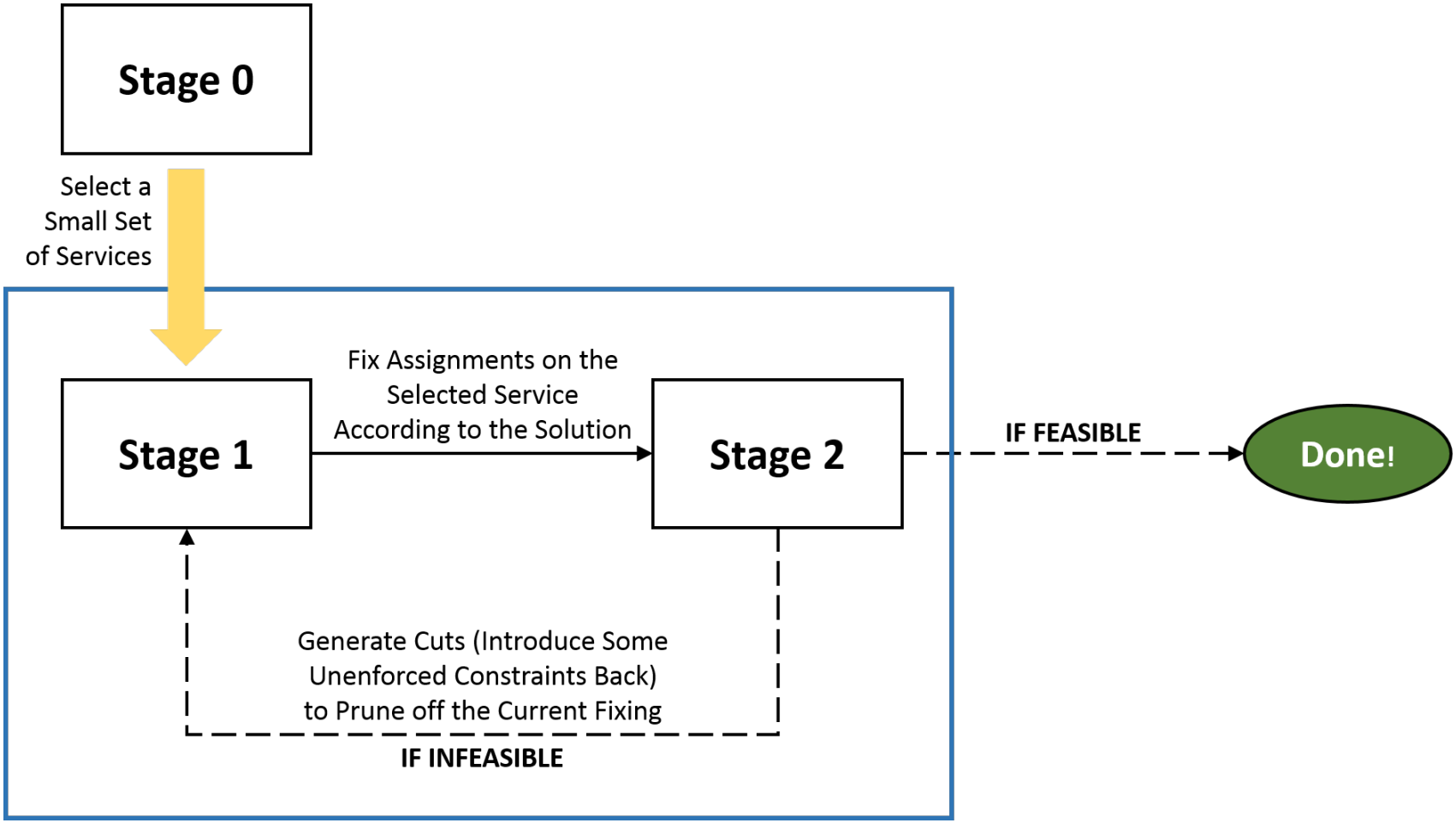
Proposed Approach

High-level Description

- **(Stage 0)** Identify the toughest services by a network-based model, which builds the relationship of different services in terms of satisfying the provided Coverage Requirements. The identification is then achieved by isolating the core nodes of the network.
- **(Stage 1)** Formulate a same integer program as the original one, but only incorporating a small subset of coverage constraints, where those selected toughest services are involved in. The objective function is designed to work as a guidance to leave as much room as possible for the assignments of the remaining services. Solve the program to get a "desirable" solution.
- **(Stage 2)** Fix the assignments of the selected services based on the solution to Stage 1 (through additional constraints), restore all unenforced coverage constraints, and try solving the updated integer program to feasibility.

Proposed Approach

High-level Description



Preliminary Experiments

- Instance Info

Residents	Services	Blocks	Colns	Rows	Runtime
249	92	24	1.2M	1.8M	1900 sec

Preliminary Experiments

- Instance Info

Residents	Services	Blocks	Colns	Rows	Runtime
249	92	24	1.2M	1.8M	1900 sec

- Results Summary (12 trials)
  - **Service Selection**: 3 - 5 among the top ones from Stage 0
  - **Config**: 3 minutes limitation with a 1% optimality gap for Stage 1
  - **Iterations**: single iteration for 10 of 12 trials; Max 3 iterations
  - **Total Time**: 500 sec in average; Min 250 sec; Max 940 Sec
  - **Improvement**: all trails achieved a better performance than the conventional approach, where the runtime is at least halved.

Future Work

- Investigation on how to more precisely isolate an effective combination of services in Stage 0
- Additional mechanisms in Stage 1 to ensure we fix the assignments of the selected services in the correct spots
- Research on different cut generation approaches when Stage 2 results in infeasibility
- Incorporation of the original objectives into the whole proposed framework, to pursue a feasible solution with better quality if possible, without impacting the computational performance.

Thank You for Your Attention

Q & A