Pediatric Emergency Department Scheduling

CENTER FOR
HEALTHCARE ENGINEERING & PATIENT SAFETY
UNIVERSITY OF MICHIGAN

Ji Wang
Dept. of Industrial and Operations Engineering

Faculty: Prof. Amy Cohn

Outline

- Background
- Methodology
- Model Description & C++ Programming
- Results

Outline

- Background
- Methodology
- Model Description & C++ Programming
- Results

Background

- C.S. Mott Children's Hospital
- Pediatric Emergency Department (PED)
- Schedules for resident physicians

- Chief Resident
  - Create schedules to assign shifts to residents
  - 25 residents
  - 35 days
  - 7 shifts
  - 20-30 hours of work

- CHEPS Goal
  - To automate the schedule building process

Monthly Schedule

- Starts from 27th of the last month
- Ends on the last day of the scheduled month
  - e.g. August Schedule: July 27th – August 31st

	27-Jul	28-Jul	29-Jul	30-Jul	31-Jul	1-Aug	2-Aug	3-Aug	4-Aug	5-Aug
7a-4p										
9a-6p										
12p-9p										
4p-1a										
5p-2a										
8p-5a										
11p-8a										

Additional Restrictions

- Residency Program Requirement:

Additional Restrictions

- Residency Program Requirement:
  - 10-Hour Rule

Additional Restrictions

- Residency Program Requirement:
  - 10-Hour Rule
  - Avoiding conferences

Additional Restrictions

- Residency Program Requirement:
  - 10-Hour Rule
  - Avoiding conferences
  - Avoiding continuity clinics

Additional Restrictions

- Residency Program Requirement:
  - 10-Hour Rule
  - Avoiding conferences
  - Avoiding continuity clinics
  - Intern restrictions

Additional Restrictions

- Residency Program Requirement:
  - 10-Hour Rule
  - Avoiding conferences
  - Avoiding continuity clinics
  - Intern restrictions
  - Etc.

Outline

- Background
- Methodology
- Model Description & C++ Programming
- Results

Methodology

- Linear Programming (LP)
  - Mathematical model to describe problems with a linear system
  - Canonical Form:

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & Ax \geq b \\ & x \geq 0. \end{aligned}$$

  - Consists of 4 general parts:
    - Inputs - c, A, b
    - Decision Variables - x
    - Objective Function
    - Constraints

Methodology

- IBM Concert Technology
  - Cplex Optimizer (Software package)
  - Solves LP problem
  - Provides interfaces to C++, Java, Python…
  - External Libraries

```
10  // include the cplex libraries
11  #include <ilconcert/ilomodel.h>
12  #include <ilcplex/ilocplex.h>
```

Methodology

- Procedure
  - Mathematical formulation
  - C++ programming
  - Solve the model

Outline

- Background
- Methodology
- Model Description & C++ Programming
- Results

Model Overview

Input

- Set:

Input

- Set:
  - R: Residents

Input

- Set:
  - R: Residents
  - S: Shift types

Input

- Set:
  - R: Residents
  - S: Shift types
  - D: Days

Input

- Set:
  - R: Residents
  - S: Shift types
  - D: Days

  - Auxiliary sets
    - Conferences
    - Continuity Clinics
    - Resident Requests
    - Etc.

Input

- Parameters
  - Stored in .csv files
  - Mostly 2-D tables
    - Row – Record
    - Column – Field/Attribute
  - Processed in 2 steps
    - File stream – store the rows into a vector of strings
    - String stream – parse the information in each row
  - Information extracted

Variables

- x_{rds}:
  - Binary
  - Decision on
    - Resident r
    - Day d
    - Shift s

```
vector<vector<vector<bool>>> x_sol;
```

Objective Function

- Feasible solution is sufficient for now
  - No optimization

Constraints - Math

- Capture the requirement in math form
- Total number: 16 constraints
  - e.g.
    - Min and max number of shifts

$$LB \leq \sum_{d \in D} \sum_{s \in S} x_{rds} \leq UB \quad \forall r \in R$$

R: Residents
S: Shift types
D: Days

LB: Lower Bound Value
UB: Upper Bound Value

Constraints - Code

- Broken down into pieces

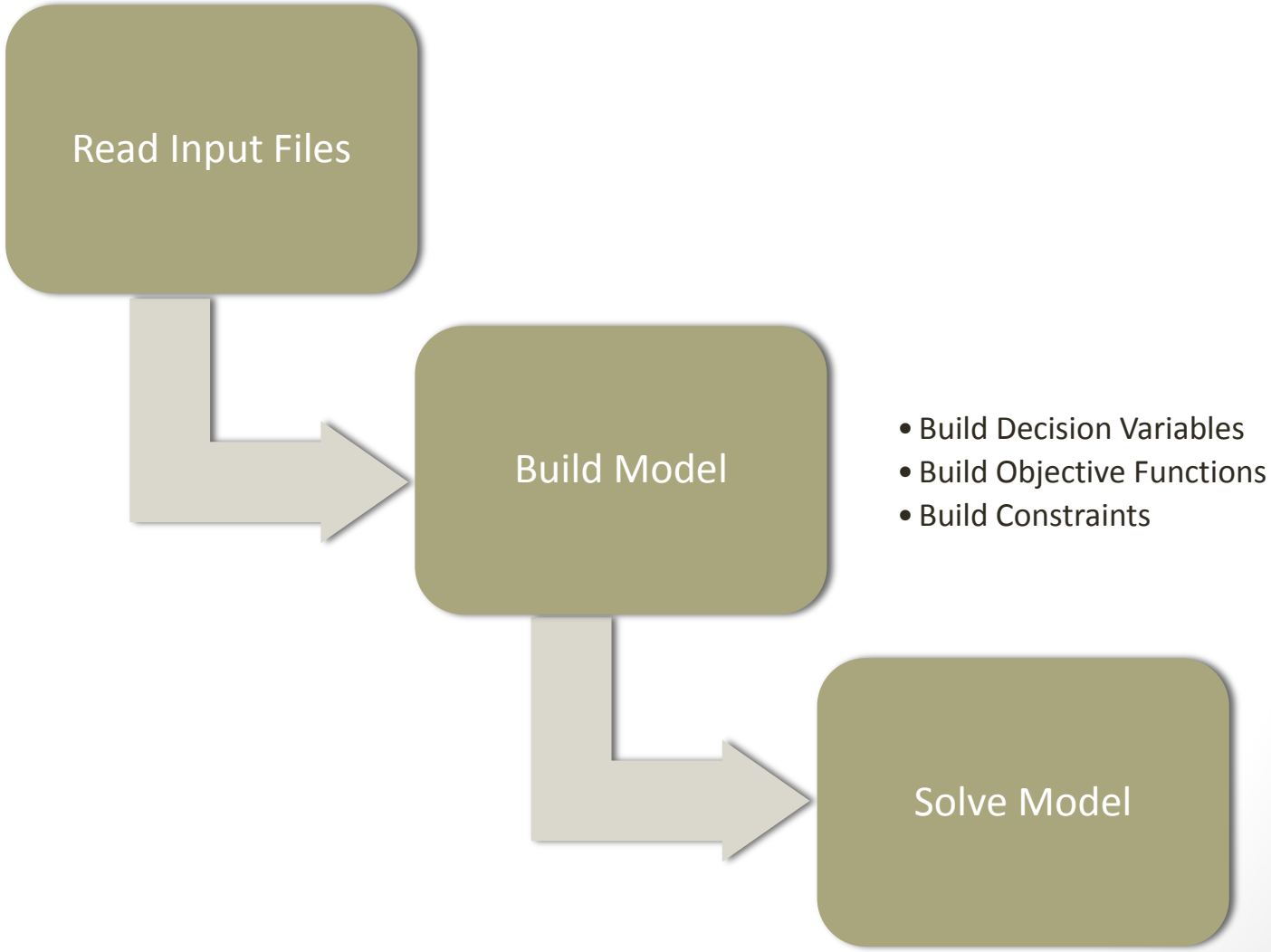
```
3  void PedsShift::buildConstr(){
4      cout << "buildConstr begins" << endl;
5
6      constrEMS();                // constraint 1
7
8      constrReq();                // constraint 2
9
10     constrBound();              // constraint 3, 4
11
12     constrConf();               // constraint 5
13
14     constrCC();                 // constraint 6
15
16     constrIntern();             // constraint 7
17
18     constrTenHour();            // constraint 8
19
20     constrShiftPairPED();       // constraint 9
21
22     constrBSP();                // constraint 10
23
24     constrTerm();               // constraint 12
25
26     constrShiftCoverage();      // constraint 11, 13
27
28     constrMaxStraight();        // constraint 14, 15, 16
29
30     cout << "buildConstr ends" << endl;
31 }
```

Solve the Model

- Cplex Optimizer solves the model

```
cplex.solve(); // solve the model
```

Procedure Summary



- Build Decision Variables
- Build Objective Functions
- Build Constraints

Outline

- Background
- Methodology
- Model Description & C++ Programming
- Results

Results

- Cplex Output
  - x_{rds}:
    - 1 – Assigned
    - 0 – Not Assigned

```
x(0_0_0) = 1
x(0_0_7) = 1
x(0_0_16) = 1
x(0_0_17) = 1
x(0_0_18) = 1
x(0_0_19) = 1
x(0_0_20) = 1
x(0_0_21) = 1
x(0_0_22) = 1
x(0_0_23) = 1
x(0_0_24) = 1
x(0_0_25) = 1
```

Results

- Visualization
  - Print in desired format
  - .csv files

```
ofstream oFile(strOutputPath + "Schedule_Resident_Month.csv");
```

Complete Schedule

- A plain view of the complete schedule

	27-Jul	28-Jul	29-Jul	30-Jul	31-Jul	1-Aug	2-Aug	3-Aug	4-Aug	5-Aug
7a-4p										
9a-6p										
12p-9p										
4p-1a										
5p-2a										
8p-5a										
11p-8a										

Complete Schedule

- A plain view of the complete schedule
- Multiple views of the schedule are produced

	27-Jul	28-Jul	29-Jul	30-Jul	31-Jul	1-Aug	2-Aug	3-Aug	4-Aug	5-Aug	6-Aug	7-Aug
7a-4p	LPH12	LPH15	LPH7	LPH4	LPH18	Si	Ma	Br	Cu	Du	EMS	Du
9a-6p	Ja	Ja	To	Ja	LPH20	Du	Wa	Wa	Co	Co	De	Co
12p-9p	Wa	LPH17	LPH16	To	LPH5	-	To	Co	Du	Le	Si	Ma
4p-1a	To	LPH10	Wa	Wa	Wa	Br	Co	To	Br	Wa	Le	EMS
5p-2a	LPH13	LPH9	LPH6	LPH11	To	Co	De	Ma	Ja	Cu	Wa	To
8p-5a	LPH8	LPH21	LPH22	LPH19	Ja	Ja	Ja	De	Ma	Br	To	Ja
11p-8a	LPH14	LPH2	LPH23	LPH3	LPH1	TPH1	Si	Si	De	Ma	Cu	De

Acknowledgements

- Mr. Tony K. Wang
- Prof. Amy Cohn
- Center for Healthcare Engineering and Patient Safety

Q & A

CENTER FOR
HEALTHCARE ENGINEERING & PATIENT SAFETY
UNIVERSITY OF MICHIGAN

Q & A

- Infeasible Hint

```
Infeasibility row 'Pair_7a_9a(10)':  0 >= 1.
Presolve time = 0.11 sec. (38.68 ticks)

Root node processing (before b&c):
  Real time             =    0.17 sec. (51.74 ticks)
Parallel b&c, 8 threads:
  Real time             =    0.00 sec. (0.00 ticks)
  Sync time (average)   =    0.00 sec.
  Wait time (average)   =    0.00 sec.
                          ------------
Total (root+branch&cut) =    0.17 sec. (51.74 ticks)
```

Q & A

- LP Display

```
Minimize
 obj: x(0_0_6) + x(0_1_6) + x(0_2_6) + x(0_3_6) + x(0_4_6) + x(0_5_6)
      + x(0_6_6) + x(1_0_3) + x(1_1_3) + x(1_2_3) + x(1_3_3) + x(1_4_3)
      + x(1_5_3) + x(1_6_3) + x(2_0_5) + x(2_1_5) + x(2_2_5) + x(2_3_5)
      + x(2_4_5) + x(2_5_5) + x(2_6_5) + x(3_0_4) + x(3_1_4) + x(3_2_4)
      + x(3_3_4) + x(3_4_4) + x(3_5_4) + x(3_6_4) + x(4_0_2) + x(4_1_2)
      + x(4_2_2) + x(4_3_2) + x(4_4_2) + x(4_5_2) + x(4_6_2) + x(5_0_1)
      + x(5_1_1) + x(5_2_1) + x(5_3_1) + x(5_4_1) + x(5_5_1) + x(5_6_1)
      + x(6_0_8) + x(6_1_8) + x(6_2_8) + x(6_3_8) + x(6_4_8) + x(6_5_8)
      + x(6_6_8) + x(7_0_9) + x(7_1_9) + x(7_2_9) + x(7_3_9) + x(7_4_9)
      + x(7_5_9) + x(7_6_9) + x(8_0_10) + x(8_1_10) + x(8_2_10) + x(8_3_10)
      + x(8_4_10) + x(8_5_10) + x(8_6_10) + x(9_0_11) + x(9_1_11) + x(9_2_11)
      + x(9_3_11) + x(9_4_11) + x(9_5_11) + x(9_6_11) + x(10_0_12) + x(10_1_12)
      + x(10_2_12) + x(10_3_12) + x(10_4_12) + x(10_5_12) + x(10_6_12)
      + x(11_0_13) + x(11_1_13) + x(11_2_13) + x(11_3_13) + x(11_4_13)
      + x(11_5_13) + x(11_6_13) + x(12_0_14) + x(12_1_14) + x(12_2_14)
      + x(12_3_14) + x(12_4_14) + x(12_5_14) + x(12_6_14) + x(13_0_15)
      + x(13_1_15) + x(13_2_15) + x(13_3_15) + x(13_4_15) + x(13_5_15)
      + x(13_6_15) + x(14_0_16) + x(14_1_16) + x(14_2_16) + x(14_3_16)
      + x(14_4_16) + x(14_5_16) + x(14_6_16) + x(15_0_17) + x(15_1_17)
      + x(15_2_17) + x(15_3_17) + x(15_4_17) + x(15_5_17) + x(15_6_17) + x113
Subject To
 shiftConstr(0_0):   x(0_0_0) + x(1_0_0) + x(2_0_0) + x(3_0_0) + x(4_0_0)
                     + x(5_0_0) + x(6_0_0) + x(7_0_0) + x(8_0_0) + x(9_0_0)
                     + x(10_0_0) + x(11_0_0) + x(12_0_0) + x(13_0_0)
                     + x(14_0_0) + x(15_0_0)  = 1
 shiftConstr(0_1):   x(0_1_0) + x(1_1_0) + x(2_1_0) + x(3_1_0) + x(4_1_0)
                     + x(5_1_0) + x(6_1_0) + x(7_1_0) + x(8_1_0) + x(9_1_0)
                     + x(10_1_0) + x(11_1_0) + x(12_1_0) + x(13_1_0)
                     + x(14_1_0) + x(15_1_0)  = 1
 shiftConstr(0_2):   x(0_2_0) + x(1_2_0) + x(2_2_0) + x(3_2_0) + x(4_2_0)
                     + x(5_2_0) + x(6_2_0) + x(7_2_0) + x(8_2_0) + x(9_2_0)
                     + x(10_2_0) + x(11_2_0) + x(12_2_0) + x(13_2_0)
                     + x(14_2_0) + x(15_2_0)  = 1
```

Q & A

- Resident schedule by month

Tomlinson	Ped1	27-Jul	11-Aug						
	27-Jul	28-Jul	29-Jul	30-Jul	31-Jul	1-Aug	2-Aug	3-Aug	4-Aug
7a-4p									
9a-6p			Tomlinson						
12p-9p				Tomlinson			Tomlinson		
4p-1a	Tomlinson							Tomlinson	
5p-2a					Tomlinson				
8p-5a									
11p-8a									

Q & A

- Resident schedule by week

Jahns	MP1	27-Jul	16-Aug		13-Aug	16-Aug	
	Sun	Mon	Tue	Wed	Thur	Fri	Sat
						27-Jul	28-Jul
7a-4p						I	I
9a-6p						Jahns	Jahns
12p-9p							
4p-1a							
5p-2a							
8p-5a							C
11p-8a						I	C I

	Sun	Mon	Tue	Wed	Thur	Fri	Sat
	29-Jul	30-Jul	31-Jul	1-Aug	2-Aug	3-Aug	4-Aug
7a-4p	C I	I	I	I	I	I	I
9a-6p	C	Jahns					
12p-9p	C						
4p-1a	C						
5p-2a	C						Jahns
8p-5a			Jahns	Jahns	Jahns		C
11p-8a	I	I	I	I	I	I	C I

	Sun	Mon	Tue	Wed	Thur	Fri	Sat
	5-Aug	6-Aug	7-Aug	8-Aug	9-Aug	10-Aug	11-Aug
7a-4p	C I	I	I	I	I	I	I
9a-6p	C						
12p-9p	C					Jahns	
4p-1a	C						
5p-2a	C						
8p-5a			Jahns				
11p-8a	I	I	I	I	I	I	I

Q & A

- Resident Metrics

Name	Longest Day Streak	Longest Night Streak	# Shift	# Night Shift	# BSP	# PCC
EMsr1	2	0	10 (10-10)	0 (0-1000)	0 (0-1000)	0 (0-1000)
Agbakwuru	3	2	10 (10-11)	4 (0-1000)	2 (0-1000)	2 (0-1000)
Gramling	5	1	10 (10-11)	1 (0-1000)	0 (0-1000)	1 (0-1000)
Patel	4	1	10 (10-11)	2 (0-1000)	0 (0-1000)	1 (0-1000)
Tomlinson	3	1	10 (10-11)	1 (0-1000)	1 (0-1000)	0 (0-1000)
Walker	3	2	10 (10-11)	2 (0-1000)	0 (0-1000)	1 (0-1000)
Cochran	5	0	10 (10-11)	0 (0-1000)	1 (0-1000)	0 (0-1000)
Dehudy	4	2	19 (19-22)	9 (0-1000)	1 (0-1000)	3 (0-1000)
Effron	4	2	10 (10-11)	3 (0-1000)	0 (0-1000)	1 (0-1000)
Foster	4	1	10 (10-11)	3 (0-1000)	1 (0-1000)	1 (0-1000)
Singla	4	2	10 (10-11)	4 (0-1000)	0 (0-1000)	1 (0-1000)
Bradley	3	2	10 (10-11)	3 (0-1000)	0 (0-1000)	1 (0-1000)

Q & A

- Residents' Request
  - Preset work terms at ED

Name	FirstDay	LastDay
EMsr1	1-Aug	31-Aug
Agbakwuru	12-Aug	26-Aug
Gramling	12-Aug	26-Aug
Patel	12-Aug	26-Aug
Tomlinson	27-Jul	11-Aug
Walker	27-Jul	11-Aug
Cochran	1-Aug	15-Aug
Dehudy	1-Aug	31-Aug
Effron	16-Aug	31-Aug
Foster	16-Aug	31-Aug
Singla	1-Aug	15-Aug
Bradley	1-Aug	15-Aug

Q & A

- Residents' Request
  - Preset work terms at ED
  - Requests for days off

Resident ID	First Day Off	Last Day Off
Ag	15-Aug	17-Aug
Wa	27-Jul	28-Jul
Wa	3-Aug	4-Aug
Wa	10-Aug	11-Aug
Co	7-Aug	10-Aug
De	8-Aug	10-Aug
Fo	16-Aug	17-Aug
Ja	13-Aug	16-Aug
Du	1-Aug	3-Aug
Le	1-Aug	3-Aug
Le	20-Aug	23-Aug
Sa	23-Aug	24-Aug

Q & A

- Decision Variable Coding

```
IloArray<IloArray<IloArray<IloNumVar>>> x;
```

Q & A

- Placeholder for optimization
  - In practice, a constant value 0 is optimized

```
// constExpr = 0;
IloObjective obj = IloMinimize(env, 0);

// add to model
model.add(obj);
```