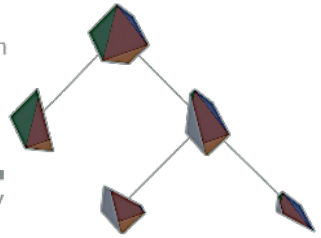
Mechanical & Industrial Engineering
UNIVERSITY OF TORONTO

Toronto Intelligent Decision
TIDEL
Engineering Laboratory



Learning Scheduling Models from Event Data

Appeared in ICAPS 2019

Arik Senderovich, Kyle E.C. Booth, J. Christopher Beck

jcb@mie.utoronto.ca

Progress Toward the Holy Grail Workshop
CP2019

Motivation: Schedule-Driven Systems

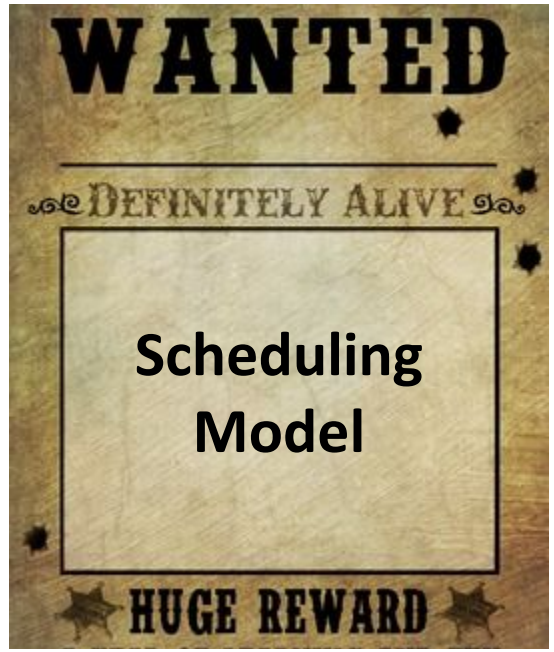
Timetables

Production plans

Appointment books

Modeling Schedule-Driven Systems

- OR analyst models the system manually
  - Interviews & time studies
  - Ad-hoc modeling decisions
- Issues with manual modeling
  - Time consuming
  - Inaccurate measurements
  - Modeling expertise matters

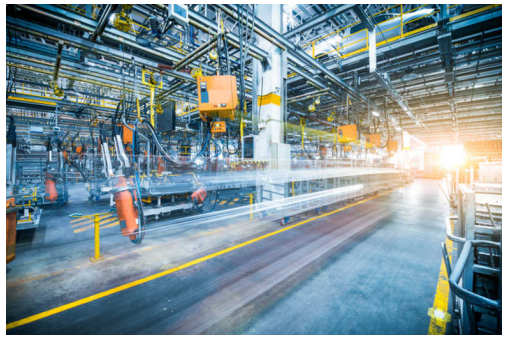


Can we automatically create a scheduling model?

Model Learning from Event Data

Assumptions

1. System is already running
2. Event data contains only feasible solutions



Optimal schedules

Scheduling Model

Model learning

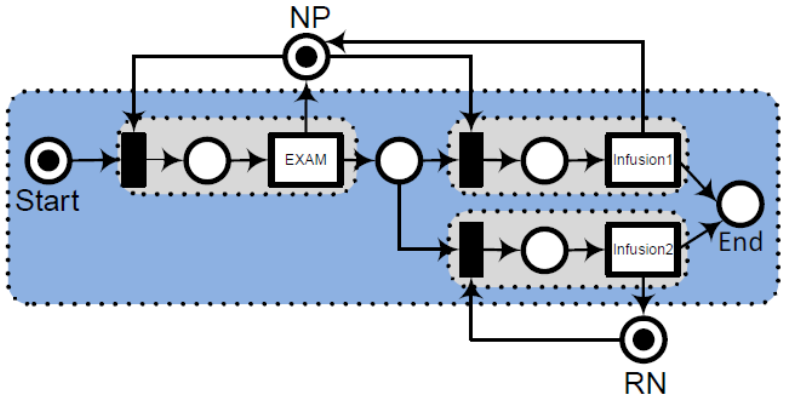
What is event data?

- Recordings of schedule executions:

Case	Activity Type	Resources	Start	Complete
pat1	Blood-Draw	RN	9:05AM	9:10AM
pat1	Exam	MD	9:55AM	10:20AM
pat2	Exam	NP	9:30AM	9:45AM
pat2	Infusion	RN	9:35AM	10:52AM
pat3	Exam	NP	12:45PM	1:10PM
pat3	Infusion	NP	9:35AM	10:32AM

Process Mining

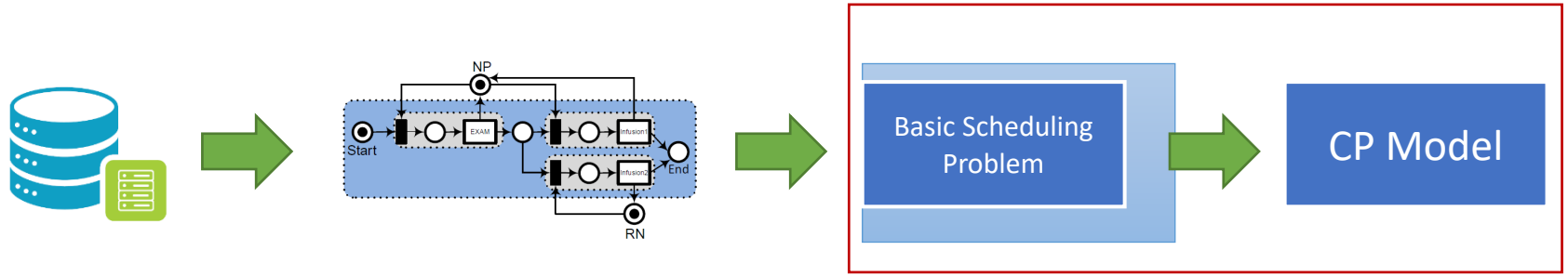
Case	Activity Type	Resources	Start	Complete
pat1	Blood-Draw	RN	9:05AM	9:10AM
pat1	Exam	MD	9:55AM	10:20AM
pat2	Exam	NP	9:30AM	9:45AM
pat2	Infusion	RN	9:35AM	10:52AM
pat3	Exam	NP	12:45PM	1:10PM
pat3	Infusion	NP	9:35AM	10:32AM



- Mining process models from event data (typically to Petri nets)
- Models are used for descriptive and predictive analysis
- In this work: process mining meets scheduling

van der Aalst (2011). *Process mining: discovery, conformance and enhancement of business processes.*

Our Approach:
Model Learning using Process Mining

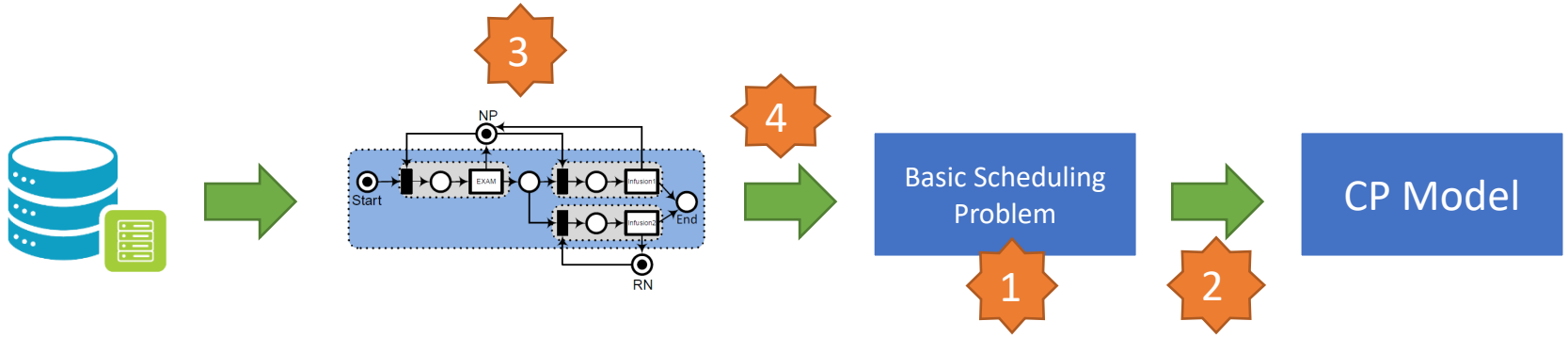


Separating problem description (BSP) from the constraint programming (CP) model that solves it

Outline

- Introduction
- Details of the Contribution
- Proof of Concept Experiments
- Future Work

Contributions



1. BSP = declarative problem representation
2. Mapping BSP into CP model
3. Defining a new type of Petri net - ARPNs
4. Mapping ARPNs to BSPs
5. End-to-end data to model solution

Basic Scheduling Problems (BSPs)

Basic Scheduling Problem

1

- Defines a family of scheduling problems

Definition (Basic Scheduling Problem (BSP))

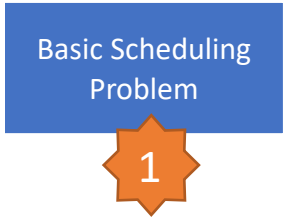
Given a set of activities to be schedule[d] … tion $\theta : \mathscr{A} \to \mathscr{T}$ that maps these activities to activity types and a function $\nu : \mathscr{A} \to \mathscr{V}$ that maps act… s, the BSP is a tuple $\langle \mathscr{A}, \theta, \nu, \mathscr{T}, \mathscr{R}, \mathscr{V}, \Pi, c, d \rangle$ with:

- $\mathscr{T}$ *being the set of …*
- $\mathscr{R}$ *being the s…*
- $\mathscr{V}$ *being the set … s a set of finite sequences over $\mathscr{T}$,*
- $\Pi = \{\Pi_v \subseteq \mathscr{T} \times \mathscr{T} \ldots$ *…ence relations between pairs of activity types, such that for all $(t, t') \in \Pi_v$, a… …re t' can start in job v,*
- $c : \mathscr{R} \to \mathbb{N}^+$ *being the … …arces to their capacities, and,*
- $d : \mathscr{T} \times \mathscr{R} \nrightarrow \mathbb{R}^+$ *being … …tial function that maps pairs of activity types and resources (that can execute these activiti… …ues in the time domain.*

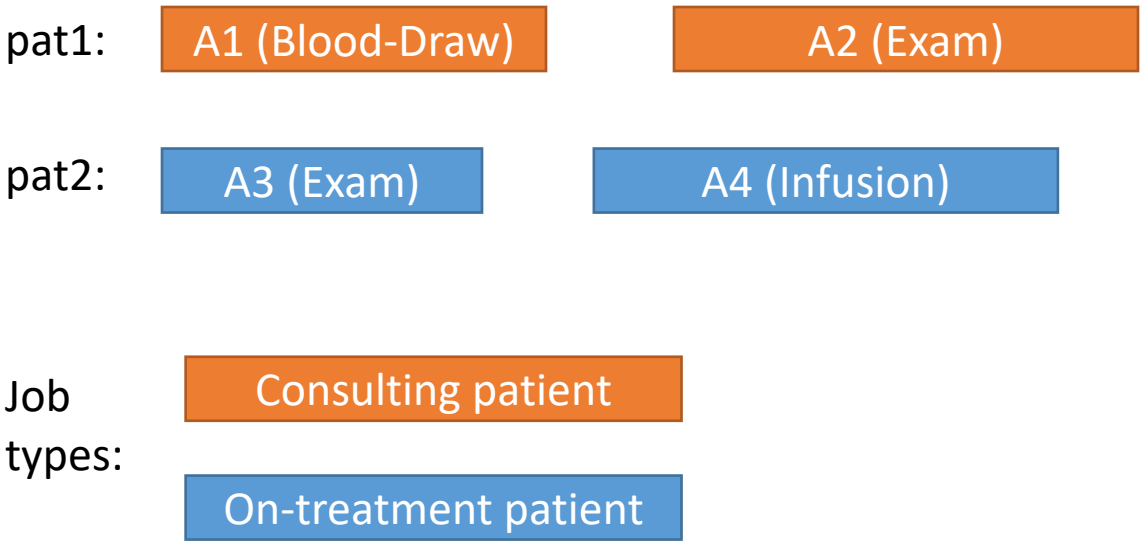
TL;DR

- Generalizes well-studied scheduling problems: single-machine, parallel machine and job-shop

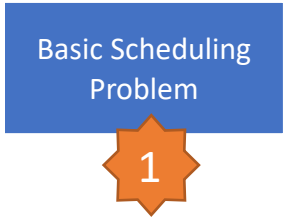
Basic Scheduling Problem (BSP)



- A BSP comprises two parts
  1. Instance-based (given for an instance)
     - Set of jobs containing activities
     - Mapping of jobs and activities to respective types

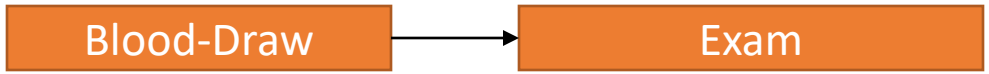


Basic Scheduling Problems (BSPs)

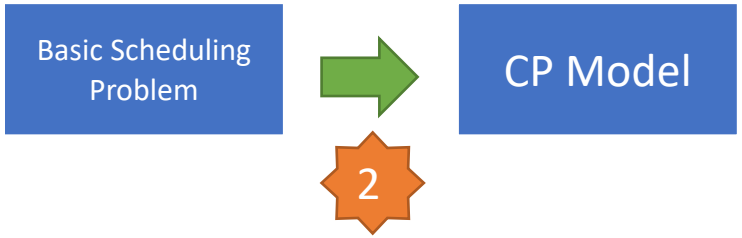


- A BSP comprises two parts

2. Parameterized (learned)
   - Precedence constraints between activity types per job type
   - Resources and their capacities: medical doctor (4), nurse (3)
   - Durations for resource & activity type combinations: exam by NP (15 min), exam by MD (25 min)

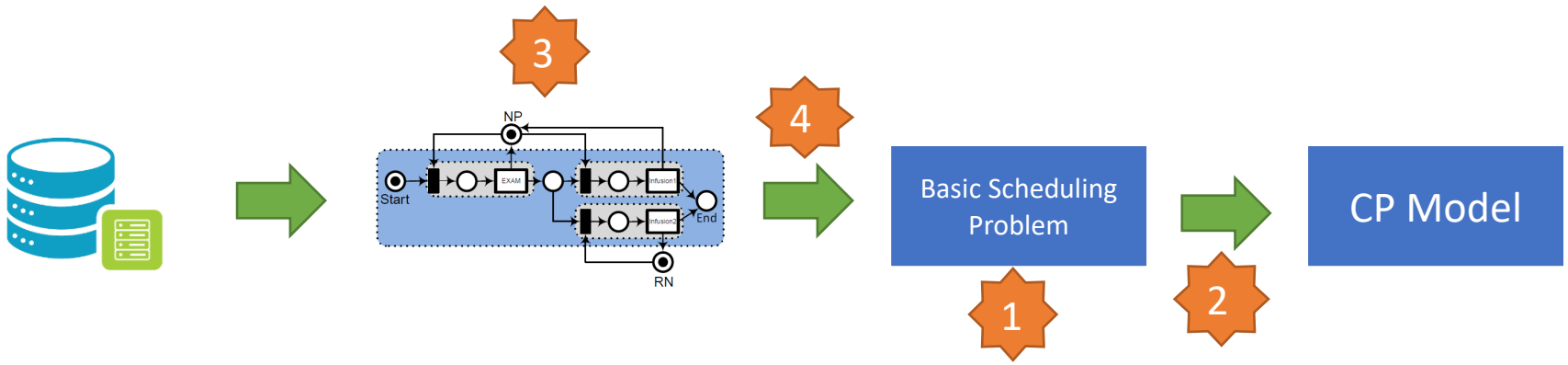


Mapping BSP to CP Model

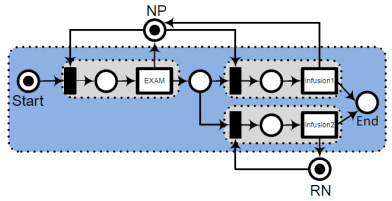


- BSP components are mapped to CP variables and constraints
  - Activities – mandatory interval variables
  - Precedence – *EndBeforeStart* constraints per job type
  - Resource assignment – optional interval variables with resource dependent durations
  - Resource capacity – *Cumulative* global constraint per resource
- Standardized CP model (no guarantees on quality of model)
- Objective function – not part of the BSP (future work)

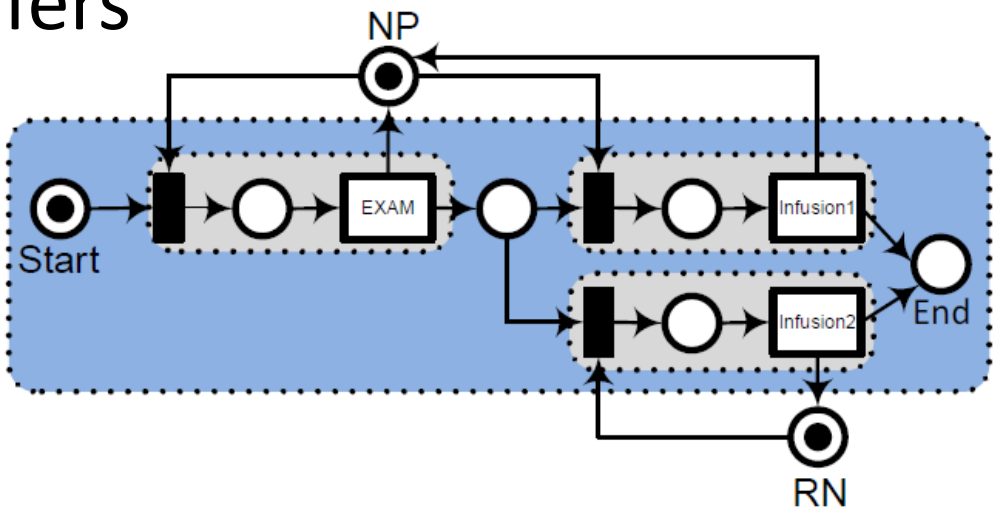
So far…


3
NP
Start
EXAM
Infusion1
Infusion2
End
RN
4
Basic Scheduling Problem
1
2
CP Model

Timed Petri net (TPN)

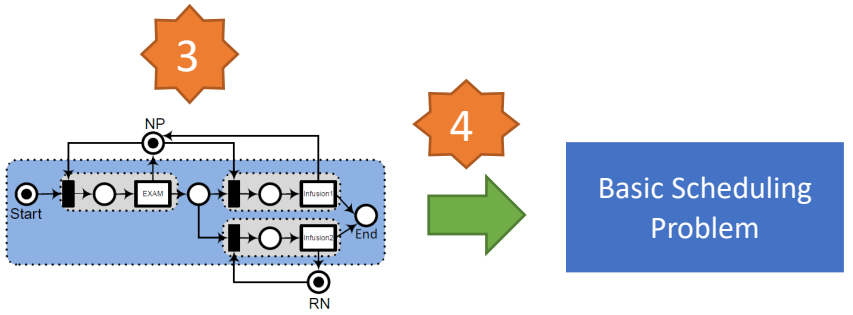


- Well-established formalism to model discrete-event dynamic systems
- Rectangles = transitions
  - Labeled transitions – activity types with specified durations
  - Silent transitions – routing (resource assignment)
- Circles = places; activity/resource buffers
- Edges = preconditions/effects
- Tokens = system state



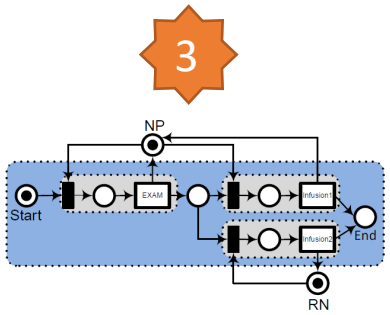
Silva (2013). *Half a century after Carl Adam Petri's Ph.D. thesis: A perspective on the field*.

Gap: Expressivity of TPNs

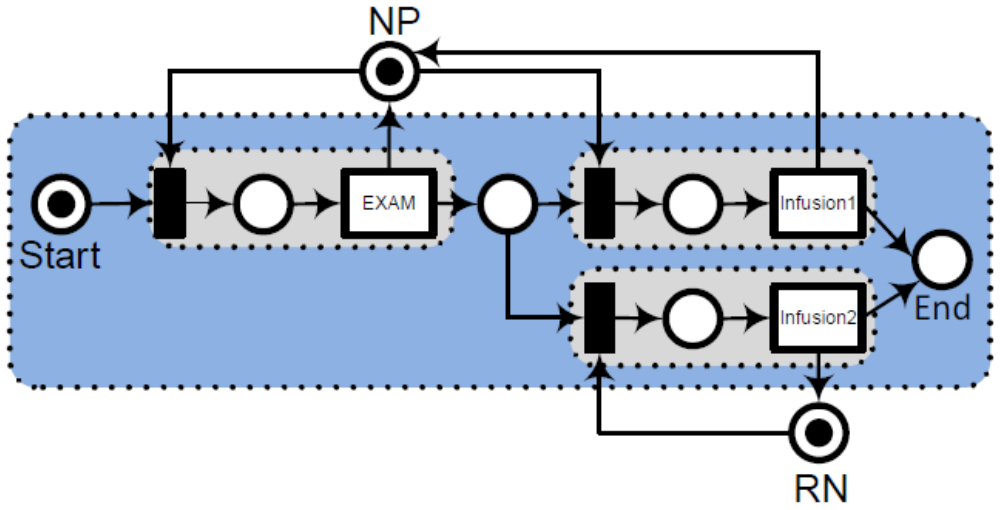


- TPNs are more expressive than BSPs
  - Shared resources per activity
  - Non-unary demand for resources
  - ….
- Mapping to BSP requires restricting the TPN

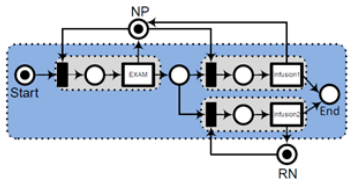
Activity-Resource Petri net (ARPN)



- Restricted TPNs
  - Activity type constructs **(grey area)**
  - Job type constructs **(blue area)** – contain only activity constructs
  - Activity-Resource PN – resource buffers connected to activity types



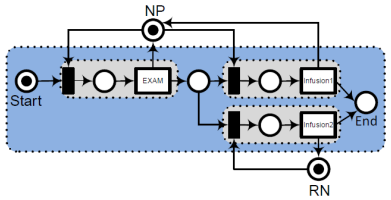
Mining TPNs from Event Data



- Produce TPNs from event data by mining
  - Temporal dependencies (precedence, overlaps,…)
  - Duration distributions (we take mean values)
  - Resource capacities and activity assignments
- Must verify that TPN is ARPN (polynomial time in size of TPN)

Senderovich et al. (2015). *Data-Driven Performance Analysis of Scheduled Processes.*

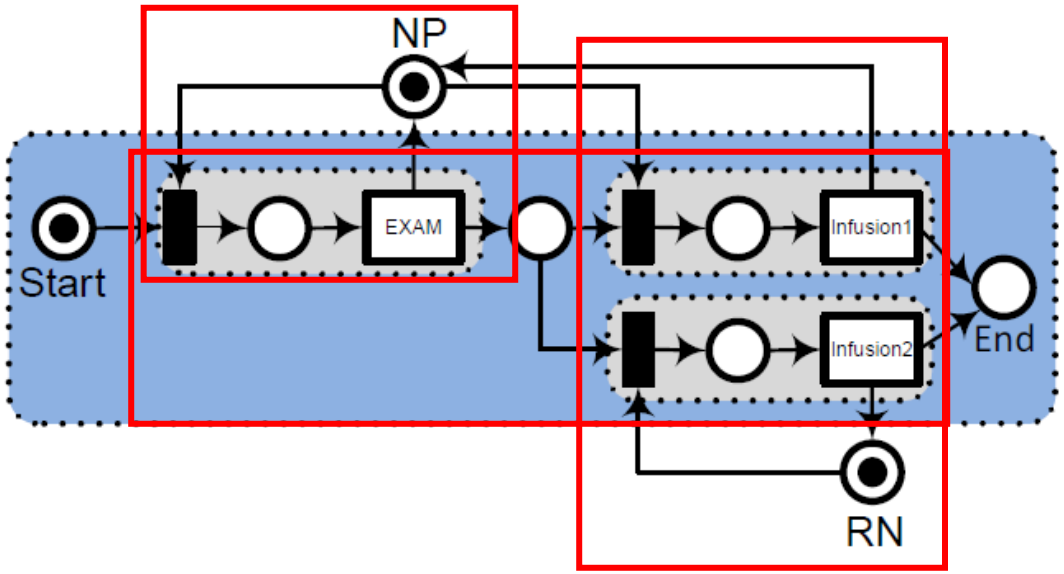
Mapping ARPN to BSP

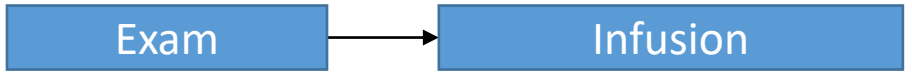


4

Basic Scheduling Problem

Example: On-treatment patient





- Exam by NP (fixed duration)
- Infusion by NP or RN
- Duration of Infusion is resource dependent

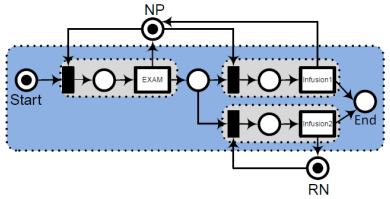
Technical details of the algorithm are in the paper

Outline

- Introduction
- Details of the Contribution
- Proof of Concept Experiments
- Future Work

Proof of Concept 1: Job-Shop Scheduling

Job-shop scheduling (JSP)
standard benchmarks



Comparison 1: BSP = JSP?

Basic Scheduling Problem

Comparison 2: CP model good enough?

CP Model

CP Optimizer
10 min time limit

Proof of Concept 1: Results

- BSP learning
  - Learned BSP is equivalent to the originating JSP
  - Learning BSP takes less than 1 second (on average)
- CP Model
  - CP solved 49/53 learned models to optimality in 15.4 seconds on average
  - Found feasible solutions for the other 4 with an average optimality gap of 6.3% (10 min time limit)

Comparison 1: BSP = JSP?

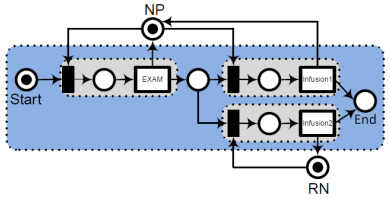
Basic Scheduling Problem

Comparison 2: CP model good enough?

CP Model

Proof of Concept 2: Appointment Scheduling

Comparisons:
CP model good enough?
How good is the schedule?



Basic Scheduling Problem

CP Model

CP Optimizer
10 min time limit

Real-world data:
30000 activities/month
240 job types
60 activity types
Ground truth schedules

Proof of Concept 2: Results

- BSP learning
  - 3 months training data (January – March 2016)
  - 1 month test data (April 2016, 21 days)
  - Learning BSP takes approx. 450 seconds

Proof of Concept 2: Results

- CP Model:

Min Makespan:

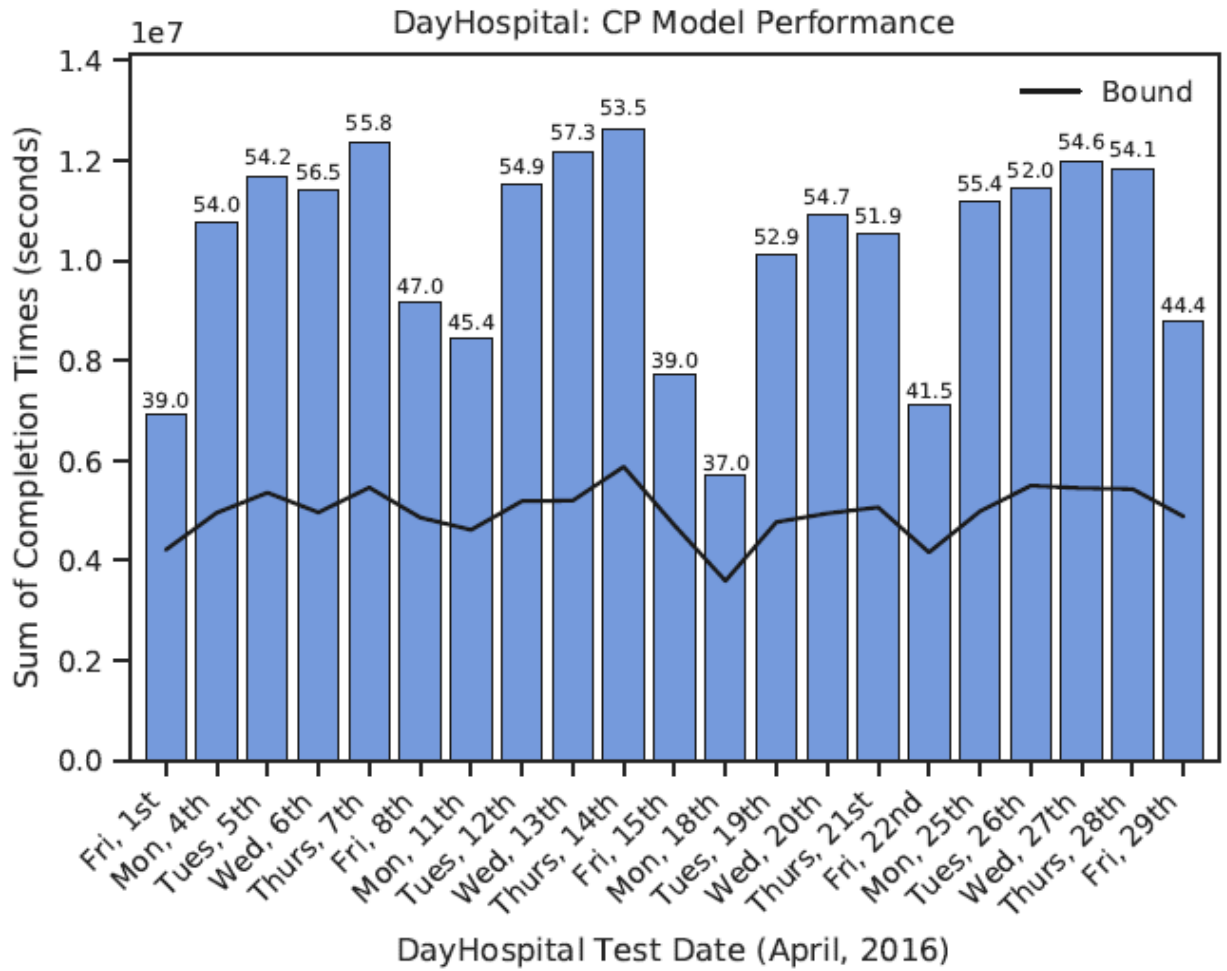
Solved to optimality

in less than 1 seconds for all test days.

Min sum of completion times:

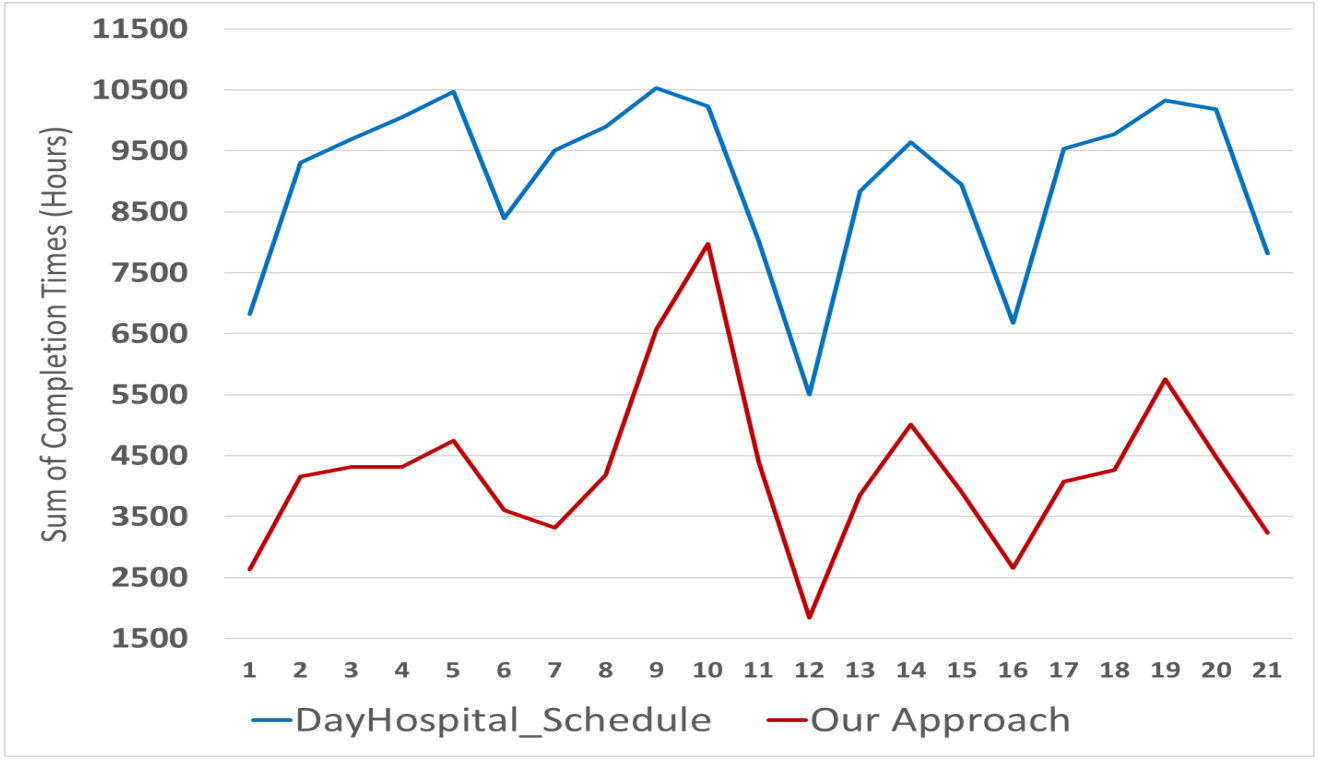
Optimality gap – 50.3% (avg)

Feasible solutions for all days



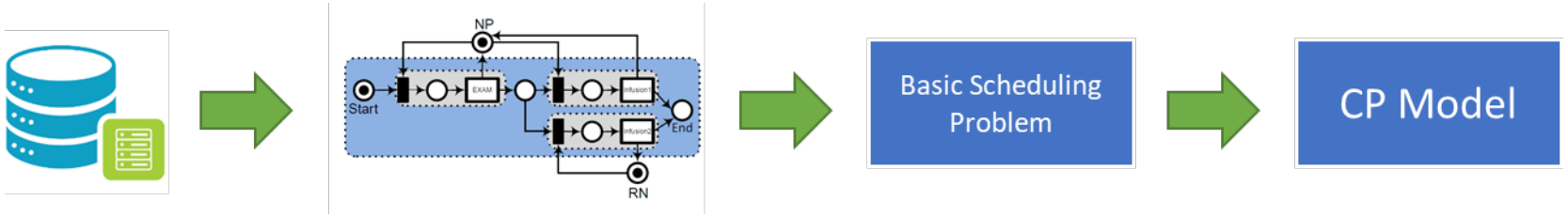
Schedule Comparison Experiment
(Not in the paper)

- Comparing our schedules to real hospital schedules
- Execution comparison: real durations and punctuality (patients are early/late)



Future Work

- Learning scheduling models with objective functions
- Extending BSP to capture more complex scheduling problems (TPN expressivity)
- Mapping learned Petri nets to stochastic scheduling problems (via stochastic Petri nets)



Learning Scheduling Models from Event Data

Arik Senderovich, Kyle E. C. Booth, J. Christopher Beck
Department of Mechanical & Industrial Engineering, University of Toronto, Toronto, ON, Canada
sariks@mie.utoronto.ca, kbooth@mie.utoronto.ca, jcb@mie.utoronto.ca

Thank you!
sariks@mie.utoronto.ca