Prediction-based Resource Allocation using LSTM and minimum cost and maximum flow algorithm

Gyunam Park, Minseok Song†

POSTECH, Pohang, South Korea

June 26, 2019

POSTECH
POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Contents

Introduction

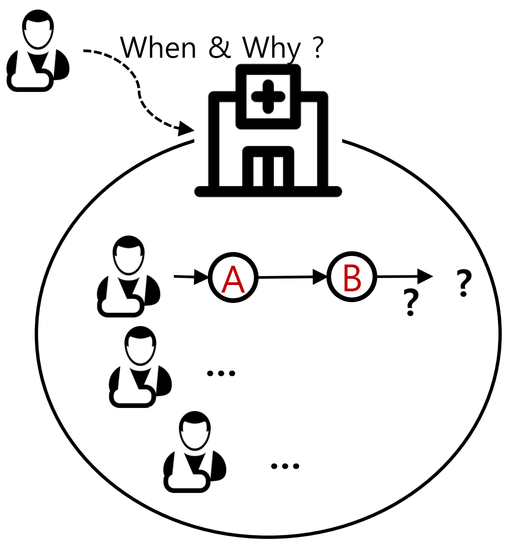
- Research Background
- Objective

Introduction – Research Background

- **Resource allocation in business process management (BPM)**
  - Resource allocation in BPM **aims at allocating appropriate resources to tasks at the correct time**, to balance the demand for process executions against the availability of these resources.
  - It has been recognized as an important issue in BPM since **efficient resource allocation improves productivity, balances resource usage, and reduces execution costs.**
  - In a more general perspective, it shares commonalities with **job-shop scheduling problem in operations research.**
    - This problem finds the job sequences on machines to achieve an objective (e.g., minimizing total completion time), which is NP-hard and computationally intractable combinatorial problem.
    - There has been considerable research in the area of job shop scheduling over the past years.
      - ✓ Dispatching rules (Huang et al., 2015)
      - ✓ Shifting bottleneck heuristics (Braune et al., 2016)
      - ✓ Local Search (Kuhpfahl et al., 2016)

Introduction – Research Background

- **Resource allocation in business process management (BPM)**
  - Among the techniques, **dispatching rules** receive massive attention from practical viewpoint since it is useful to find **a reasonably good solution in a relatively short time**.
  - However, they are applicable **only if the required parameters** such as the release time, the processing time, and the sequence of operations of jobs **are known in advance**.
  - Instead, we have **limited information** about the scheduling parameters in many circumstances.



<Emergency department>

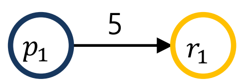
Unaware of,
1. When and why a patient would come into the department
2. Clinical procedures
3. Processing time taken to finish an operation

Non-clairvoyant Online Job Shop Scheduling Problem

Prediction can play a key role in this problem

Introduction – Research Background

- **Motivating example**
  - Suppose we find optimal resource allocation (in terms of **total weighted completion time**) for "MRI" operation in emergency department.



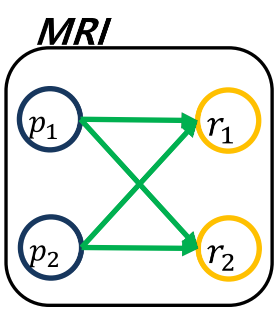
<Notation>

	p_1	p_2	p_3
Weight	1	1	10

<patient weights (Urgency)>

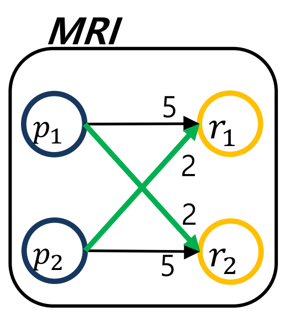
Initial Setting

At $T = t$,



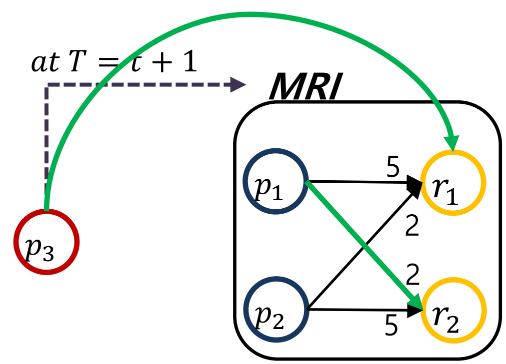
Allocation

Predicting the processing time → Assigning most efficient resource



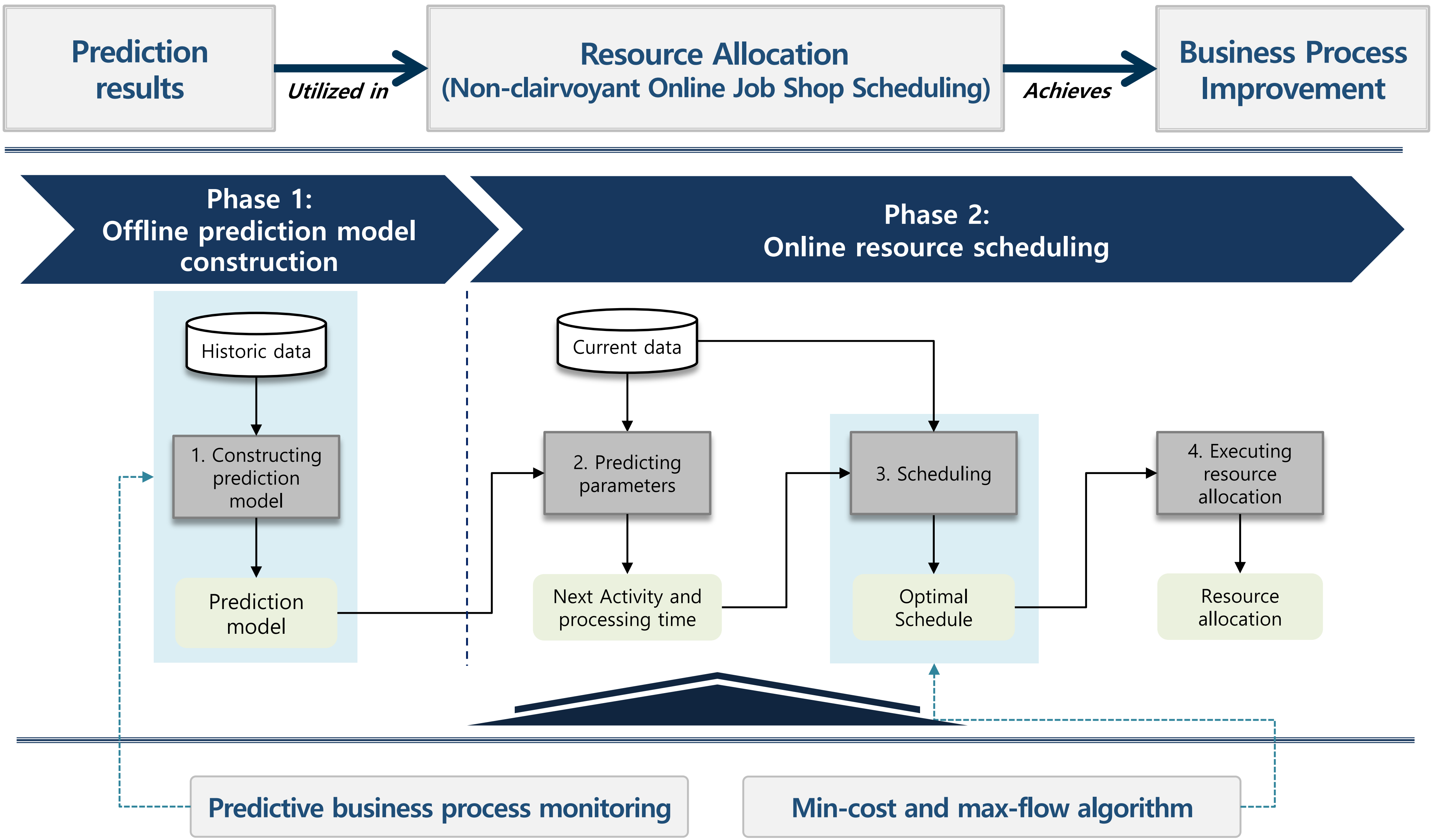
Allocation

Predicting that p_3 will require "MRI" → Reserving r_1 for p_3



Allocation

Introduction – Objective

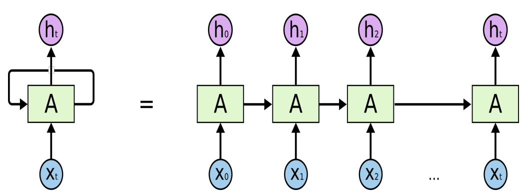


Background

- Preliminaries
- Problem Statement
- Baseline approach

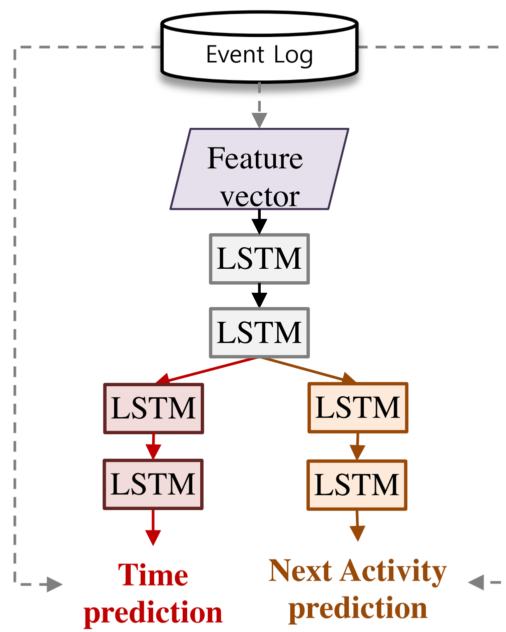
Background – Preliminaries

- **Predictive business process monitoring**
  - Predictive business process monitoring aims at **providing timely information** that enable **proactive and corrective actions** to improve process performance and mitigate risks.
    - Next event prediction: predicting the next event of a running instance such as **next activity**.
    - Time prediction: predicting time-related properties of a running instance such as **remaining time and processing time**.
  - Tax et al. (2017) propose an approach that predicts both the next activity and its timestamp using LSTM (Long-Short Term Memory Neural Network).



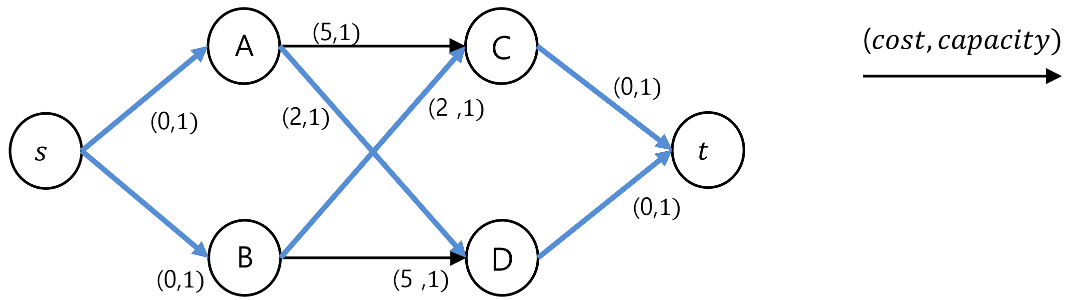
LSTM
(Long-Short Term Memory)

- Sequence learning tasks (e.g., Natural language processing (NLP))
- Learning temporal dynamics



Background – Preliminaries

- **Minimum cost and maximum flow problem**
  - Minimum cost and maximum flow problem is a way of **minimizing the cost required to deliver maximum amount of flow possible in the network**.
    - E.g., A directed graph $G = (V, E)$ with a source node $s \in V$ and a sink node $t \in V$, where each edge $(u, v) \in E$ has cost and capacity.



<Minimum cost and maximum flow of G>

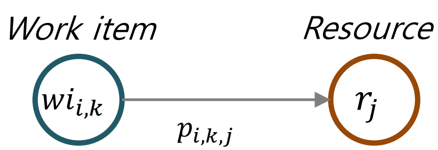
  - It can be solved in polynomial time using the network simplex algorithm.

Background – Problem Statement

- **Non-clairvoyant Online Job Shop Scheduling Problem**
  - Given a set of instances I, this problem **finds an optimal scheduling of all operations** within instances while **minimizing total weighted completion time $\Sigma_i w_i C_i$**,
    - w_i: weight of I_i
    - C_i: difference between the finish time F_i and start time S_i of an instance I_i.
  - Assumptions:
    1. **Unaware of the information** regarding an instance **except the weight** of it.
    2. Find out the **next operation of an instance** only if the instance finishes its current operation.
    3. Each operation has **a specific set of resources** with whom it needs to be processed.
    4. **Only one operation** within an instance can be processed at a given time.
    5. Once processing begins on an operation, **it cannot be stopped** until completion.

Background – Problem Statement

- **Running Example**
  - Suppose there are 5 instances and 3 resources in the process.
    - $I_1, \ldots, I_4$ are ready for the allocation at $T = t$ → We don't know the processing time.
    - I_5 is currently doing its 2nd operation (i.e., $wi_{5,2}$) at $T = t$ → We don't know the next activity (and required resource).



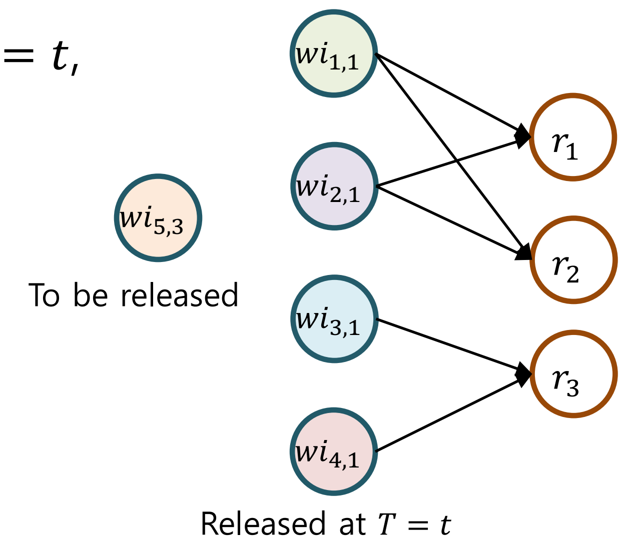
<Notation>

→ $\boldsymbol{wi_{i,k}}$ (k^{th} operation of instance I_i) can be processed by $\boldsymbol{r_j}$ in $\boldsymbol{p_{i,k,j}}$ (processing time)

	I_1	I_2	I_3	I_4	I_5
Weight	1	1	1	5	10

<Instance weights>

At $T = t$,



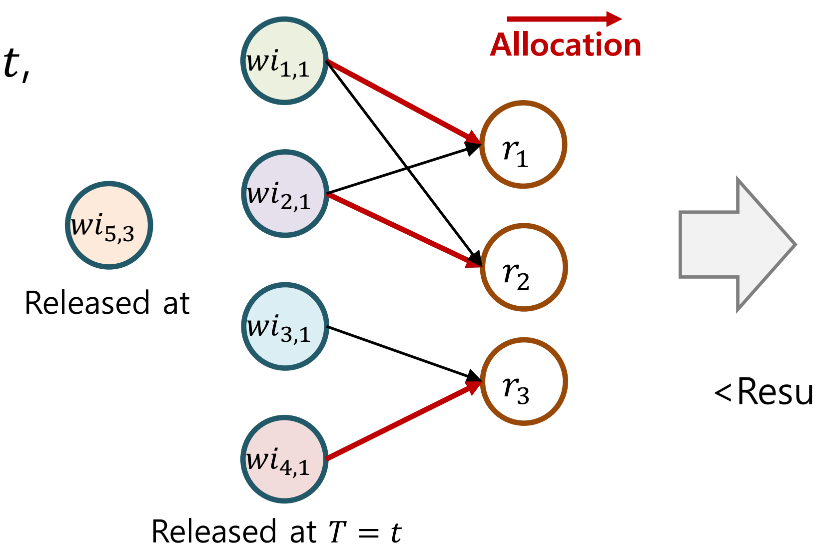
Background – Baseline approach

- **Baseline Approach (*WeightGreedy*)**
  1. Each work item is assigned to an available resource in a "**first come, first served**" manner.
  2. If there exist conflicting demands for the same resource, the work item with **higher weight is served first**.
  3. If the competing work items have the same instance weights, the **tie is broken at random**.

	I_1	I_2	I_3	I_4	I_5
Weight	1	1	1	5	10

<Instance weights>

At $T = t$,



Here

	t	$t+1$	$t+2$	$t+3$	$t+4$	$t+5$	$t+6$	$\sum w_i C_i$
r_1	$wi_{1,1}$					$wi_{5,3}$		65
r_2	$wi_{2,2}$							5
r_3	$wi_{4,1}$		$wi_{3,1}$					15

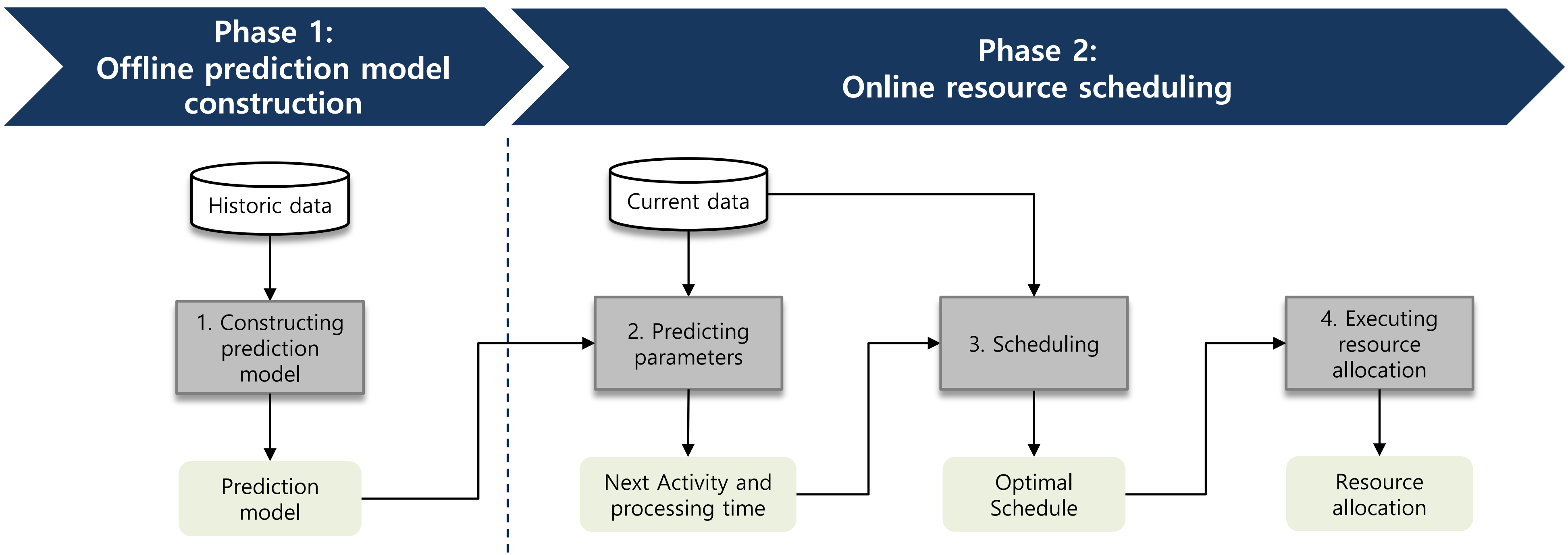
85

<Result of resource allocation based on *WeightGreedy*>

Method

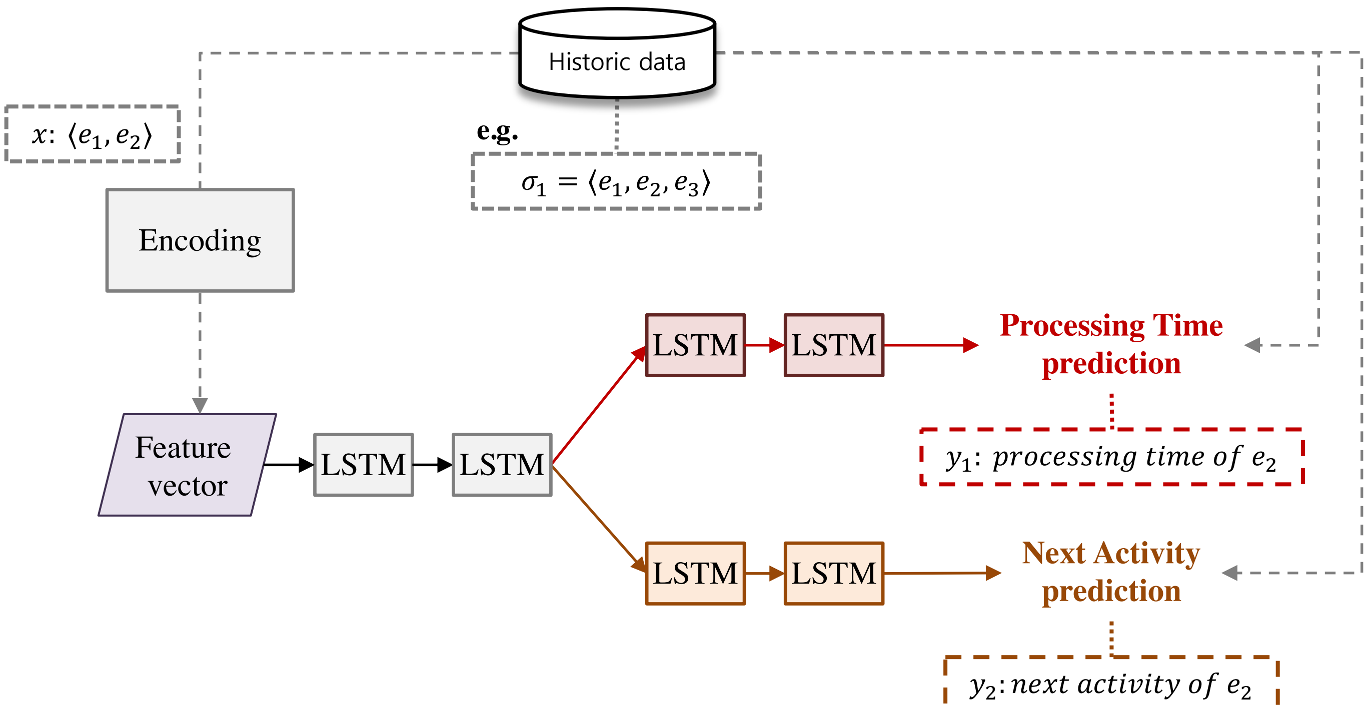
- Overview
- Steps

Method – Overview



Method – Phase 1

- **Step 1: Constructing Prediction Model**
  - In this step, we aim at building a model to predict the **processing time** and the **next activity** of a running instance, which is based on LSTM (Tax et al, 2017).
  - We learn the model with all traces in the historic data.
    - E.g., Training with a trace $\sigma_1 = \langle e_1, e_2, e_3 \rangle$

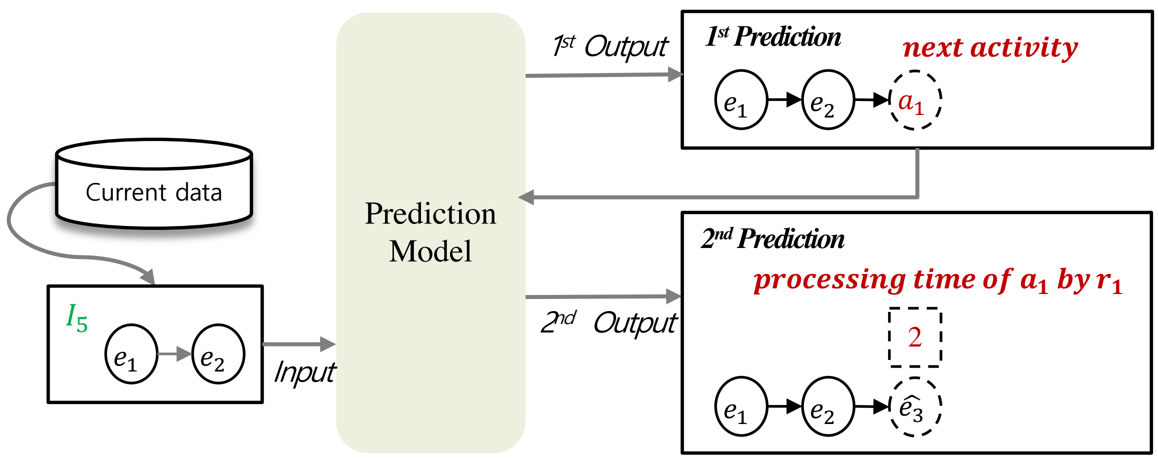


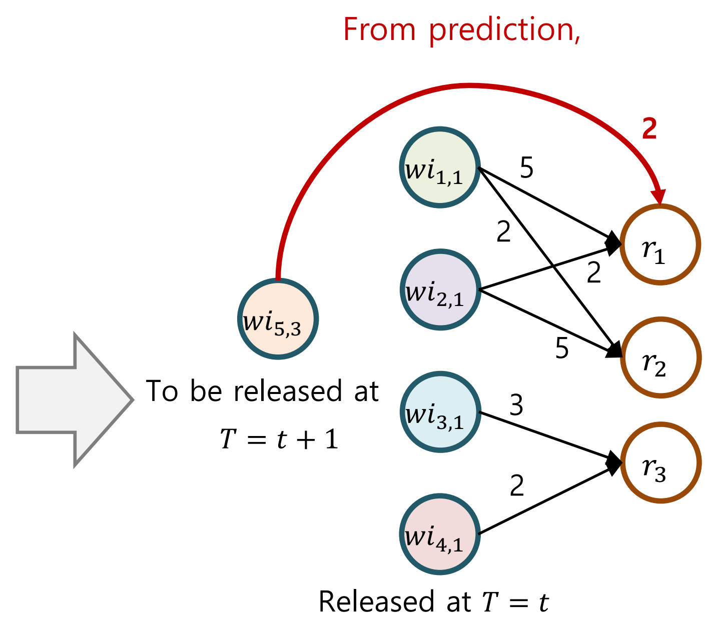
Method – Phase 2

- **Step 2: Predicting parameters**
  - Based on the prediction model we construct in the previous step, we predict the **next activity and processing time** of ongoing instances from the current data.
  - We conduct **two consecutive predictions** for a running instance.
    1. Predict the next activity of it.
    2. Predict the processing time of the activity by available resources.

E.g.,

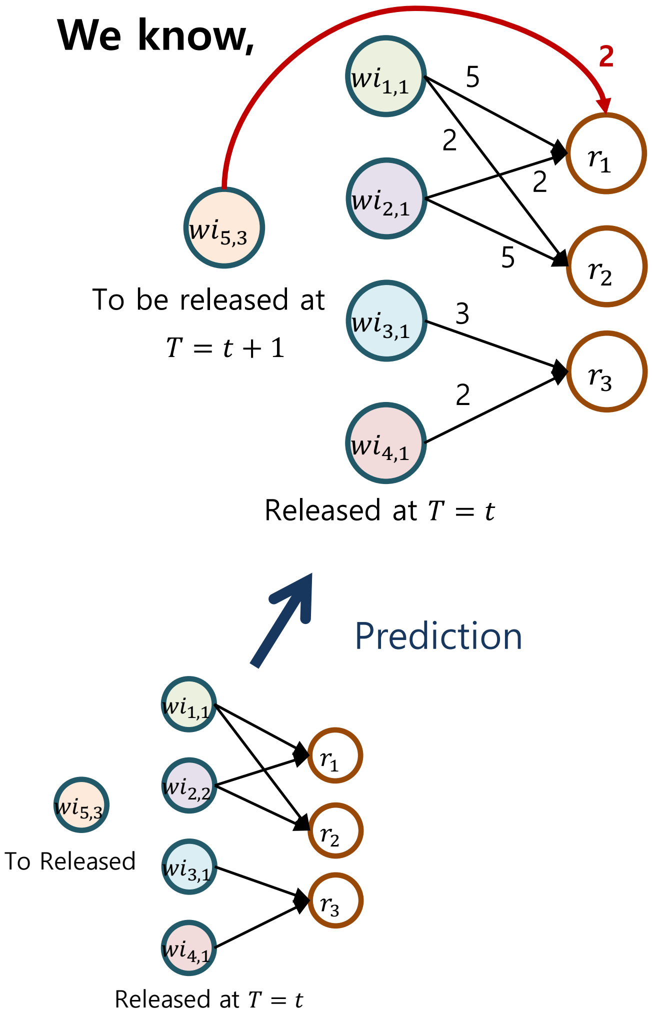
- I_5 is currently at its 2nd operation.
- We first predict its next activity a_1
- Next, we predict the processing time of a_1 by resource r_1.





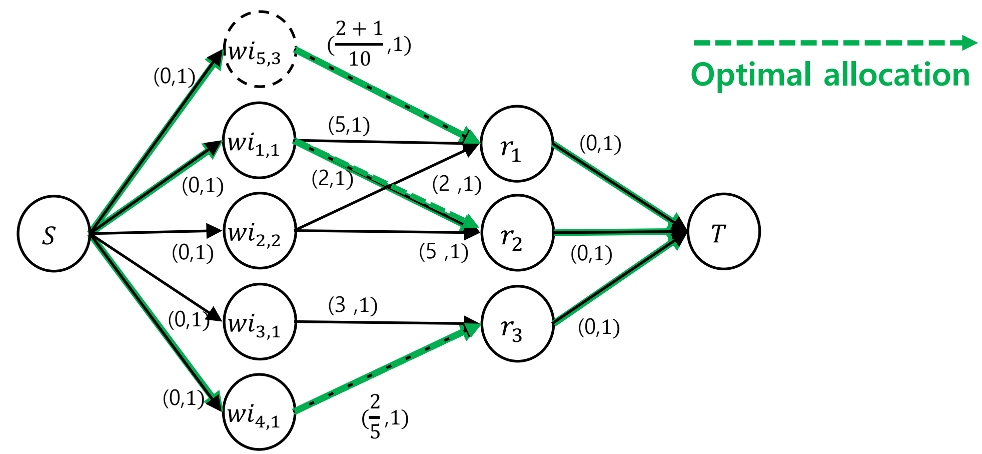
Method – Phase 2

- **Step 3: Scheduling**
  - In this step, we find an optimal scheduling by solving a min-cost max-flow network problem.



Cost function is designed to minimize total weighted completion time

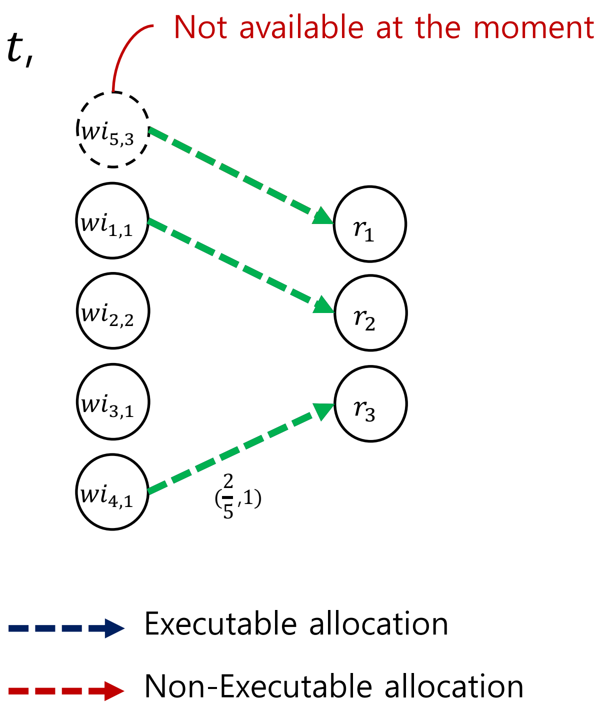
1. Connect source(sink) node to $\widehat{WI}(\hat{R})$. Edges have cost of 0 and capacity of 1.
2. If a work item can be processed by a resource, add edges with (*cost*, *capacity=1*).
3. Apply min-cost max-flow algorithm to find the optimal allocations.



Method – Phase 2

- **Step 4: Executing resource allocation**
  - In this step, we classify the optimal allocations into **executable and non-executable allocations** and then execute only the executable allocations.
    - *Executable allocation* : both instance and resource are available at the moment
    - *Non-executable allocation* : either instance or resource is not available at the moment

At $T = t$,



	t	$t+1$	$t+2$	$t+3$	$t+4$	$t+5$	$t+6$	$\sum w_i C_i$
r_1	$wi_{1,1}$					$wi_{5,3}$		65
r_2	$wi_{2,2}$							5
r_3	$wi_{4,1}$		$wi_{3,1}$					15

85

improve

Here

	t	$t+1$	$t+2$	$t+3$	$t+4$	$\sum w_i C_i$
r_1		$wi_{5,3}$		$wi_{2,2}$		25
r_2	$wi_{1,1}$					2
r_3	$wi_{4,1}$		$wi_{3,1}$			15

42

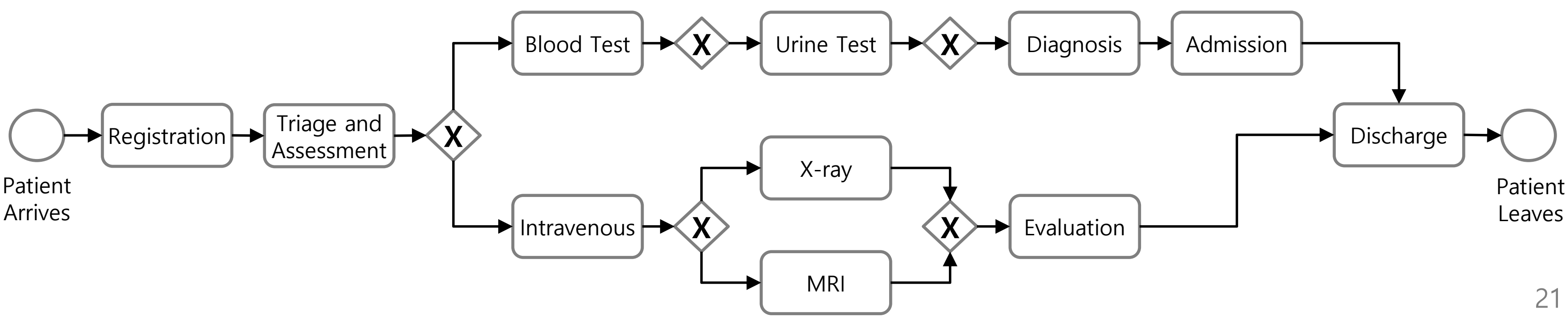
<Result of resource allocation>

Evaluation

- Artificial event log
- Real-life event log

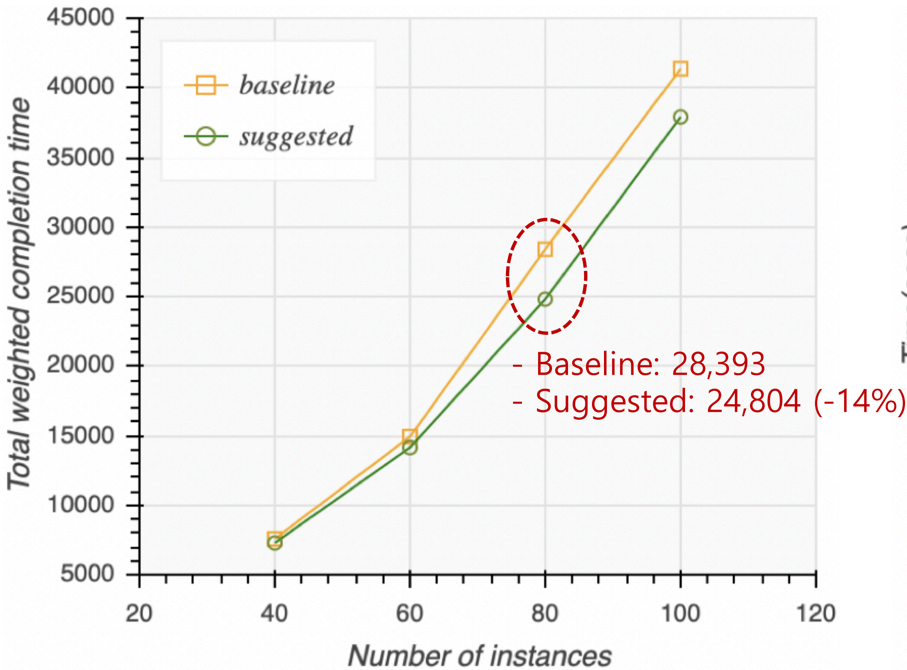
Evaluation – Artificial event log

- **Experimental design**
  - Procedure
    1. Design a business process and generate historic data and current data by simulating it.
    2. Compare our proposed method with baseline approach in terms of **total weighted completion time and computation time by varying the number of instances**.
  - Process description
    - Emergency treatment process at a hospital with 11 activities and 25 resources
    - Each resource has different skills and proficiency level.
    - Patients with different weights (1~10) come into the process in a regular interval.
  - Log Generation
    - Historic data: 7 days, 1,000 instances
    - Current data: 6 hours, 40~120 instances

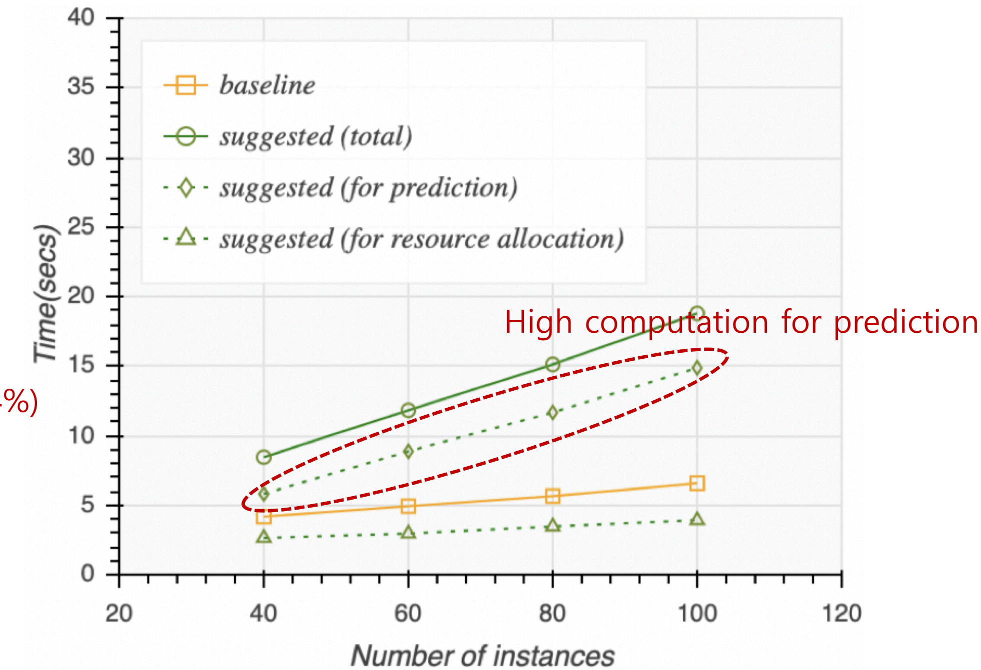


Evaluation – Artificial event log

- **Results**
  - Total weighted completion time and computation time, given the different number of instances.



<Total weighted completion time of varying |I|>



<Computation time of varying |I|>

Evaluation – Real-life event log

- **Experimental design**
  - Procedure
    1. Generate historic data and current data by splitting the real-life log.
    2. Compare our proposed method with baseline approach in terms of **total weighted completion time and computation time**
  - Process description
    - Application procedure for a personal loan at a global financing organization (BPIC'12)
    - 7 activities and 48 resources
    - 13,087 cases and 262,200 events from Oct. 2011 to Mar. 2012
    - According to the case attribute "*AMOUNT_REQ*", we assign the weight (1~10) to each instance.
  - Log split
    - Historic data: events before 10th Mar. 2012
    - Current data: 10th Mar. 2012
      - ✓ contains 110 instances, each conducting 3 activities on average

Evaluation – Real-life event log

- **Results**
  - Total weighted completion time and computation time.
    - Total weight completion time of the proposed method is **42 percent lower** than the one of baseline approach.
      - ✓ assigning the most efficient resources and reserving some resources for future allocation
    - The computation time is much higher in the proposed method.
      - ✓ each work item has many resource options → high computation for predicting the parameters (110.1 out of 115.6)

<Experimental result on real-life event log>

Method	Total weighted completion time	Computation time(secs)
Baseline	1479	7.6
Suggested	1038 (-42%)	115.6

For prediction: 110.1 secs
For scheduling: 5.5 secs

Conclusion

- Contribution
- Limitation
- Future works

Conclusion

- **Contribution**
  - In this paper, we suggest a **concrete method to improve a business process using results from predictive business process monitoring**.
  - To this end, we adopt **the time and next event prediction technique based on LSTM** and **min-cost max-flow algorithm** to optimize online resource scheduling.
  - We verify the effectiveness and efficiency of the proposed method on **both an artificial log and a real-life log**.

- **Limitation**
  - Our proposed method relies heavily on the **performance of the prediction model**.
  - The **computation time** is relatively higher than the baseline approach.

Conclusion

- **Future work**
  - We will conduct additional experiments such as **the effect of the prediction accuracy on the performance**.
  - We will extend this two-phase method to achieve **another goal** such as minimizing the potential risks in the business process **by predicting other relevant parameters and defining a relevant cost function of network arcs**.
  - Another direction for future work is to extend the proposed method by **adopting advanced dispatching techniques**.

Q&A

POSTECH
POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY