POMS 2017

28th annual conference

University of Michigan

Multi-Job Production Systems

Pooya Alavian[1] | Peter Denno[2] | Semyon M. Meerkov[1]

1 University of Michigan, Ann Arbor, MI

2 National institute of Standards and Technology (NIST), Gaithersburg, MD

Outline

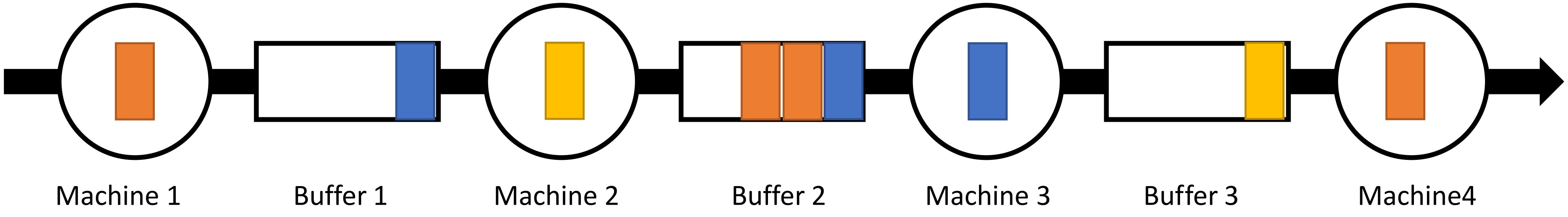
- Introduction: MJP systems
- Problems
- Solutions and results
- Case study
- Summary and future work

Introduction

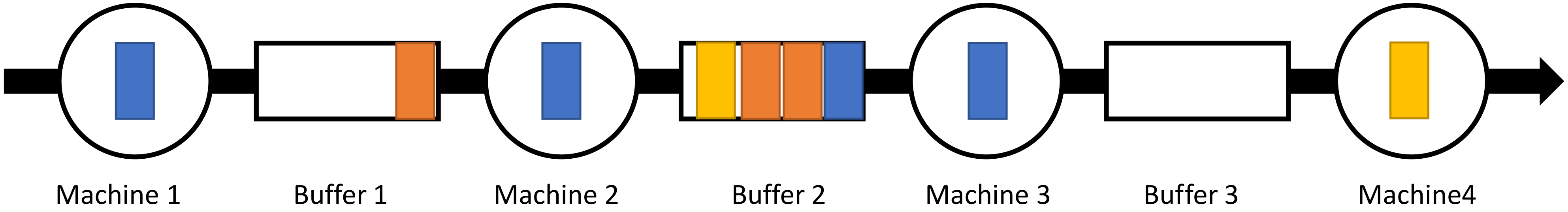
Multi-Job Production (MJP) Systems

- Jobs are released one-by-one according to a product-mix and build schedule.
- Different job-types require different amount of work at some or all operations.
- Processing times are deterministic and job-dependent.
- The setup times are zero.
- In-process buffers are non-dedicated.
- Jobs are processed on a FIFO basis.
- The machines are unreliable and experience random breakdowns.

MJP Systems illustration



MJP Systems illustration



Who uses MJP

- Auto industry
  - Assembly plants
  - Engine and transmission plants
  - Battery plants
  - Suppliers
- Household appliance industry
- Computer industry

Problems

The Problem

- The relation between throughput, bottleneck and product-mix is not known.
- Performance losses of 10-20% consistently observed in MJP plants
- No theory exists to address these losses

MJP Theory

- MJP Theory addresses the following problems

Performance analysis

Calculate performance characteristics of MJP systems (e.g., TP, WIP, probability of machine blockage and starvation) as functions of the product-mix.

Continuous improvement

Determine MJP system bottlenecks, indicate a way to alleviate them, and quantify the resulting performance improvement as functions of the product-mix.

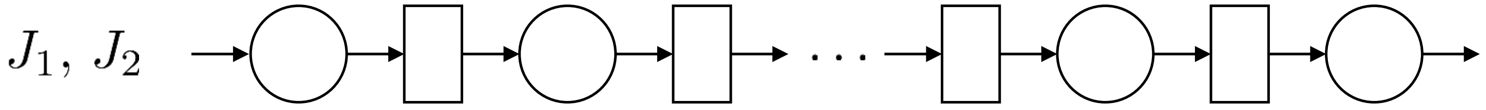
- MJP Toolbox

MJP Toolbox

Provide a software package to make the theory usable to industry and academia, via both real-time and off-line analysis capabilities.

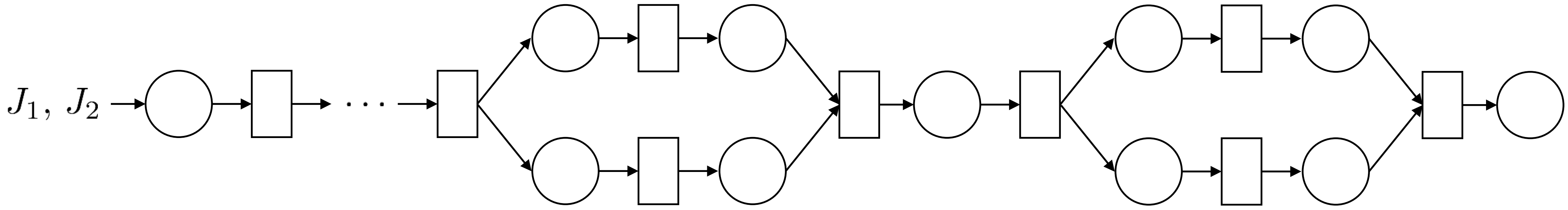
Different types of MJP systems

- Serial line



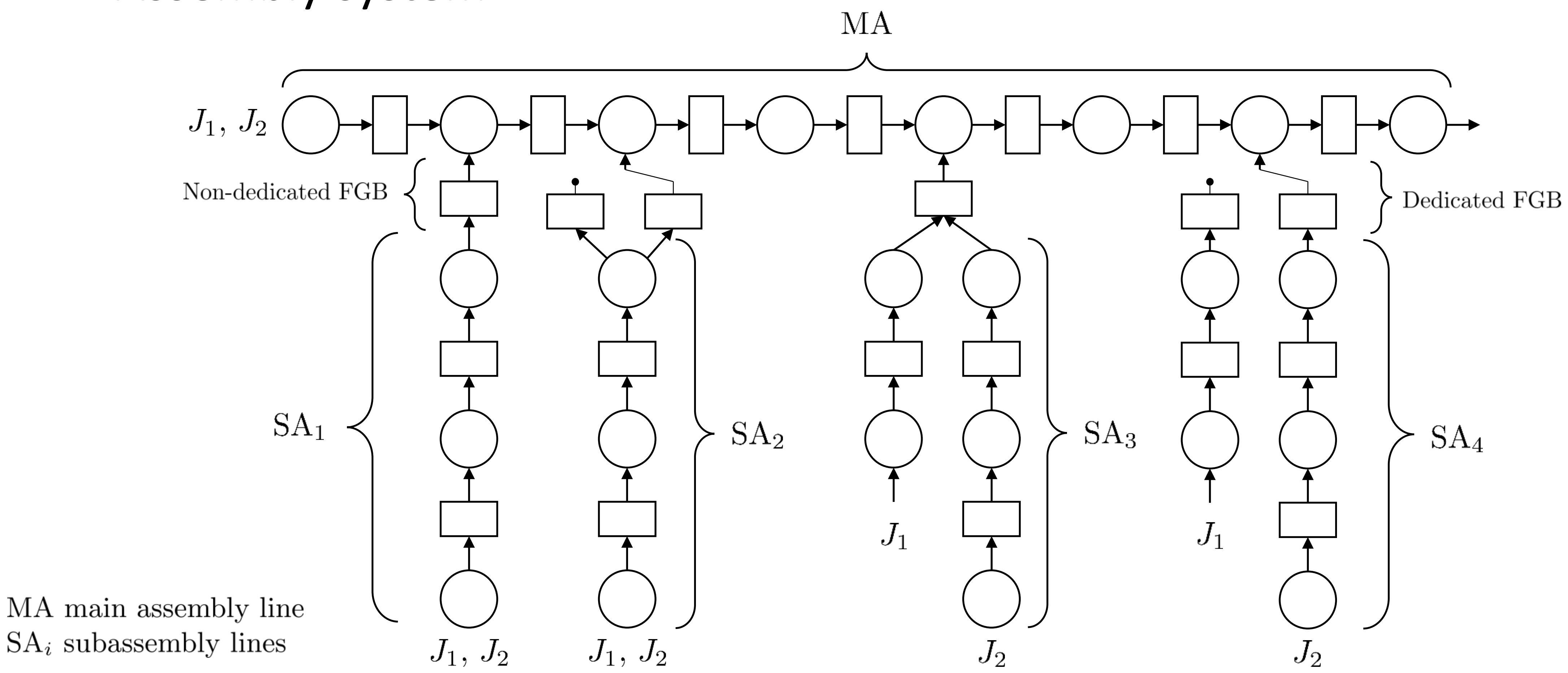
Different types of MJP systems

- Hybrid serial line



Different types of MJP systems

- Assembly system



Solutions and results

Work-based model

- Work-based model
  - Machines produce work
    - Work capacity of machine i: W_i
  - Jobs require work
    - Work-requirement of job j at machine i: w_{ij}
  - Different job-types require different amount of work.
    - Processing time of job-type j at machine i: $\tau_{ij} = w_{ij}/W_i$

  - Suitable for analysis of MJP and Single-Job Production (SJP) systems

Performance analysis for serial lines

- Three-step procedure

Performance analysis for serial lines

- Three-step procedure

1. Virtual SJP system
2. Performance estimation
3. Conversion to MJP

For the given product-mix, create a virtual SJP system processing a virtual job whose work-requirements are weighted (by product mix) average of jobs' work requirements ($w_{i,v} = \sum_j r_j w_{ij}$)

Performance analysis for serial lines

- Three-step procedure

Performance of the virtual system is estimated using the methods available in the literature. We used aggregation method from Li and Meerkov (2007).

Performance analysis for serial lines

- Three-step procedure

Estimate the performance metrics of the original MJP line from the virtual SJP system.

Accuracy

- Accuracy of the conversion from MJP to virtual SJP is very high
  - Less than 0.2% error in TP for 10-machine lines
- Accuracy of step 2 depends on the method used
  - We used aggregation method by Li and Meerkov, 2009
  - Around 4% error in TP for 10-machine lines
- Conversion from SJP back to MJP does not introduce additional errors.

Classification of MJP systems

- Two distinct behaviors of TP as function of product-mix

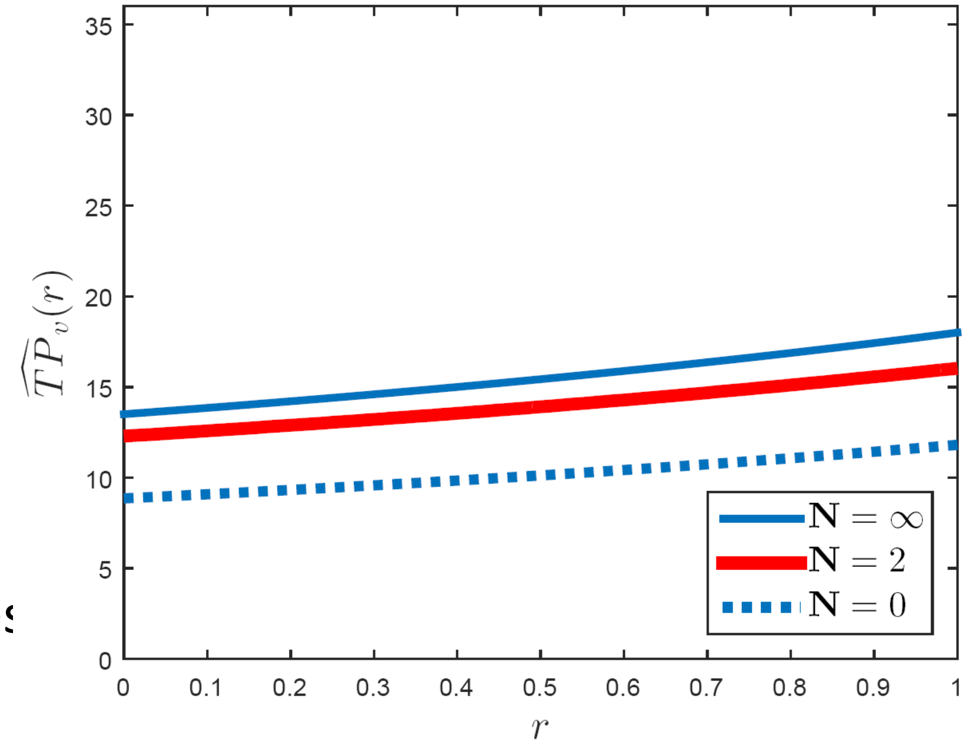
- **DEFINITION.** Two job-types are non-conflicting if $BN_{J1} = BN_{J2}$
- Otherwise, they are conflicting.

For illustration purposes, systems with 2 job-types are presented in the rest of the presentation.

Product-mix is denoted as $r_1 = r, r_2 = 1 - r$

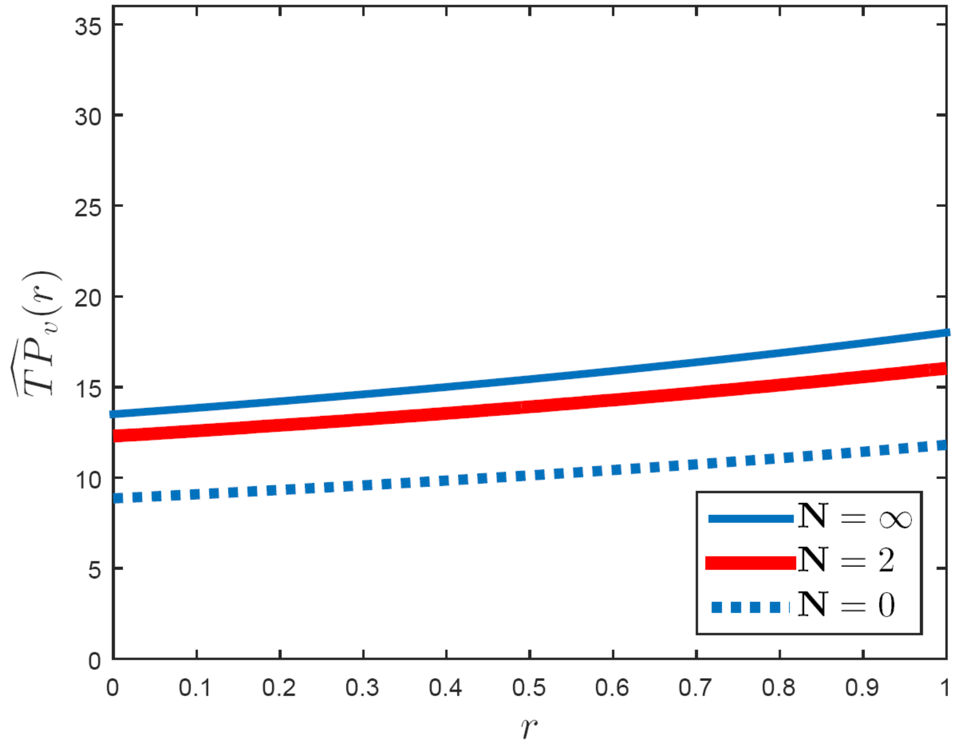
Non-conflicting jobs

- **Theorem.** If J1 and J2 are non-conflicting with common bottleneck m_k, and all buffers are of zero or infinite capacity
- m_k is BN for any r
- $TP_v(r)$ is given by
  - $TP_v(r) = \frac{1}{\frac{r}{TP_{J1}} + \frac{1-r}{TP_{J2}}}$
- $TP_v(r)$ is
  - Strictly monotonically increasing if $TP_{J1} > TP_{J2}$
  - Strictly monotonically decreasing if $TP_{J1} < TP_{J2}$
  - Constant if $TP_{J1} = TP_{J2}$
- **Numerical fact.** The above results hold true for lines with any buffers. Analyzing 25000 lines:
  - BN remaining the same machine in 99.2% of cases
  - TP is monotonic in 92.8% of cases
  - Accuracy of (b) is above 99% in 95.8% of cases



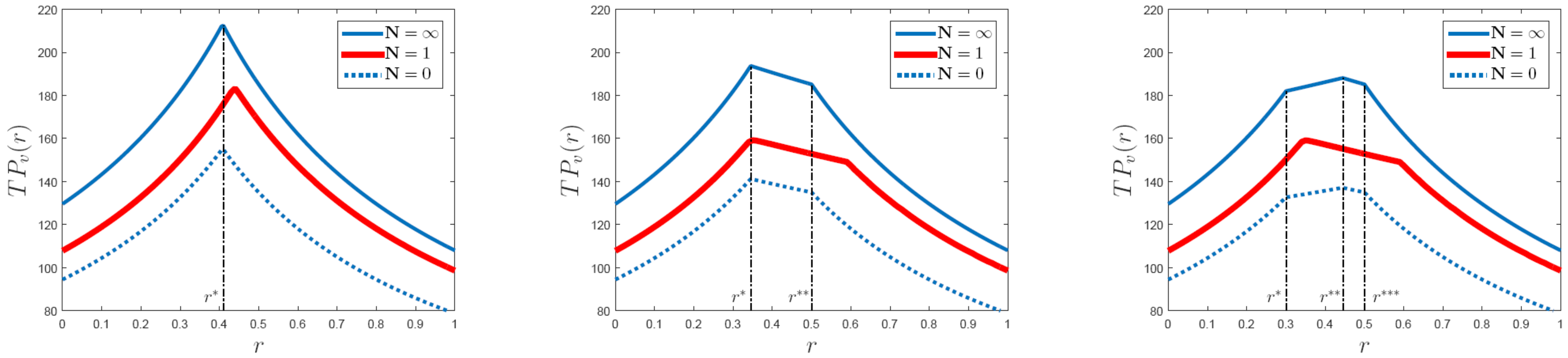
Non-conflicting jobs

- Intuitive behavior
- Not linear
- No BN switches
- Easily expandable to more than two job-types



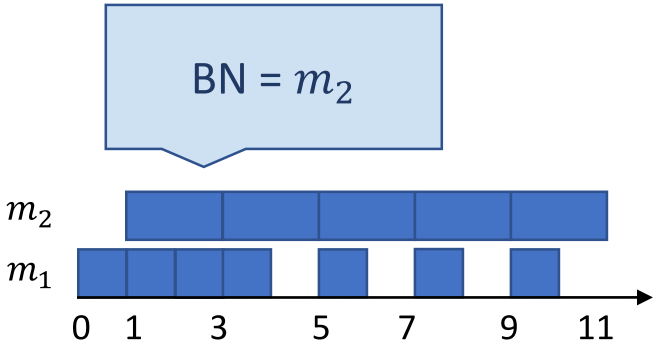
Conflicting jobs

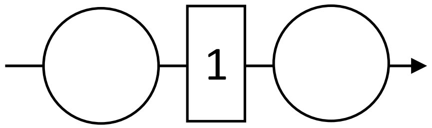
- **Theorem.** Consider an M-machine line producing conflicting jobs J1 and J2, i.e., $BN_{J1} \neq BN_{J2}$. If buffers are $N = 0$ or $N = \infty$, then
- BN has at most $M - 1$ switches in the interval of r
- If there are K BN switches, then $TP(r)$ has $K + 1$ intervals of continuous differentiability.
- If $w_{BN_{J1},1} > w_{BN_{J1},2}$ and $w_{BN_{J2},1} < w_{BN_{J2},2}$, then there is an interval $R \subset [0,1]$, at which $TP(r) > TP_{Jj}$



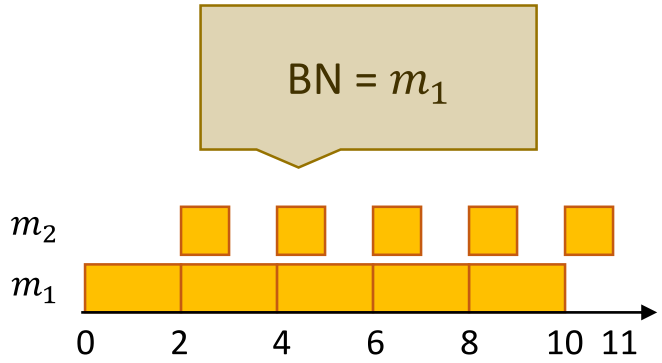
Conflicting jobs

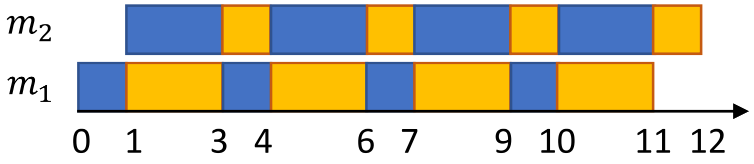
- Why $TP(r)$ acts non-monotonically?





Cycle times			TP
Job-type1	1	2	0.5
Job-type2	2	1	0.5





Both BNs are utilized

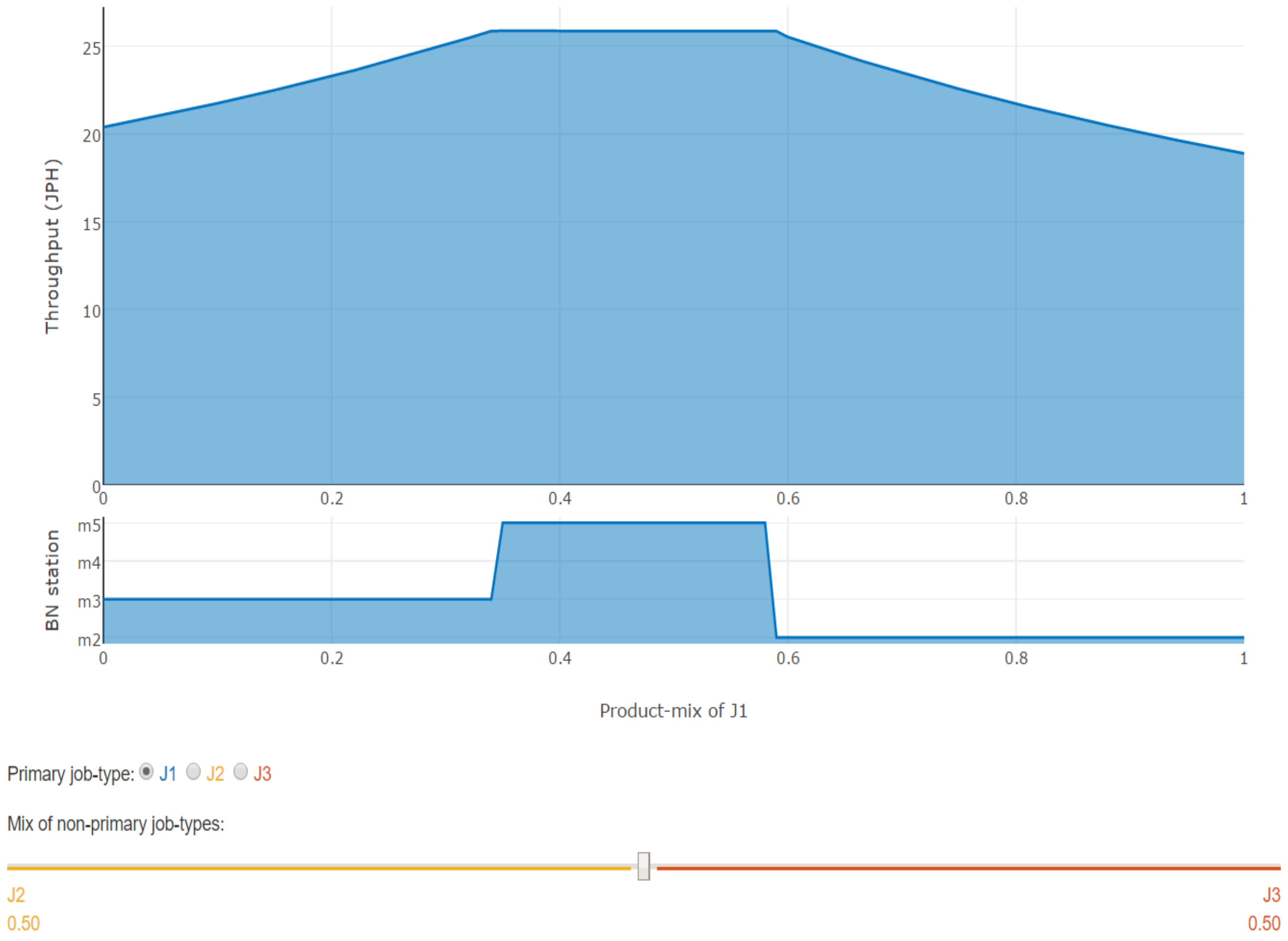
$$TP = \frac{8}{12} = 0.66$$

33% more production

Performance portrait

- Graphically presents throughput and bottlenecks as functions of the product-mix

- Works with systems with more than two job-types by introducing additional degrees of freedom.

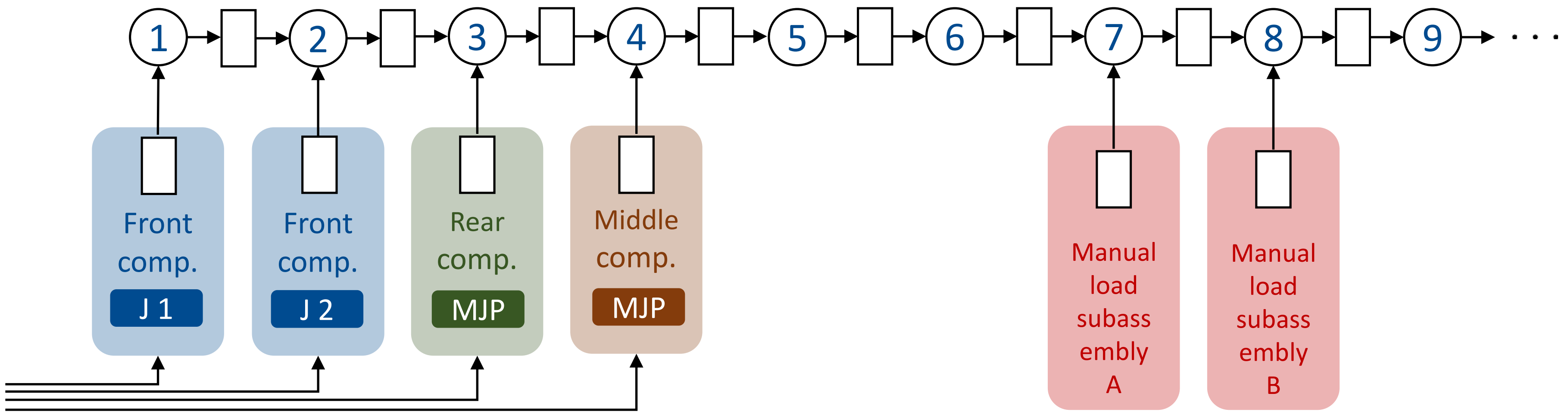


Case study

The data presented in the following slides are modified due to confidentiality purposes

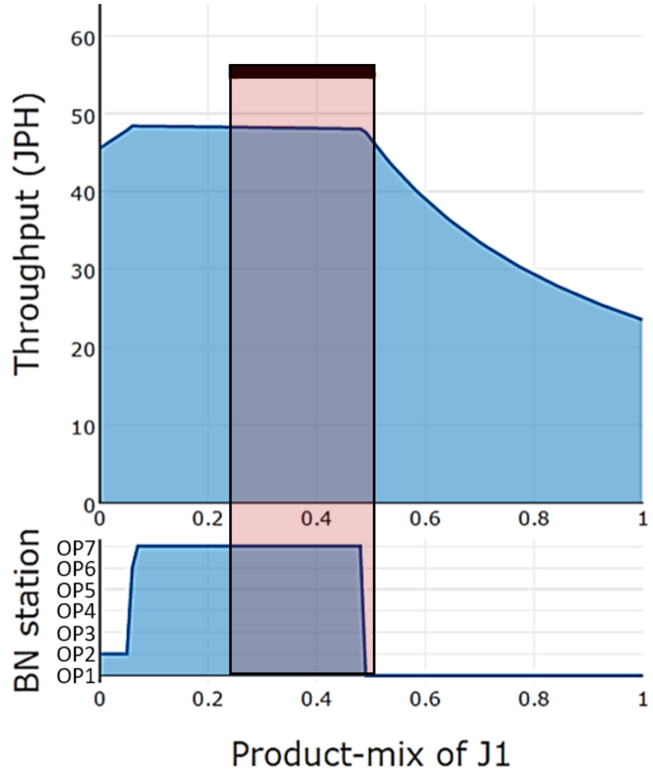
Production System

- Automotive assembly plant > Body shop > Underbody
- Two job-types
- Study period: 14 weeks



Current system

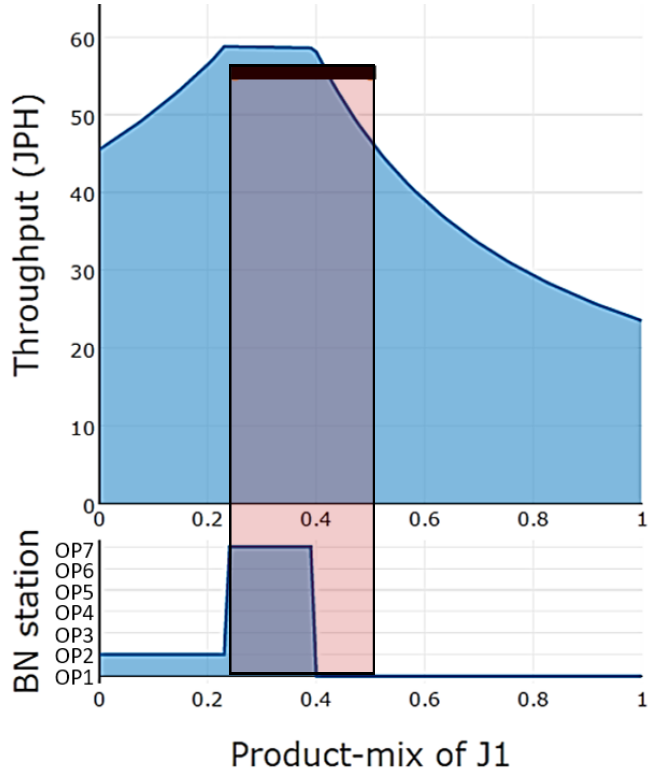
- Body shop target TP: 55 JPH
- Body shop current TP: 49 JPH
- Typical range of product-mix of job 1
  - $r_1 \in [25\%, 50\%]$
- Bottleneck OP7



Continuous improvement project

- Improve OP7
  - Prioritize maintenance
  - Prioritize delivery
  - Improve manual loading

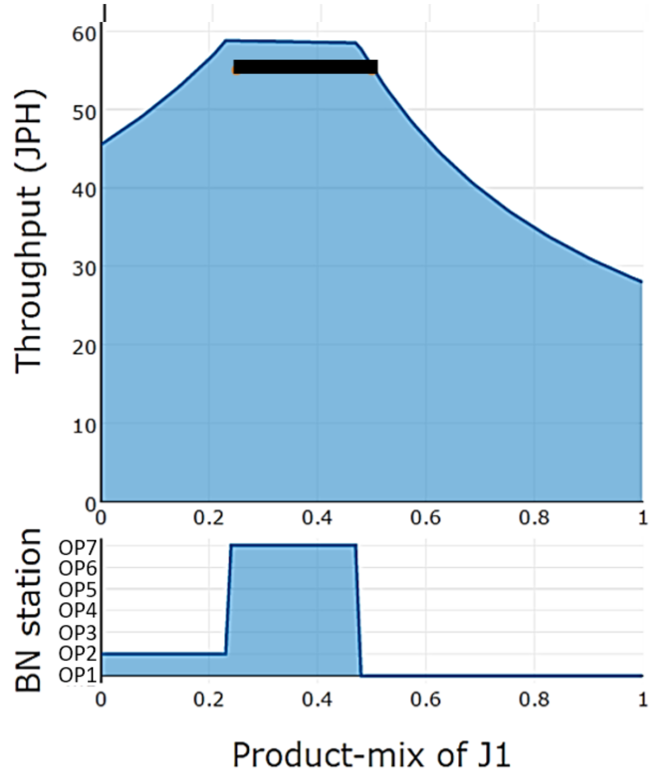
- Result



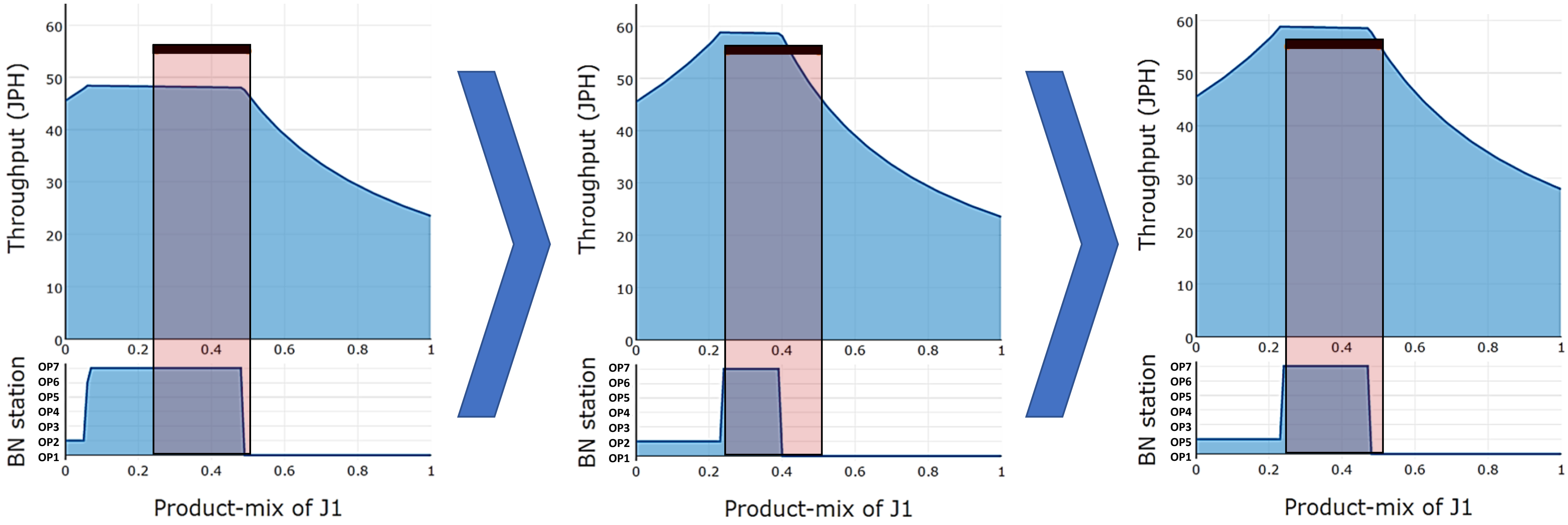
Continuous improvement project

- Improve OP1
  - Look into Front comp. line for J1
  - Model Front comp. line
  - Find BNs
  - Design improvement plan

- Result



Continuous improvement project



Summary

Summary

- MJP systems are widely used in the industries
- 15%-20% of their capacity is wasted
- MJP theory helps recover these losses
- Performance portraits are quick and easy way to understand system behavior and plan continuous improvement projects

Future work

- Develop the theory for the assembly systems
- Improve overall MJP accuracy by improving the accuracy of current methods for SJP systems analyses
- Integrating MJP tools especially performance portrait into plants' production management dashboard

Thank you

Continuous improvement

- Bottleneck identification
- Bottleneck: machine whose capacity affects throughput the most
- m_i is bottleneck iff $w_{i,v}(r)\frac{\partial TP}{\partial W_i} > \frac{\partial TP}{\partial W_k} w_{k,v}(r)$, for all machine $k \neq i$
- Alternate approach: arrow method
  - Use blockage and starvation data
  - Compare BL_i with ST_{i+1} and draw an arrow toward the smaller
  - The machine(s) with no emanating arrows is the bottleneck

Work-based model

- WBM characterizes the production system components
- Machines (Operations)
  - Work capacity
  - Reliability characteristics (MTBF/MTTR/…)
  - Starvation/blockage measurement policy
- Buffers
  - Non-dedicated vs. dedicated
- Jobs
  - Work requirement at each operation
- Release
  - According to product-mix or build-schedule

Accuracy

- Step one has high accuracy
  - Less than 0.6% discrepancy in BN identification of 5-machine lines
- Step two
  - Lower accuracy; 10% discrepancy in BN identification of 5-machine lines
- Step three
  - No additional errors are introduced

Performance analysis for serial lines

- Three-step procedure
  - For any product-mix, create the corresponding virtual SJP system
  - Calculate (estimate) performance of the virtual system using the methods available in the literature
  - Convert the results back to original MJP system coordinates