An inspection-based compositional approach to the quantitative evaluation of assembly lines

Marco Biagi[1] Laura Carnevali[1]
Tommaso Papini[1] Kumiko Tadano[2] Enrico Vicario[1]

[1] Department of Information Engineering, University of Florence, Italy
[2] Data Science Research Labs, NEC Corporation, Kawasaki, Japan

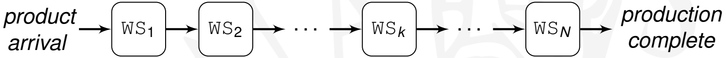
IMT Lucca,
31st October 2017

- Assembly line analysis
  - for transient measures evaluation
  - subsequent to an inspection at steady-state
- subject to ambiguity
  - on the logical state
  - and on the remaining times

Why analyse assembly lines?

- Maximize throughput and efficiency
  - speeding-up/slowing-down workstations in order to balance throughput
  - identify bottlenecks and focus resources on them
- Minimize resources and energy consumption
  - lower power consumption before bottlenecks
- Adapt production during runtime
  - Industrie 4.0
  - e.g. schedule personnel/resources in a tactic perspective

Assembly line

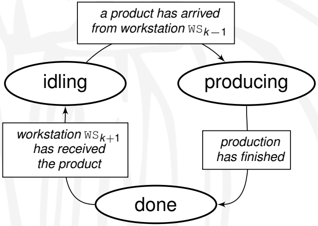


N sequential workstations $\mathtt{WS}_1, \dots, \mathtt{WS}_N$

- with transfer blocking
- and no buffering capacity

Workstation $\mathtt{WS}_k$ can be in one of three states

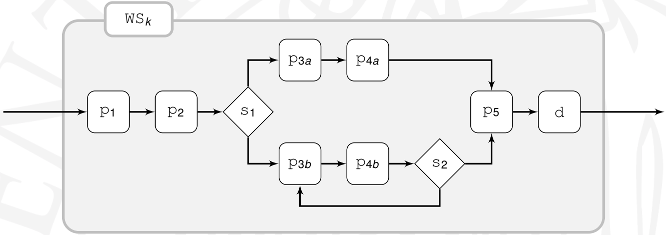
- *producing*: $\mathtt{WS}_k$ is working on a product
- *done*: $\mathtt{WS}_k$ is done working on a product
- *idling*: $\mathtt{WS}_k$ is waiting for a new product



Workstation

Each workstation $\mathtt{WS}_k$

- has no internal parallelism
  - at most one item being processed in each workstation
- can implement complex workflows
  - sequential/alternative/cyclic phases with random choices
- and has GEN phases' durations



The last phase has no duration and encodes the *done* state

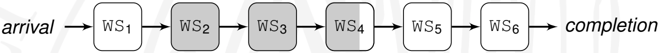
Underlying stochastic process

The underlying stochastic process of each isolated workstation is a Semi Markov Process (SMP)

- due to GEN durations
- and the absence of internal parallelism

The whole assembly line finds a renewal in any case where

- every *done* station is in a queue before a bottleneck
- and everything else is *idling*



Inspection with partial observability

The assembly line can be inspected by external observers

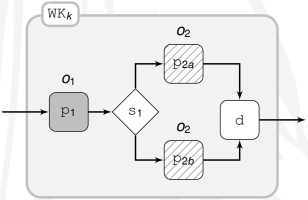
- the line can be considered at steady-state at inspection
- there can be ambiguity about the current phase

An observation is a tuple $\omega = \langle \omega_0, \omega_1, \ldots, \omega_N \rangle$

- ω_0 indicates if a new product is ready to enter the line or not
- $\omega_k = \langle \sigma_k, \phi_k \rangle$ refers to $\mathtt{WS}_k$
  - σ_k indicates if $\mathtt{WS}_k$ is *idle*/*producing*/*done*
  - ϕ_k identifies the set of possible current phases

Two kinds of uncertainty

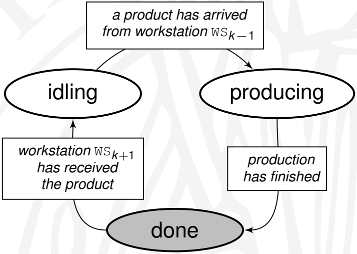
- about the actual current phase
  - discrete
- about the remaining time in the current phase
  - continuous



Performance measures

Time To Done

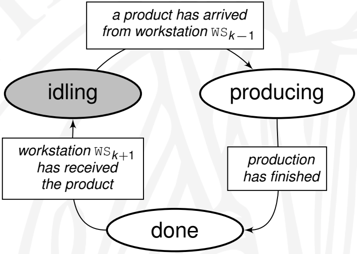
The remaining time until workstation k,
according to observation ω, reaches the *done* state



Performance measures

Time To Idle

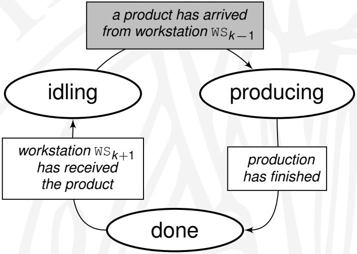
The remaining time until workstation k,
according to observation ω, reaches the *idling* state



Performance measures

Time To Start Next

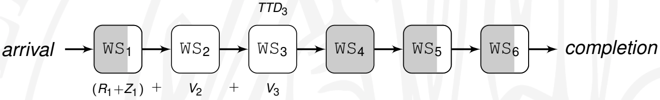
The remaining time until workstation k,
according to observation ω, starts the production of a new product



Time To Done

$$
\mathrm{TTD}(k,\omega) := \begin{cases} \sum_{\gamma \in \phi_k} P_{k,\gamma,\omega} \cdot (R(k,\gamma) + Z(k,\gamma)), & \text{if } \sigma_k = \textit{producing} \\ \mathrm{TTD}(k-1,\omega) + V(k), & \text{if } \sigma_k = \textit{idling} \\ 0, & \text{if } \sigma_k = \textit{done} \end{cases}
$$

- $P_{k,\gamma,\omega}$ steady-state probability that $\texttt{WS}_k$ is in phase γ according to ω
- $R(k,\gamma)$ *remaining time* in phase γ of $\texttt{WS}_k$
- $Z(k,\gamma)$ *execution time* of phases of $\texttt{WS}_k$ that follow γ
- $V(k)$ *production time* of $\texttt{WS}_k$

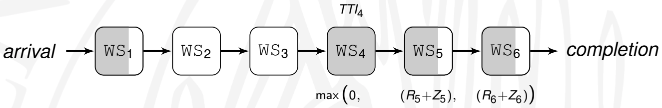


Backward recursive evaluation

Time To Idle

$$\mathrm{TTI}(k,\omega) := \begin{cases} \max\{\mathrm{TTD}(k,\omega), \mathrm{TTI}(k+1,\omega)\}, & \text{if } \sigma_k \in \{\textit{producing}, \textit{done}\} \\ 0, & \text{if } \sigma_k = \textit{idling} \end{cases}$$

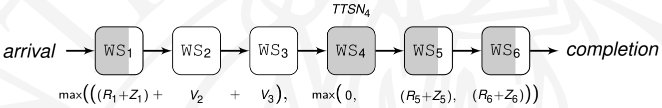
- $\mathrm{TTI}(k,\omega) = \max\{\mathrm{TTD}(k,\omega), \ldots, \mathrm{TTD}(k+n,\omega)\}$
  - $\mathtt{WS}_j$ producing/done $\forall j \in [k, k+n]$
  - either $\mathtt{WS}_{k+n}$ last workstation or $\mathtt{WS}_{k+n+1}$ idling
- $\mathtt{WS}_k$ becomes idle when the bottleneck finishes its production



Forward recursive evaluation

Time To Start Next

$$\mathrm{TTSN}(k,\omega) := \max\{\mathrm{TTI}(k,\omega), \mathrm{TTD}(k-1,\omega)\}$$



Forward and backward recursive evaluation

Disambiguation of observed phases

Resolve observed (producing) phases' ambiguity

- steady-state probability that $\texttt{WS}_k$ is in phase γ according to ω

Given observation ϕ_k for workstation $\texttt{WS}_k$

- we compute probability $P_{k,\gamma,\omega}$
- that it was actually γ that produced ϕ_k

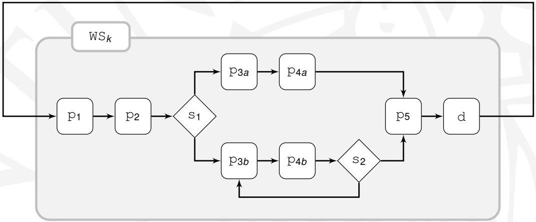
$$P_{k,\gamma,\omega} = \frac{\tilde{\pi}(\gamma)}{\sum_{\gamma' \in \phi_k} \tilde{\pi}(\gamma')}$$

- $\tilde{\pi}(\gamma)$ steady-state probability of phase γ in an isolated model of $\texttt{WS}_k$

Isolated workstation model

The isolated workstation model represents a workstation repeatedly processing a product

- one product being processed
- after its production, it's moved back to the entry point of the workstation



It can be used for two reasons

- steady-state probabilities of producing phases are independent
- the inspection is at steady-state
  - arrivals and productions can be considerer in equilibrium

Remaining time

Evaluation of $F_{R(k,\gamma)}(t) =$ CDF of $R(k,\gamma)$

- $R(k,\gamma)$ *remaining time* in phase γ of $\mathtt{WS}_k$

Problem!

- remaining times of enabled GEN transitions are *dependent*
- joint probabilities don't allow for a compositional approach

$^1/_3$ *Immediate* approximation

- assume that phase γ is inspected at its ending
  - $\tilde{F}_{R(k,\gamma)}(t) = 1 \quad \forall t$
- represents an *upper bound*

$^2/_3$ *Newly enabled* approximation

- assume that phase γ is inspected at its beginning
  - $\tilde{F}_{R(k,\gamma)}(t) = F_\gamma(t)$
  - $F_\gamma(t)$ original CDF of the duration of γ
- represents a *lower bound*

Remaining time

$^3/_3$ *Independent remaining times* approximation

- consider the remaining times of ongoing phases as *independent*
- represents a (better) *lower bound*

Theorem: positive correlation & stochastic order

If $\hat{R}$ is an independent version of vector R of positively correlated remaining times of ongoing phases, then $\hat{R} \geq_{st} R$

Steady-state distribution of $\hat{R}(k, \gamma)$
computed according to the Key Renewal Theorem[1]

$$\tilde{F}_{R(k,\gamma)}(t) = \frac{1}{\mu} \int_0^t [1 - F_\gamma(s)] ds$$

- μ expected value of $F_\gamma(t)$

[1] Serfozo, R., 2009. Basics of applied stochastic processes. Springer Science & Business Media.

Execution and production time

Evaluation of $F_{Z(k,\gamma)}(t)$ and $F_{V(k)}$

- $Z(k,\gamma)$ *execution time* of phases of $\mathtt{WS}_k$ that follow γ
- $V(k)$ *production time* of $\mathtt{WS}_k$

CDFs of $Z(k,\gamma)$ and $V(k)$ are computed as transient probabilities

- $F_{Z(k,\gamma)}$ transient probability from phase after γ to final phase of $\mathtt{WS}_k$
- $F_{V(k)}$ transient probability from first phase to final phase of $\mathtt{WS}_k$

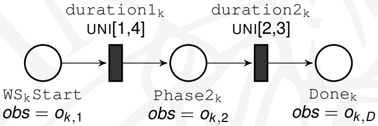
Upper/lower bounds for *TTD*, *TTI* and *TTSN* can be evaluated

- *convolution* and *max* operations maintain stochastic order

Experimentation

Models and analyses implemented through the Oris tool[2,3] APIs

- modelled through *PO-sTPN*
  - Partially Observable - stochastic Time Petri Nets
  - workstations are *state machine PNs* and *1-safe*



Experiments performed on a

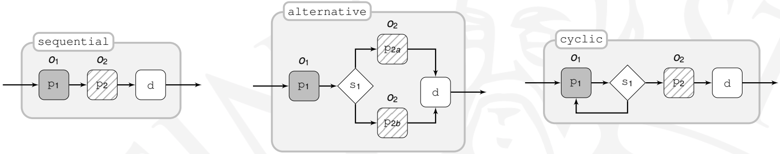
- single core of an Intel Core i5-6600K processor
- with 16GB of RAM

[2] http://www.oris-tool.org/

[3] L. Carnevali, L. Ridi, and E. Vicario, "A Framework for Simulation and Symbolic State Space Analysis of Non-Markovian Models", inInternational Conference on Computer Safety, Reliability, and Security (SAFECOMP), Lecture Notes in Computer Science, pp. 409-422,Springer Berlin Heidelberg, 2011.

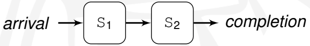
Case study assembly lines

Sequential, alternative and cyclic workstations



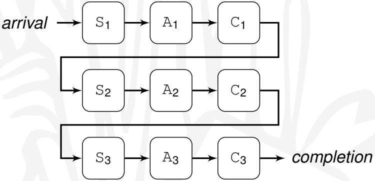
Simple assembly line

- two sequential workstations
- both in phase $\mathtt{p}_1$ at inspection



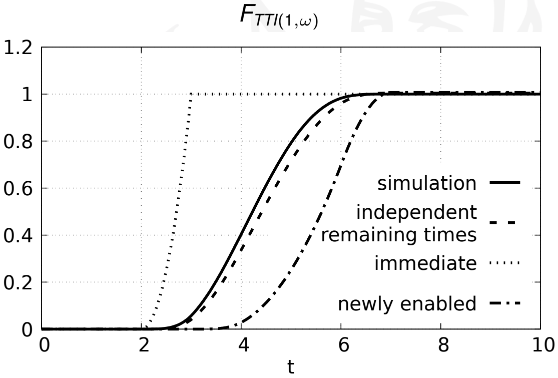
Complex assembly line

- three repetitions
- of sequential/alternative/cyclic ws
- all observed in *producing*



Simple assembly line

TTIdle



TTIdle computed in

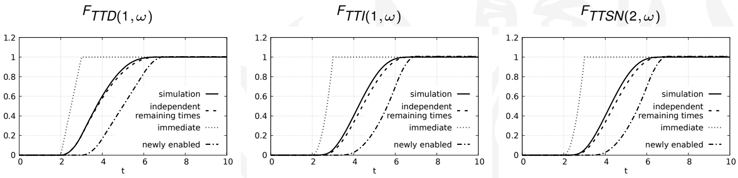
- 45 min for simulation
- 0.18 s for bounds

Due to steady-state probabilities

- steady-state probability of 0.208
- 79.2% of runs are discarded

Simple assembly line

TTDone, TTIdle, TTStartNext



TTD, *TTI* and *TTSN* computed in

- 41/45/42 min for simulation
- 0.15/0.18/0.10 s for bounds

Very good approximation results

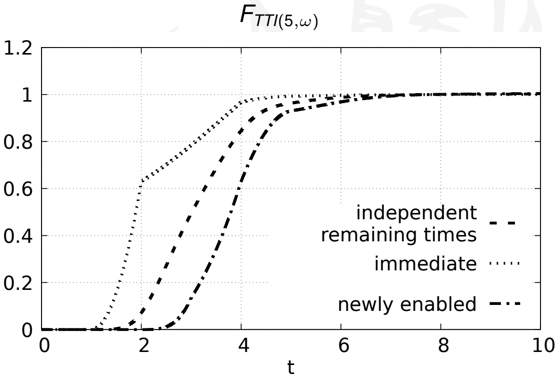
- especially for *independent remaining times*

Feasible approach

- very fast bounds evaluation
- compared to simulation

Complex assembly line

TTIdle



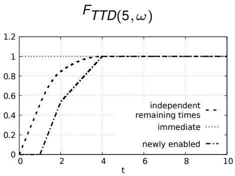
TTIdle computed in

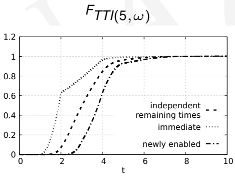
- 0.123 s for bounds

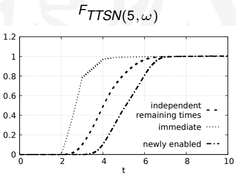
Simulation would be too computationally expensive

Complex assembly line

TTDone, TTIdle, TTStartNext







TTD, *TTI* and *TTSN* computed in

- 0.126/0.123/0.75 s for bounds

Scalable solution

- in a complex scenario
- simulation would be infeasible

Summary

We have proposed a compositional approach

- for analysis of assembly lines
  - with transfer blocking and no buffer capacity
- after inspection
  - subject to discrete/continuous ambiguity

Leveraging the positive correlation among remaining times

- we have derived stochastic bounds of performance measures
- with good and scalable results

Future directions

- introduce buffer capacity
- derive additional performance measures
  - e.g. time to complete the production of a certain product
  - or of the next N products