Performance Evaluation of Workflows Using Continuous Petri Nets with Interval Firing Speeds

Kunihiko Hiraishi
School of Information Science,
Japan Advanced Institute of Science and Technology

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY
1990

Motivation

- State space explosion in discrete-state systems
  - Necessity of methods *scalable* for the size of the model.
- Continuous relaxation in optimization problems
  - Necessary condition for the feasibility.
- Fluidification (or continuization):
  Approximating discrete transitions by fluid-flow.
  - Approximation of the reachability set.
  - Approximation of transient behavior of state variables.

Fluidification in Petri Nets

- Formalism
  - Continuous Petri Nets (Autonomous / Timed),
  - Hybrid Petri Nets,
  - Fluid Stochastic Petri Nets, etc.
- Analysis methods
  - Steady-state analysis.
  - Numerical simulations for transient behavior analysis.

Proposed approach

GSPN (Generalized Stochastic Petri Nets)

↓ Fluidification

TCPN (Timed Continuous Petri Nets)

↓ Transformation

PWLS (PieceWise Linear Systems)

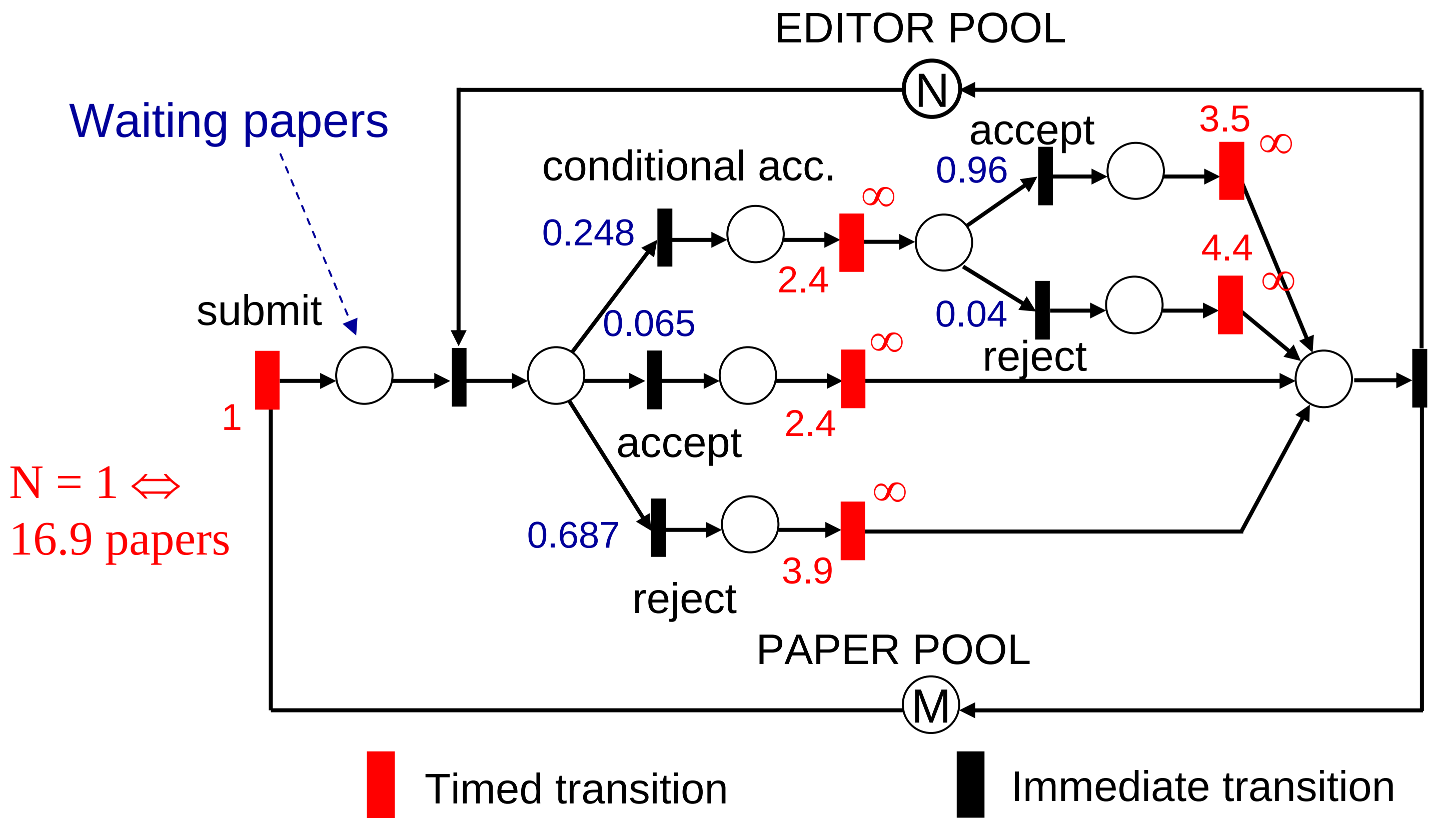
↓ Interval Method

Transient behavior
(Guaranteed enclosures for state vector)

What's new is the approach

- Computation of guaranteed enclosure in transient analysis: all possibilities are computed at the same time.
- Interval firing speeds for approximating probabilistic deviation.
- Using place invariants in piecewise interval computation as constraints.
- A prototyping computer tool.

GSPN model



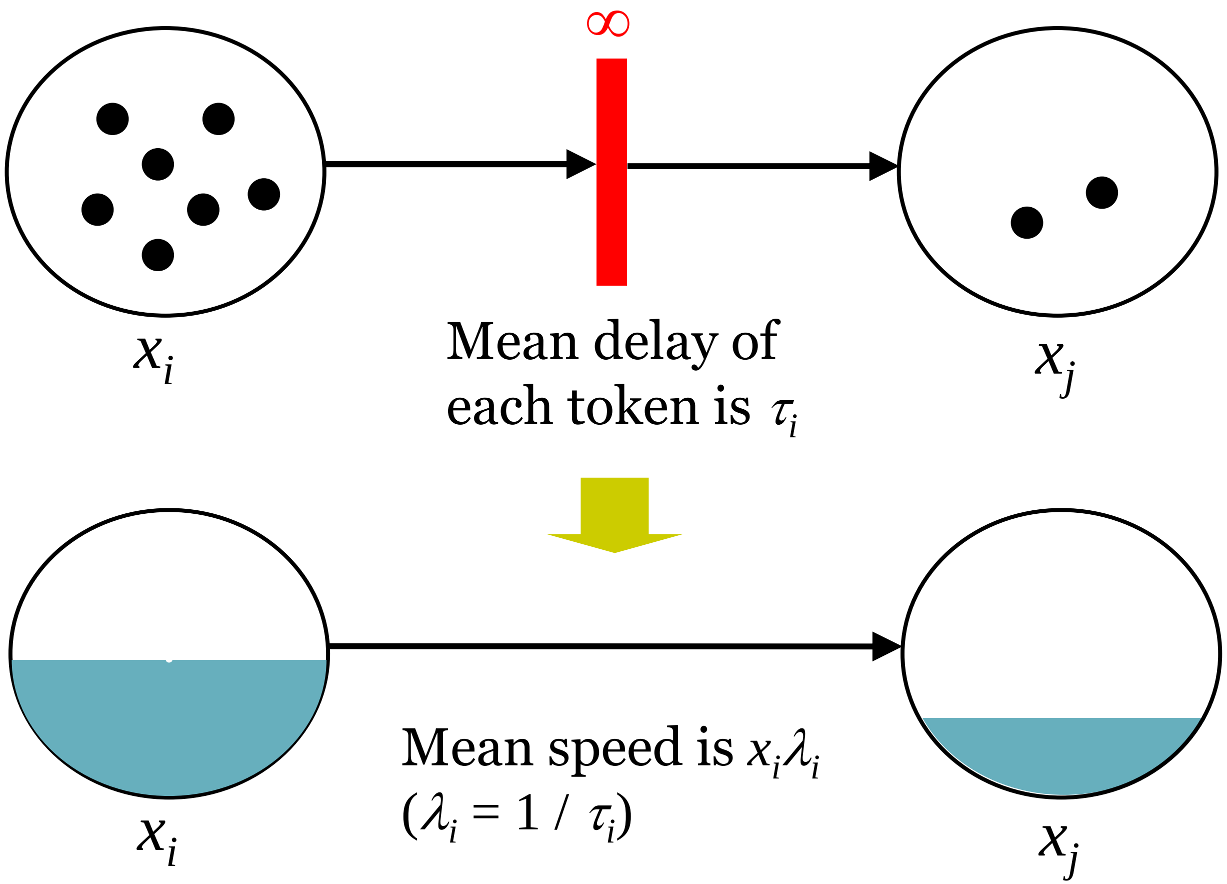
Paper review workflow

Computation results by DSPNExpress-NG

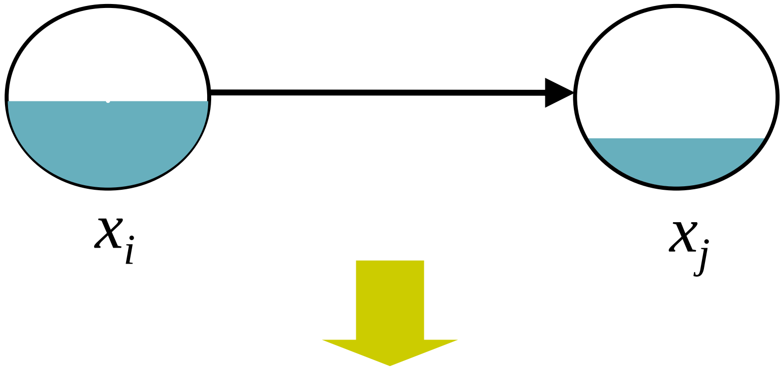
N	#states	CPU Time (sec.)	#Waiting papers	p(#paper pool = 0)
3	2926	0.3	10.18	0.30
4	8866	0.7	5.94	0.094
5	23023	2.3	1.99	0.013
6	53053	6.2	0.63	0.0021
7	110968	15	0.21	0.00049
8	213928	29	0.08	0.00020
9	384098	58	0.03	0.00010

Itanium2 1.6GHz/9MBCache, 16GB Memory

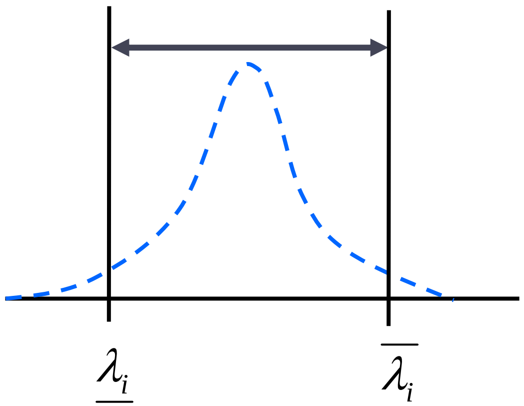
Fluidification



Approximating probabilistic deviation



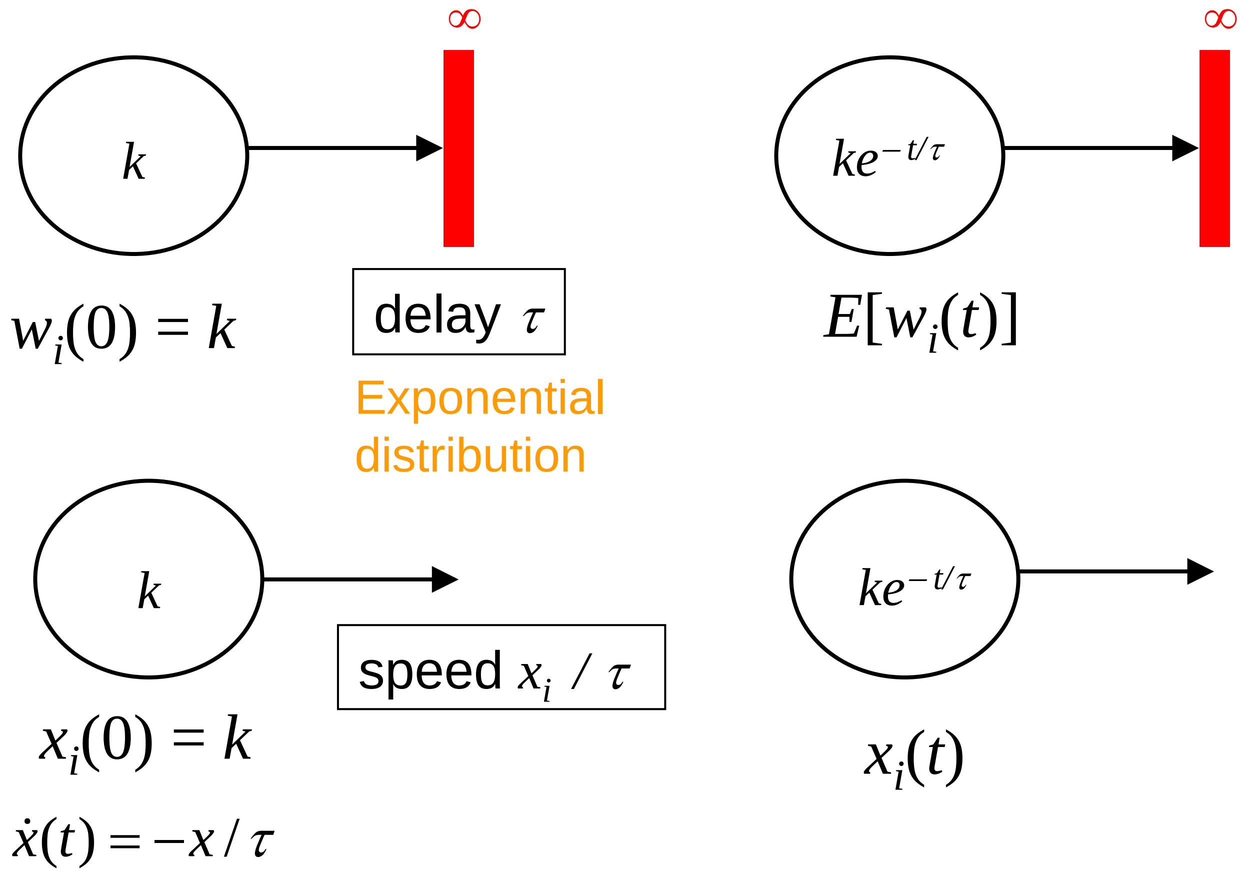
$$\dot{x}_i = -[\lambda_i] \cdot x_i$$



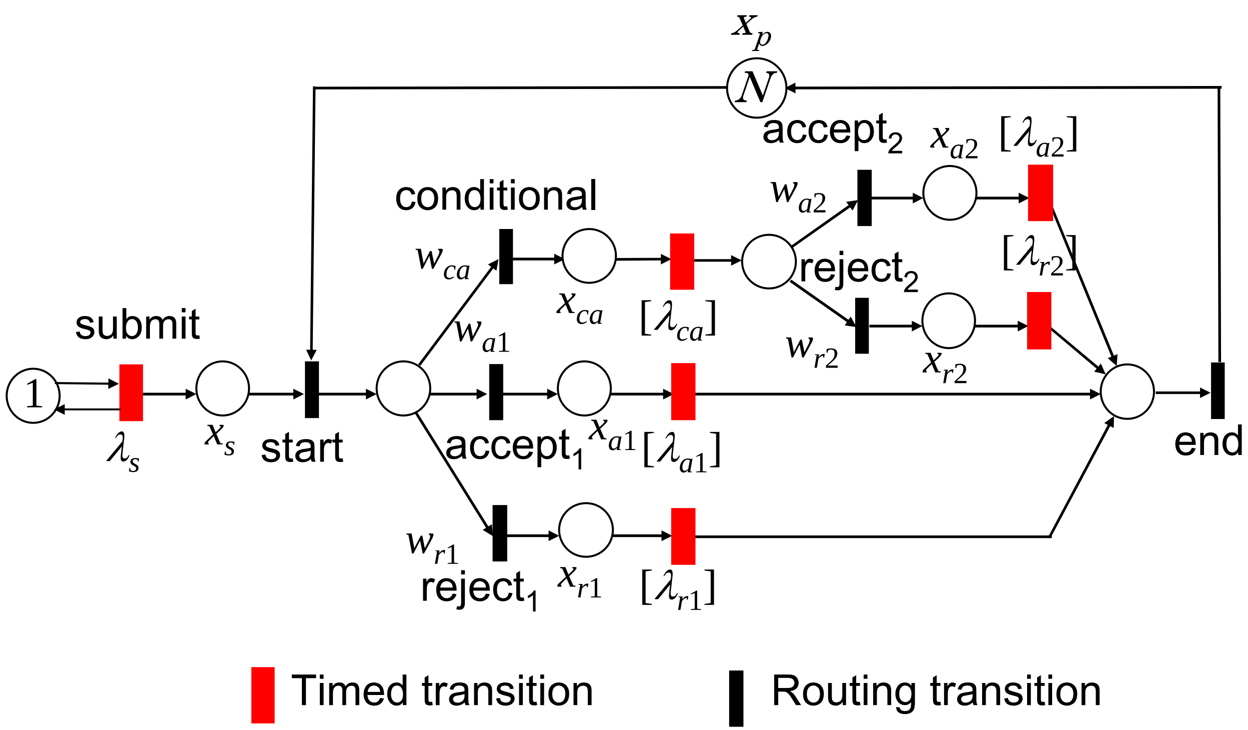
$$[\lambda_i] = [\underline{\lambda_i}, \overline{\lambda_i}]$$

Interval firing speed

Preserving expected values

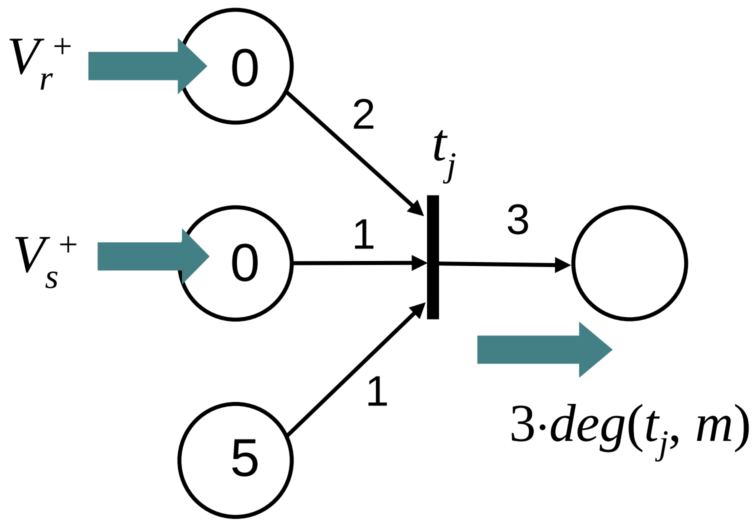


TCPN model

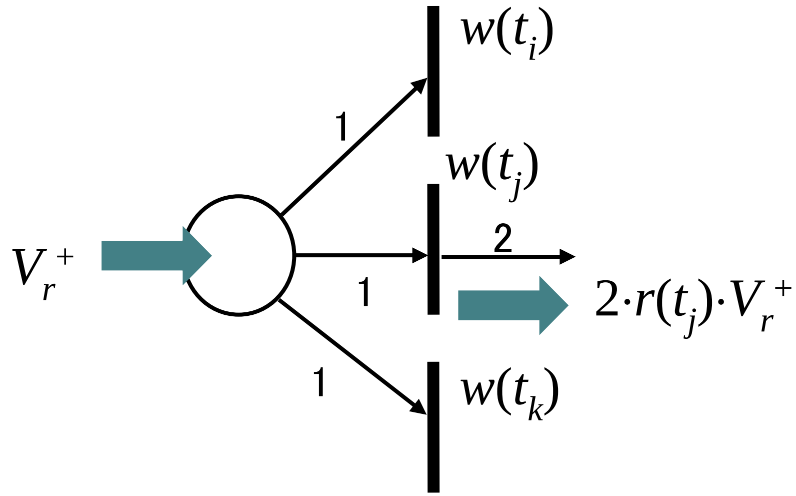


x_p
N
accept$_2$
x_{a2}
$[\lambda_{a2}]$
conditional
w_{a2}
$[\lambda_{r2}]$
w_{ca}
reject$_2$
x_{ca}
$[\lambda_{ca}]$
submit
w_{a1}
w_{r2}
x_{r2}
1
λ_s
x_s
start
accept$_1$
x_{a1}
$[\lambda_{a1}]$
end
w_{r1}
reject$_1$
x_{r1}
$[\lambda_{r1}]$
Timed transition
Routing transition

Routing transitions



$$deg(t_j, m) = min\{ V_r^+ / 2, V_s^+ \}$$



$$r(t_j) = w(t_j) / (w(t_i) + w(t_j) + w(t_k))$$

Representation by a PWL system

$$
\begin{aligned}
\dot{x}_s &= \lambda_s - R(x) \\
\dot{x}_p &= 0 \\
\dot{x}_{ca} &= \rho_{ca} R(x) - [\lambda_{ca}] x_{ca} \\
\dot{x}_{a1} &= \rho_{a1} R(x) - [\lambda_{a1}] x_{a1} \\
\dot{x}_{r1} &= \rho_{r1} R(x) - [\lambda_{r1}] x_{r1} \\
\dot{x}_{a2} &= \rho_{a2} [\lambda_{ca}] x_{ca} - [\lambda_{a2}] x_{a2} \\
\dot{x}_{r2} &= \rho_{r2} [\lambda_{ca}] x_{ca} - [\lambda_{r2}] x_{r2} \\
x_p &= 0, x_s > 0
\end{aligned}
$$

$$\xrightarrow{x_s = 0} \qquad \xleftarrow[x_p = 0]{}$$

$$
\begin{aligned}
\dot{x}_s &= 0 \\
\dot{x}_p &= R(x) - \lambda_s \\
\dot{x}_{ca} &= \rho_{ca} \lambda_s - [\lambda_{ca}] x_{ca} \\
\dot{x}_{a1} &= \rho_{a1} \lambda_s - [\lambda_{a1}] x_{a1} \\
\dot{x}_{r1} &= \rho_{r1} \lambda_s - [\lambda_{r1}] x_{r1} \\
\dot{x}_{a2} &= \rho_{a2} [\lambda_{ca}] x_{ca} - [\lambda_{a2}] x_{a2} \\
\dot{x}_{r2} &= \rho_{r2} [\lambda_{ca}] x_{ca} - [\lambda_{r2}] x_{r2} \\
x_p &> 0, x_s = 0
\end{aligned}
$$

$$R(x) = [\lambda_{a1}] x_{a1} + [\lambda_{r1}] x_{r1} + [\lambda_{a2}] x_{a2} + [\lambda_{r2}] x_{r2}$$

$$\rho_j = w_j / (w_{ca} + w_{a1} + w_{r1}) \; (j \in \{ca, a1, r1\})$$

$$\rho_j = w_j / (w_{a2} + w_{r2}) \; (j \in \{a2, r2\})$$

Interval method (1)

Ordinary differential equation.

$$\dot{x} = f(x,\theta),\ x(0) = x_0$$

θ: the vector of interval system parameters, $\theta \in [\theta]$.

Taylor series expansion.

$$x(t_{k+1}) = x(t_k) + \sum_{r=1}^{\gamma} \frac{h^r}{r!} f^{(r-1)}(x(t_k),\theta) + e(x(\eta),\theta) \quad (h = t_{k+1} - t_k,\ t_k \le \eta \le t_{k+1})$$

(γ = 1 is used in the experiments since the system is piecewise linear.)

Estimation of discretizing error by Bounding Box.

$$e(x(\eta),\theta) \subseteq [e_k] = \frac{h^{\gamma+1}}{(\gamma+1)!} F^{(\gamma)}([B_k],[\theta]), \quad [B_k]: \forall t_k \le t \le t_{k+1}.\ x(t) \in [B_k]$$

Interval method (2)

Computation of Bounding Box: iteration of Picard operator.

$$\Phi([B_k]) := [x_k] + [0, h] \cdot F([B_k], [\theta]) \subseteq [B_k]$$

Piecewise interval method:
intervals are computed by solving optimization problems.

$$[x(t_{k+1})]_i = \left[\inf_{x \in [x(t_k)] \cap \Theta, \theta \in [\theta]} \varphi_i(x, \theta), \sup_{x \in [x(t_k)] \cap \Theta, \theta \in [\theta]} \varphi_i(x, \theta) \right] + [e_k]_i$$

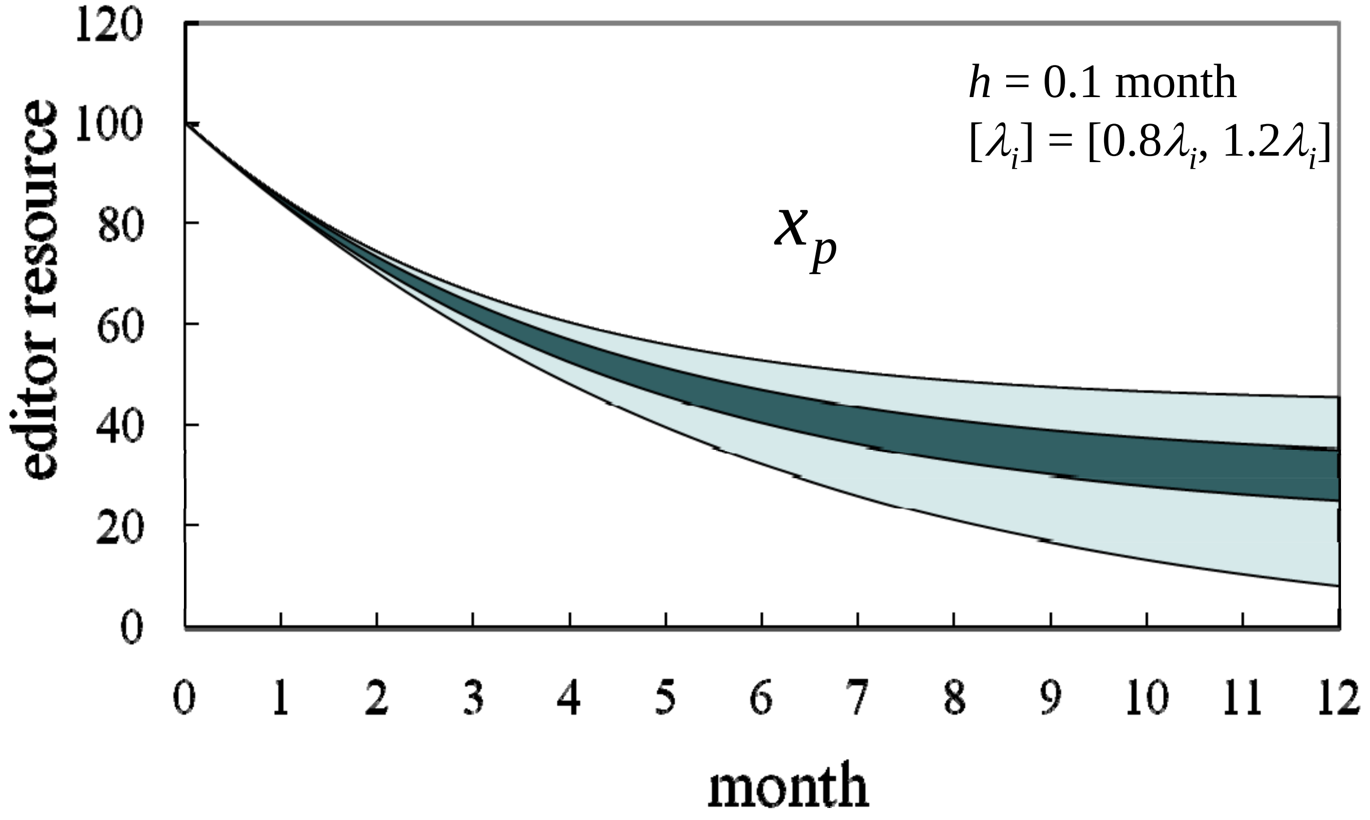
$$\varphi_i(x, \theta) := (x + \Sigma_{r=1,t} (h^r/r!) f^{(r-1)}(x, \theta)) |_i$$

Θ : constraints on the state space $\rightarrow$ we use place invariants.

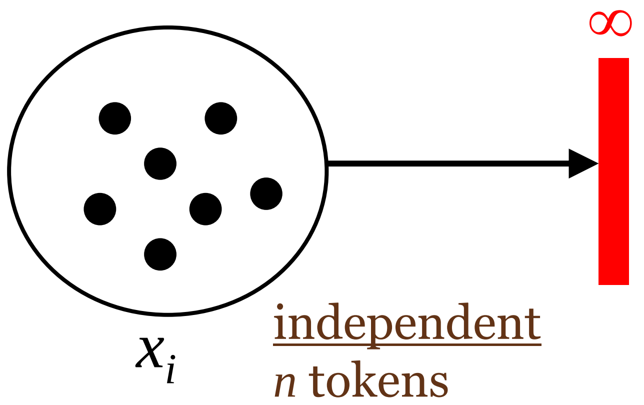
Implementation

- KCLP-HS =Prolog Interpreter
  + Linear Constraint Solver
  + Quadratic Programming Solver
  + Manipulation of Convex Polyhedra
  + Interval Arithmetic

Computation results (1)



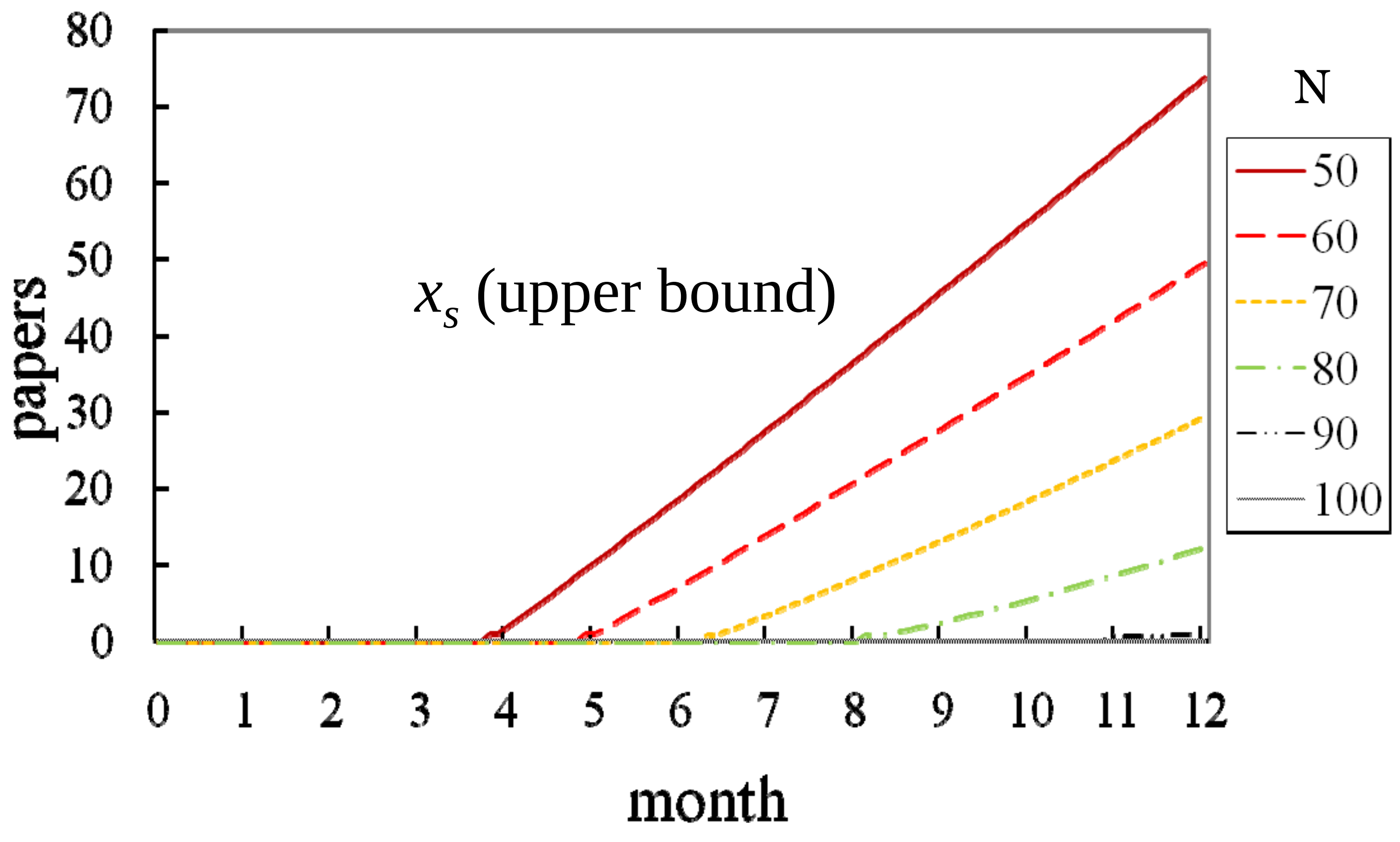
Central limit theorem



Distribution of the delay of each token: (any) identical distribution with mean τ_i, variance σ^2	→	Distribution of *the mean delay of n tokens*: normal distribution with mean τ_i, variance σ^2/n.

The interval $[\lambda]$ for each firing speed may be narrowed to $[\lambda]/\sqrt{n}$.

Computation results (2)



Conculusion

- Proposed approach
  - Computation based on linear calculation.
  - Scalable for the number of workflow instances.
  - Guaranteed enclosures for the state vector.
- Future work
  - Application to real-time systems with interval timing constraints.
  - Computation of transient behavior = Bounded Model Checking.