RUHR-UNIVERSITÄT BOCHUM

NON-LINEAR PROGRAMMING FOR THE NETWORK CALCULUS ANALYSIS OF FIFO FEEDFORWARD NETWORKS

Lukas Herll and **Steffen Bondorf** (Ruhr University Bochum)

Outline aka Bisecting the Paper Title

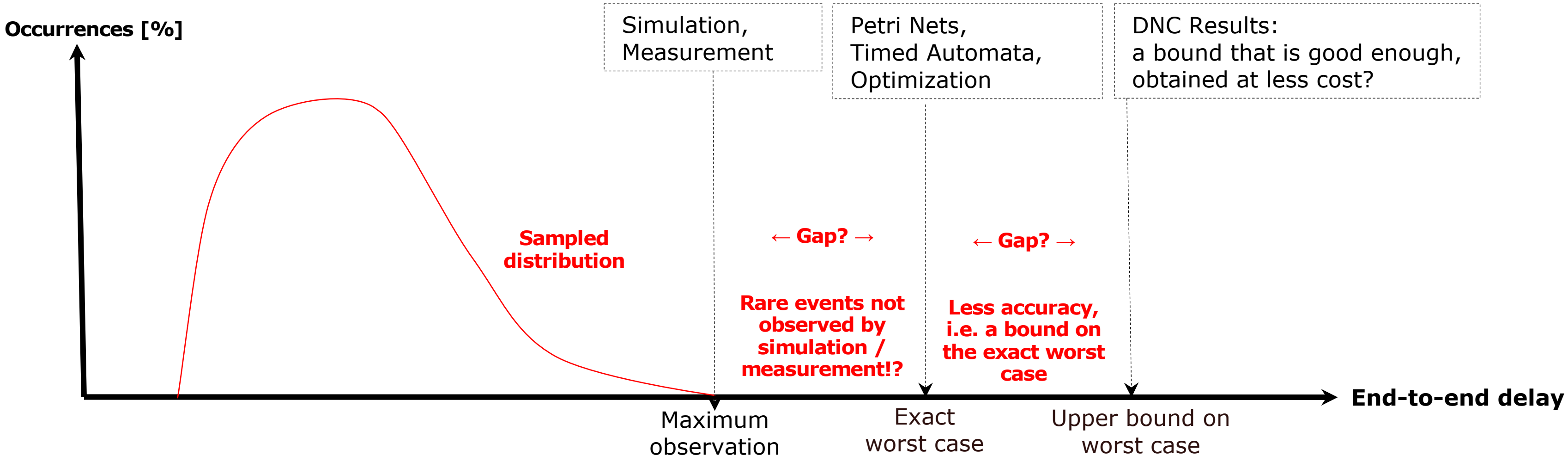
(1) Non-linear Programming
(2) for the Network Calculus
(3) Analysis of
(4) FIFO
(5) Feedforward Networks

becomes

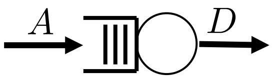
I. (2 & 3) Deterministic Network Calculus (DNC)
II. (3 & 1) the DNC extension “DiffNC”
III. (4) Challenge: FIFO Multiplexing and Queueing
IV. (3 & 4) Applying Gradient-based Non-linear Programming
V. (5) Evaluation

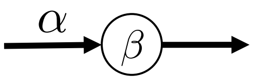
DNC Motivation and Basics

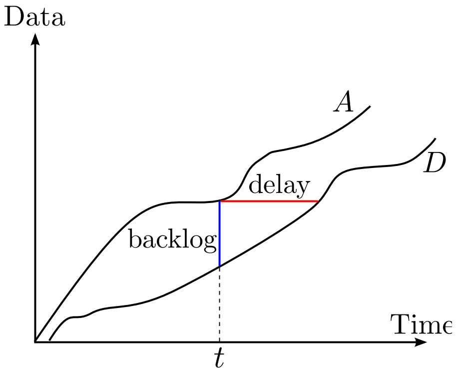
- **Theory of deterministic queueing systems [Cruz91]**
  - Metric: end-to-end communication delay of a data crossing a network
  - DNC: a worst-case bound on the end-to-end delay of a specific data flow

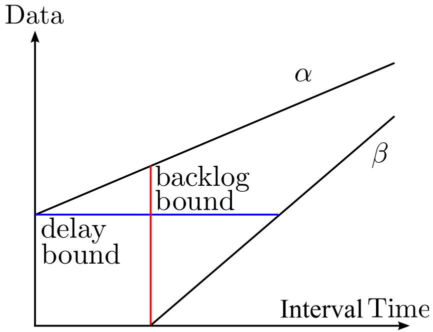


DNC Modeling: Bounding Curves in Interval Time

System: 

Model: 



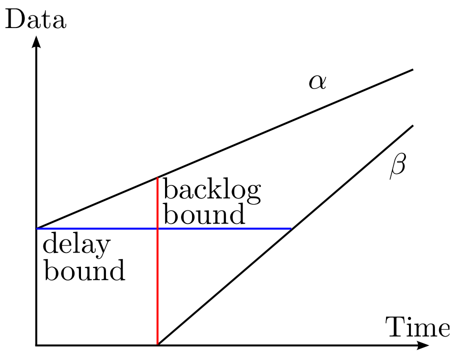


- **Arrival Curve** $\alpha(d)$ $\forall\, 0 \leq d \leq t \;:\; A(t) - A(t-d) \leq \alpha(d)$ **(derived from traffic regulation)**
- **Service Surve** $\beta(d)$ $\forall t : A'(t) \geq \inf_{0 \leq d \leq t} \{A(t-d) + \beta(d)\}$ **(derived from scheduler)**

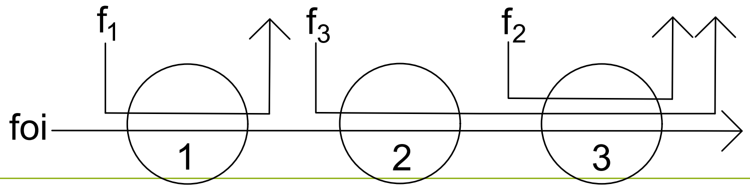
DNC Analysis: A (min,plus)-algebraic Term

- **(min,plus) Operations (complexity depends on curve shapes)**

- Concatenation of servers $\beta_1 \otimes \beta_2 = \beta_{1,2}$
- Output bound $\alpha'(t) = \alpha \oslash \beta(t) := \sup_{u \geq 0}\{\alpha(t+u) - \beta(u)\}$
- Delay bound $hdev(\alpha, \beta) = \inf\{d \geq 0 : (\alpha \oslash \beta)(-d) \leq 0\}$
- Left-over service $\beta(t) \ominus \alpha(t) = \max\{0, \beta(t) - \alpha(t)\}$



- **Example: end-to-end delay bound for data of the flow of interest (foi) crossing 3 servers**



$$\mathrm{hdev}(\alpha_{\mathrm{foi}}, (\beta_1 \ominus \alpha_1) \otimes ((\beta_2 \otimes (\beta_3 \ominus \alpha_2)) \ominus \alpha_3))$$

RUHR UNIVERSITÄT BOCHUM RUB

DNC Intro Wrap-Up

- **The Good**
  - a quite powerful methodolody for worst-case modeling and analysis
  - has found application in the industry (certification of Airbus AFDX network)

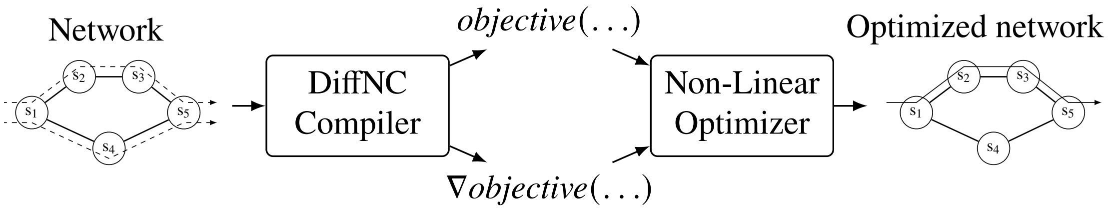
- **The Bad**
  - analysis of non-feedforward networks is not yet as advanced (not part of this paper)

- **The „Ugly“**
  - DNC is a tool for analysis, not for synthesis,
    i.e., you need a fully specified model, you cannot optimize for open parameters,
    you can only sample your network design space

DiffNC [Geyer22]: DNC-based Parameter Synthesis

- **Idea**
- derive your DNC term
- leave some parameters open and/or add binary ones for design alternatives
- differentiate w.r.t. these parameters
- let a solver do the heavy lifting, which turned out to be quite efficient

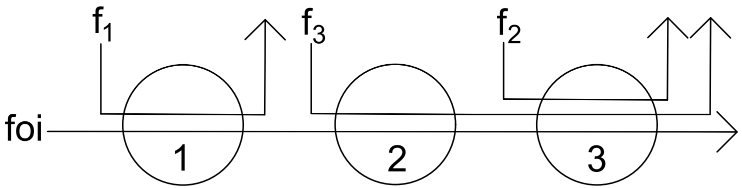
Example: Find delay-bound-minimal flow paths



This Paper: Use the DiffNC Idea Differently

This example actually assumed strict priority between flows:

foi < f_1 < f_3 < f_2



$$\text{hdev}(\alpha_{\text{foi}}, (\beta_1 \ominus \alpha_1) \otimes ((\beta_2 \otimes (\beta_3 \ominus \alpha_2)) \ominus \alpha_3))$$

$$\beta(t) \ominus \alpha(t) = \max\{0, \beta(t) - \alpha(t)\}$$

The left-over service curve computation for FIFO systems is more complex …

$$\beta_{f_1}^{\text{l.o.}}(t,\theta) = [\beta(t) - \alpha_2(t-\theta)]^{\uparrow} \cdot 1_{\{t>\theta\}} \forall \theta \geq 0$$

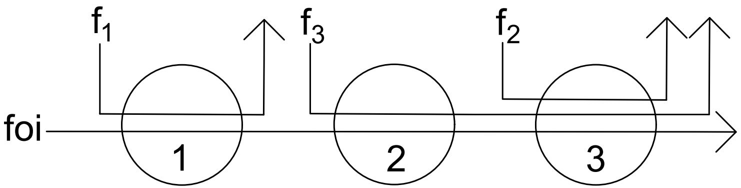
$$[g(x)]^{\uparrow} = \sup_{0 \leq z \leq x} g(z)$$

$$1_{\{t>\theta\}} := \begin{cases} 1, & t > \theta \\ 0, & t \leq \theta \end{cases}$$

This Paper: Use the DiffNC Idea Differently

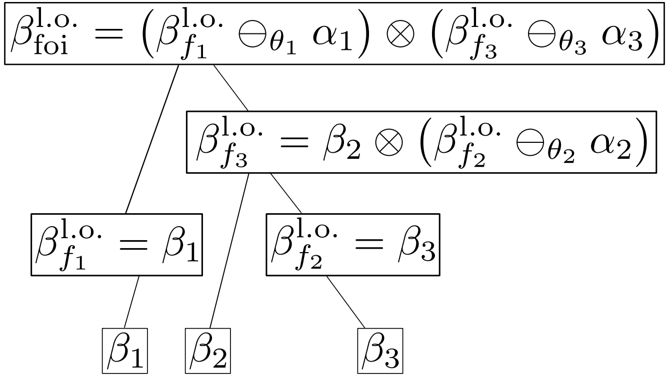
This example in a FIFO system:

Analysis called LUDB (Least Upper Delay Bound) [Bisti08]



$$\mathrm{hdev}(\alpha_{\mathrm{foi}}, (\beta_1 \ominus_{\theta_1} \alpha_1) \otimes ((\beta_2 \otimes (\beta_3 \ominus_{\theta_2} \alpha_2)) \ominus_{\theta_3} \alpha_3))$$

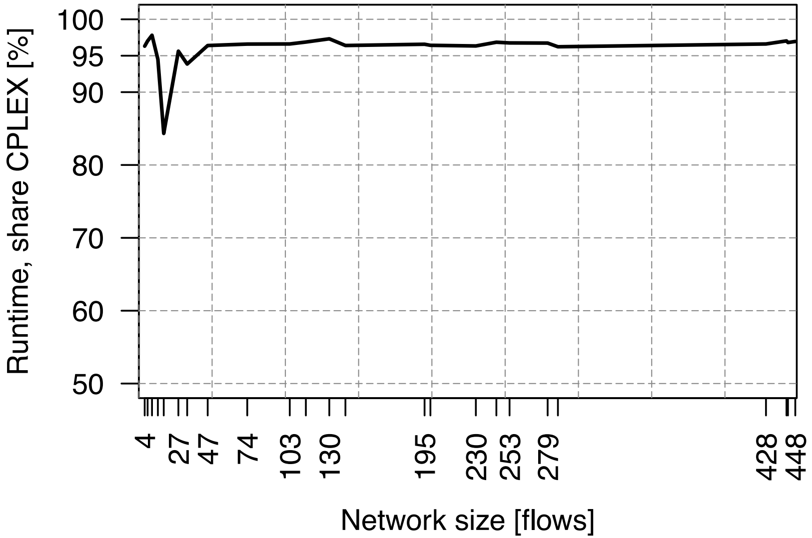
The term "hides" a complex dependency structure, finding optimal theta values is hard …



RUHR UNIVERSITÄT BOCHUM RUB

Related Work: Original LUDB

Idea: convert into a linear problem and let a solver do the heavy lifting [Bisti08]

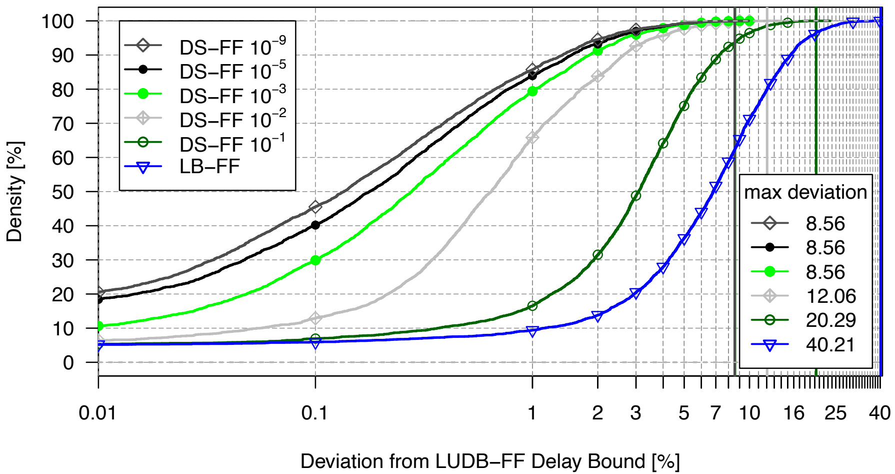


Insight: does not scale, the solver is the bottleneck [Scheffler21]

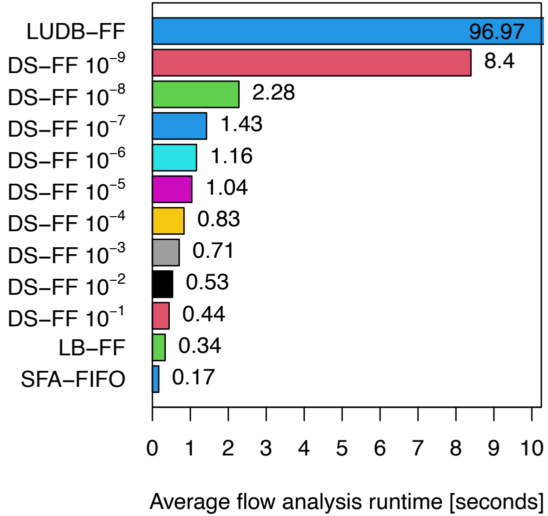
Related Work: Use a Search Algorithm

Idea: assume the problem is convex and use a gradient descent algorithm with some termination criterion [Scheffler22]

Deviation from LUDB delay bounds



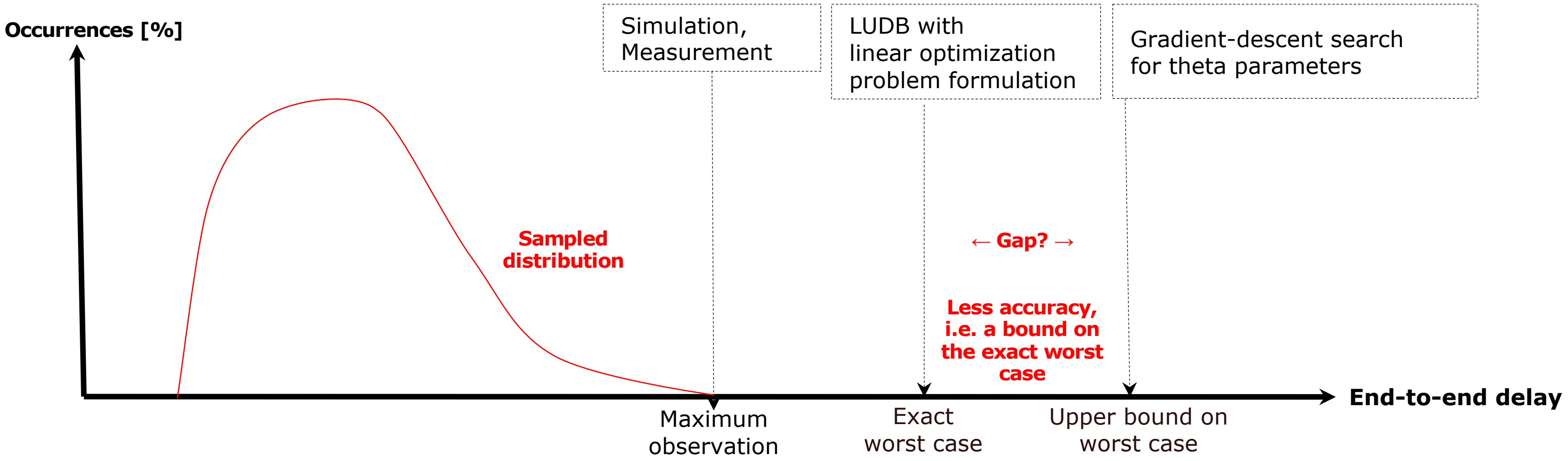
Computation time



Problem: termination criterion must be pre-defined **(here: smallest step size in search)**

Related Work Summary / Paper Motivation

- **Can we close the gap to LUDB even further? And maybe even do so at less cost?**



RUHR UNIVERSITÄT BOCHUM

Applying Gradient-based Non-linear Programming

This paper: Apply the DiffNC idea to the DNC FIFO Analysis (LUDB)

Challenges: $\mathrm{hdev}(\alpha_{\mathrm{foi}}, (\beta_1 \ominus_{\theta_1} \alpha_1) \otimes ((\beta_2 \otimes (\beta_3 \ominus_{\theta_2} \alpha_2)) \ominus_{\theta_3} \alpha_3))$

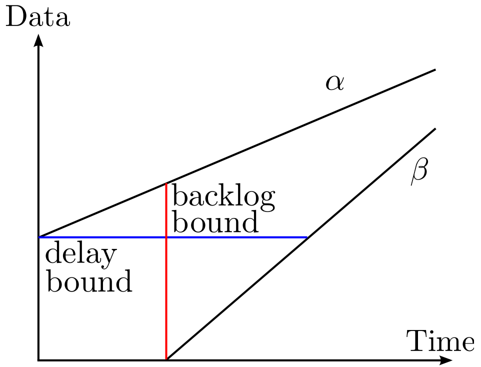
- **There is no solver that can use DNC (min,plus)-algebraic terms as objective function**
- **There is no automatic differentiation software that can use such a term either**

Solution:

- ~~**Implement all required tools ourselves**~~
- **Convert the (min,plus)-algebraic term into a regular (plus,time)-algebraic one**

Applying Gradient-based Non-linear Programming

Using the simple curve shapes (an LUDB restriction, too),



the operations become:

concatenation: $\beta_{R_1,L_1} \otimes \beta_{R_2,L_2} = \beta_{\min(R_1,R_2),L_1+L_2}$

output bounding: $\gamma_{\sigma,\rho} \oslash \beta_{R,L} = \gamma_{\sigma+\rho\cdot L,\rho}$

left-over [2]: $\beta_{R,L} \ominus_\theta \gamma_{\sigma,\rho} = \gamma_{R[\theta-(L+\frac{\sigma}{R})],R-\rho}(t-\theta),$

$$\theta \geq L + \frac{\sigma}{R}$$

$$\overset{s:=\theta-\left(L+\frac{\sigma}{R}\right)}{=} \delta_{s+L+\frac{\sigma}{R}} \otimes \gamma_{Rs,R-\rho}$$

$$=: \beta_{R,L} \ominus_s \gamma_{\sigma,\rho}, \quad s \geq 0$$

delay bound: $hDev(\gamma_{\sigma,\rho}, \beta_{R,L}) = \sigma/R + L$

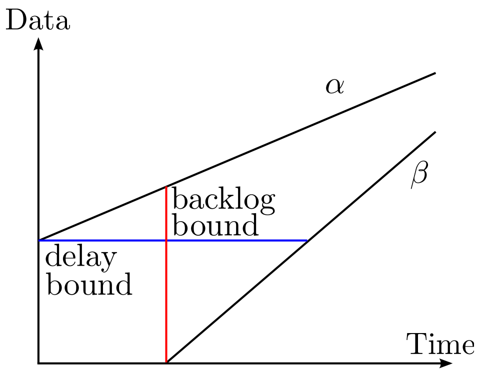
Applying Gradient-based Non-linear Programming

However … *left-over [2]:* $\beta_{R,L} \ominus_\theta \gamma_{\sigma,\rho} = \gamma_{R[\theta-(L+\frac{\sigma}{R})],R-\rho}(t-\theta),$

$$\theta \geq L + \frac{\sigma}{R}$$

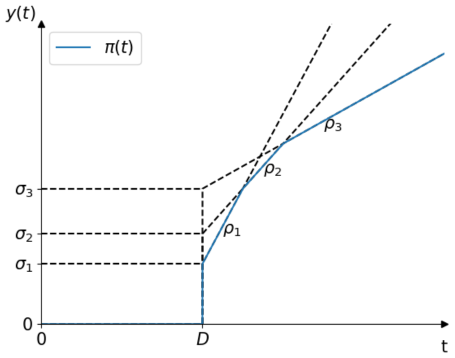
$$\overset{s:=\theta-\left(L+\frac{\sigma}{R}\right)}{=} \delta_{s+L+\frac{\sigma}{R}} \otimes \gamma_{Rs,R-\rho}$$

$$=: \beta_{R,L} \ominus_s \gamma_{\sigma,\rho}, \quad s \geq 0$$



… the FIFO left-over service curve operation is not closed in the simple curve shapes ☹. It becomes this:

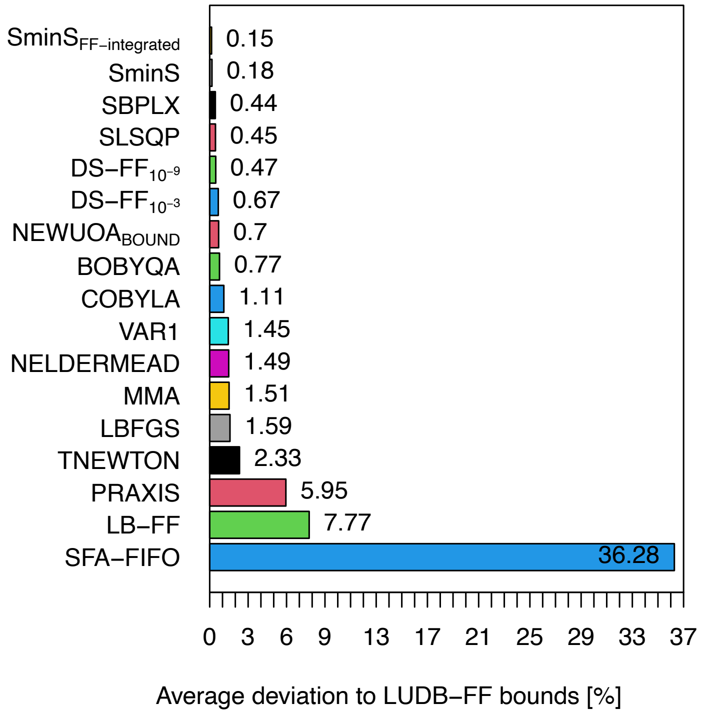
(all further details are in the paper)

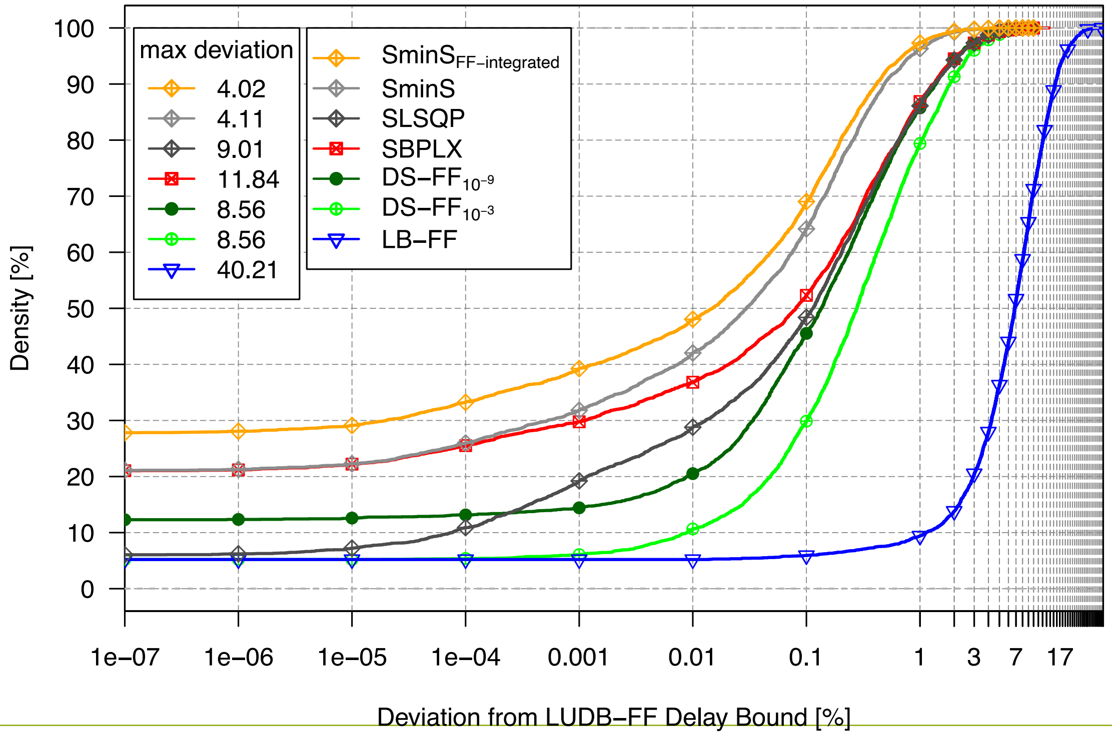


Evaluation

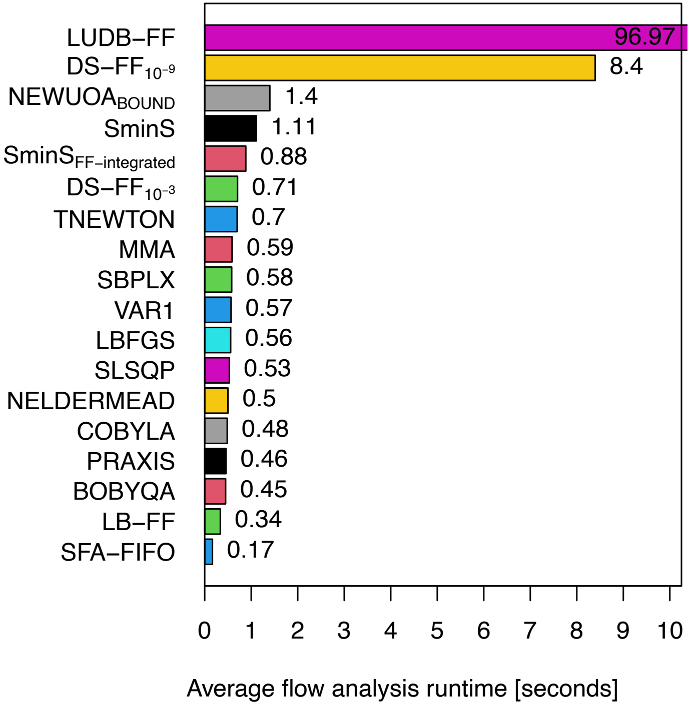
- **Networks to be analyzed: (taken from [Scheffler22], see [Scheffler22dataset])**
  - 31 random feedforward networks following Erdös-Rényi model with a total of 4479 flows
  - Arrival curves set to token bucket $\gamma_{\rho,\sigma} = \gamma_{1,1}$
  - Service curve set to rate latency $\beta_{R,T}$ with $T = 0$ and R set to achieve a desired utilization of the server between 50% and 99%
- **Evalutation machine: (same as in [Scheffler22])**
  - LenovoThinkStation P620, AMD Ryzen Threadripper PRO 3955WX
  - Ubuntu22.04.1 LTS, OpenJDK17
- **Two main proposals for NLP algorithms**
  - SminS (minimum of NLP algorithms SLSQP and Subplex's **overall results**)
  - $\text{SminS}_{\text{FF-integrated}}$ (as before, yet already for all **intermediate results** in a DNC analysis)

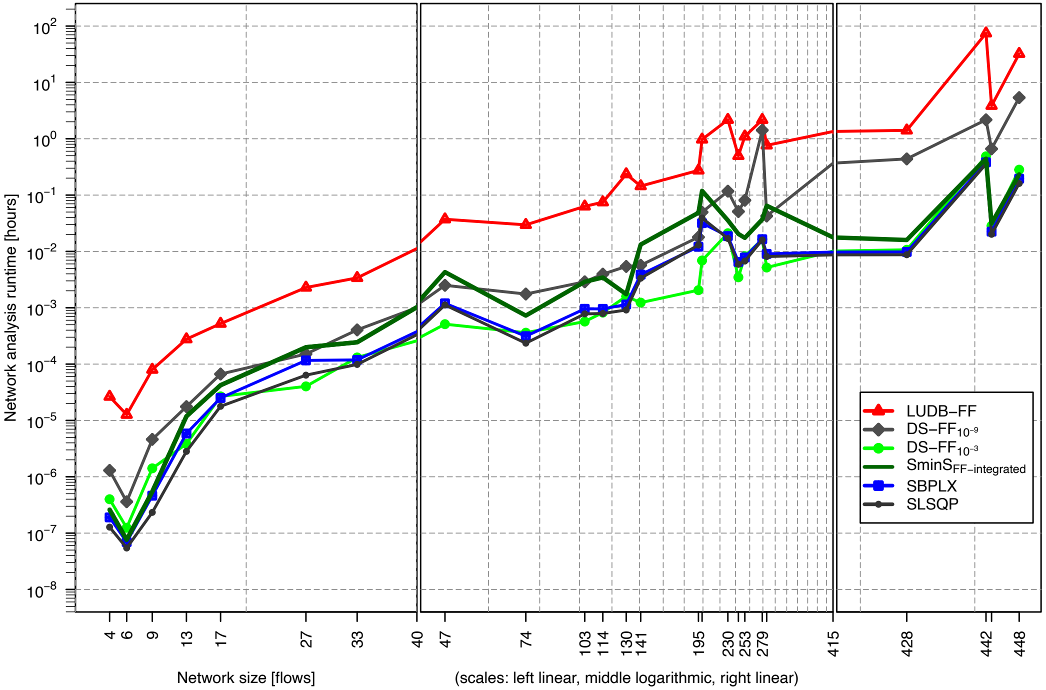
Evaluation: Delay Bounds (Metric: deviation from LUDB [Bisti08])





Evaluation: Computation Times





RUHR UNIVERSITÄT BOCHUM RUB

Conclusion

- **(Gradient-based) NLP optimization can be applied to aid the LUDB analysis**
  - even though convexity of the problem was not (yet) proven
  - Which may be irrelevant as the Subplex algorithm is gradient-free anyways
- **Choosing the right algorithms computes more accurate results in shorter times**
  - Linear programming and search-based algorithms are outperformed

Future Work

- Proof convexity of the problem
- Use our new tool chain for synthesis of networks configurations
  Code is open-source, available at [Herll25dataset]

References

[Herll25dataset] Available online: github.com/Lukasssssssssss/ICPE2025-Non-linear-Programming-for-the-Network-Calculus-Analysis-of-FIFO-Feedforward-Networks

[Scheffler22] A. Scheffler and S. Bondorf. *Network Calculus for Bounding Delays in Feedforward Networks of FIFO Queueing Systems*. In Proc. of RTNS, 2022.

[Scheffler22dataset] Available online: github.com/alexscheffler/dataset-rtns2022

[Geyer22] F. Geyer and S. Bondorf. *Network Synthesis under Delay Constraints: The Power of Network Calculus Differentiability.* In Proc. of INFOCOM, 2022.

[Scheffler21] A. Scheffler, J, Schmitt, and S. Bondorf. *Searching for Upper Delay Bounds in FIFO Multiplexing Feedforward Networks*. In Proc. of RTNS, 2022.

[Bisti08] L. Bisti, L. Lenzini, E. Mingozzi, and G. Stea. *Estimating the worst-case delay in FIFO tandems using network calculus*. In Proc. of ValueTools, 2008.

[Cruz91] R. L. Cruz. *A Calculus for Network Delay, Part I: Network Elements in Isolation* and *A Calculus for Network Delay, Part II: Network Analysis.* In IEEE Transactions on Information Theory, 1991.