Shape-Aware Analysis of End-to-End Latency Under LET

Mario Günzel[1], Matthias Becker[2], Daniel Casini[3]

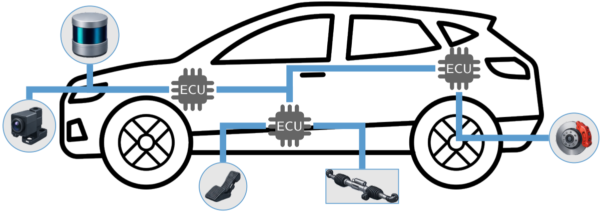
[1]Max Planck Institute for Software Systems (MPI-SWS)/TU Dortmund University,
[2]KTH Royal Institute of Technology Stockholm,
[3]Scuola Superiore Sant'Anna

May 12, 2026 @RTAS 26

This result is part of a project (PropRT) that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 865170). This work has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project No. 569077889 (PEACH). This work was partially funded by the Swedish Research Council (VR) under the project nr. 2023-04773 and by Sweden's Innovation Agency via the NFFP8 project 2024-01267: PARTI. This work has been partially supported by the project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union – NextGenerationEU.

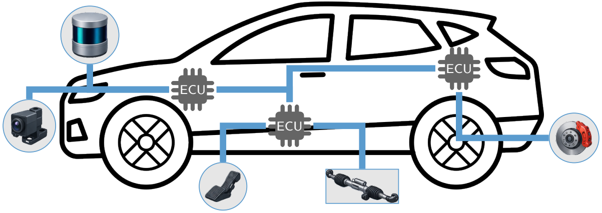
Motivation: Cause-Effect Chains in Automotive Systems

- Automotive Systems: Tasks work together to perform functionality



Motivation: Cause-Effect Chains in Automotive Systems

- Automotive Systems: Tasks work together to perform functionality

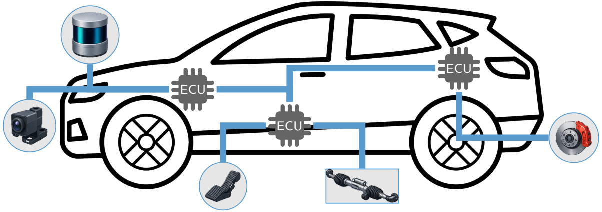


- Example: Automatic brake system

(= Cause-effect chain $E = (\tau_1 \to \tau_2 \to \tau_3)$)

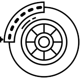
Motivation: Cause-Effect Chains in Automotive Systems

- Automotive Systems: Tasks work together to perform functionality



- Example: Automatic brake system

$\Longrightarrow$ Get Camera Image $\rightarrow$ Image Recognition $\rightarrow$ Danger Analysis $\Longrightarrow$



(= Cause-effect chain $E = (\tau_1 \rightarrow \tau_2 \rightarrow \tau_3)$)

- Want: Analyze timing behavior of full cause-effect chain

Motivation: Many Metrics for Reaction Time

Reaction Time: "How fast does the system *react* to a change?"

Motivation: Many Metrics for Reaction Time

Reaction Time: "How fast does the system *react* to a change?"

Metrics of interest for real-world systems:

Motivation: Many Metrics for Reaction Time

Reaction Time: "How fast does the system *react* to a change?"

Metrics of interest for real-world systems:

- Maximum (and Minimum) Reaction Time
- Average Reaction Time

Motivation: Many Metrics for Reaction Time

Reaction Time: "How fast does the system *react* to a change?"

Metrics of interest for real-world systems:

- Maximum (and Minimum) Reaction Time
- Average Reaction Time
- Throughput

Motivation: Many Metrics for Reaction Time

Reaction Time: "How fast does the system *react* to a change?"

Metrics of interest for real-world systems:

- Maximum (and Minimum) Reaction Time
- Average Reaction Time
- Throughput
- Tolerance to bounded violations
- ...

Motivation: Many Metrics for Reaction Time

Reaction Time: "How fast does the system *react* to a change?"

Metrics of interest for real-world systems:

- Maximum (and Minimum) Reaction Time } $\exists$ Analysis in literature
- Average Reaction Time
- Throughput
- Tolerance to bounded violations
- ...

Motivation: Many Metrics for Reaction Time

Reaction Time: "How fast does the system *react* to a change?"

Metrics of interest for real-world systems:

- Maximum (and Minimum) Reaction Time } $\exists$ Analysis in literature
- Average Reaction Time
- Throughput
- Tolerance to bounded violations
- ...

} Largely open (Average Reaction Time, Throughput, Tolerance to bounded violations, ...)

Motivation: Many Metrics for Reaction Time

Reaction Time: "How fast does the system *react* to a change?"

Metrics of interest for real-world systems:

- Maximum (and Minimum) Reaction Time } $\exists$ Analysis in literature
- Average Reaction Time
- Throughput
- Tolerance to bounded violations
- . . .

} Largely open (Average Reaction Time, Throughput, Tolerance to bounded violations, . . .)

Do we need to build a new analysis for *each metric from scratch*?

Motivation: Many Metrics for Reaction Time

Reaction Time: "How fast does the system *react* to a change?"

Metrics of interest for real-world systems:

- Maximum (and Minimum) Reaction Time } $\exists$ Analysis in literature
- Average Reaction Time
- Throughput
- Tolerance to bounded violations
- ...

(Average Reaction Time, Throughput, Tolerance to bounded violations, ...) } Largely open

Do we need to build a new analysis for *each metric from scratch*?

This paper:

- No! We can analyze the *full shape once* and derive metrics from that.

Motivation: Many Metrics for Reaction Time

Reaction Time: "How fast does the system *react* to a change?"

Metrics of interest for real-world systems:

- Maximum (and Minimum) Reaction Time } $\exists$ Analysis in literature
- Average Reaction Time
- Throughput
- Tolerance to bounded violations
- ...

(Average Reaction Time, Throughput, Tolerance to bounded violations, ...) } Largely open

Do we need to build a new analysis for *each metric from scratch*?

This paper:

- No! We can analyze the *full shape once* and derive metrics from that.
- For many metrics, we provide *first* analysis

Outline

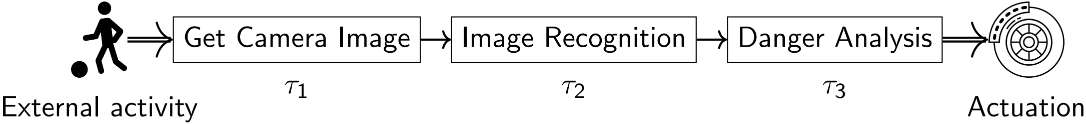
End-to-End Latency of Cause-Effect Chains

Shape-Aware Analysis of Reaction Time

Evaluation

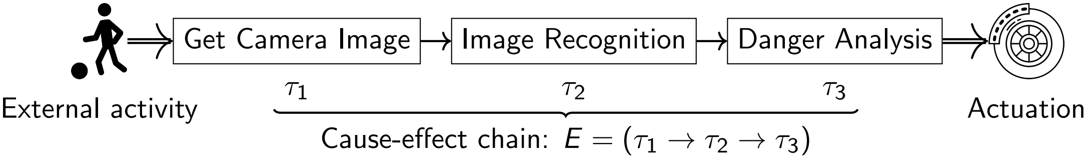
Cause-Effect Chains

- Functionality is described by a sequence of tasks



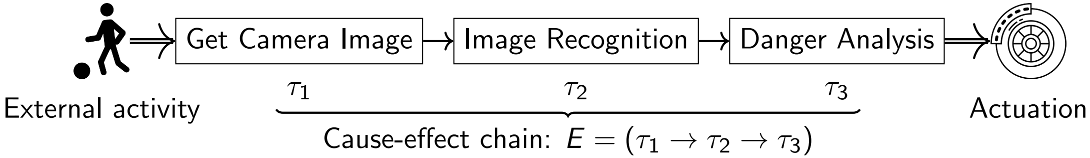
Cause-Effect Chains

- Functionality is described by a sequence of tasks



Cause-Effect Chains

- Functionality is described by a sequence of tasks

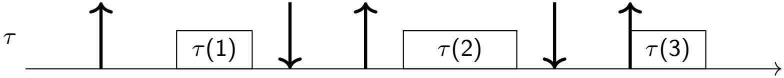


Cause-Effect Chain

Sequence $E = (\tau_1 \to \tau_2 \to \cdots \to \tau_n)$ of tasks τ_i

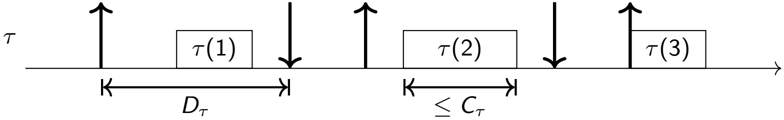
(Real-Time) Task Model

$\mathbb{T}$ task set, $\tau \in \mathbb{T}$ task, releases jobs $\tau(i)$



(Real-Time) Task Model

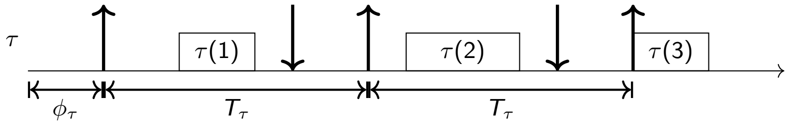
$\mathbb{T}$ task set, $\tau \in \mathbb{T}$ task, releases jobs $\tau(i)$



- Worst-case execution time (WCET) C_τ
- Relative deadline D_τ

(Real-Time) Task Model

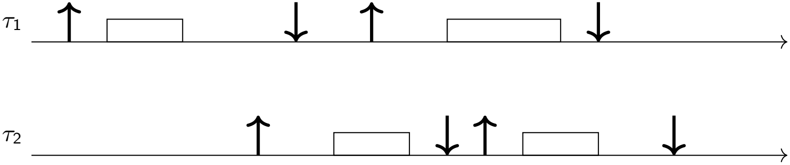
$\mathbb{T}$ task set, $\tau \in \mathbb{T}$ task, releases jobs $\tau(i)$



- Worst-case execution time (WCET) C_τ
- Relative deadline D_τ
- Period T_τ
- Phase ϕ_τ

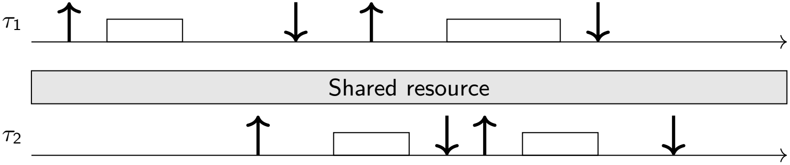
Communication Under Logical Execution Time (LET)

- Communication modeled through shared resource



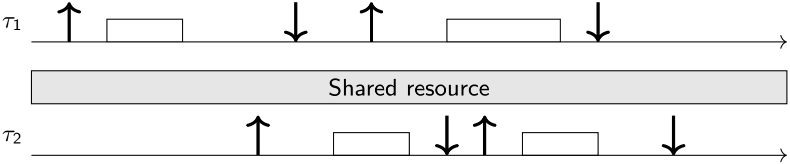
Communication Under Logical Execution Time (LET)

- Communication modeled through shared resource



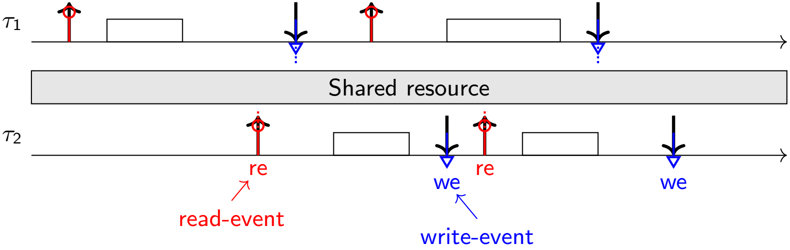
Communication Under Logical Execution Time (LET)

- Communication modeled through shared resource
- LET: Read-events (re) and write-events (we) at release and deadline



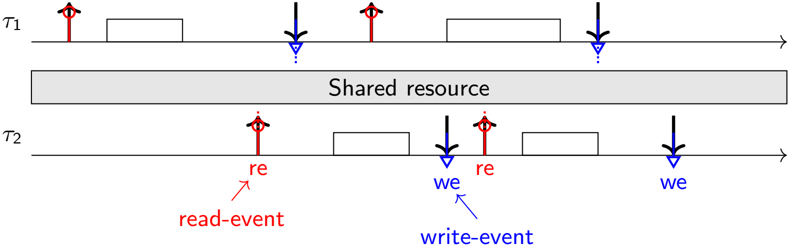
Communication Under Logical Execution Time (LET)

- Communication modeled through shared resource
- LET: Read-events (re) and write-events (we) at release and deadline

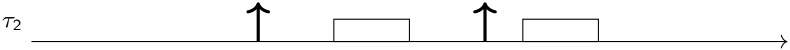


Communication Under Logical Execution Time (LET)

- Communication modeled through shared resource
- LET: Read-events (re) and write-events (we) at release and deadline

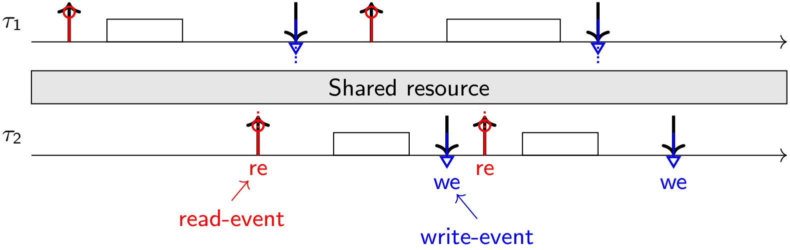


- For simplicity of this presentation: Consider implicit deadline ($D_i = T_i$)

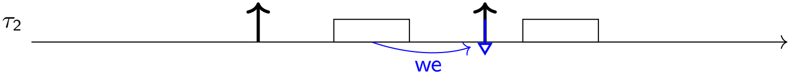


Communication Under Logical Execution Time (LET)

- Communication modeled through shared resource
- LET: Read-events (re) and write-events (we) at release and deadline

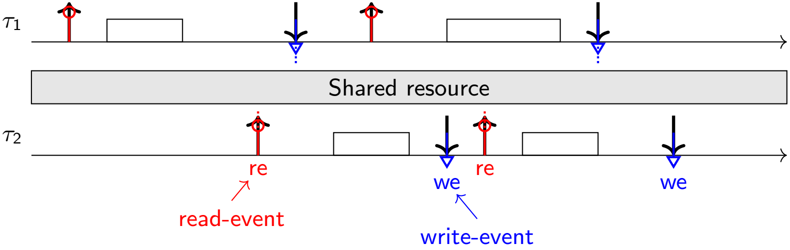


- For simplicity of this presentation: Consider implicit deadline ($D_i = T_i$)

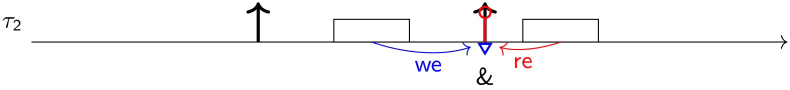


Communication Under Logical Execution Time (LET)

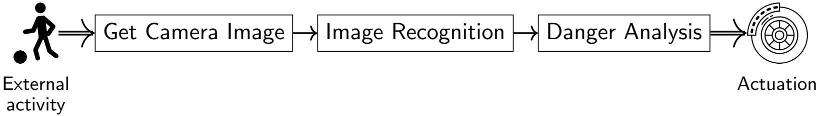
- Communication modeled through shared resource
- LET: Read-events (re) and write-events (we) at release and deadline



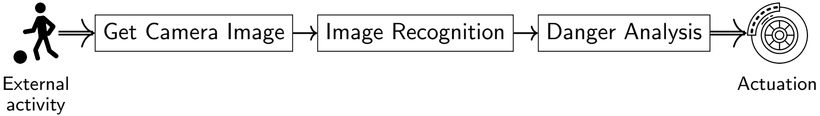
- For simplicity of this presentation: Consider implicit deadline ($D_i = T_i$)



Reaction Time (RT)



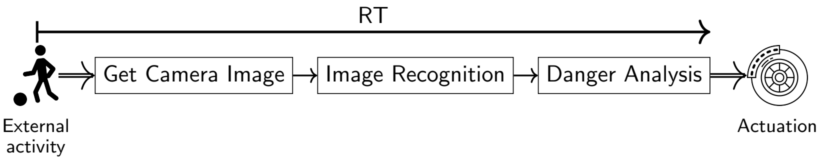
Reaction Time (RT)



Reaction Time $RT(z)$

Given an external activity at time z, how long does it take until this external activity is fully processed?

Reaction Time (RT)



Reaction Time $RT(z)$

Given an external activity at time z, how long does it take until this external activity is fully processed?

Reaction Time (RT) under LET: An Example

Reaction Time $RT(z)$

Given an external activity at time z, how long does it take until this external activity is fully processed?

Reaction Time (RT) under LET: An Example

Reaction Time $RT(z)$

Given an external activity at time z, how long does it take until this external activity is fully processed?

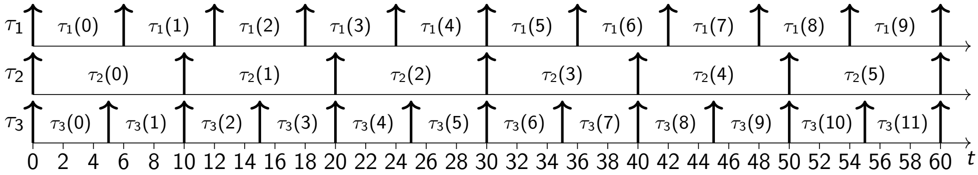
Running Example: $\phi_1 = \phi_2 = \phi_3 = 0,\ T_1 = D_1 = 6,\ T_2 = D_2 = 10,\ T_3 = D_3 = 5$

Reaction Time (RT) under LET: An Example

Reaction Time $RT(z)$

Given an external activity at time z, how long does it take until this external activity is fully processed?

Running Example: $\phi_1 = \phi_2 = \phi_3 = 0,\ T_1 = D_1 = 6,\ T_2 = D_2 = 10,\ T_3 = D_3 = 5$

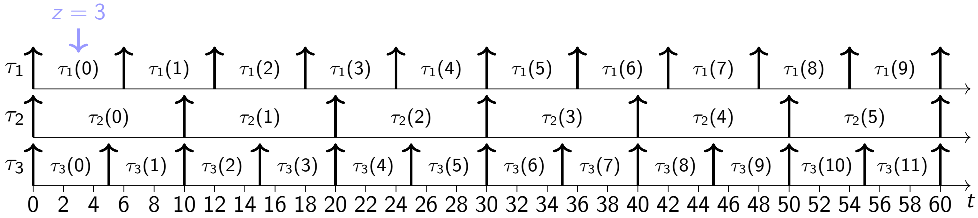


Reaction Time (RT) under LET: An Example

Reaction Time $RT(z)$

Given an external activity at time z, how long does it take until this external activity is fully processed?

Running Example: $\phi_1 = \phi_2 = \phi_3 = 0,\ T_1 = D_1 = 6,\ T_2 = D_2 = 10,\ T_3 = D_3 = 5$

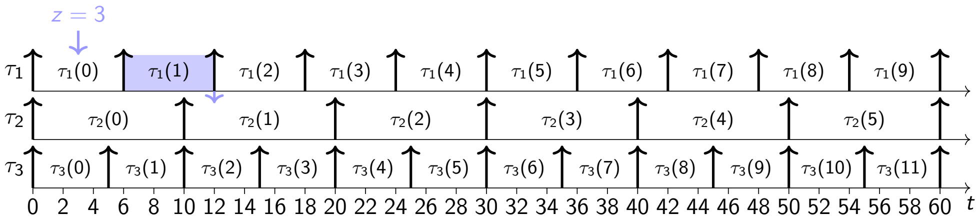


Reaction Time (RT) under LET: An Example

Reaction Time $RT(z)$

Given an external activity at time z, how long does it take until this external activity is fully processed?

Running Example: $\phi_1 = \phi_2 = \phi_3 = 0,\ T_1 = D_1 = 6,\ T_2 = D_2 = 10,\ T_3 = D_3 = 5$

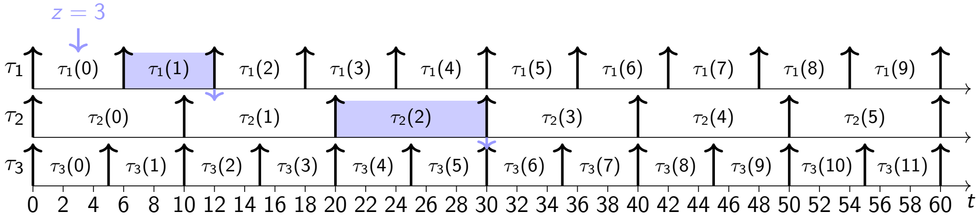


Reaction Time (RT) under LET: An Example

Reaction Time $RT(z)$

Given an external activity at time z, how long does it take until this external activity is fully processed?

Running Example: $\phi_1 = \phi_2 = \phi_3 = 0,\ T_1 = D_1 = 6,\ T_2 = D_2 = 10,\ T_3 = D_3 = 5$

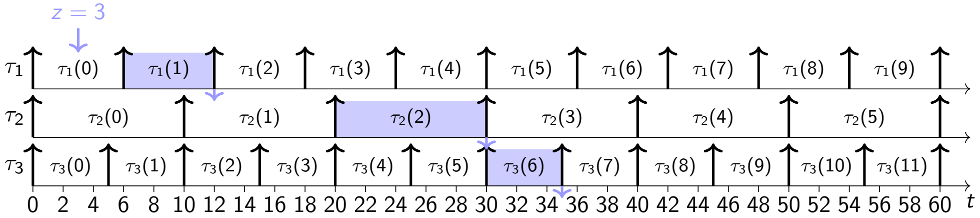


Reaction Time (RT) under LET: An Example

Reaction Time $RT(z)$

Given an external activity at time z, how long does it take until this external activity is fully processed?

Running Example: $\phi_1 = \phi_2 = \phi_3 = 0,\ T_1 = D_1 = 6,\ T_2 = D_2 = 10,\ T_3 = D_3 = 5$



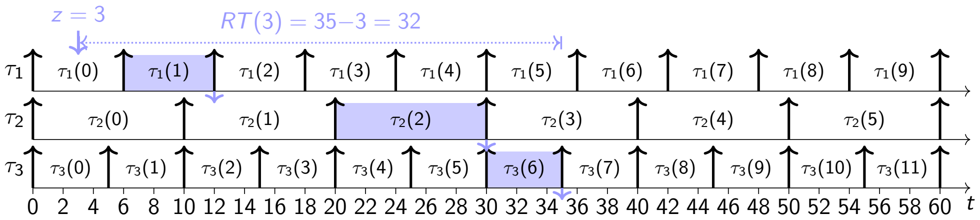
MAX PLANCK INSTITUTE FOR SOFTWARE SYSTEMS

Reaction Time (RT) under LET: An Example

Reaction Time $RT(z)$

Given an external activity at time z, how long does it take until this external activity is fully processed?

Running Example: $\phi_1 = \phi_2 = \phi_3 = 0,\ T_1 = D_1 = 6,\ T_2 = D_2 = 10,\ T_3 = D_3 = 5$

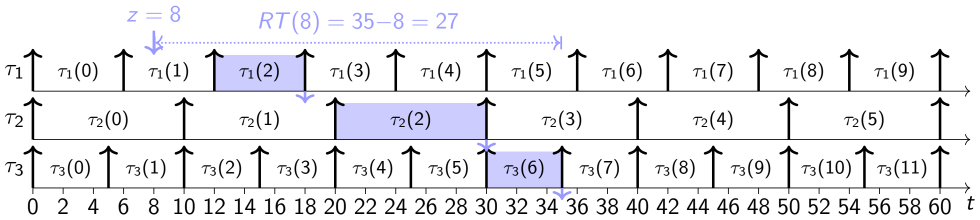


Reaction Time (RT) under LET: An Example

Reaction Time $RT(z)$

Given an external activity at time z, how long does it take until this external activity is fully processed?

Running Example: $\phi_1 = \phi_2 = \phi_3 = 0,\ T_1 = D_1 = 6,\ T_2 = D_2 = 10,\ T_3 = D_3 = 5$

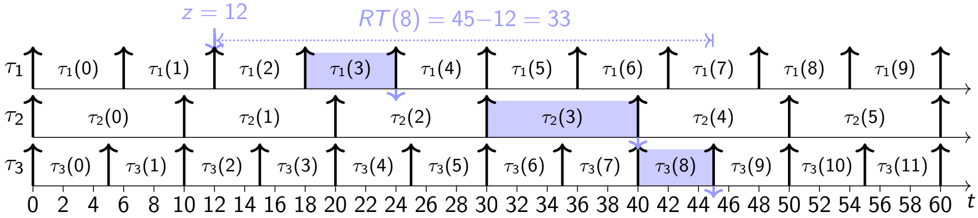


Reaction Time (RT) under LET: An Example

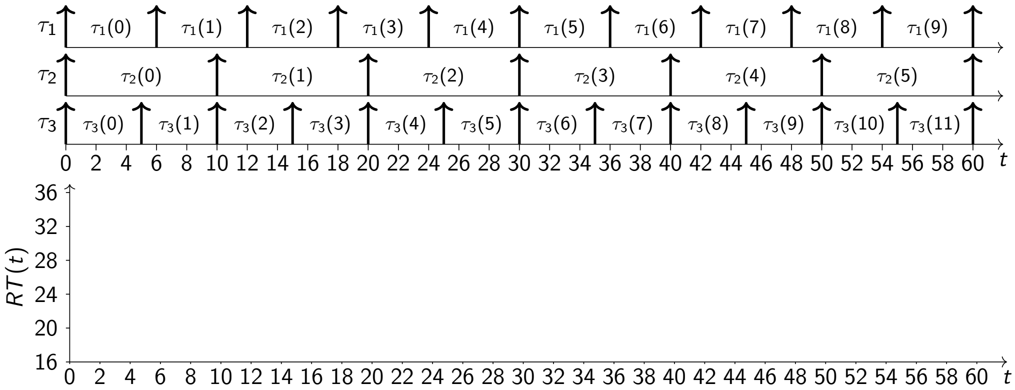
Reaction Time $RT(z)$

Given an external activity at time z, how long does it take until this external activity is fully processed?

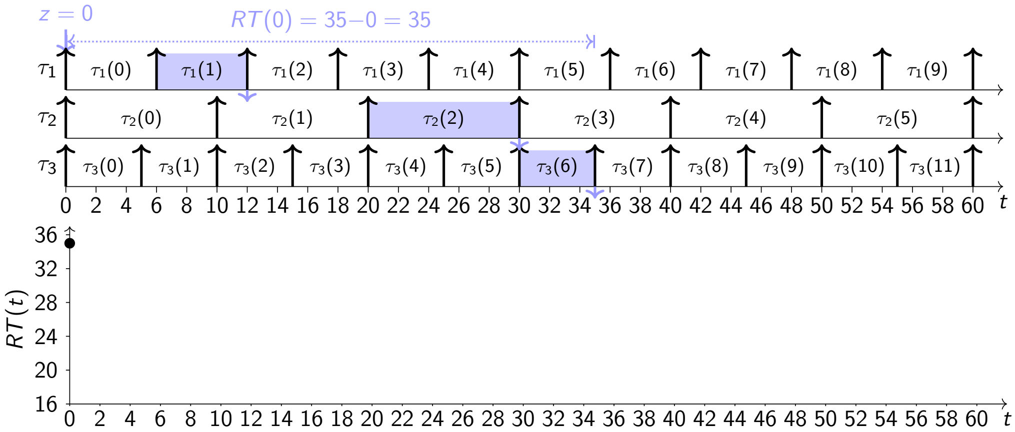
Running Example: $\phi_1 = \phi_2 = \phi_3 = 0,\ T_1 = D_1 = 6,\ T_2 = D_2 = 10,\ T_3 = D_3 = 5$



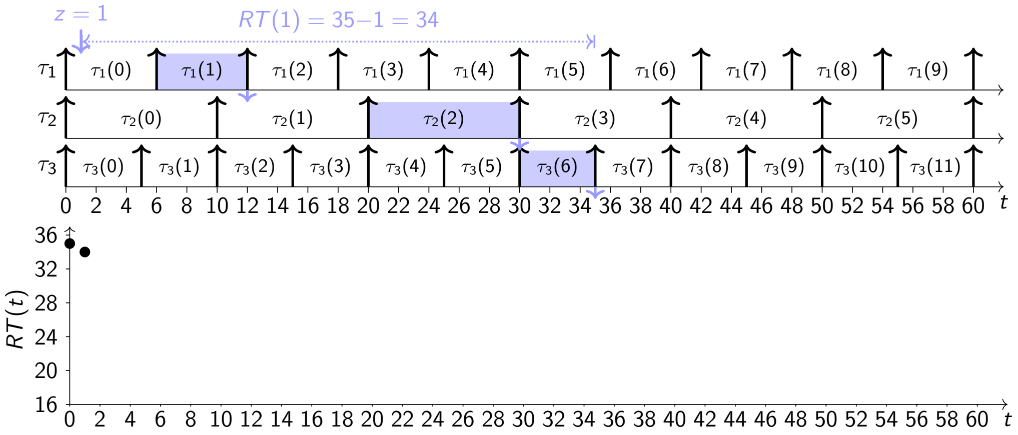
How Does the Shape of RT Look Like?



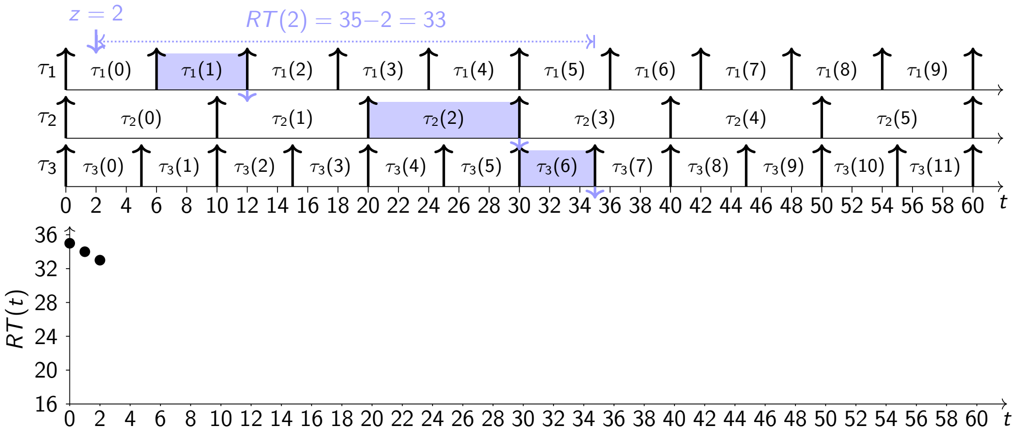
How Does the Shape of RT Look Like?



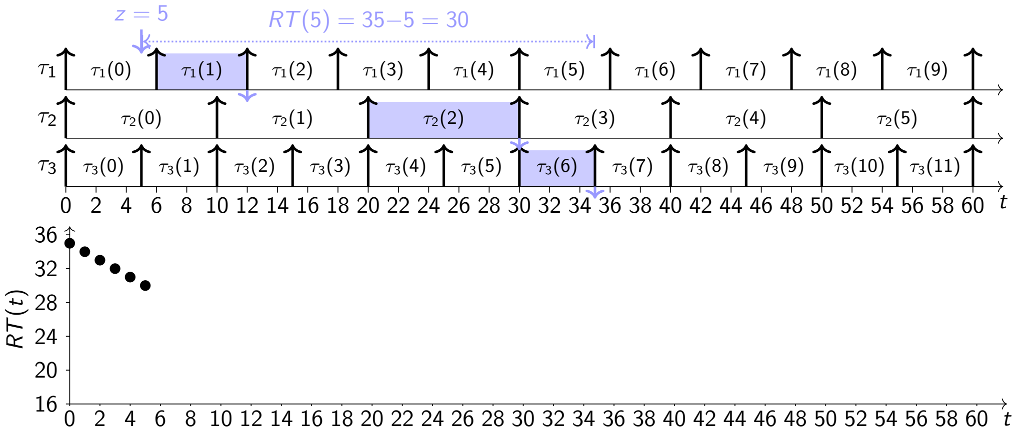
How Does the Shape of RT Look Like?



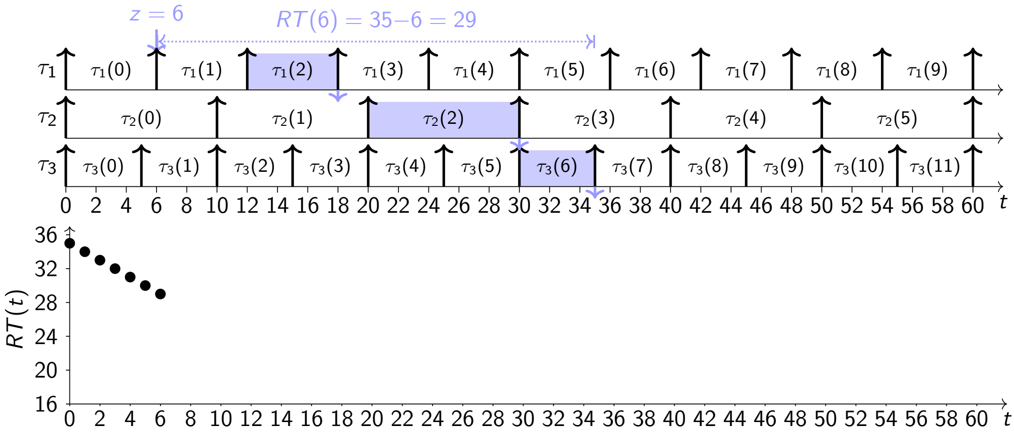
How Does the Shape of RT Look Like?



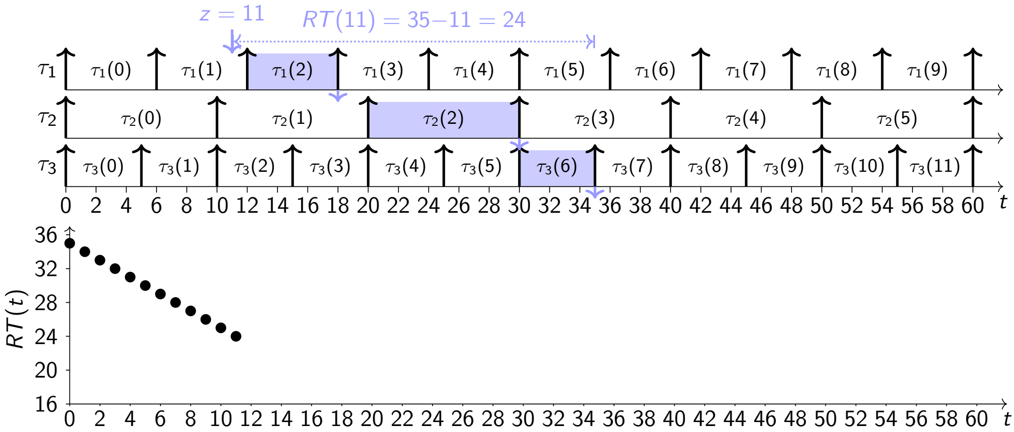
How Does the Shape of RT Look Like?



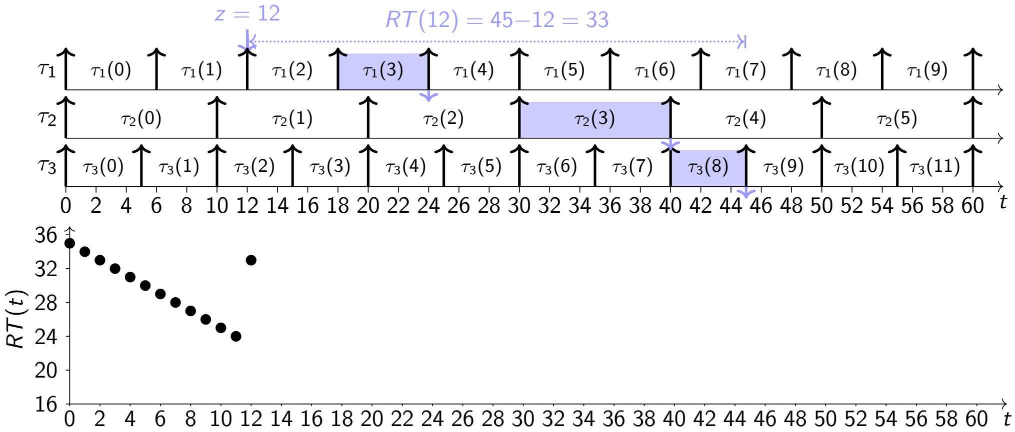
How Does the Shape of RT Look Like?



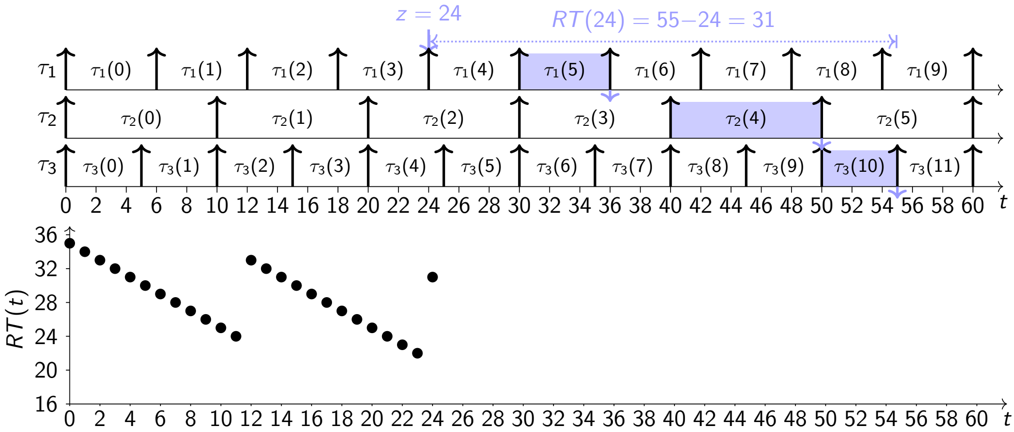
How Does the Shape of RT Look Like?



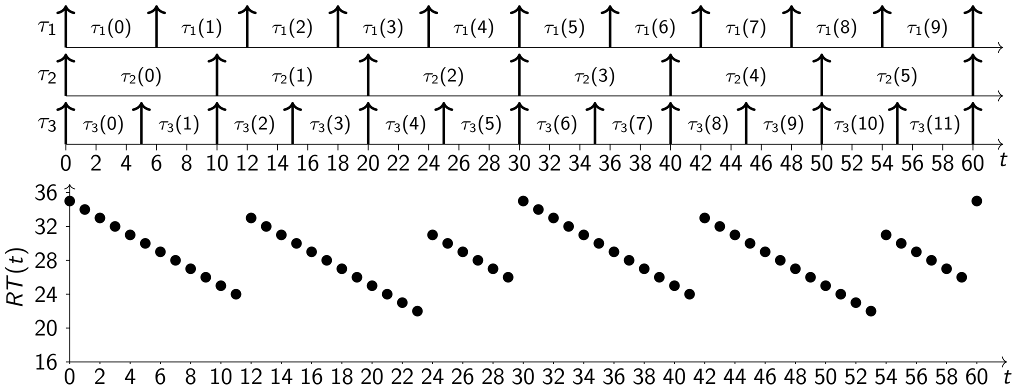
How Does the Shape of RT Look Like?



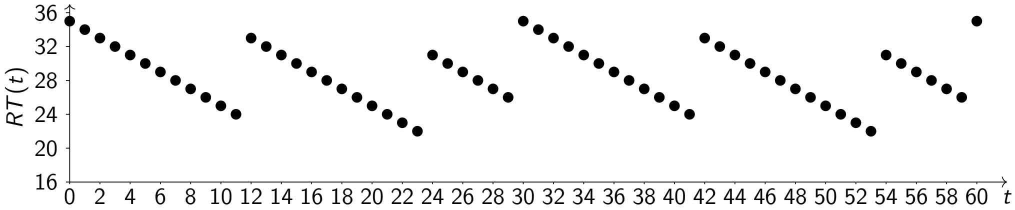
How Does the Shape of RT Look Like?



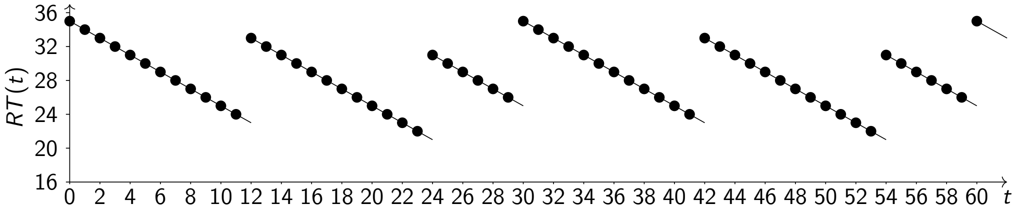
How Does the Shape of RT Look Like?



Observations

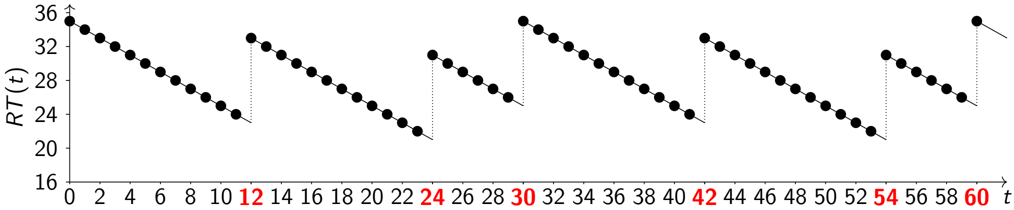


Observations



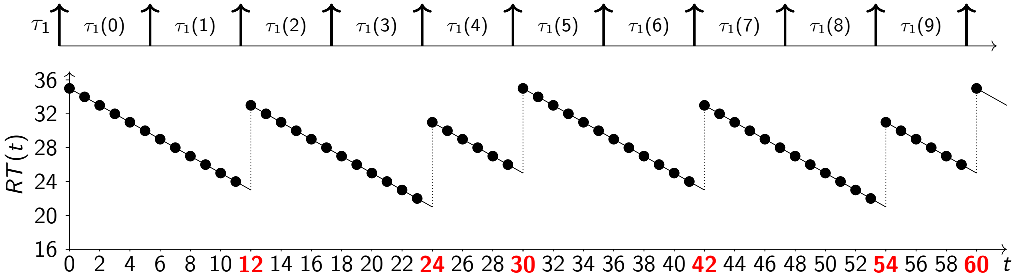
1. RT Linearly decreasing

Observations



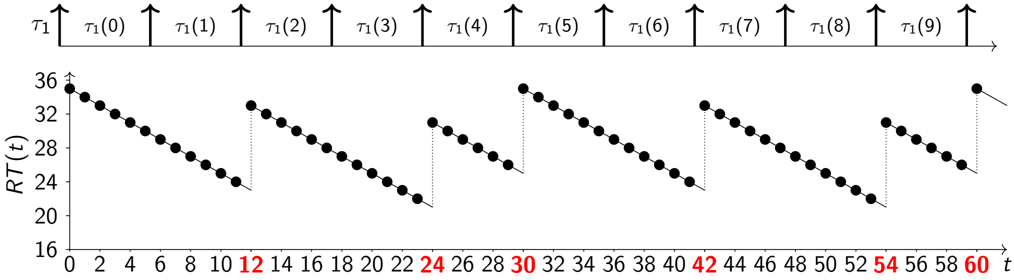
1. *RT* Linearly decreasing
2. Jumps up can only occur at multiples of 6

Observations



1. RT Linearly decreasing
2. Jumps up can only occur at multiples of 6

Observations



1. RT Linearly decreasing
2. Jumps up can only occur at multiples of 6 (= read-events of task τ_1)

Formalization of Observations

Lemma 8

RT restricted to the interval $[\mathrm{re}(\tau_1(m)), \mathrm{re}(\tau_1(m+1)))$, $m \in \mathbb{N}$ is linearly decreasing:

Formalization of Observations

Lemma 8

RT restricted to the interval $[\mathrm{re}(\tau_1(m)), \mathrm{re}(\tau_1(m+1)))$, $m \in \mathbb{N}$ is linearly decreasing:

$$RT(t) = RT(\mathrm{re}(\tau_1(m))) - (t - \mathrm{re}(\tau_1(m)))$$

$\forall\ t \in [\mathrm{re}(\tau_1(m)), \mathrm{re}(\tau_1(m+1)))$

Formalization of Observations

Lemma 8

RT restricted to the interval $[\mathrm{re}(\tau_1(m)), \mathrm{re}(\tau_1(m+1)))$, $m \in \mathbb{N}$ is linearly decreasing:

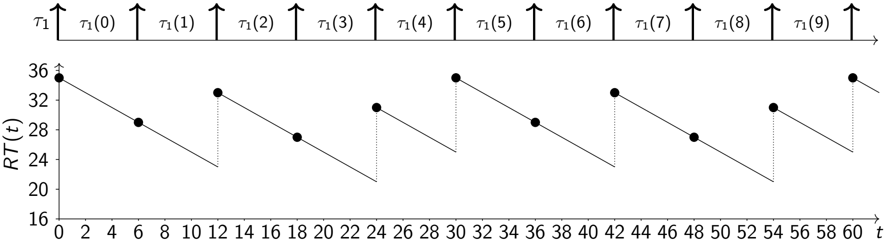
$$RT(t) = RT(\mathrm{re}(\tau_1(m))) - (t - \mathrm{re}(\tau_1(m)))$$

$$\forall\ t \in [\mathrm{re}(\tau_1(m)), \mathrm{re}(\tau_1(m+1)))$$

$\Rightarrow$ Full shape of RT can be determined by evaluating at read-events of τ_1

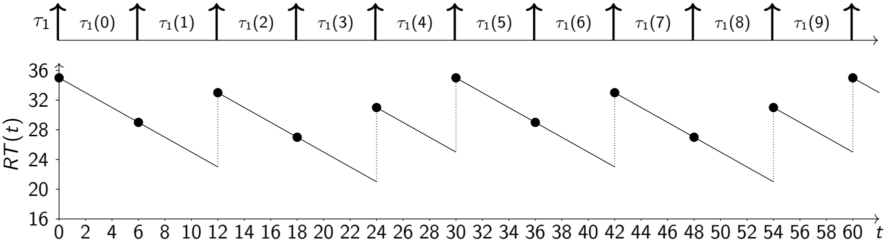
Shape of RT: Specification by Anchor Points

This gives us the following shape:



Shape of RT: Specification by Anchor Points

This gives us the following shape:

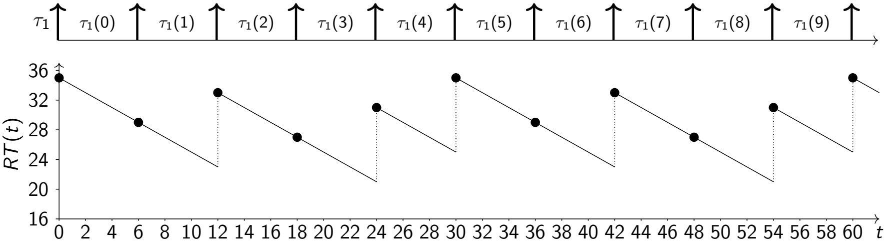


Definition: Anchor Points

$$\mathcal{A}^1 := \left\{ \left(x, RT(x)\right) \,\middle|\, x = \mathrm{re}(\tau_1(m)), m \in \mathbb{N} \right\}$$

Shape of RT: Specification by Anchor Points

This gives us the following shape:

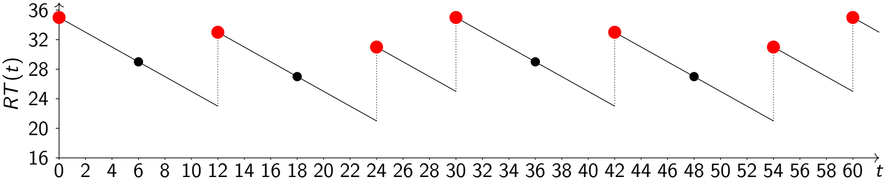


Definition: Anchor Points

$$\mathcal{A}^1 := \{(x, RT(x)) \mid x = \text{re}(\tau_1(m)), m \in \mathbb{N}\}$$

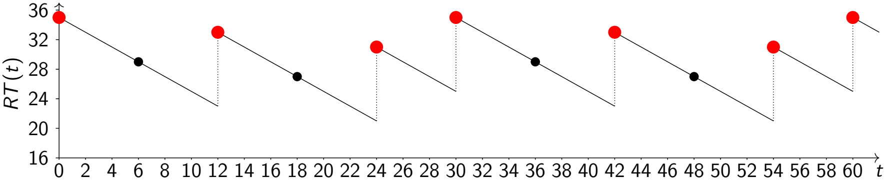
$$\mathcal{A}^1 = \left\{ \begin{array}{l} (0,35), (6,29), (12,33), \\ (18,27), (24,31), (30,35), \\ (36,29), (42,33), (48,27), \\ (54,31), (60,35), \ldots \end{array} \right\}$$

Minimal Set of Anchor Points



$$\mathcal{A}^1 = \{(0,35),(6,29),(12,33),(18,27),(24,31),(30,35),(36,29),(42,33),(48,27),(54,31),(60,35),\ldots\}$$

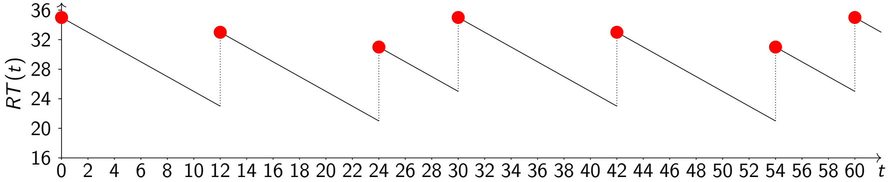
Minimal Set of Anchor Points



$\mathcal{A}^1 = \{(0,35),(6,29),(12,33),(18,27),(24,31),(30,35),(36,29),(42,33),(48,27),(54,31),(60,35),\ldots\}$

1. Remove redundant anchor points from $\mathcal{A}^1$

Minimal Set of Anchor Points

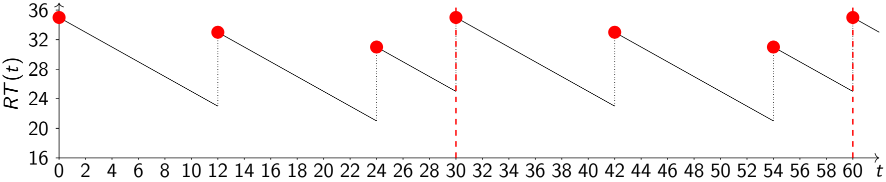


$$\mathcal{A}^1 = \{(0,35),(6,29),(12,33),(18,27),(24,31),(30,35),(36,29),(42,33),(48,27),(54,31),(60,35),\dots\}$$

$$\mathcal{A}^* = \{(0,35),(12,33),(24,31),(30,35),(42,33),(54,31),(60,35),\dots\}$$

1. Remove redundant anchor points from $\mathcal{A}^1$

Minimal Set of Anchor Points

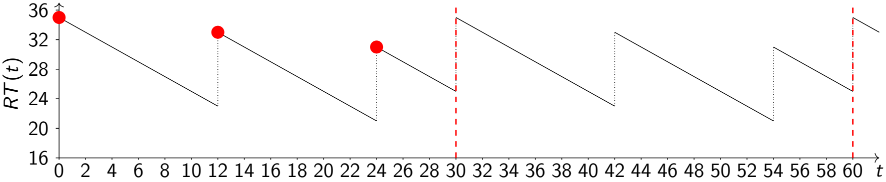


$$\mathcal{A}^1 = \{(0,35),(6,29),(12,33),(18,27),(24,31),(30,35),(36,29),(42,33),(48,27),(54,31),(60,35),\dots\}$$

$$\mathcal{A}^* = \{(0,35),(12,33),(24,31),(30,35),(42,33),(54,31),(60,35),\dots\}$$

1. Remove redundant anchor points from $\mathcal{A}^1$
2. Consider first hyperperiod $[0, H)$ with $H = \text{LCM}(T_1, \dots, T_n)$ $(= 30$ for example$)$

Minimal Set of Anchor Points

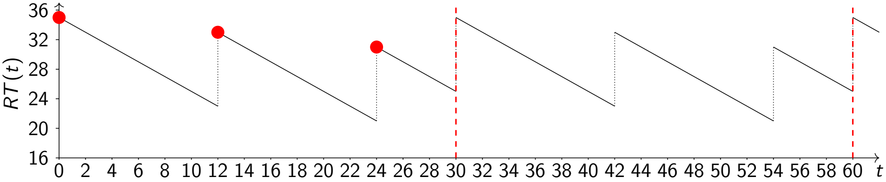


$$\mathcal{A}^1 = \{(0,35),(6,29),(12,33),(18,27),(24,31),(30,35),(36,29),(42,33),(48,27),(54,31),(60,35),\dots\}$$

$$\mathcal{A}^* = \{(0,35),(12,33),(24,31),(30,35),(42,33),(54,31),(60,35),\dots\} \qquad \mathcal{A}^*|_H = \{(0,35),(12,33),(24,31)\}$$

1. Remove redundant anchor points from $\mathcal{A}^1$
2. Consider first hyperperiod $[0, H)$ with $H = \text{LCM}(T_1, \dots, T_n)$ ($= 30$ for example)

Minimal Set of Anchor Points



$$\mathcal{A}^1 = \{(0,35),(6,29),(12,33),(18,27),(24,31),(30,35),(36,29),(42,33),(48,27),(54,31),(60,35),\dots\}$$

$$\mathcal{A}^* = \{(0,35),(12,33),(24,31),(30,35),(42,33),(54,31),(60,35),\dots\} \qquad \mathcal{A}^*|_H = \{(0,35),(12,33),(24,31)\}$$

1. Remove redundant anchor points from $\mathcal{A}^1$
2. Consider first hyperperiod $[0, H)$ with $H = \text{LCM}(T_1, \dots, T_n)$ (= 30 for example)

$\Rightarrow \mathcal{A}^*|_H$ is sufficient to describe the whole shape of RT

Analysis of Metrics Becomes Simple

Given: Anchor points $\mathcal{A}^*|_H = \{(x_0, y_0), \dots, (x_g, y_g)\}$ over one hyperperiod $[0, H)$

Analysis of Metrics:

Analysis of Metrics Becomes Simple

Given: Anchor points $\mathcal{A}^*|_H = \{(x_0, y_0), \dots, (x_g, y_g)\}$ over one hyperperiod $[0, H)$

Analysis of Metrics:

- Maximum: $MaxRT = \max_{i=0,\dots,g} y_i$

Analysis of Metrics Becomes Simple

Given: Anchor points $\mathcal{A}^*|_H = \{(x_0, y_0), \dots, (x_g, y_g)\}$ over one hyperperiod $[0, H)$

Analysis of Metrics:

- Maximum: $MaxRT = \max_{i=0,\dots,g} y_i$

Running example: $\mathcal{A}^*|_H = \{(0, 35), (12, 33), (24, 31)\}$

- $MaxRT = \max(35, 33, 31) = 35$

Analysis of Metrics Becomes Simple

Given: Anchor points $\mathcal{A}^*|_H = \{(x_0, y_0), \dots, (x_g, y_g)\}$ over one hyperperiod $[0, H)$

Analysis of Metrics:

- Maximum: $MaxRT = \max_{i=0,\dots,g} y_i$
- Minimum: $MinRT = \min_{i=0,\dots,g}(y_i - (x_{i+1} - x_i))$

Running example: $\mathcal{A}^*|_H = \{(0, 35), (12, 33), (24, 31)\}$

- $MaxRT = \max(35, 33, 31) = 35$

Analysis of Metrics Becomes Simple

Given: Anchor points $\mathcal{A}^*|_H = \{(x_0, y_0), \ldots, (x_g, y_g)\}$ over one hyperperiod $[0, H)$

Analysis of Metrics:

- Maximum: $MaxRT = \max_{i=0,\ldots,g} y_i$
- Minimum: $MinRT = \min_{i=0,\ldots,g}(y_i - (x_{i+1} - x_i))$

Running example: $\mathcal{A}^*|_H = \{(0, 35), (12, 33), (24, 31)\}$

- $MaxRT = \max(35, 33, 31) = 35$
- $MinRT = \min(35 - (12 - 0), 33 - (24 - 12), 31 - (30 - 24)) = \min(23, 21, 25) = 21$

Analysis of Metrics Becomes Simple

Given: Anchor points $\mathcal{A}^*|_H = \{(x_0, y_0), \dots, (x_g, y_g)\}$ over one hyperperiod $[0, H)$

Analysis of Metrics:

- Maximum: $MaxRT = \max_{i=0,\dots,g} y_i$
- Minimum: $MinRT = \min_{i=0,\dots,g}(y_i - (x_{i+1} - x_i))$
- Average: $AvRT = \frac{1}{2H} \sum_{i=0}^{g}(x_{i+1} - x_i) \cdot (y_i + \hat{y}_i)$

Running example: $\mathcal{A}^*|_H = \{(0, 35), (12, 33), (24, 31)\}$

- $MaxRT = \max(35, 33, 31) = 35$
- $MinRT = \min(35 - (12 - 0), 33 - (24 - 12), 31 - (30 - 24)) = \min(23, 21, 25) = 21$

Analysis of Metrics Becomes Simple

Given: Anchor points $\mathcal{A}^*|_H = \{(x_0, y_0), \dots, (x_g, y_g)\}$ over one hyperperiod $[0, H)$

Analysis of Metrics:

- Maximum: $MaxRT = \max_{i=0,\dots,g} y_i$
- Minimum: $MinRT = \min_{i=0,\dots,g}(y_i - (x_{i+1} - x_i))$
- Average: $AvRT = \frac{1}{2H} \sum_{i=0}^{g} (x_{i+1} - x_i) \cdot (y_i + \hat{y}_i)$

Running example: $\mathcal{A}^*|_H = \{(0, 35), (12, 33), (24, 31)\}$

- $MaxRT = \max(35, 33, 31) = 35$
- $MinRT = \min(35 - (12 - 0), 33 - (24 - 12), 31 - (30 - 24)) = \min(23, 21, 25) = 21$
- $AvRT = \frac{1}{2 \cdot 30} \cdot (12 \cdot 58 + 12 \cdot 54 + 6 \cdot 56) = \frac{1}{60} \cdot 1680 = 28$

Analysis of Metrics Becomes Simple

Given: Anchor points $\mathcal{A}^*|_H = \{(x_0, y_0), \dots, (x_g, y_g)\}$ over one hyperperiod $[0, H)$

Analysis of Metrics:

- Maximum: $MaxRT = \max_{i=0,\dots,g} y_i$
- Minimum: $MinRT = \min_{i=0,\dots,g}(y_i - (x_{i+1} - x_i))$
- Average: $AvRT = \frac{1}{2H} \sum_{i=0}^{g}(x_{i+1} - x_i) \cdot (y_i + \hat{y}_i)$
- Throughput: $Thr = \frac{1}{H} \cdot |\mathcal{A}^*|_H|$

Running example: $\mathcal{A}^*|_H = \{(0, 35), (12, 33), (24, 31)\}$

- $MaxRT = \max(35, 33, 31) = 35$
- $MinRT = \min(35 - (12 - 0), 33 - (24 - 12), 31 - (30 - 24)) = \min(23, 21, 25) = 21$
- $AvRT = \frac{1}{2 \cdot 30} \cdot (12 \cdot 58 + 12 \cdot 54 + 6 \cdot 56) = \frac{1}{60} \cdot 1680 = 28$

Analysis of Metrics Becomes Simple

Given: Anchor points $\mathcal{A}^*|_H = \{(x_0, y_0), \ldots, (x_g, y_g)\}$ over one hyperperiod $[0, H)$

Analysis of Metrics:

- Maximum: $MaxRT = \max_{i=0,\ldots,g} y_i$
- Minimum: $MinRT = \min_{i=0,\ldots,g}(y_i - (x_{i+1} - x_i))$
- Average: $AvRT = \frac{1}{2H} \sum_{i=0}^{g} (x_{i+1} - x_i) \cdot (y_i + \hat{y}_i)$
- Throughput: $Thr = \frac{1}{H} \cdot |\mathcal{A}^*|_H|$

Running example: $\mathcal{A}^*|_H = \{(0, 35), (12, 33), (24, 31)\}$

- $MaxRT = \max(35, 33, 31) = 35$
- $MinRT = \min(35 - (12 - 0), 33 - (24 - 12), 31 - (30 - 24)) = \min(23, 21, 25) = 21$
- $AvRT = \frac{1}{2 \cdot 30} \cdot (12 \cdot 58 + 12 \cdot 54 + 6 \cdot 56) = \frac{1}{60} \cdot 1680 = 28$
- $Thr = \frac{1}{30} \cdot 3 = 0.1$

More Metrics and Insights in the Paper

More Metrics and Insights in the Paper

- Variants of MRT
  - Reactive Time
  - Maximum Reduced Reaction Time (MRRT)

More Metrics and Insights in the Paper

- Variants of MRT
  - Reactive Time
  - Maximum Reduced Reaction Time (MRRT)

- Metrics for Bounded Violations of RT constraints $\Rightarrow$ Two proposals:

More Metrics and Insights in the Paper

- Variants of MRT
  - Reactive Time
  - Maximum Reduced Reaction Time (MRRT)

- Metrics for Bounded Violations of RT constraints $\Rightarrow$ Two proposals:
  - Weakly-Hard Chain-Level (m, k) Constraints
    "m out of k data propagations can exceed the constraint"

More Metrics and Insights in the Paper

- Variants of MRT
  - Reactive Time
  - Maximum Reduced Reaction Time (MRRT)

- Metrics for Bounded Violations of RT constraints $\Rightarrow$ Two proposals:
  - Weakly-Hard Chain-Level (m, k) Constraints
    "m out of k data propagations can exceed the constraint"
  - Longest Consecutive Exceedance
    "largest time interval that constraint is exceeded"

More Metrics and Insights in the Paper

- Variants of MRT
  - Reactive Time
  - Maximum Reduced Reaction Time (MRRT)

- Metrics for Bounded Violations of RT constraints $\Rightarrow$ Two proposals:
  - Weakly-Hard Chain-Level (m, k) Constraints
    "m out of k data propagations can exceed the constraint"
  - Longest Consecutive Exceedance
    "largest time interval that constraint is exceeded"

- Determine $\mathcal{A}^*|_H$ efficiently using *partitioned job chains*
  $\Rightarrow$ Speedup of $\approx$1000x for automotive tasks

Implementation in Python

https://github.com/marioguenzel/Shape-Aware-E2E.git

Implementation in Python

https://github.com/marioguenzel/Shape-Aware-E2E.git

- Simple specification of cause-effect chains:

```
{"ID": 0, "tasks": [{"phase": 0, "period": 6, "deadline": 6}, {"phase": 0,
 "period": 10, "deadline": 10}, {"phase": 0, "period": 5, "deadline": 5}]}
```

Implementation in Python

https://github.com/marioguenzel/Shape-Aware-E2E.git

- Simple specification of cause-effect chains:

```
{"ID": 0, "tasks": [{"phase": 0, "period": 6, "deadline": 6}, {"phase": 0,
 "period": 10, "deadline": 10}, {"phase": 0, "period": 5, "deadline": 5}]}
```

- One command to run the analysis:

```
> python analysis.py chains/paper_running_example.jsonl

{"ID": 0, "MaxRT": 35, "MaxRedRT": 29, "Reac": 31, "MinRT": 21, "AvRT":
28.0, "throughp": 0.1, "analysis_time_sec": 4.1961669921875e-05}
```

Implementation in Python

https://github.com/marioguenzel/Shape-Aware-E2E.git
(Artifact Evaluated!)

- Simple specification of cause-effect chains:

```
{"ID": 0, "tasks": [{"phase": 0, "period": 6, "deadline": 6}, {"phase": 0,
 "period": 10, "deadline": 10}, {"phase": 0, "period": 5, "deadline": 5}]}
```

- One command to run the analysis:

```
> python analysis.py chains/paper_running_example.jsonl

{"ID": 0, "MaxRT": 35, "MaxRedRT": 29, "Reac": 31, "MinRT": 21, "AvRT":
28.0, "throughp": 0.1, "analysis_time_sec": 4.1961669921875e-05}
```

Evaluation of Case Studies

Case Study	Max	Min	Av	Thr	(m,k)	LE
Wat17-C1	50	40	45.0	0.100	(0,10)	2.50
Wat17-C2	212	112	162.0	0.010	(0,10)	10.60
Wat19-C1	908	470	689.0	0.003	(1,10)	45.40
Wat19-C2	855	445	650.0	0.003	(4,10)	42.75
Wat19-C3	65	45	55.0	0.067	(0,10)	3.25
Wat19-C4	98	53	75.5	0.030	(0,10)	4.90
Wat19-C5	164	86	125.0	0.015	(0,10)	8.20
Wat19-C6	430	220	325.0	0.005	(0,10)	21.50
RTSS-C1	610	510	560.0	0.010	(0,10)	30.50
RTSS-C2	608	476	542.0	0.010	(0,10)	30.40
RTSS-C3	710	610	660.0	0.010	(0,10)	35.50
RTSS-C4	410	310	360.0	0.010	(0,10)	20.50
RTSS-C5	320	220	270.0	0.010	(1,10)	16.00
APD	275	225	250.0	0.020	(0,10)	13.75
Bec24	360	240	282.0	0.017	(0,10)	18.00
Gem21-UP	19	13	16.0	0.200	(0,10)	0.95
Gem21-LP	31	21	26.0	0.100	(0,10)	1.55
Iye20	360	310	335.0	0.020	(2,10)	18.00
Fre10-C1	45	35	40.0	0.100	(0,10)	2.25
Fre10-C2	35	25	30.0	0.100	(0,10)	1.75
Fre10-C3	55	45	50.0	0.100	(0,10)	2.75
Fre10-C4	45	35	40.0	0.100	(0,10)	2.25
Pag14-C1	70	50	60.0	0.050	(0,10)	3.50
Pag14-C2	50	30	40.0	0.050	(0,10)	2.50

Collection of Benchmarks:

- 24 cause-effect chains from 9 papers
- Representing industrial case studies and applications

Evaluation of Case Studies

Case Study	Max	Min	Av	Thr	(m,k)	LE
Wat17-C1	50	40	45.0	0.100	(0,10)	2.50
Wat17-C2	212	112	162.0	0.010	(0,10)	10.60
Wat19-C1	908	470	689.0	0.003	(1,10)	45.40
Wat19-C2	855	445	650.0	0.003	(4,10)	42.75
Wat19-C3	65	45	55.0	0.067	(0,10)	3.25
Wat19-C4	98	53	75.5	0.030	(0,10)	4.90
Wat19-C5	164	86	125.0	0.015	(0,10)	8.20
Wat19-C6	430	220	325.0	0.005	(0,10)	21.50
RTSS-C1	610	510	560.0	0.010	(0,10)	30.50
RTSS-C2	608	476	542.0	0.010	(0,10)	30.40
RTSS-C3	710	610	660.0	0.010	(0,10)	35.50
RTSS-C4	410	310	360.0	0.010	(0,10)	20.50
RTSS-C5	320	220	270.0	0.010	(1,10)	16.00
APD	275	225	250.0	0.020	(0,10)	13.75
Bec24	360	240	282.0	0.017	(0,10)	18.00
Gem21-UP	19	13	16.0	0.200	(0,10)	0.95
Gem21-LP	31	21	26.0	0.100	(0,10)	1.55
Iye20	360	310	335.0	0.020	(2,10)	18.00
Fre10-C1	45	35	40.0	0.100	(0,10)	2.25
Fre10-C2	35	25	30.0	0.100	(0,10)	1.75
Fre10-C3	55	45	50.0	0.100	(0,10)	2.75
Fre10-C4	45	35	40.0	0.100	(0,10)	2.25
Pag14-C1	70	50	60.0	0.050	(0,10)	3.50
Pag14-C2	50	30	40.0	0.050	(0,10)	2.50

Collection of Benchmarks:

- 24 cause-effect chains from 9 papers
- Representing industrial case studies and applications

Observations:

- **Few anchor points**:
  For 19 out of 24 chains: 3 or less anchor points suffice to describe the full shape of RT

Evaluation of Case Studies

Case Study	Max	Min	Av	Thr	(m,k)	LE
Wat17-C1	50	40	45.0	0.100	(0,10)	2.50
Wat17-C2	212	112	162.0	0.010	(0,10)	10.60
Wat19-C1	908	470	689.0	0.003	(1,10)	45.40
Wat19-C2	855	445	650.0	0.003	(4,10)	42.75
Wat19-C3	65	45	55.0	0.067	(0,10)	3.25
Wat19-C4	98	53	75.5	0.030	(0,10)	4.90
Wat19-C5	164	86	125.0	0.015	(0,10)	8.20
Wat19-C6	430	220	325.0	0.005	(0,10)	21.50
RTSS-C1	610	510	560.0	0.010	(0,10)	30.50
RTSS-C2	608	476	542.0	0.010	(0,10)	30.40
RTSS-C3	710	610	660.0	0.010	(0,10)	35.50
RTSS-C4	410	310	360.0	0.010	(0,10)	20.50
RTSS-C5	320	220	270.0	0.010	(1,10)	16.00
APD	275	225	250.0	0.020	(0,10)	13.75
Bec24	360	240	282.0	0.017	(0,10)	18.00
Gem21-UP	19	13	16.0	0.200	(0,10)	0.95
Gem21-LP	31	21	26.0	0.100	(0,10)	1.55
Iye20	360	310	335.0	0.020	(2,10)	18.00
Fre10-C1	45	35	40.0	0.100	(0,10)	2.25
Fre10-C2	35	25	30.0	0.100	(0,10)	1.75
Fre10-C3	55	45	50.0	0.100	(0,10)	2.75
Fre10-C4	45	35	40.0	0.100	(0,10)	2.25
Pag14-C1	70	50	60.0	0.050	(0,10)	3.50
Pag14-C2	50	30	40.0	0.050	(0,10)	2.50

Collection of Benchmarks:

- 24 cause-effect chains from 9 papers
- Representing industrial case studies and applications

Observations:

- **Few anchor points**:
  For 19 out of 24 chains: 3 or less anchor points suffice to describe the full shape of RT
- **Low Runtime**:
  Total analysis time: 0.062 seconds (single core with 2.25 GHz)

Runtime Comparison with SOTA

Two benchmarks:

Runtime Comparison with SOTA

Two benchmarks:

- WATERS: Periods in $\{1, 2, 5, 10, 20, 50, 100, 200, 1000\}$ with given distribution[1]

[1] Kramer et al. "Real world automotive benchmarks for free", WATERS Workshop, 2015.

Runtime Comparison with SOTA

Two benchmarks:

- WATERS: Periods in $\{1, 2, 5, 10, 20, 50, 100, 200, 1000\}$ with given distribution[1]
- UNIFORM: Periods drawn uniformly from $\{10, 20, \ldots, 200\}$

[1] Kramer et al. "Real world automotive benchmarks for free", WATERS Workshop, 2015.

Runtime Comparison with SOTA

Two benchmarks:

- WATERS: Periods in $\{1, 2, 5, 10, 20, 50, 100, 200, 1000\}$ with given distribution[1]
- UNIFORM: Periods drawn uniformly from $\{10, 20, \ldots, 200\}$

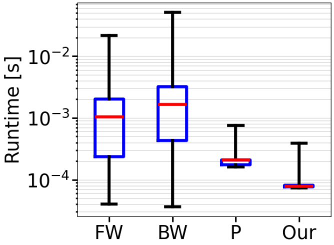
Approaches for Comparison:

Paper	**Approach**	**Metrics**
RTNS 2023 (FW)	Forward	MaxRT
RTAS 2023 (BW)	Backward	Reac
ECRTS 2023 (P)	Partitioned	MaxRT, MaxRedRT
This paper (Our)	Partitioned	MaxRT, MinRT, MaxRedRT, Reac, AvRT, Thr, (m,k), LE

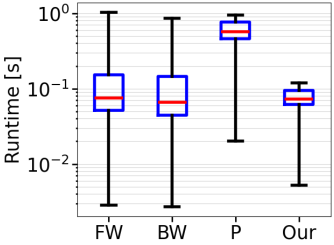
[1] Kramer et al. "Real world automotive benchmarks for free", WATERS Workshop, 2015.

Runtime Comparison with SOTA

Plots with $N = 50$ tasks per chain (more in the paper):



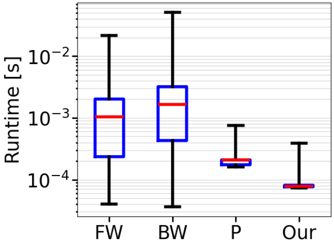
(WATERS)



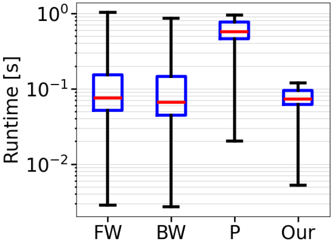
(UNIFORM)

Runtime Comparison with SOTA

Plots with $N = 50$ tasks per chain (more in the paper):



(WATERS)

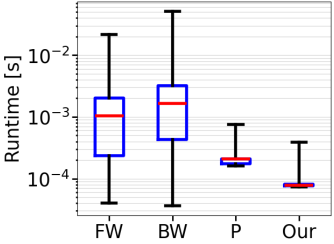


(UNIFORM)

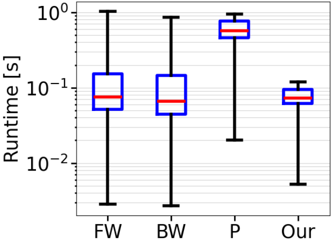
- Comparable runtime for UNIFORM and improved runtime for WATERS

Runtime Comparison with SOTA

Plots with $N = 50$ tasks per chain (more in the paper):



(WATERS)



(UNIFORM)

- Comparable runtime for UNIFORM and improved runtime for WATERS
- **Note:** Major effort (= determining anchor points) only *once*!
  $\Rightarrow$ Our analyzes *multiple* metrics in similar time

Conclusion and Future Work

- Cause-effect chains under LET

Conclusion and Future Work

- Cause-effect chains under LET
- Analysis that describes *full shape* of reaction time

Conclusion and Future Work

- Cause-effect chains under LET
- Analysis that describes *full shape* of reaction time
- Demonstrate: Allows simple (and fast!) analysis of multiple metrics

Conclusion and Future Work

- Cause-effect chains under LET
- Analysis that describes *full shape* of reaction time
- Demonstrate: Allows simple (and fast!) analysis of multiple metrics

However, this is only the first step!

⇒ *It is easy to integrate further metrics into the framework!*

Conclusion and Future Work

- Cause-effect chains under LET
- Analysis that describes *full shape* of reaction time
- Demonstrate: Allows simple (and fast!) analysis of multiple metrics

However, this is only the first step!

Framework[2]:

> ⇒ *It is easy to integrate further metrics into the framework!*

[2] `https://github.com/marioguenzel/Shape-Aware-E2E.git`

Conclusion and Future Work

- Cause-effect chains under LET
- Analysis that describes *full shape* of reaction time
- Demonstrate: Allows simple (and fast!) analysis of multiple metrics

However, this is only the first step!

$\Rightarrow$ *It is easy to integrate further metrics into the framework!*

Framework[2]:

Future work:

- Further metrics and exploitation of shape for design choices

[2] https://github.com/marioguenzel/Shape-Aware-E2E.git

Conclusion and Future Work

- Cause-effect chains under LET
- Analysis that describes *full shape* of reaction time
- Demonstrate: Allows simple (and fast!) analysis of multiple metrics

However, this is only the first step!

Framework[2]:

> ⇒ *It is easy to integrate further metrics into the framework!*

Future work:

- Further metrics and exploitation of shape for design choices
- Data-age related metrics

[2] `https://github.com/marioguenzel/Shape-Aware-E2E.git`

Conclusion and Future Work

- Cause-effect chains under LET
- Analysis that describes *full shape* of reaction time
- Demonstrate: Allows simple (and fast!) analysis of multiple metrics

However, this is only the first step!

Framework[2]:

> ⇒ *It is easy to integrate further metrics into the framework!*

Future work:

- Further metrics and exploitation of shape for design choices
- Data-age related metrics
- Other communication paradigms such as implicit communication

[2] https://github.com/marioguenzel/Shape-Aware-E2E.git