In Search of Butterflies: Exceedance Analysis for Real-Time Systems under Transient Overload

M. Zini[1], F. Marković[2], D. Casini[1], A. Biondi[1], and B. Brandenburg[2]

[1]*Scuola Superiore Sant'Anna, Pisa, Italy*

[2]*Max Planck Institute for Software Systems, Kaiserslautern, Germany*

Retis Real-Time Systems Laboratory

MAX PLANCK INSTITUTE FOR SOFTWARE SYSTEMS

WCET

- The theoretical analysis of real-time systems often relies on the concept of **worst-case execution time** (WCET).

WCET can be found with:

WCET

- The theoretical analysis of real-time systems often relies on the concept of **worst-case execution time** (WCET).

WCET can be found with:

Analytical methods

WCET

- The theoretical analysis of real-time systems often relies on the concept of **worst-case execution time** (WCET).

WCET can be found with:

Analytical methods

Empirical methods

Analytical Methods

Analytical methods

Analytical Methods

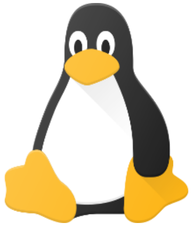
Analytical methods

PROBLEM

High complexity of modern hardware and software stacks.

Linux-based solutions in critical systems:

- Unmanned aerial vehicles
- Autonomous driving (e.g., Tesla)
- Spacecrafts (e.g., SpaceX)



Experimental Methods

Empirical methods

Experimental Methods

WCET is estimated with an experimental approach

Empirical methods

Experimental Methods

WCET is estimated with an experimental approach

The bound is not provably safe!!!

Empirical methods

Experimental Methods

WCET is estimated with an experimental approach

The bound is not provably safe!!!

Empirical methods

In practice, we are not dealing with a **worst-case execution time**, but rather with a

Nominal execution time (NET)

Real-Time Systems Laboratory

Nominal Execution Time (NET)

In practice, we are not dealing with a **worst-case execution time**, but rather with a

Nominal execution time (NET)

Nominal Execution Time (NET)

In practice, we are not dealing with a **worst-case execution time**, but rather with a

Nominal execution time (NET)

NETs can be **exceeded at run-time** due to many factors:

- Unaccounted interference effects
- Intentionally under-provisioned systems
  - The WATERS 2017 challenge's task set is unschedulable with WCETs
  - E.g., NET = 99^{th} percentile of observed execution times
- …

Real-Time Systems Laboratory

Nominal Execution Time (NET)

NETs cannot be trusted!

QUESTION

What happens if jobs **exceed** their NET at runtime?

Nominal Execution Time (NET)

NETs cannot be trusted!

QUESTION

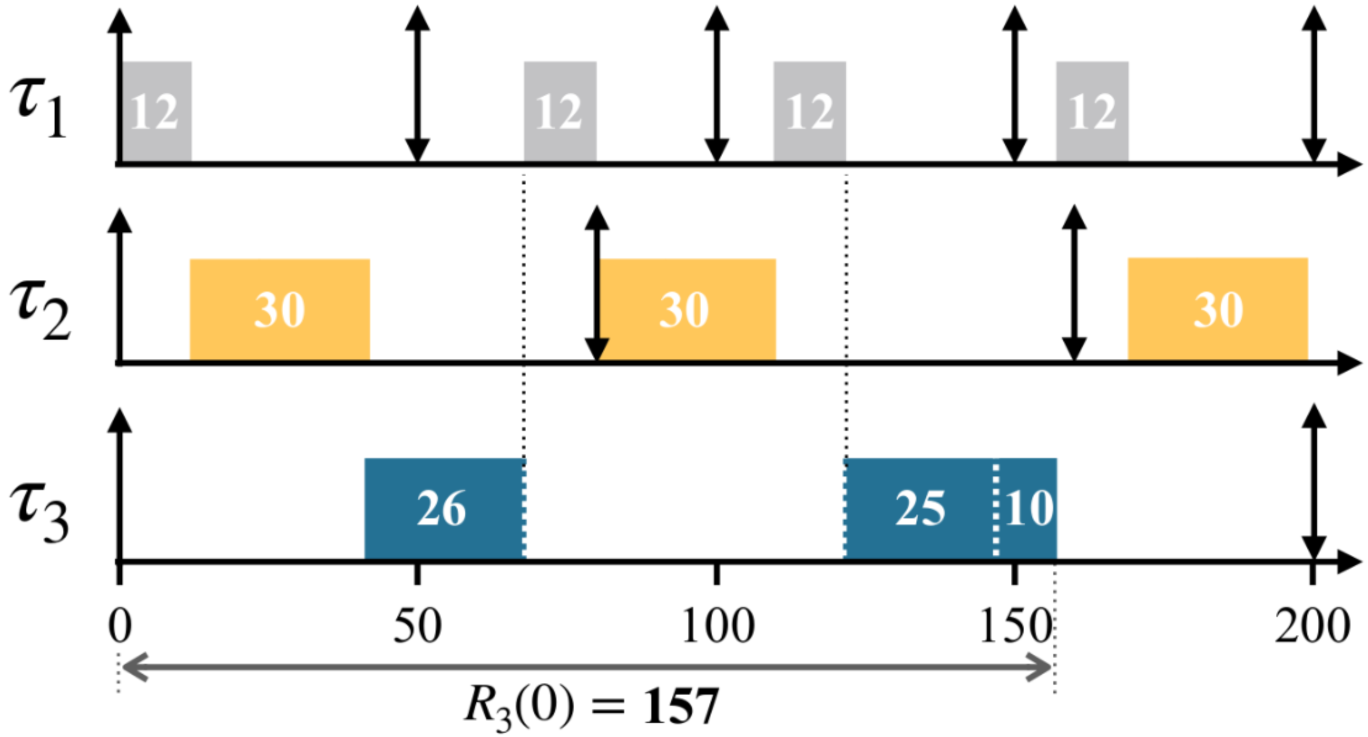
What happens if jobs **exceed** their NET at runtime?

We refer to the extra execution time as **exceedance**

Motivating Example

For example, consider this simple limited-preemptive taskset:

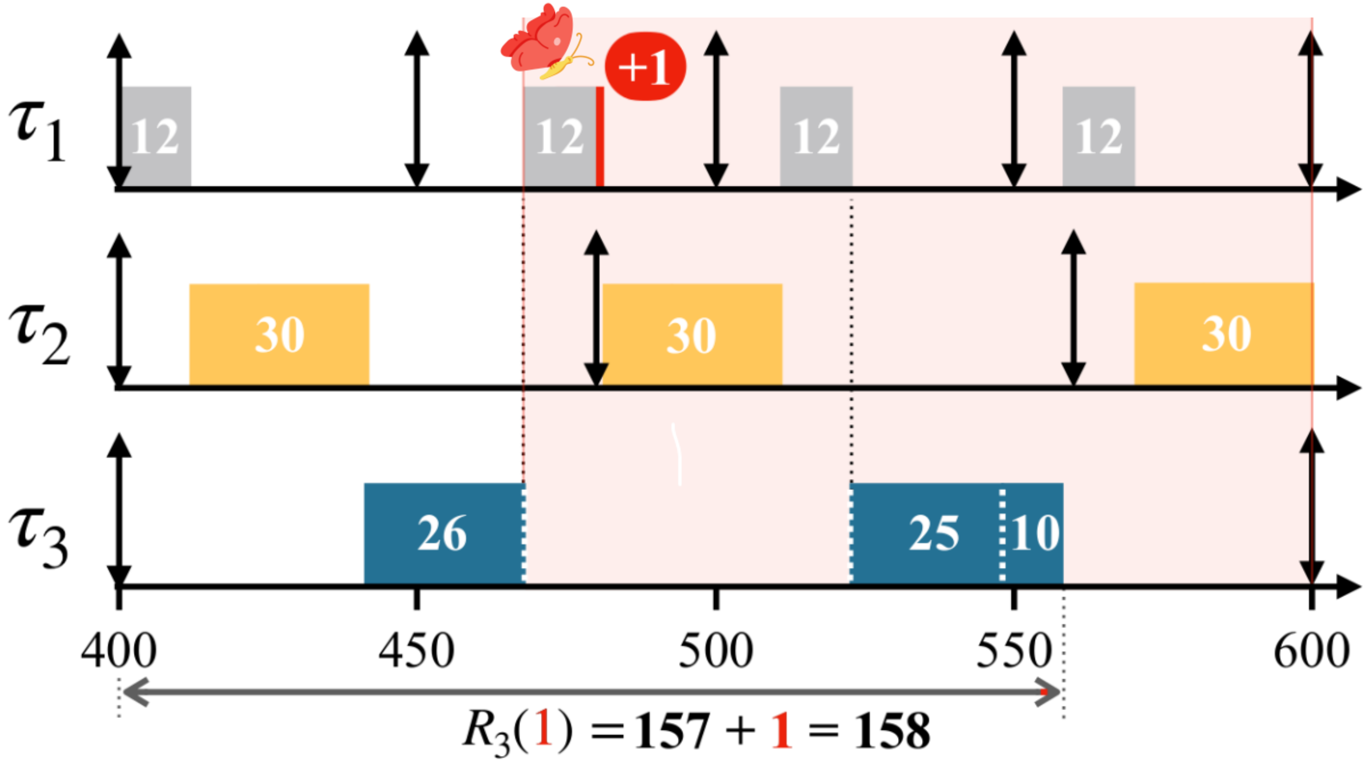
Task	Period	NET
τ_1	50	<12>
τ_2	80	<30>
τ_3	200	<26,25,10>



Motivating Example

We **add 1 unit of exceedance** to the second job of task τ_1

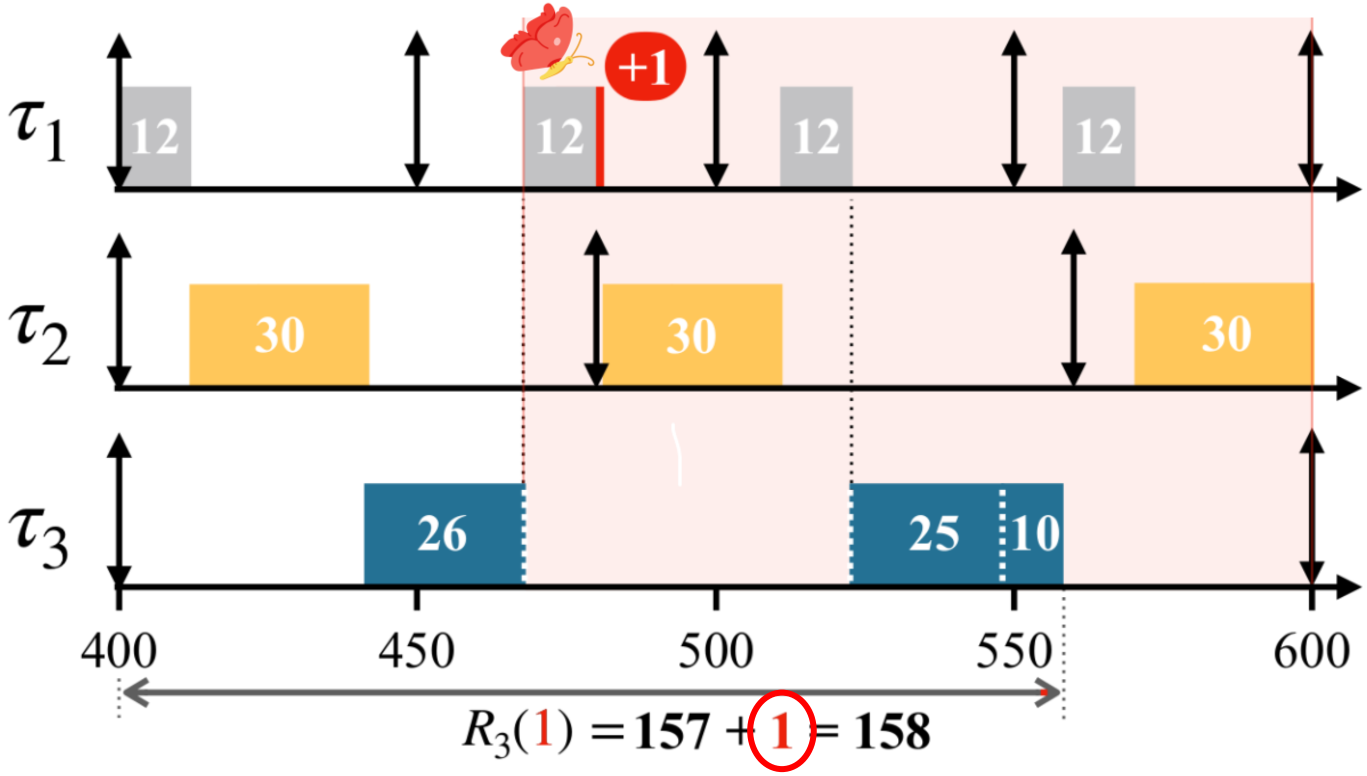
Task	Period	NET
τ_1	50	<12>
τ_2	80	<30>
τ_3	200	<26,25,10>



Motivating Example

We **add 1 unit of exceedance** to the second job of task τ_1

Task	Period	NET
τ_1	50	<12>
τ_2	80	<30>
τ_3	200	<26,25,10>

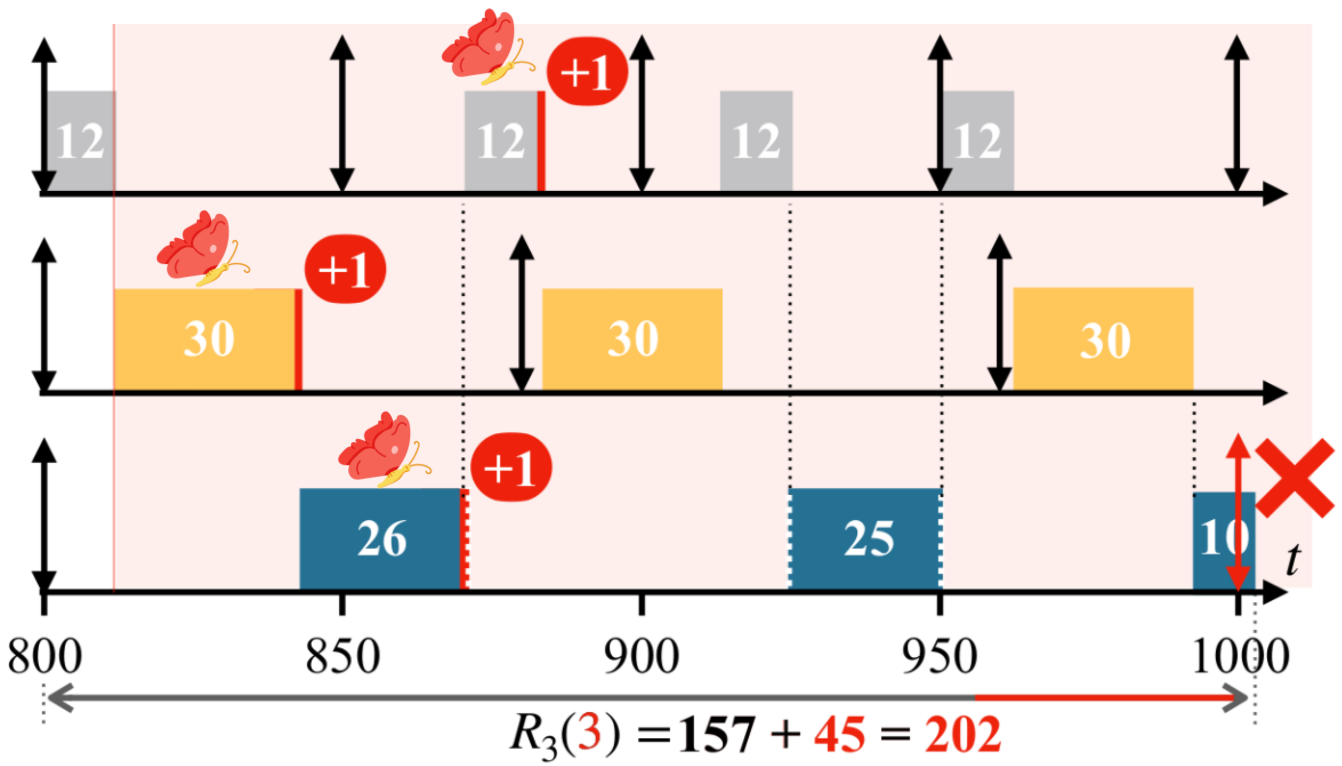


τ_3's response time increased by 1 time unit

Motivating Example

We add 1 unit of exceedance to the first job of task τ_2 and τ_3

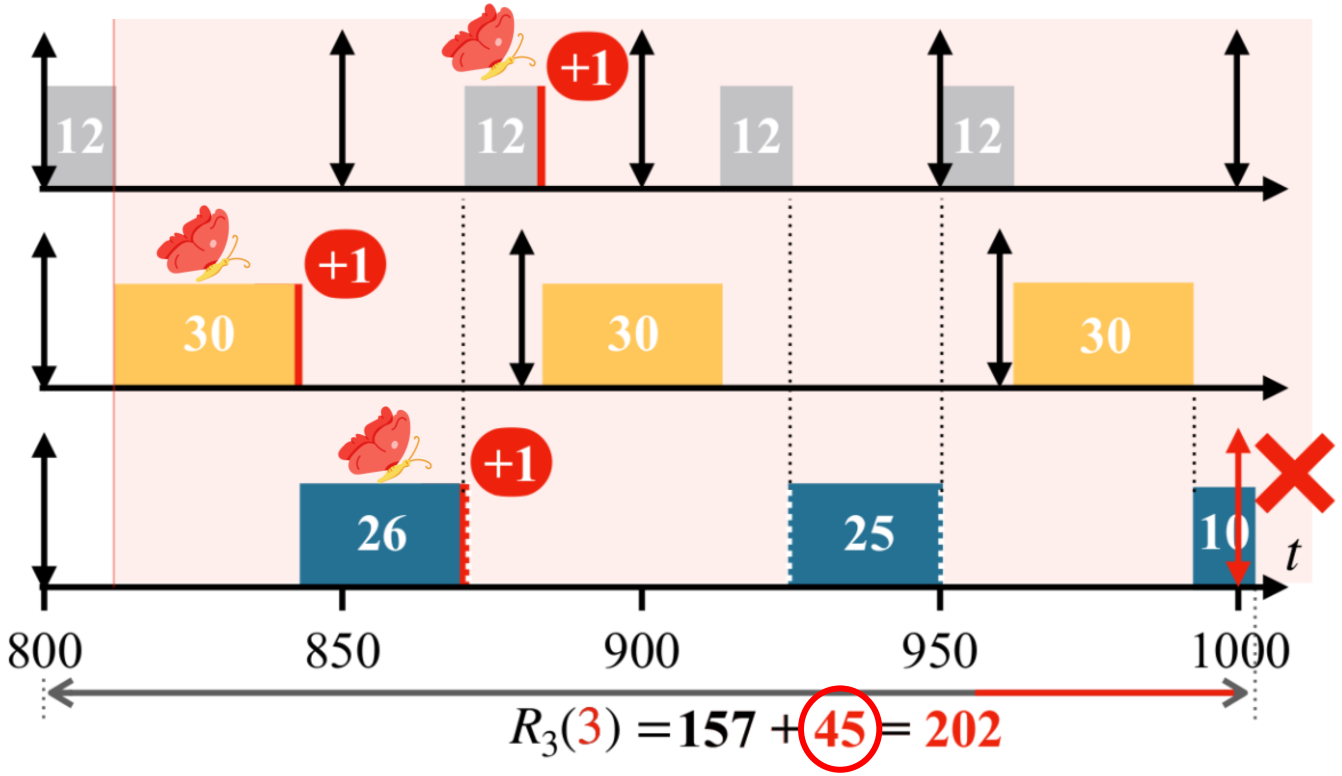
Task	Period	NET
τ_1	50	<12>
τ_2	80	<30>
τ_3	200	<26,25,10>



Motivating Example

We add 1 unit of exceedance to the first job of task τ_2 and τ_3

Task	Period	NET
τ_1	50	<12>
τ_2	80	<30>
τ_3	200	<26,25,10>



τ_3's response time increased by additional 43 time units!!!

Response-Time Nonlinearities

The consequences of **NET exceedance** are not easy to predict:

- NET + 1 ⟹ Response time + 1
- NET + 2 ⟹ Response time + 2
- NET + 3 ⟹ Response time **+ 45**
- …

Response-Time Nonlinearities

The consequences of **NET exceedance** are not easy to predict:

- NET + 1 ⟹ Response time + 1
- NET + 2 ⟹ Response time + 2
- NET + 3 ⟹ Response time **+ 45** → **Nonlinear** increase!
- …

Response-Time Nonlinearities

The consequences of **NET exceedance** are not easy to predict:

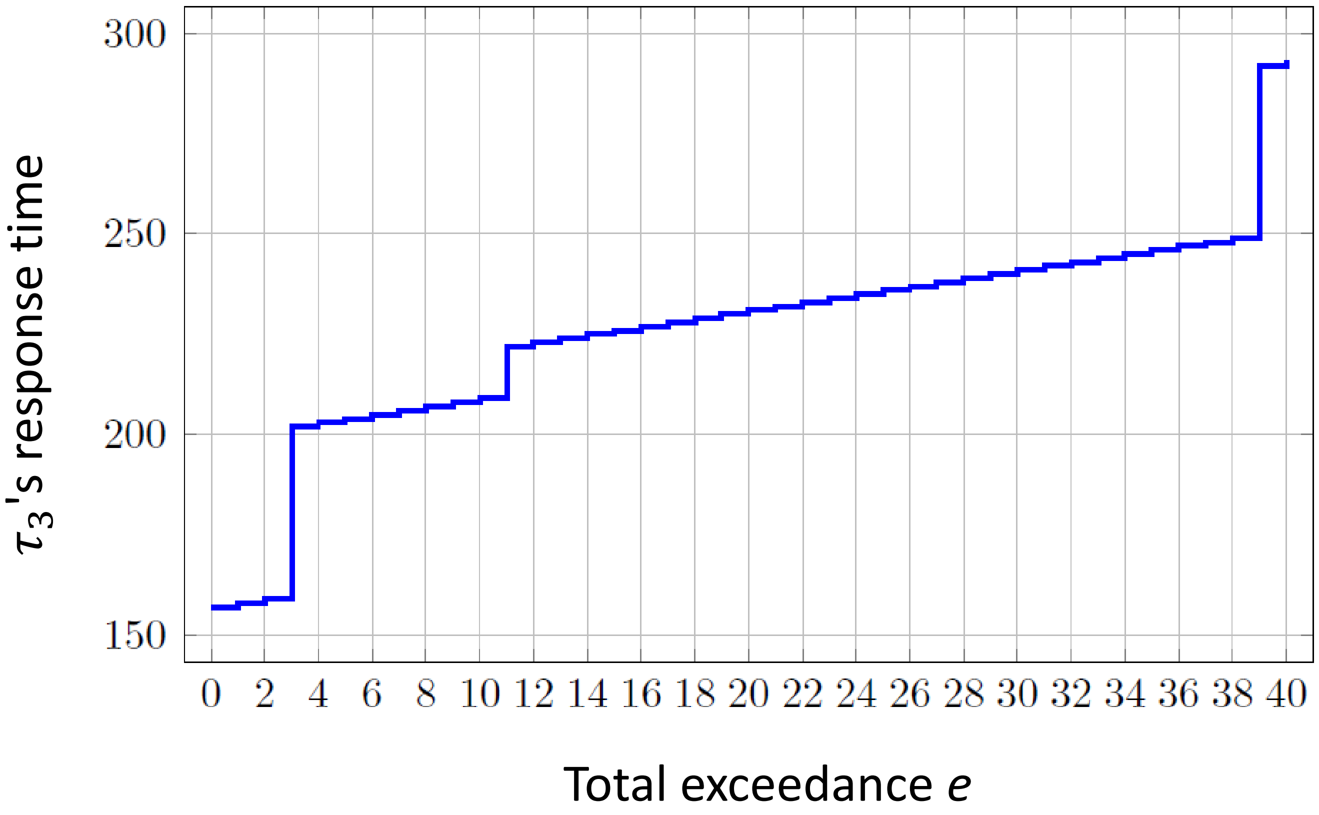
- NET + 1 ⟹ Response time + 1
- NET + 2 ⟹ Response time + 2
- NET + 3 ⟹ Response time **+ 45** → **Nonlinear** increase!
- …

If we are neglect this phenomena, we might **over-estimate the system's temporal safety margin**

Response-Time Nonlinearities

But response-time nonlinearities are difficult to predict

Task	Period	NET
τ_1	50	<12>
τ_2	80	<10, 20>
τ_3	200	<26,25,10>



Response-Time Nonlinearities

But response-time nonlinearities are difficult to predict

Task	Period	NET
τ_1	50	<12>
τ_2	80	<10, 20>
τ_3	200	<26,25,10>

Response-Time Nonlinearities

We understood that **nonlinearities** are:

- Dangerous for schedulability
- Difficult to predict

Real-Time Systems Laboratory

Response-Time Nonlinearities

We understood that **nonlinearities** are:

- Dangerous for schedulability
- Difficult to predict

We need a strategy to understand

how much the tasks can exceed the NETs

before generating a nonlinearity

Is It Sensitivity Analysis?

A similar problem has already been faced in literature with **sensitivity analysis**.

But sensitivity analysis:

Is It Sensitivity Analysis?

A similar problem has already been faced in literature with **sensitivity analysis**.

But sensitivity analysis:

Lacks support for:

- limited-preemptive
- fully non-preemtpive

Real-Time Systems Laboratory

Is It Sensitivity Analysis?

A similar problem has already been faced in literature with **sensitivity analysis**.

But sensitivity analysis:

Lacks support for:

- limited-preemptive
- fully non-preemtpive

Does not consider:

- Arbitrary arrival curves
- Multiple scheduling policies
- Arbitrary deadlines for FP

Is It Sensitivity Analysis?

A similar problem has already been faced in literature with **sensitivity analysis**.

But sensitivity analysis:

Lacks support for:

- limited-preemptive
- fully non-preemtpive

Does not consider:

- Arbitrary arrival curves
- Multiple scheduling policies
- Arbitrary deadlines for FP

Yields a scaling factor / additive constant

↓

Considers permanent overload

Retis Real-Time Systems Laboratory

Is It Sensitivity Analysis?

A similar problem has already been faced in literature with **sensitivity analysis**.

But sensitivity analysis:

Lacks support for:
- limited-preemptive
- fully non-preemtpive

Does not consider:
- Arbitrary arrival curves
- Multiple scheduling policies
- Arbitrary deadlines for FP

Yields a scaling factor / additive constant
↓
Considers permanent overload

Gives no information on nonlinearities before or after the deadlines

Real-Time Systems Laboratory

Exceedance-Aware RTA

Our solution is based on the **abstract Response-Time Analysis** framework [1] by Bozhko et al.

[1] S. Bozhko and B. B. Brandenburg. Abstract Response-Time Analysis: A Formal Foundation for the Busy-Window Principle, *Proceedings of the 32nd Euromicro Conference on Real-Time Systems (ECRTS 2020)*

Real-Time Systems Laboratory

Exceedance-Aware RTA

Our solution is based on the **abstract Response-Time Analysis** framework [1] by Bozhko et al.

Provides a unified and general *abstract response-time analysis*, independent of specific scheduling policies, workload models, and preemption policies

[1] S. Bozhko and B. B. Brandenburg. Abstract Response-Time Analysis: A Formal Foundation for the Busy-Window Principle, *Proceedings of the 32nd Euromicro Conference on Real-Time Systems (ECRTS 2020)*

Real-Time Systems Laboratory

Exceedance-Aware RTA

Our solution is based on the **abstract Response-Time Analysis** framework [1] by Bozhko et al.

Provides a unified and general *abstract response-time analysis*, independent of specific scheduling policies, workload models, and preemption policies

We extended the framework to support the **presence of exceedance** and provide concrete instantiations for **3 scheduling policies and 4 preemption models**:

[1] S. Bozhko and B. B. Brandenburg. Abstract Response-Time Analysis: A Formal Foundation for the Busy-Window Principle, *Proceedings of the 32nd Euromicro Conference on Real-Time Systems (ECRTS 2020)*

Retis
Real-Time Systems Laboratory

Exceedance-Aware RTA

Our solution is based on the **abstract Response-Time Analysis** framework [1] by Bozhko et al.

Provides a unified and general *abstract response-time analysis*, independent of specific scheduling policies, workload models, and preemption policies

We extended the framework to support the **presence of exceedance** and provide concrete instantiations for **3 scheduling policies and 4 preemption models**:

- Fixed priority
- EDF
- FIFO

- Fully preemptive
- Fully non-preemptive
- Segmented non-preemptive
- Floating non-preemptive

[1] S. Bozhko and B. B. Brandenburg. Abstract Response-Time Analysis: A Formal Foundation for the Busy-Window Principle, *Proceedings of the 32nd Euromicro Conference on Real-Time Systems (ECRTS 2020)*

Real-Time Systems Laboratory

Search for Nonlinearities

The function $\boldsymbol{R_i(e)}$ yields the response time of task τ_i with exceedance $\boldsymbol{e}$.

Search for Nonlinearities

The function $\boldsymbol{R_i(e)}$ yields the response time of task τ_i with exceedance $\boldsymbol{e}$.

The **"jumps"** of this function are the **nonlinearities** we are looking for!

Search for Nonlinearities

The function $\boldsymbol{R_i(e)}$ yields the response time of task τ_i with exceedance $\boldsymbol{e}$.

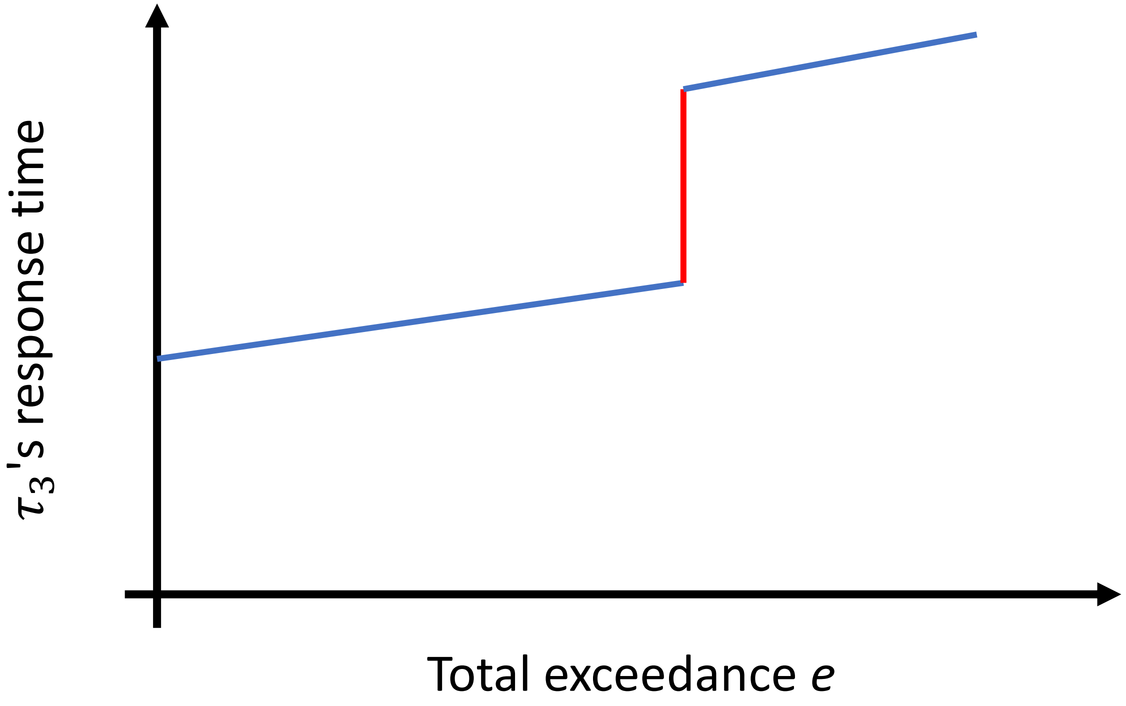
The **”jumps”** of this function are the **nonlinearities** we are looking for!

We developed an algorithm to **find such ”jumps”** efficiently

Search Algorithm

The search algorithm is based on:

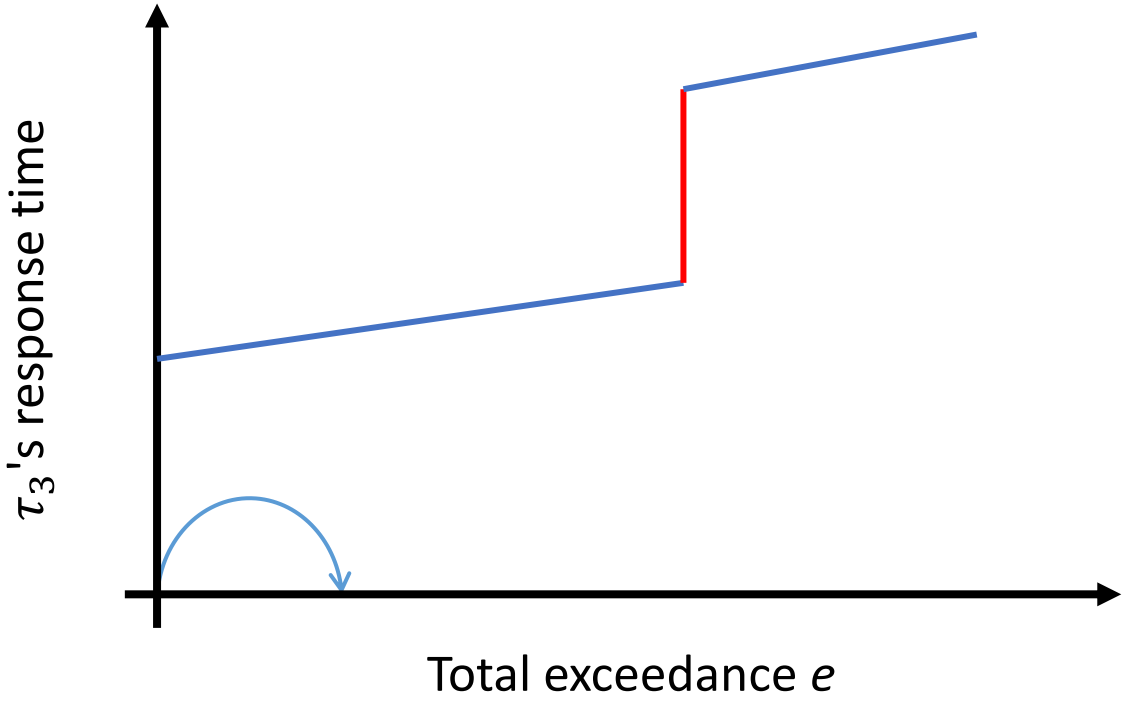
1. An exponential search of the **nonlinearity interval**



Search Algorithm

The search algorithm is based on:

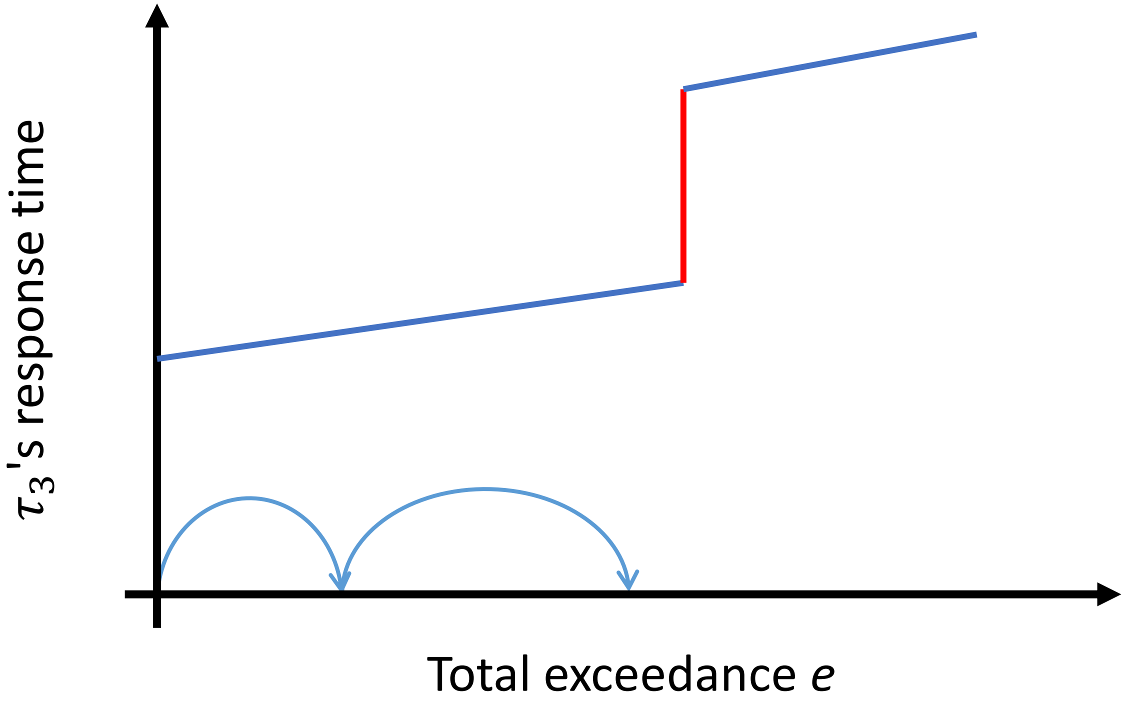
1. An exponential search of the **nonlinearity interval**



Search Algorithm

The search algorithm is based on:

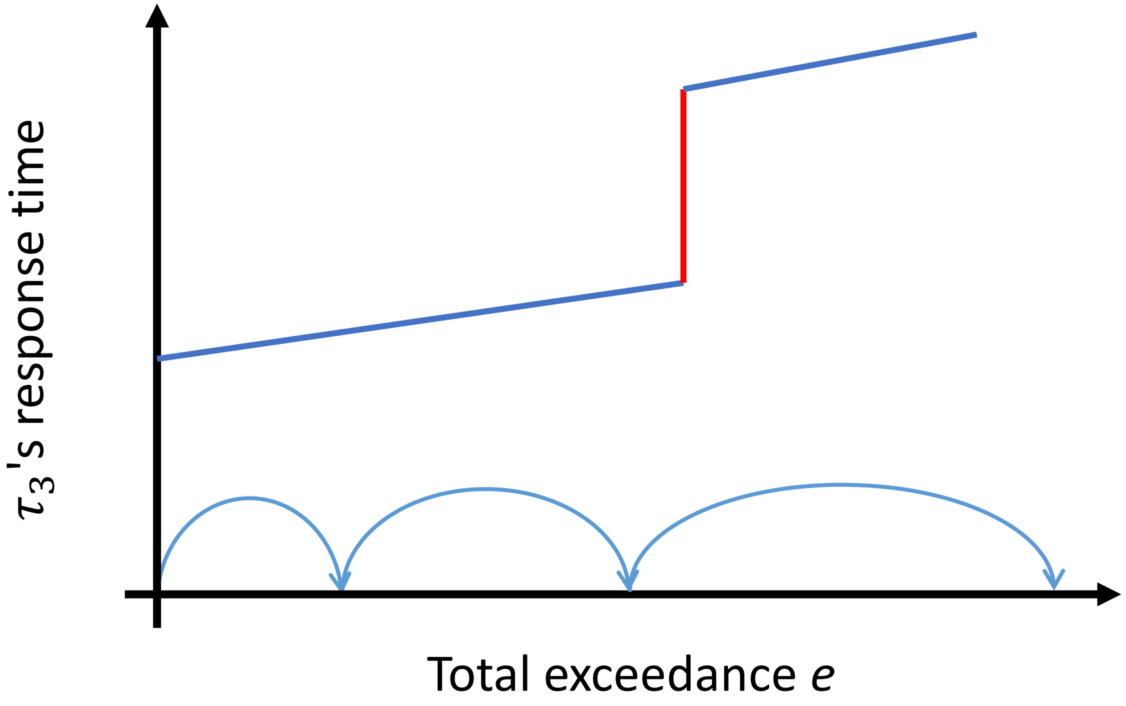
1. An exponential search of the **nonlinearity interval**



Search Algorithm

The search algorithm is based on:

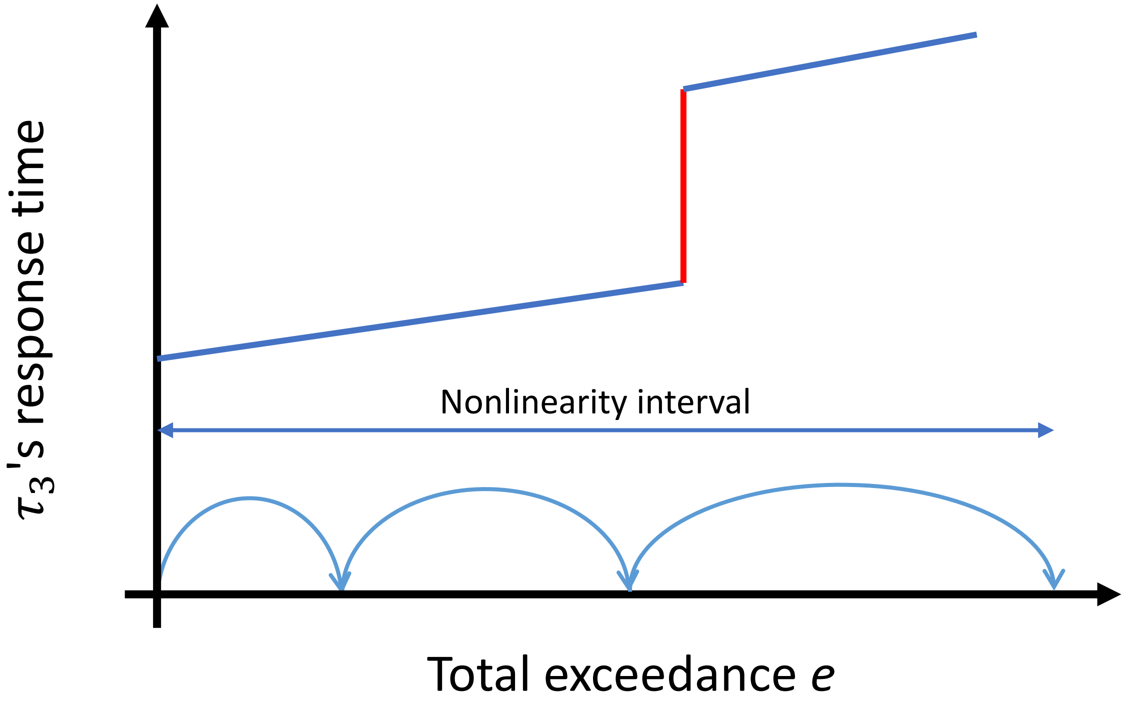
1. An exponential search of the **nonlinearity interval**



Search Algorithm

The search algorithm is based on:

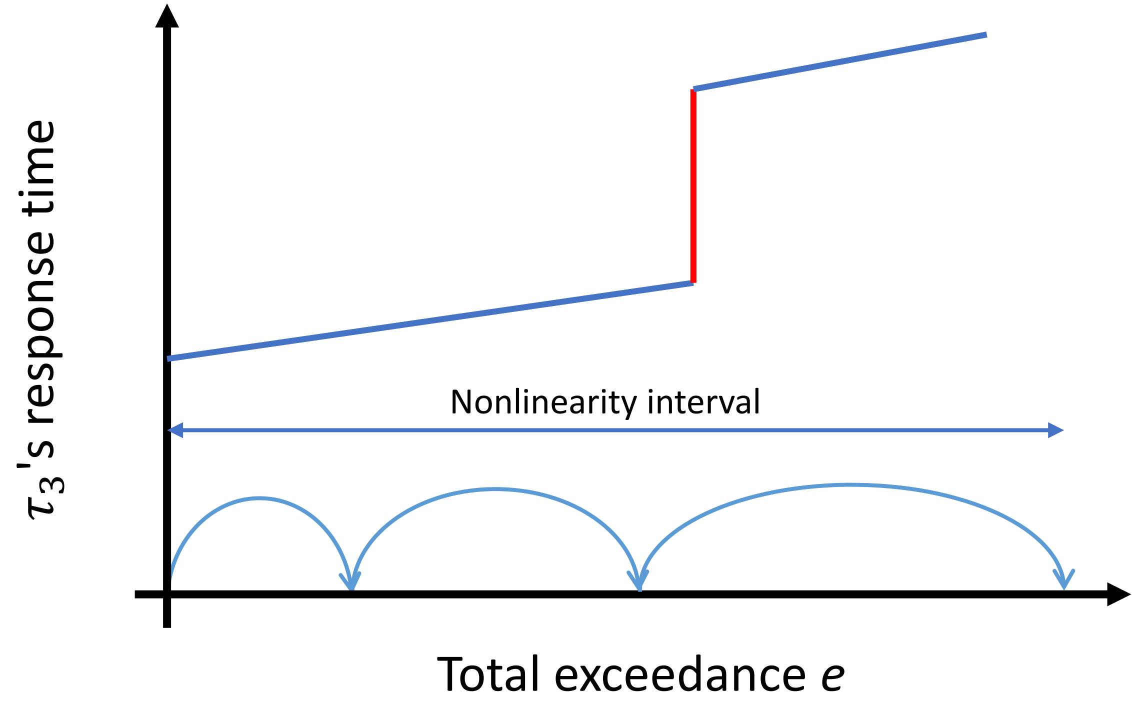
1. An exponential search of the **nonlinearity interval**



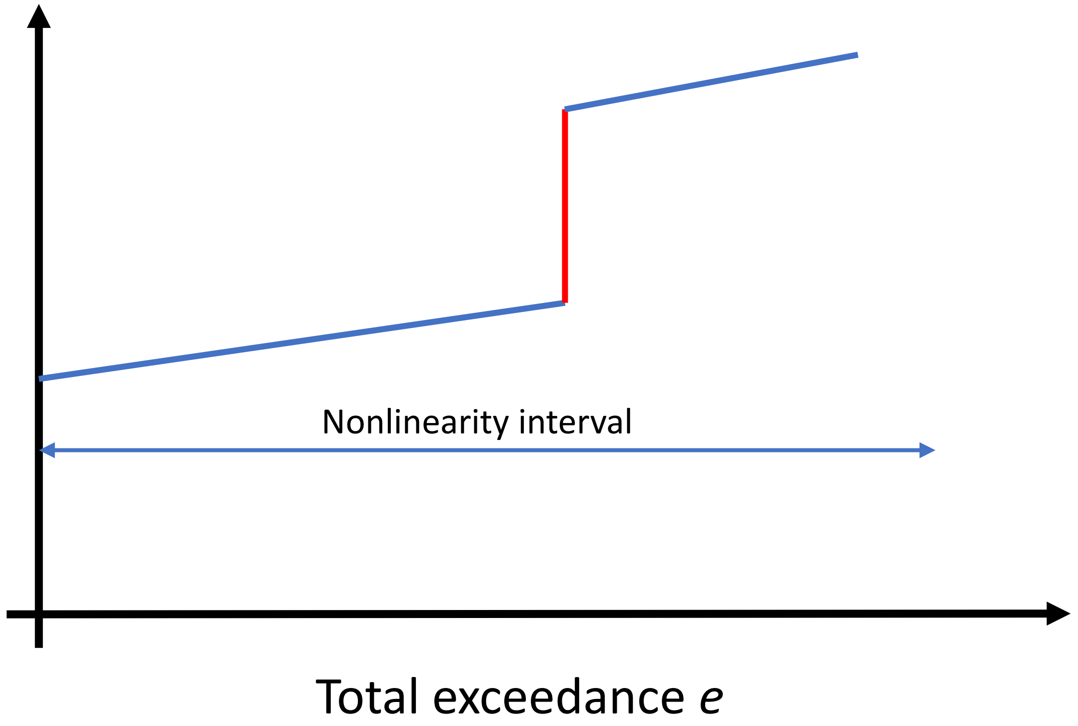
Search Algorithm

The search algorithm is based on:

1. An exponential search of the **nonlinearity interval**



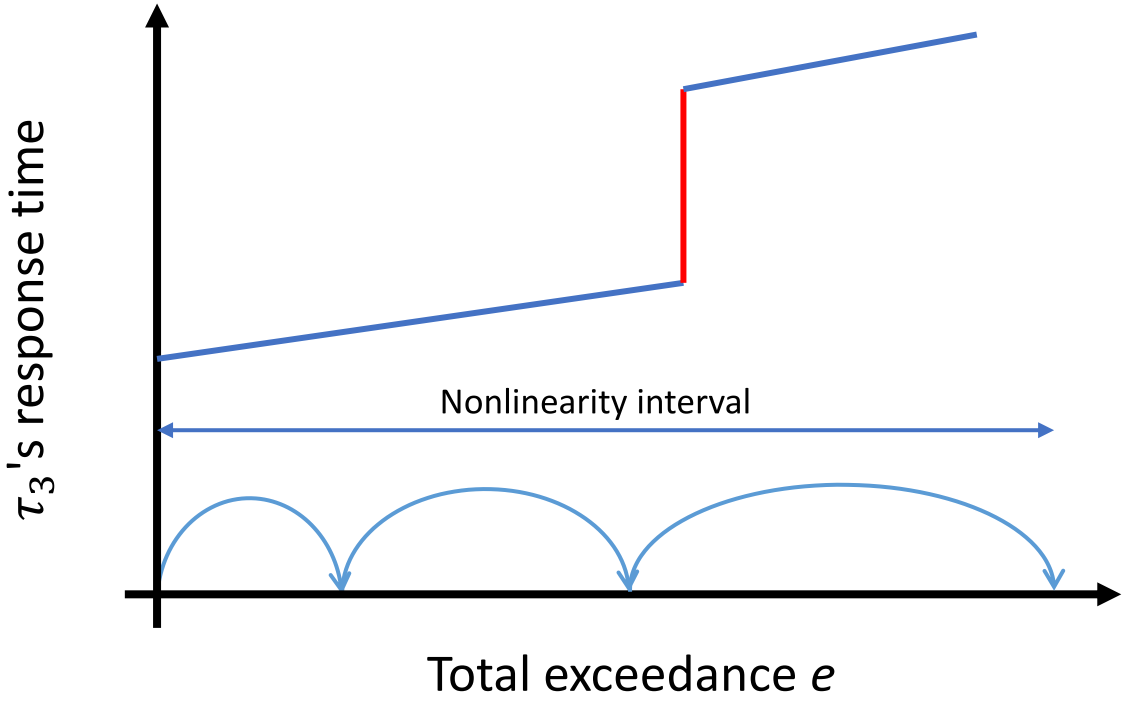
2. A binary search on the nonlinearity interval



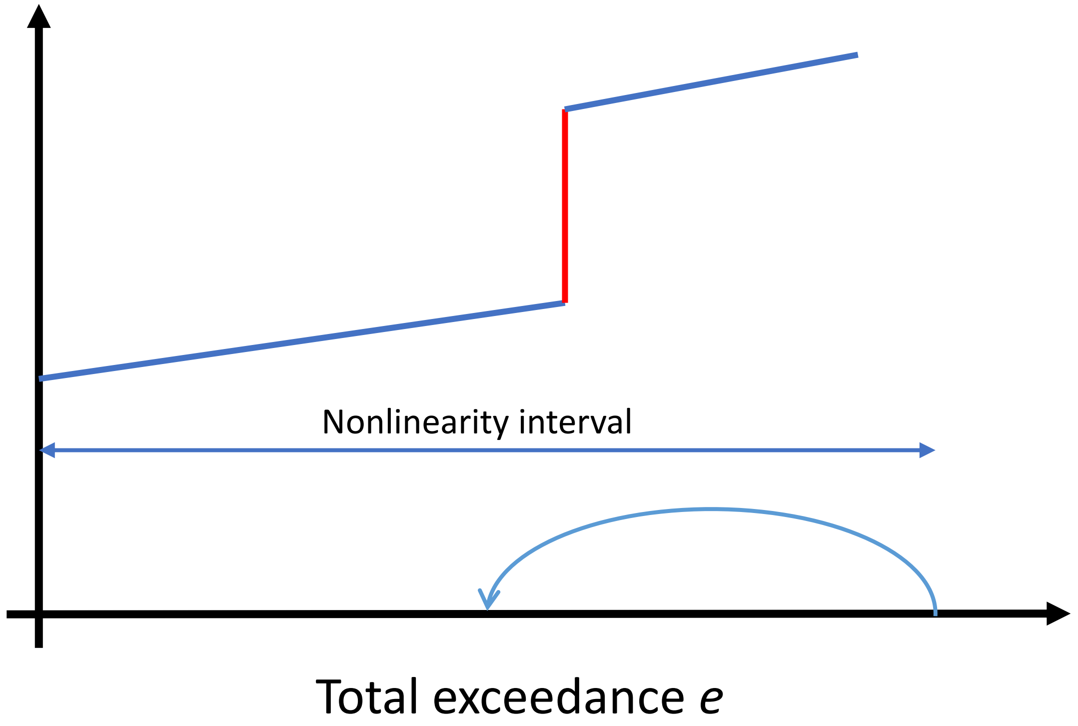
Search Algorithm

The search algorithm is based on:

1. An exponential search of the **nonlinearity interval**



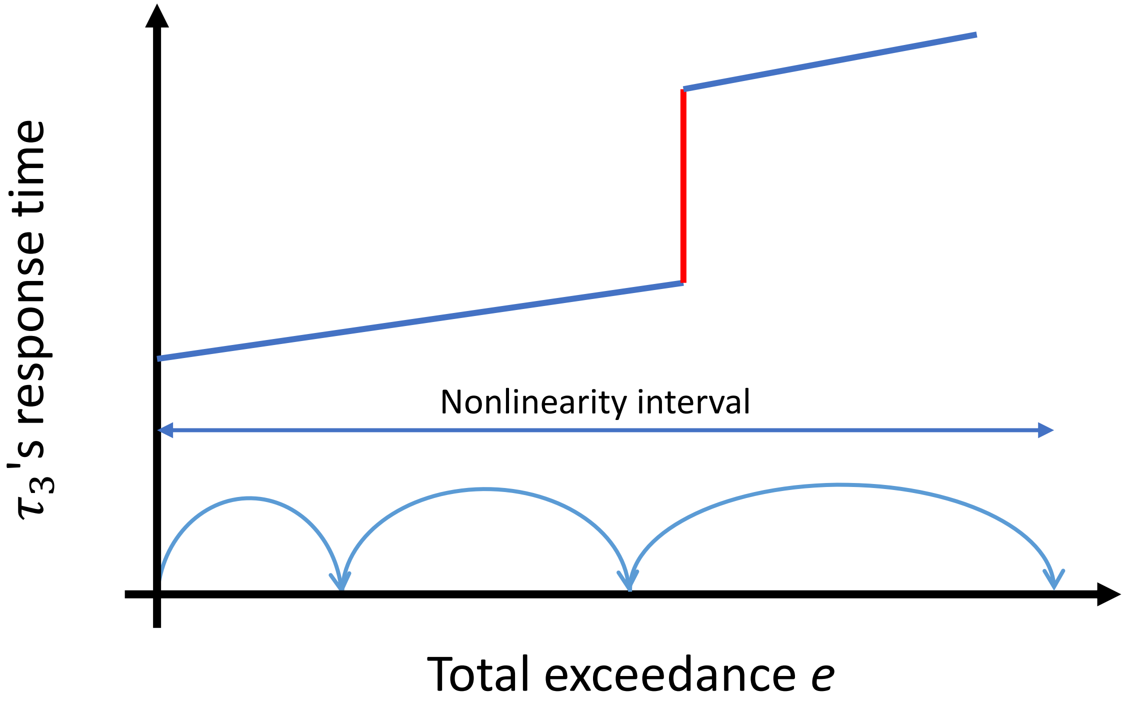
2. A binary search on the nonlinearity interval



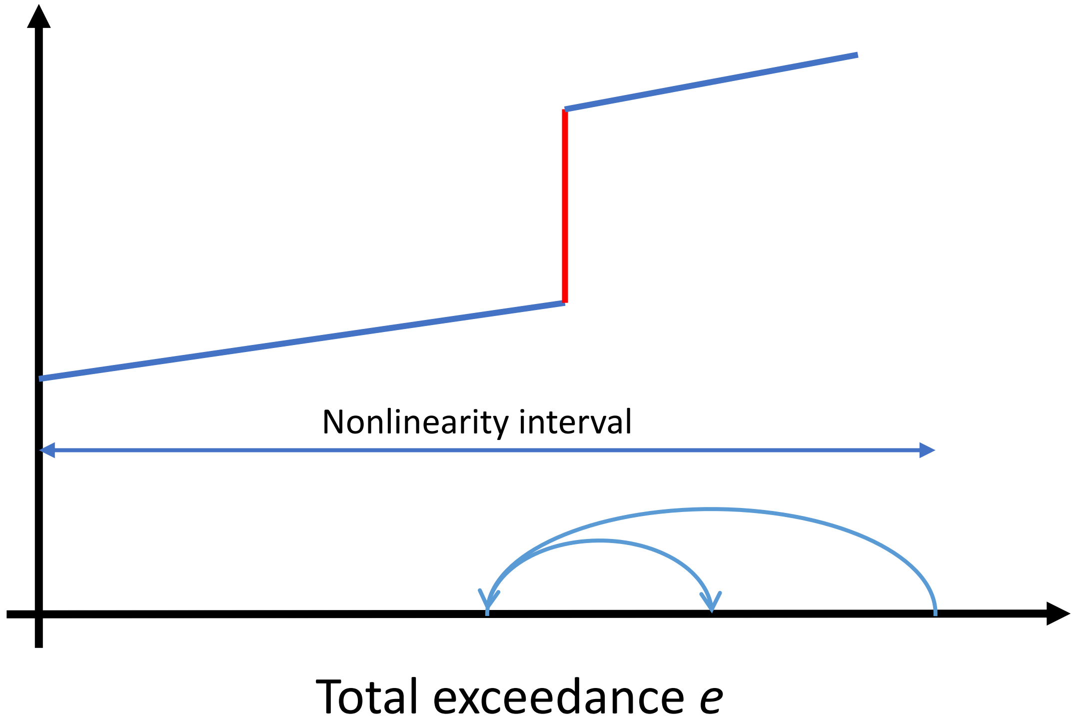
Search Algorithm

The search algorithm is based on:

1. An exponential search of the **nonlinearity interval**



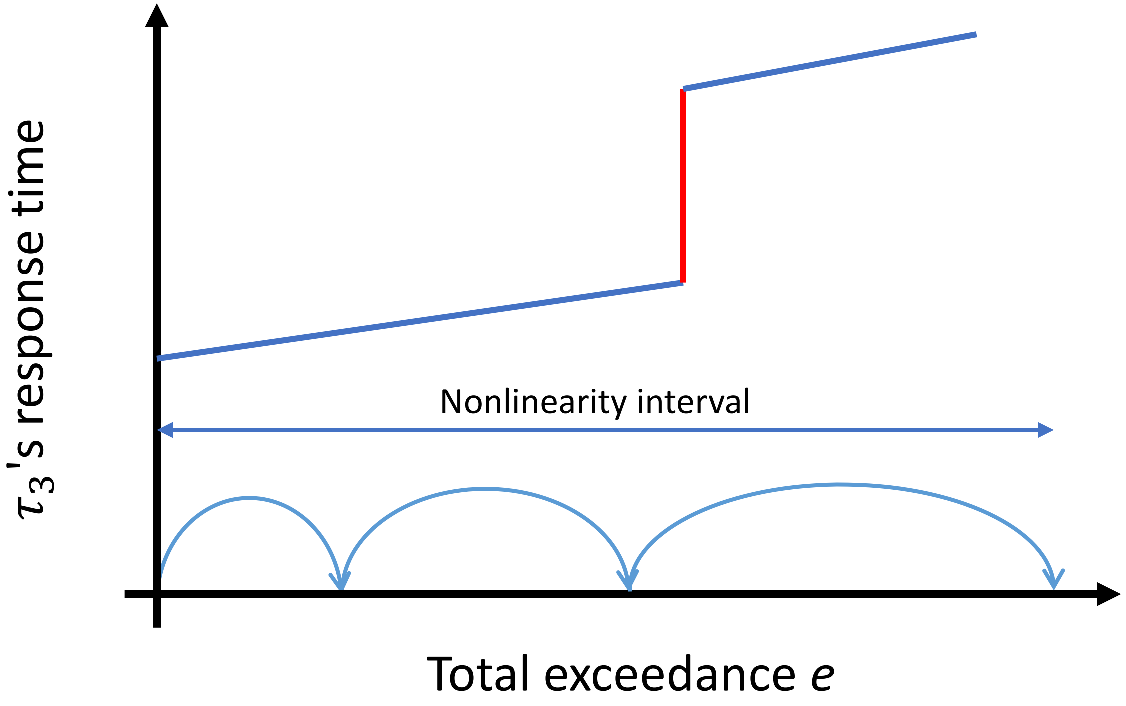
2. A binary search on the nonlinearity interval



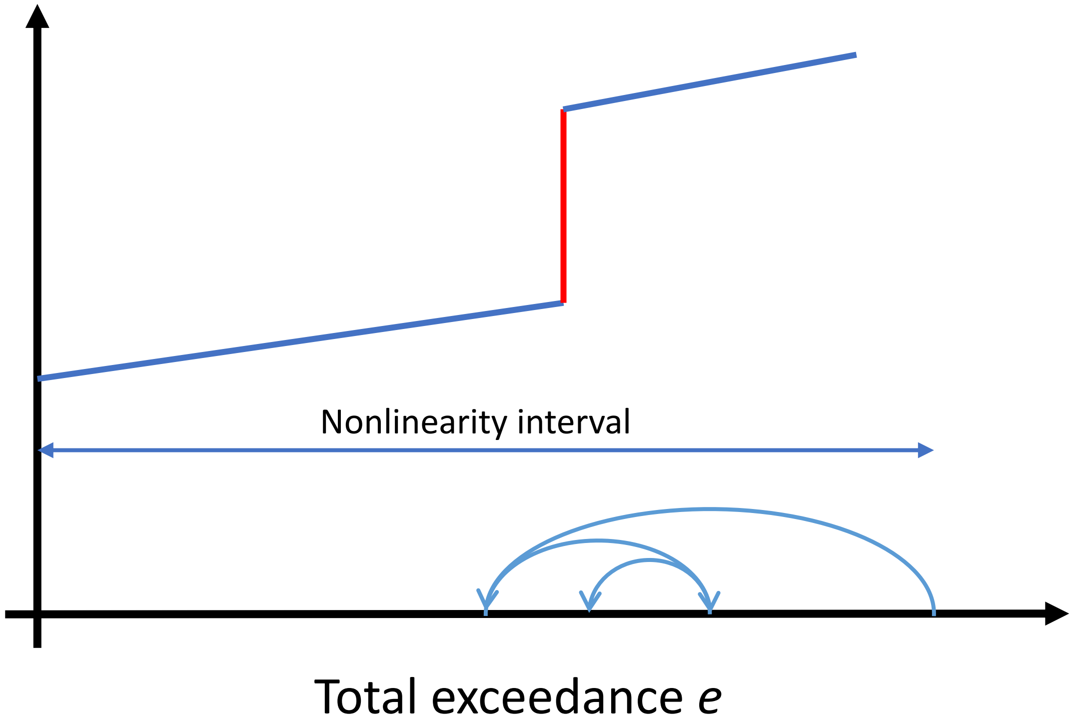
Search Algorithm

The search algorithm is based on:

1. An exponential search of the **nonlinearity interval**



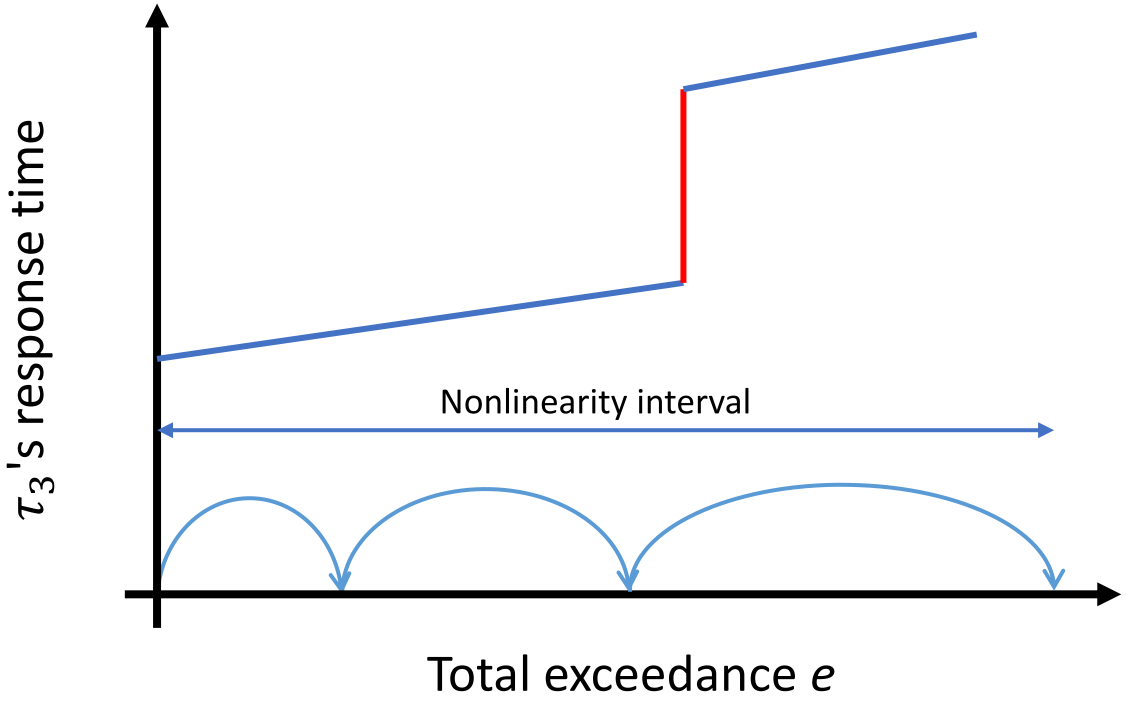
2. A binary search on the nonlinearity interval



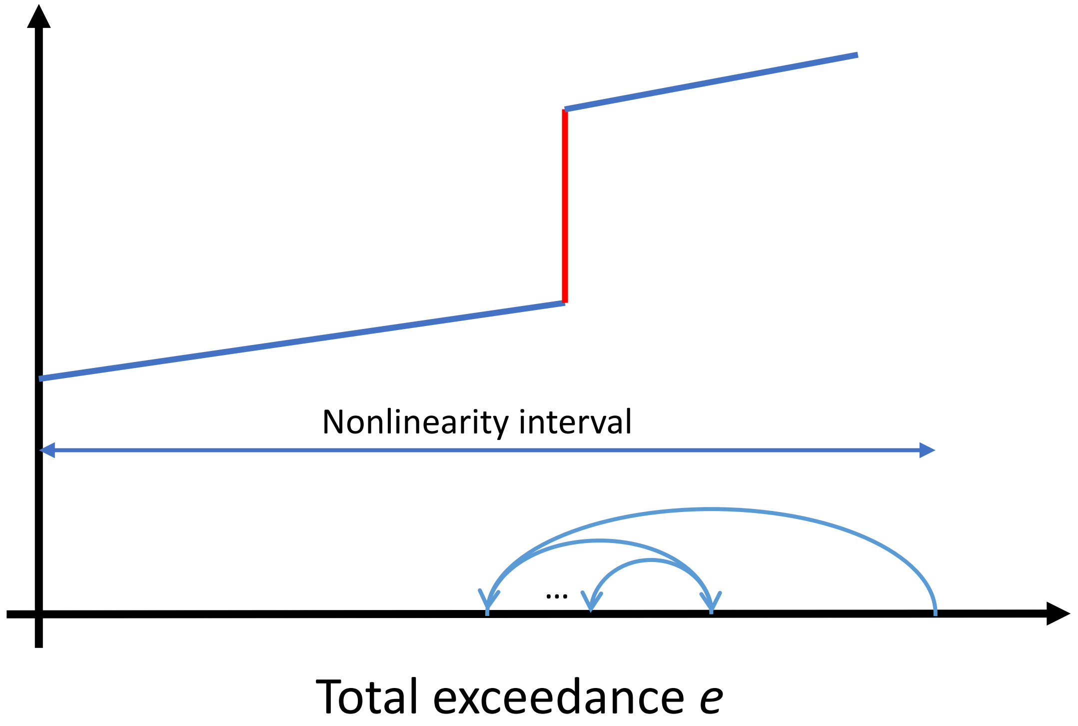
Search Algorithm

The search algorithm is based on:

1. An exponential search of the **nonlinearity interval**



2. A binary search on the nonlinearity interval



Who Generates the Exceedance?

Once we know the amount of exceedance that produces a nonlinearity, we still have **many scenarios** that can produce such an amount of exceedance:

Who Generates the Exceedance?

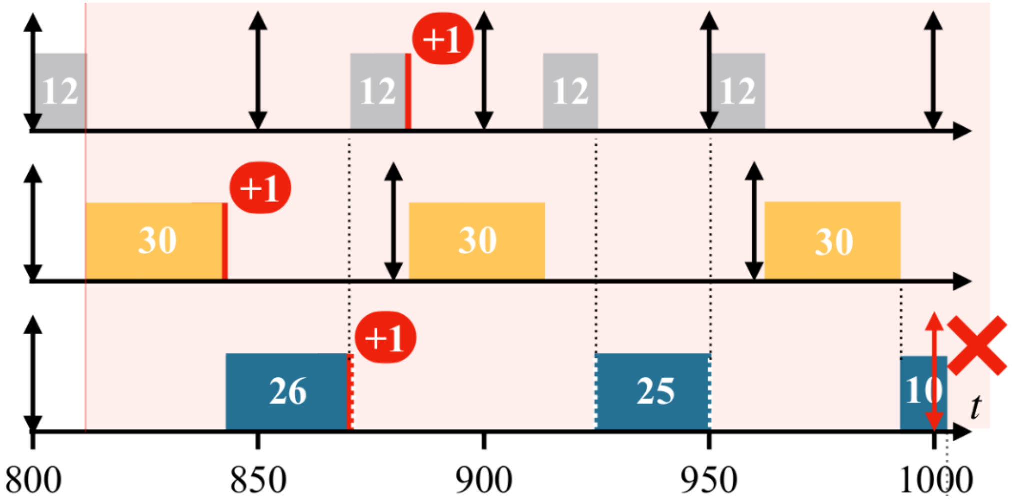
Once we know the amount of exceedance that produces a nonlinearity, we still have **many scenarios** that can produce such an amount of exceedance:

Exceedance:
3 time units

Who Generates the Exceedance?

Once we know the amount of exceedance that produces a nonlinearity, we still have **many scenarios** that can produce such an amount of exceedance:

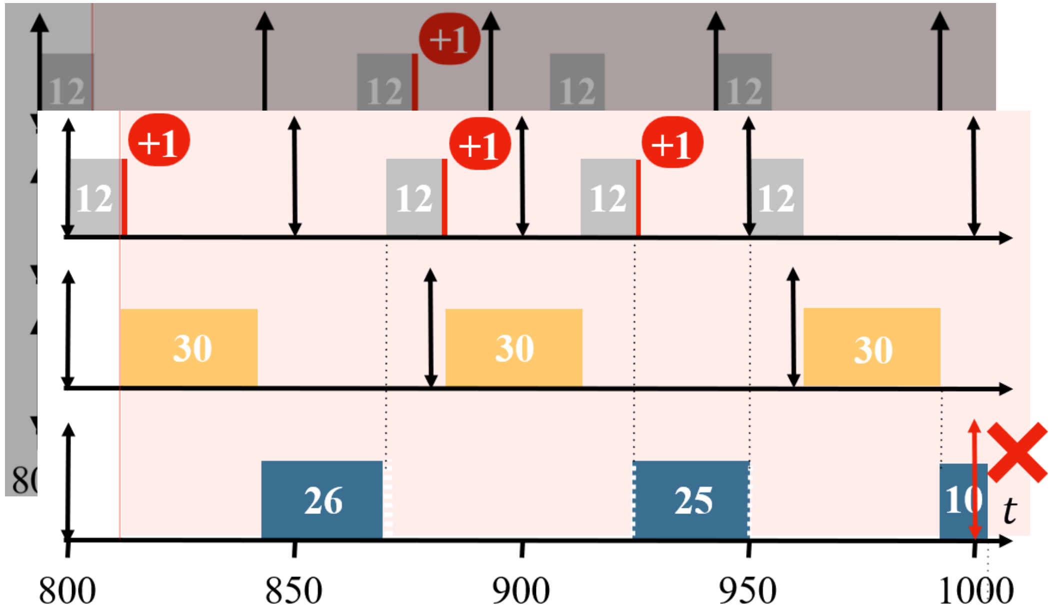
Exceedance: 3 time units



Who Generates the Exceedance?

Once we know the amount of exceedance that produces a nonlinearity, we still have **many scenarios** that can produce such an amount of exceedance:

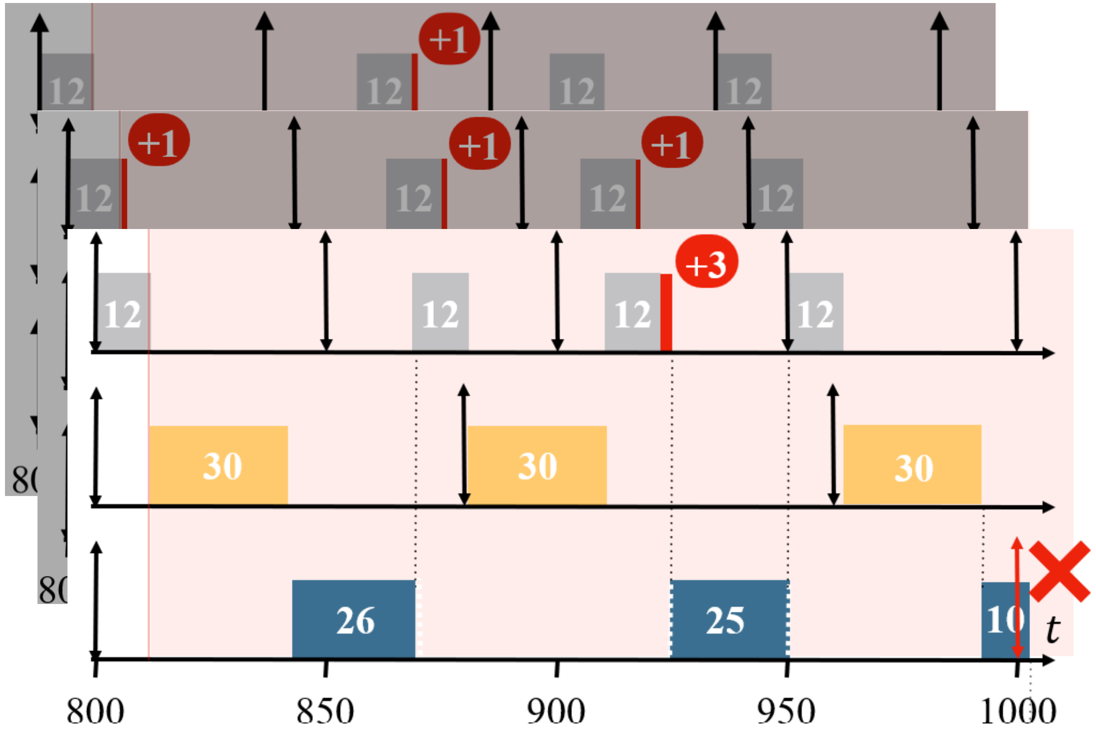
Exceedance: 3 time units



Who Generates the Exceedance?

Once we know the amount of exceedance that produces a nonlinearity, we still have **many scenarios** that can produce such an amount of exceedance:

Exceedance: 3 time units



Solution-Space Analysis

We need an instrument to **explore the space of interesting scenarios**

Solution-Space Analysis

We need an instrument to **explore the space of interesting scenarios**

QUESTION

What is an **interesting scenario**?

Real-Time Systems Laboratory

Solution-Space Analysis

We need an instrument to **explore the space of interesting scenarios**

QUESTION

What is an **interesting scenario**?

This may vary based on:

- The specific system
- The workload
- The designer’s preferences
- …

Solution Configurability

We developed a **configurable MILP-based tool** that produces execution traces that trigger the nonlinearities.

Solution Configurability

We developed a **configurable MILP-based tool** that produces execution traces that trigger the nonlinearities.

The tool has **three main tuning knobs**:

Solution Configurability

Real-Time Systems Laboratory

We developed a **configurable MILP-based tool** that produces execution traces that trigger the nonlinearities.

The tool has **three main tuning knobs**:

NET trustworthiness

Is the task recently developed or well-known?

Solution Configurability

We developed a **configurable MILP-based tool** that produces execution traces that trigger the nonlinearities.

The tool has **three main tuning knobs**:

NET trustworthiness

Is the task recently developed or well-known?

Min / Max exeedance per job

Support for budget enforcement or self-aborting jobs.
Exploration of specific scenarios.

Solution Configurability

We developed a **configurable MILP-based tool** that produces execution traces that trigger the nonlinearities.

The tool has **three main tuning knobs**:

NET trustworthiness

Is the task recently developed or well-known?

Min / Max exeedance per job

Support for budget enforcement or self-aborting jobs.
Exploration of specific scenarios.

Exeedance balancer

A few jobs exceed a lot
vs
many jobs exceed a little

Experimental Evaluation

The search algorithm and the exceedance-distribution tool were **evaluated** with a set of experiments:

Experimental Evaluation

The search algorithm and the exceedance-distribution tool were **evaluated** with a set of experiments:

- Evaluation on synthetic tasksets
  - Dirichlet-Rescale-based (DRS) workload
  - Automotive-based workload

Real-Time Systems Laboratory

Experimental Evaluation

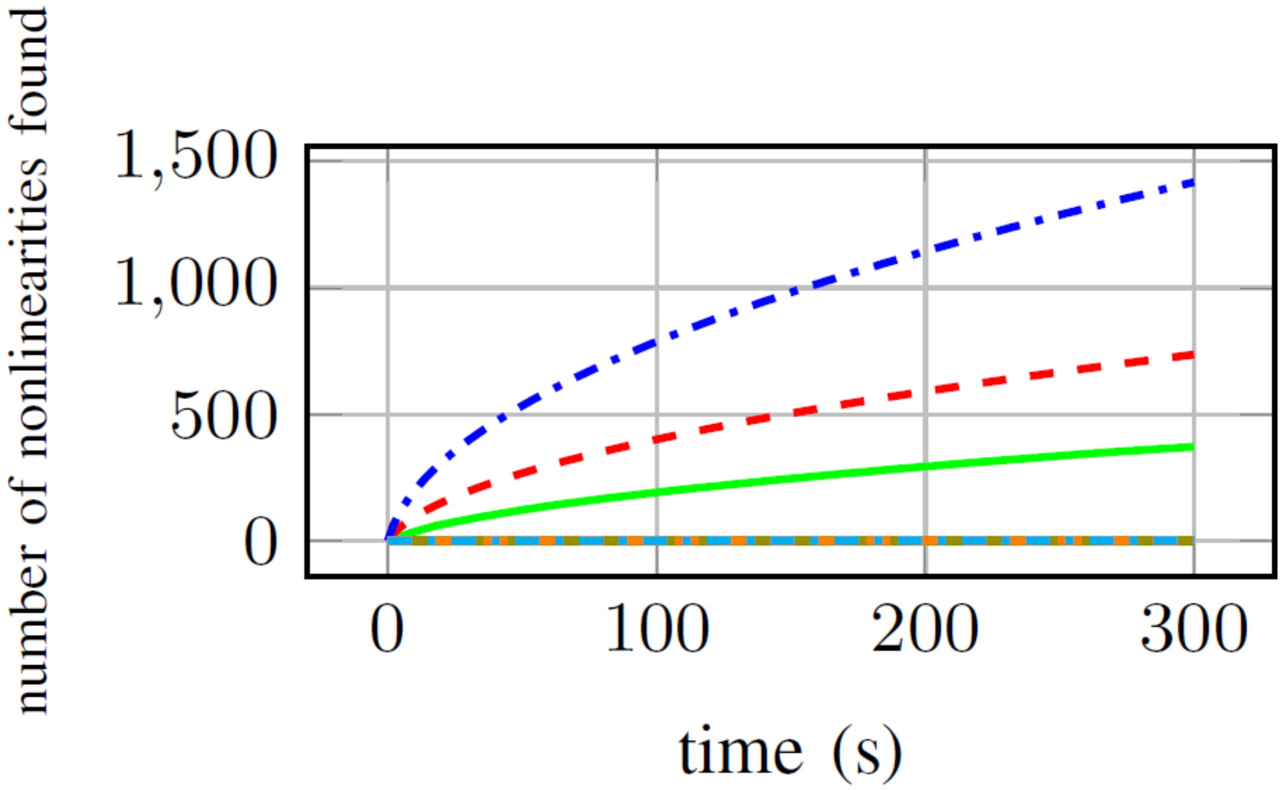
The search algorithm and the exceedance-distribution tool were **evaluated** with a set of experiments:

- Evaluation on synthetic tasksets
  - Dirichlet-Rescale-based (DRS) workload
  - Automotive-based workload
- Case study
  - WATERS 2017 industrial challenge

Search Performance

The **nonlinearities search algorithm** was compared with a brute-force solution

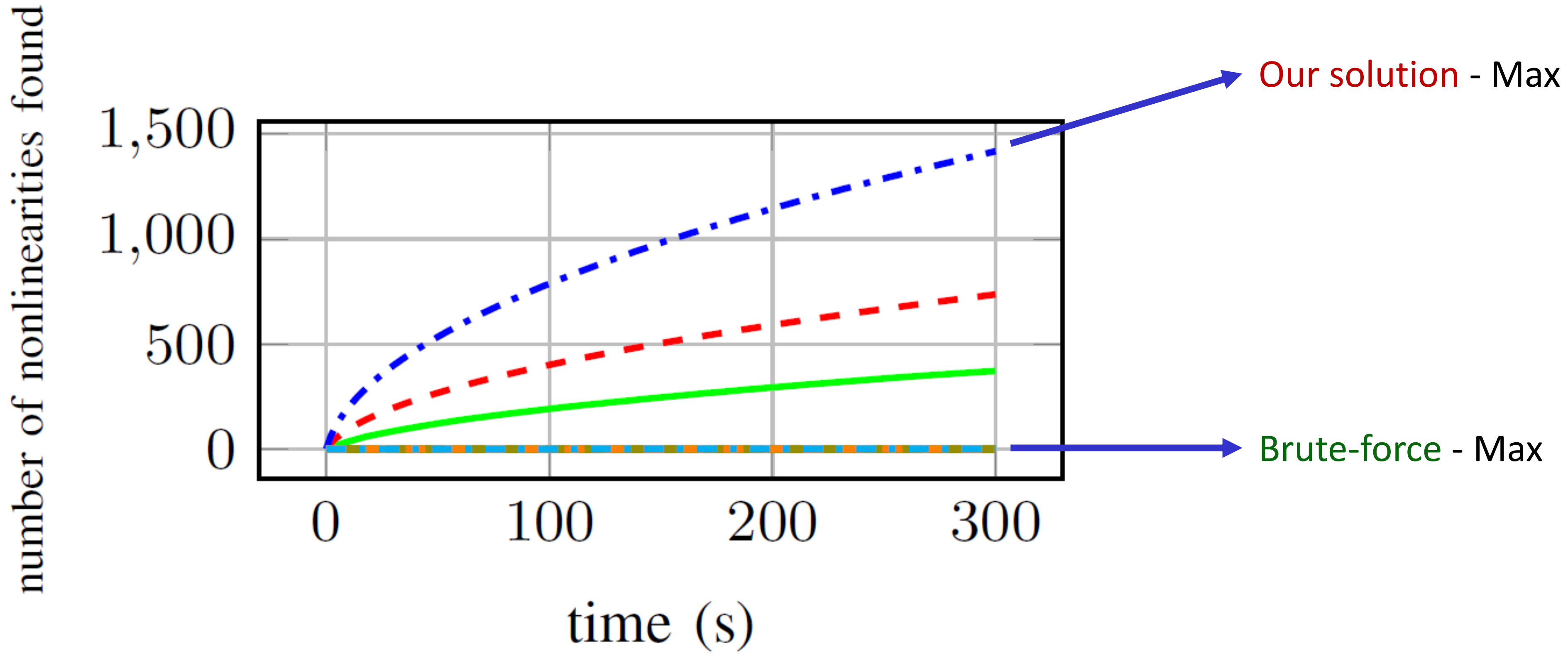
DRS-based, fully non-preemtpive workload – 3600 tasksets



Search Performance

The **nonlinearities search algorithm** was compared with a brute-force solution

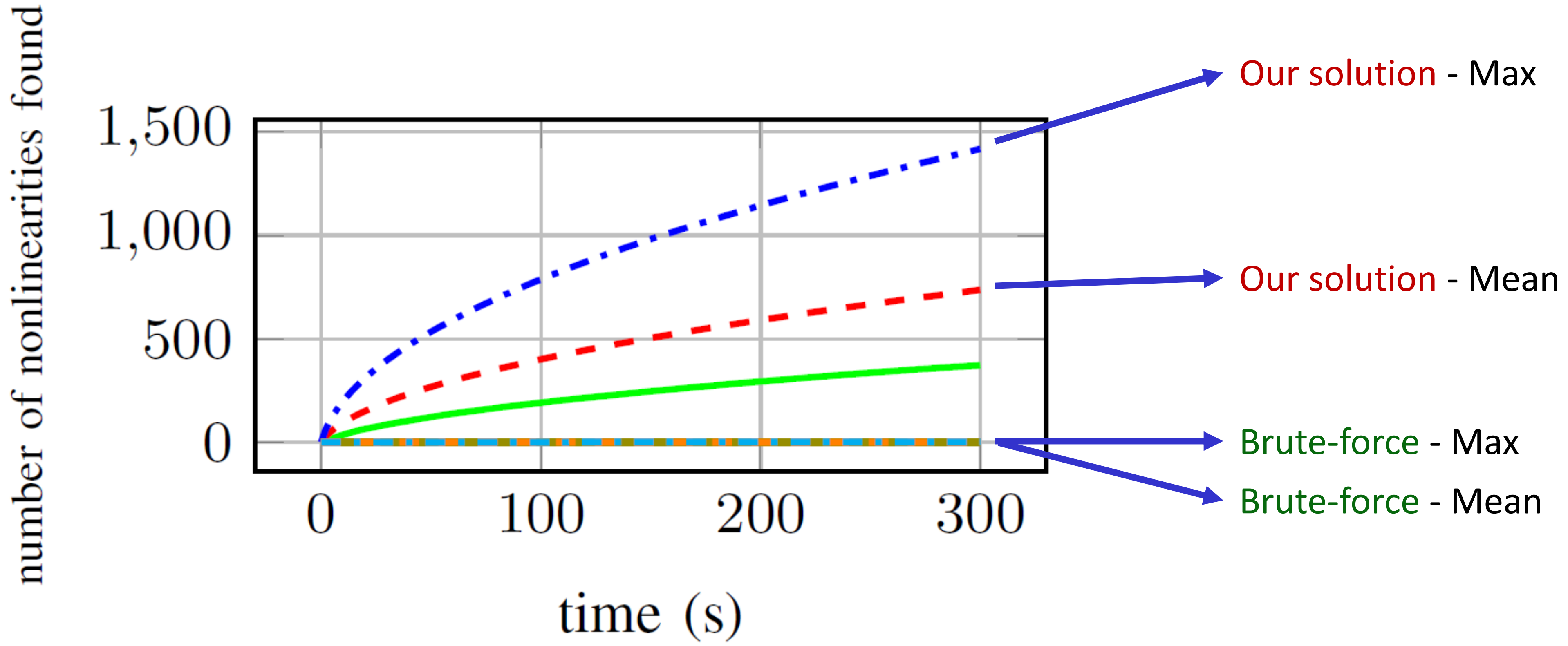
DRS-based, fully non-preemtpive workload – 3600 tasksets



Search Performance

The **nonlinearities search algorithm** was compared with a brute-force solution

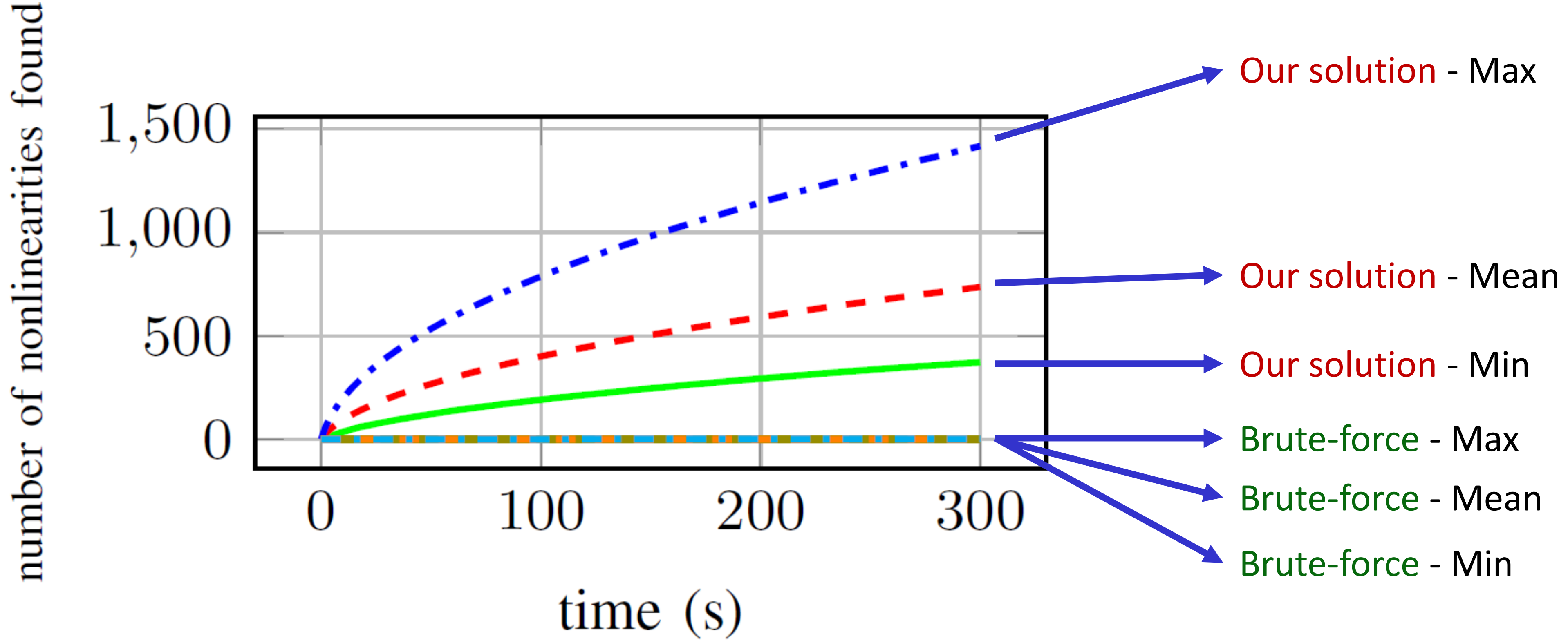
DRS-based, fully non-preemtpive workload – 3600 tasksets



Search Performance

The **nonlinearities search algorithm** was compared with a brute-force solution

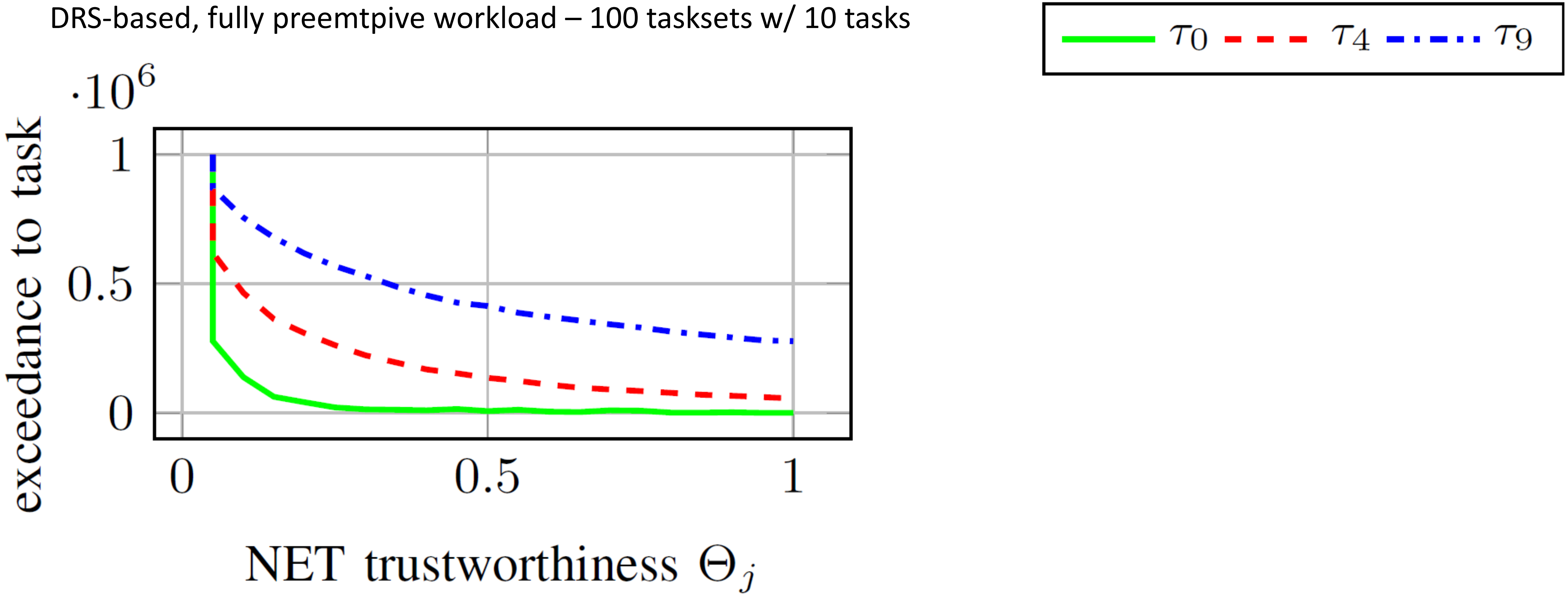
DRS-based, fully non-preemtpive workload – 3600 tasksets



NET Trustworthiness

The **NET trustworthiness** parameter affects the amount of exceedance assigned to a task

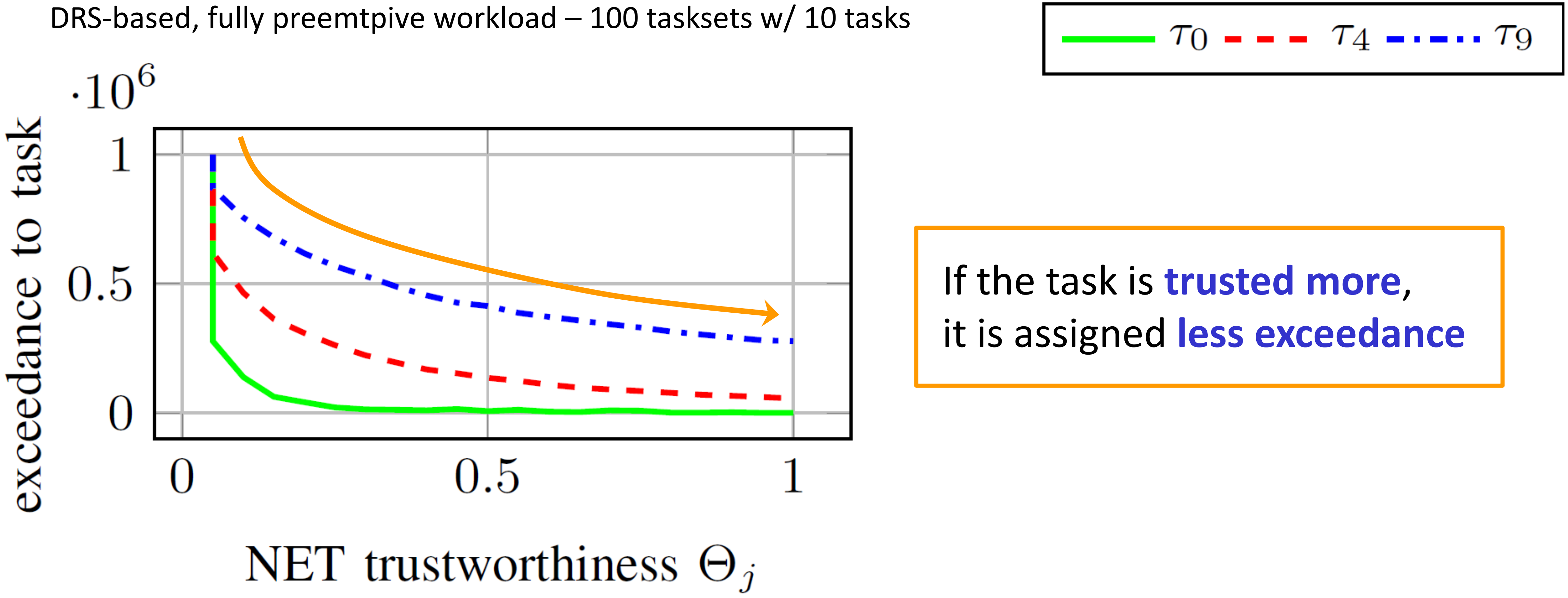
DRS-based, fully preemtpive workload – 100 tasksets w/ 10 tasks



NET Trustworthiness

The **NET trustworthiness** parameter affects the amount of exceedance assigned to a task

DRS-based, fully preemtpive workload – 100 tasksets w/ 10 tasks

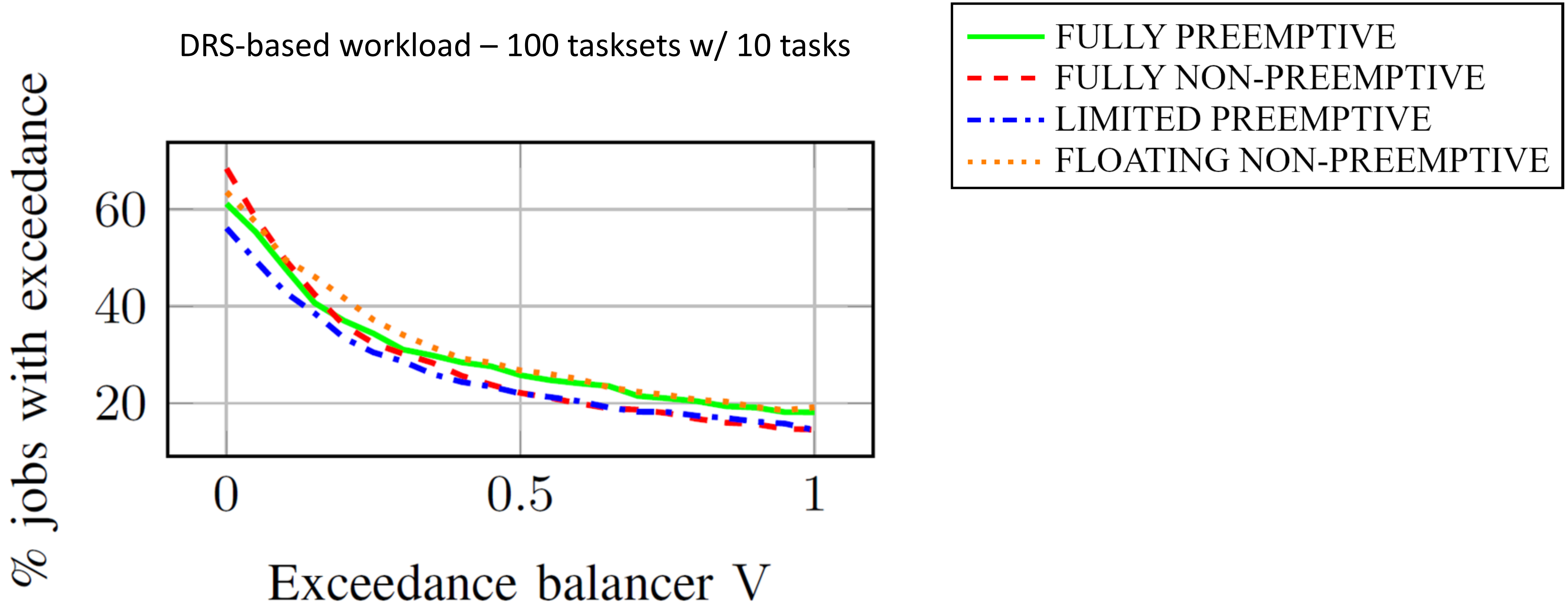


If the task is **trusted more**,
it is assigned **less exceedance**

Balancer Effects

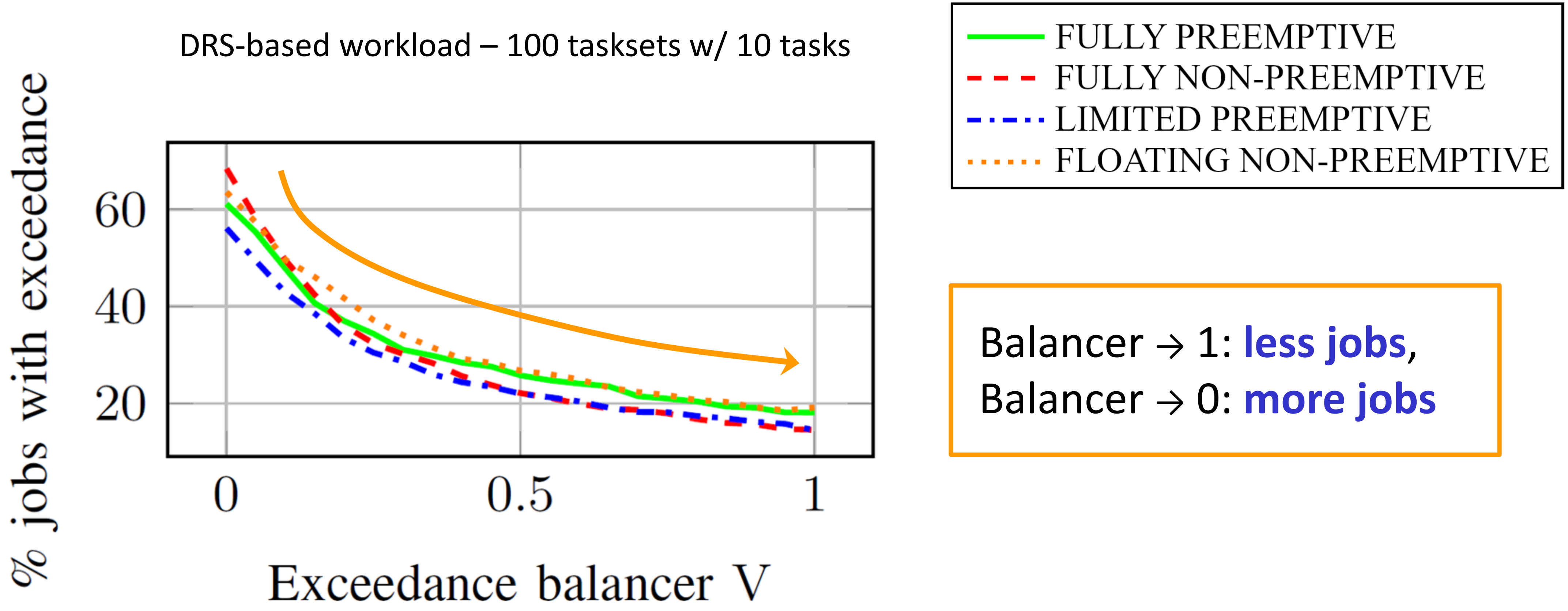
The **exceedance balancer** parameter affects the number of jobs that are assigned exceedance

DRS-based workload – 100 tasksets w/ 10 tasks



Balancer Effects

The **exceedance balancer** parameter affects the number of jobs that are assigned exceedance



Balancer → 1: **less jobs**,
Balancer → 0: **more jobs**

Case Study

The optimization problem can be **easily extended** to consider additional metrics.

Case Study

The optimization problem can be **easily extended** to consider additional metrics.

Case study: WATERS 2017 challenge - NET = 0.9 · WCET

NOTE:
Original taskset not schedulable with WCETs!

Case Study

The optimization problem can be **easily extended** to consider additional metrics.

Case study: WATERS 2017 challenge - NET = 0.9 · WCET

Slowdown = exceedance / NET

NOTE:
Original taskset not schedulable with WCETs!

Case Study

Retis Real-Time Systems Laboratory

The optimization problem can be **easily extended** to consider additional metrics.

Case study: WATERS 2017 challenge - NET = 0.9 · WCET

NOTE:
Original taskset not schedulable with WCETs!

Slowdown = exceedance / NET

The MILP was modified to look for the slowdown that makes the taskset unschedulable.

Task	Period	NET	e	$\min\max x_h^s$	$L_7(e)$
τ_1	2 ms	0.364 ms	1.636 ms	450.06%	97.59 ms
τ_2	5 ms	0.838 ms	3.071 ms	159.24%	99.39 ms
τ_3	20 ms	9.421 ms	3.591 ms	21.89%	99.91 ms
τ_4	50 ms	2.776 ms	14.407 ms	16.99%	399.06 ms
τ_5	100 ms	8.476 ms	3.929 ms	4.09%	179.91 ms
τ_6	200 ms	0.124 ms	7.733 ms	4.02%	279.91 ms
τ_7	1000 ms	0.123 ms	38.542 ms	4.01%	1079.91 ms

Case Study

Real-Time Systems Laboratory

The optimization problem can be **easily extended** to consider additional metrics.

Case study: WATERS 2017 challenge - NET = 0.9 · WCET

NOTE:
Original taskset not schedulable with WCETs!

Slowdown = exceedance / NET

The MILP was modified to look for the slowdown that makes the taskset unschedulable.

Exceedance to deadline miss

Task	Period	NET	e	$\min\max x_h^s$	$L_7(e)$
τ_1	2 ms	0.364 ms	1.636 ms	450.06%	97.59 ms
τ_2	5 ms	0.838 ms	3.071 ms	159.24%	99.39 ms
τ_3	20 ms	9.421 ms	3.591 ms	21.89%	99.91 ms
τ_4	50 ms	2.776 ms	14.407 ms	16.99%	399.06 ms
τ_5	100 ms	8.476 ms	3.929 ms	4.09%	179.91 ms
τ_6	200 ms	0.124 ms	7.733 ms	4.02%	279.91 ms
τ_7	1000 ms	0.123 ms	38.542 ms	4.01%	1079.91 ms

Real-Time Systems Laboratory

Case Study

The optimization problem can be **easily extended** to consider additional metrics.

Case study: WATERS 2017 challenge - NET = 0.9 · WCET

NOTE:
Original taskset not schedulable with WCETs!

Slowdown = exceedance / NET

The MILP was modified to look for the slowdown that makes the taskset unschedulable.

Exceedance to deadline miss

Slowdown necessary for a deadline miss to occur

Task	Period	NET	e	$\min \max x_h^s$	$L_7(e)$
τ_1	2 ms	0.364 ms	1.636 ms	450.06%	97.59 ms
τ_2	5 ms	0.838 ms	3.071 ms	159.24%	99.39 ms
τ_3	20 ms	9.421 ms	3.591 ms	21.89%	99.91 ms
τ_4	50 ms	2.776 ms	14.407 ms	16.99%	399.06 ms
τ_5	100 ms	8.476 ms	3.929 ms	4.09%	179.91 ms
τ_6	200 ms	0.124 ms	7.733 ms	4.02%	279.91 ms
τ_7	1000 ms	0.123 ms	38.542 ms	4.01%	1079.91 ms

Case Study

The optimization problem can be **easily extended** to consider additional metrics.

Case study: WATERS 2017 challenge - NET = 0.9 · WCET

NOTE:
Original taskset not schedulable with WCETs!

Slowdown = exceedance / NET

The MILP was modified to look for the slowdown that makes the taskset unschedulable.

Exceedance to deadline miss — Slowdown necessary for a deadline miss to occur — Bound on the recovery time

Task	Period	NET	e	$\min \max x_h^s$	$L_7(e)$
τ_1	2 ms	0.364 ms	1.636 ms	450.06%	97.59 ms
τ_2	5 ms	0.838 ms	3.071 ms	159.24%	99.39 ms
τ_3	20 ms	9.421 ms	3.591 ms	21.89%	99.91 ms
τ_4	50 ms	2.776 ms	14.407 ms	16.99%	399.06 ms
τ_5	100 ms	8.476 ms	3.929 ms	4.09%	179.91 ms
τ_6	200 ms	0.124 ms	7.733 ms	4.02%	279.91 ms
τ_7	1000 ms	0.123 ms	38.542 ms	4.01%	1079.91 ms

Case Study

The optimization problem can be **easily extended** to consider additional metrics.

Case study: WATERS 2017 challenge - NET = 0.9 · WCET

NOTE:
Original taskset not schedulable with WCETs!

Slowdown = exceedance / NET

The MILP was modified to look for the slowdown that makes the taskset unschedulable.

Exceedance to deadline miss

Slowdown necessary for a deadline miss to occur

Bound on the recovery time

Task	Period	NET	e	$\min \max x_h^s$	$L_7(e)$
τ_1	2 ms	0.364 ms	1.636 ms	450.06%	97.59 ms
τ_2	5 ms	0.838 ms	3.071 ms	159.24%	99.39 ms
τ_3	20 ms	9.421 ms	3.591 ms	21.89%	99.91 ms
τ_4	50 ms	2.776 ms	14.407 ms	16.99%	399.06 ms
τ_5	100 ms	8.476 ms	3.929 ms	4.09%	179.91 ms
τ_6	200 ms	0.124 ms	7.733 ms	4.02%	279.91 ms
τ_7	1000 ms	0.123 ms	38.542 ms	4.01%	1079.91 ms

High-priority tasks have a big safety margin

Case Study

The optimization problem can be **easily extended** to consider additional metrics.

Case study: WATERS 2017 challenge - NET = 0.9 · WCET

NOTE:
Original taskset not schedulable with WCETs!

Slowdown = exceedance / NET

The MILP was modified to look for the slowdown that makes the taskset unschedulable.

Exceedance to deadline miss

Slowdown necessary for a deadline miss to occur

Bound on the recovery time

Task	Period	NET	e	$\min \max x_h^s$	$L_7(e)$
τ_1	2 ms	0.364 ms	1.636 ms	450.06%	97.59 ms
τ_2	5 ms	0.838 ms	3.071 ms	159.24%	99.39 ms
τ_3	20 ms	9.421 ms	3.591 ms	21.89%	99.91 ms
τ_4	50 ms	2.776 ms	14.407 ms	16.99%	399.06 ms
τ_5	100 ms	8.476 ms	3.929 ms	4.09%	179.91 ms
τ_6	200 ms	0.124 ms	7.733 ms	4.02%	279.91 ms
τ_7	1000 ms	0.123 ms	38.542 ms	4.01%	1079.91 ms

High-priority tasks have a big safety margin

Low-priority tasks have a very small safety margin!

Conclusions

Conclusions

- When using NETs, exceedance effects must be analyzed carefully due to the presence of **hard-to-predict nonlinearities**.

Conclusions

- When using NETs, exceedance effects must be analyzed carefully due to the presence of **hard-to-predict nonlinearities**.

- Exceedance analysis helps to **efficiently explore** the space of exceedance effects in order to:
  - Explore hypothetical scenarios;
  - Assess the system’s safety margin w.r.t. transient execution-time fluctuations.

Conclusions

- When using NETs, exceedance effects must be analyzed carefully due to the presence of **hard-to-predict nonlinearities**.

- Exceedance analysis helps to **efficiently explore** the space of exceedance effects in order to:
  - Explore hypothetical scenarios;
  - Assess the system's safety margin w.r.t. transient execution-time fluctuations.

- The results of the analysis can be presented visually and **easily understood** by the system's designer.

Real-Time Systems Laboratory

Future work

Future work

- The approach can be extended to **other scheduling policies**
  - Support for locking protocols
  - Self-suspending tasks
  - …

Future work

- The approach can be extended to **other scheduling policies**
  - Support for locking protocols
  - Self-suspending tasks
  - …

- The approach can be extended to **other task parameters:**
  - Release jitter
  - Critical-section length
  - Supply-bound functions
  - …

Thank you!