On Strong and Weak Sustainability,
with an Application to Self-Suspending Real-Time Tasks

Felipe Cerqueira, Geoffrey Nelissen, Björn Brandenburg

ECRTS'18, Barcelona
July 6, 2018

MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

CISTER
Research Center in
Real-Time & Embedded
Computing Systems

MAX-PLANCK-GESELLSCHAFT

formally proven
schedulability analysis | PROSA

http://prosa.mpi-sws.org

Partially funded by Deutsche Forschungsgemeinschaft (DFG) — 391919384

This Paper in a Nutshell

This Paper in a Nutshell

Identify an ambiguity in the definition of sustainability,
a central notion in real-time scheduling theory and practice

This Paper in a Nutshell

**Identify an ambiguity in the definition of sustainability,
a central notion in real-time scheduling theory and practice**

Precise, formal theory that covers all types of sustainability

This Paper in a Nutshell

Identify an ambiguity in the definition of sustainability, a central notion in real-time scheduling theory and practice

Precise, formal theory that covers all types of sustainability

New notion of weak sustainability that enables more efficient analysis for policies that are not sustainable

This Paper in a Nutshell

Identify an ambiguity in the definition of sustainability, a central notion in real-time scheduling theory and practice

Precise, formal theory that covers all types of sustainability

New notion of weak sustainability that enables more efficient analysis for policies that are not sustainable

Uniprocessor JLFP scheduling of self-suspending tasks is weakly weakly sustainable w.r.t. execution times and variable suspension times

formally proven schedulability analysis | PROSA

This Talk

This Talk

Ambiguity in the Current Definitions

This Talk

Ambiguity in the Current Definitions

Strong and Weak Sustainability

This Talk

Ambiguity in the Current Definitions

Strong and Weak Sustainability

Weak Sustainability of Self-Suspending Tasks

This Talk

Ambiguity in the Current Definitions

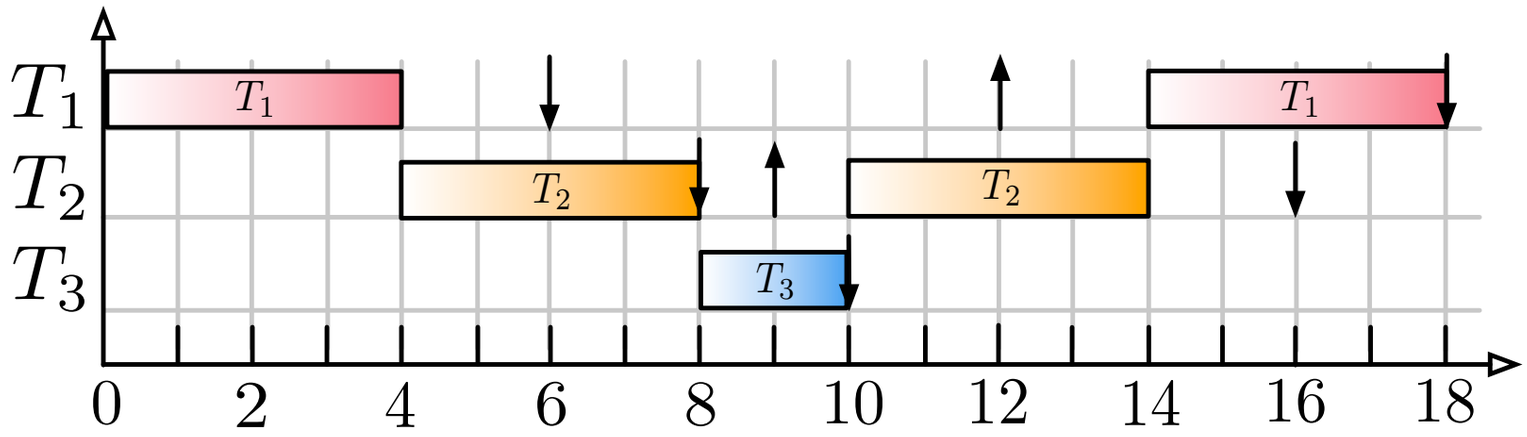
Strong and Weak Sustainability

Weak Sustainability of Self-Suspending Tasks

What really is sustainability?

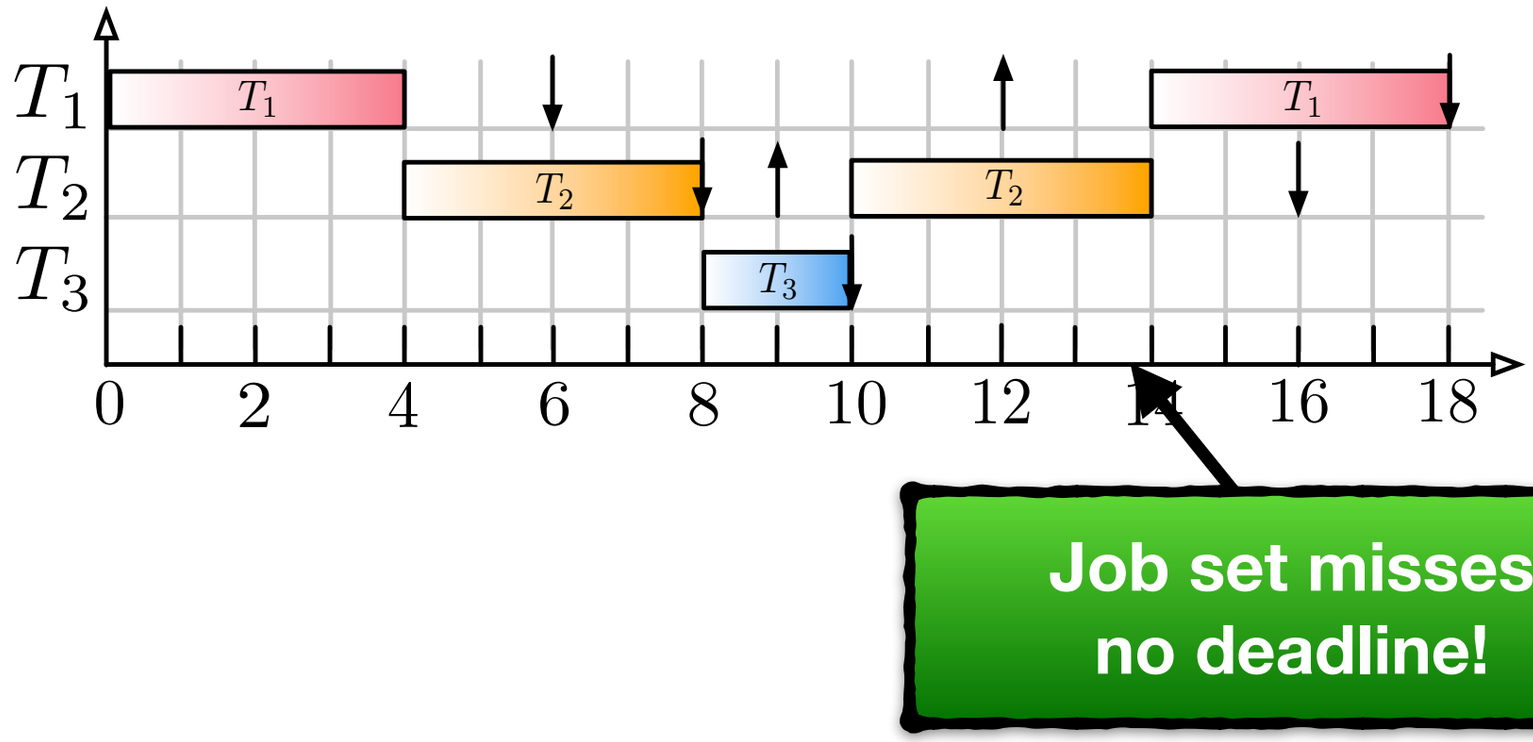
Sustainability of Sporadic Tasks

Original Job Set



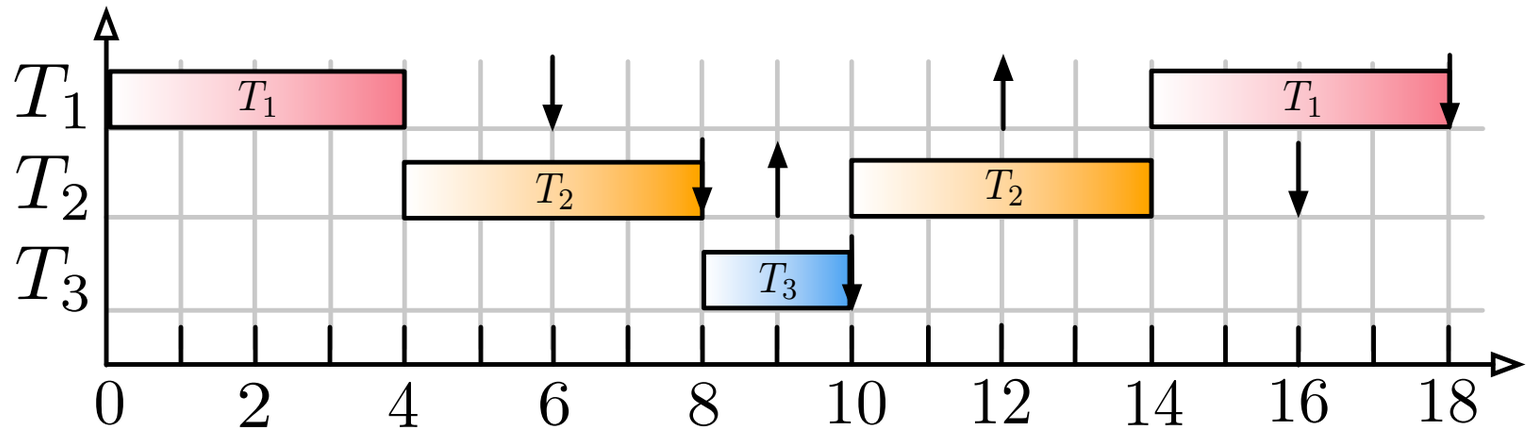
Sustainability of Sporadic Tasks

Original Job Set

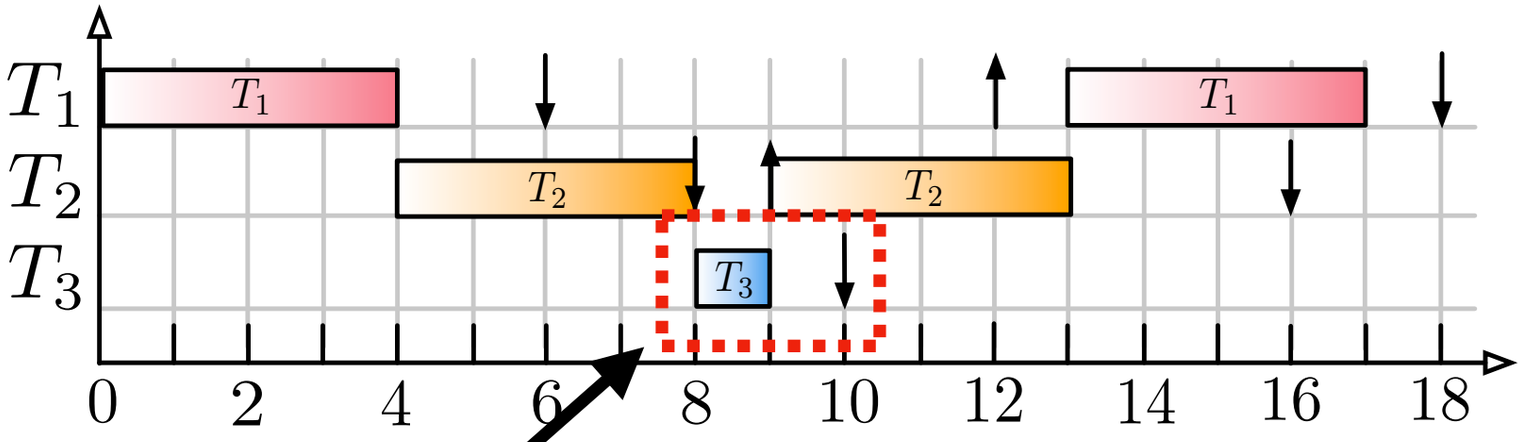


Sustainability of Sporadic Tasks

Original Job Set



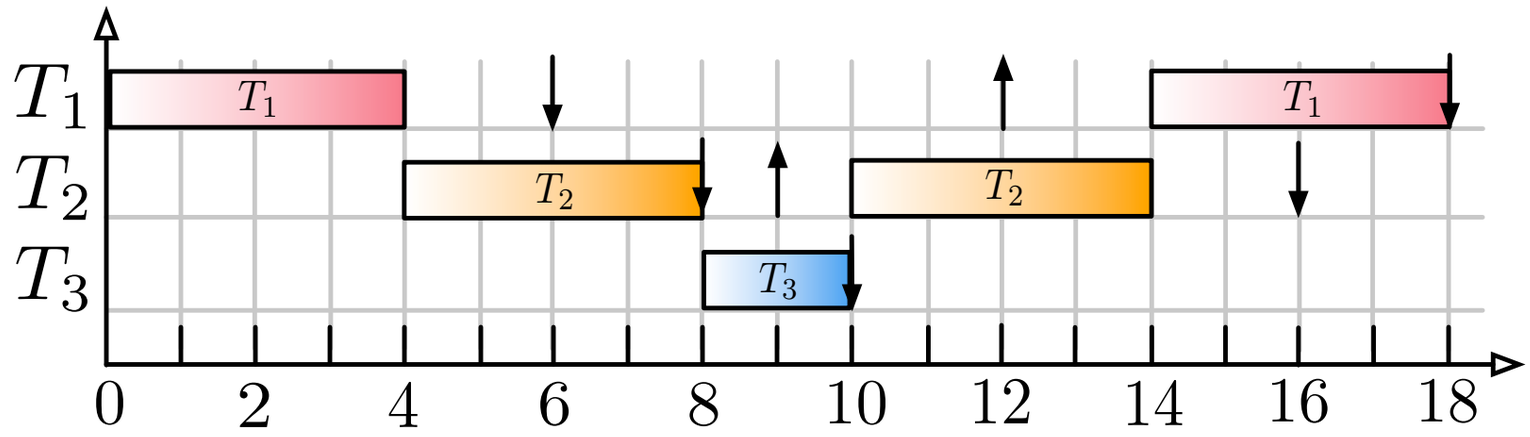
Reducing cost of T_3



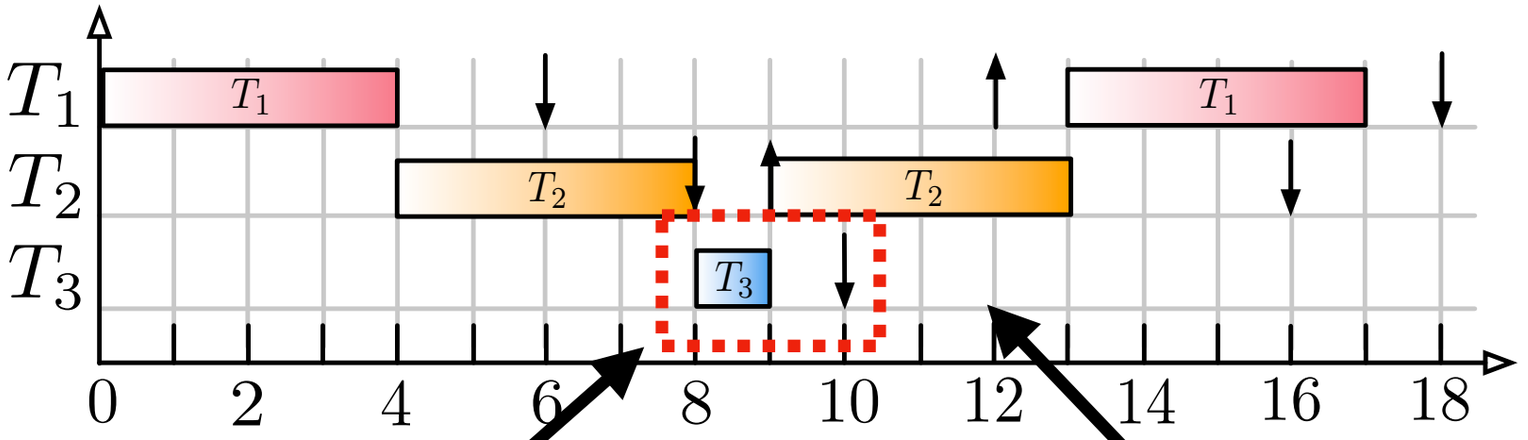
Assume T_3 completes one unit earlier

Sustainability of Sporadic Tasks

Original Job Set



Reducing cost of T_3

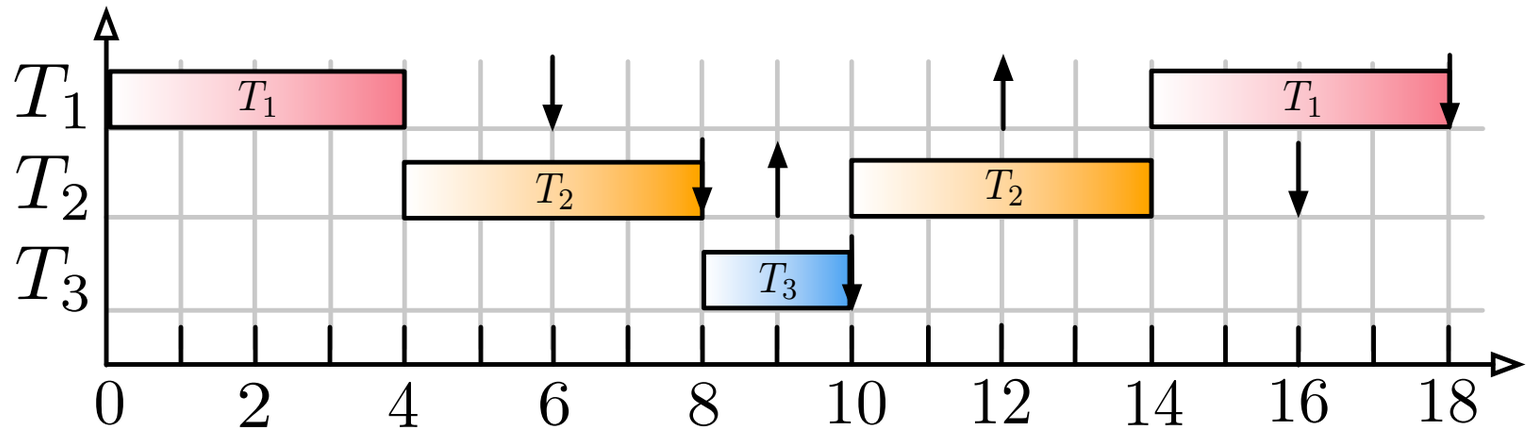


Assume T_3 completes one unit earlier

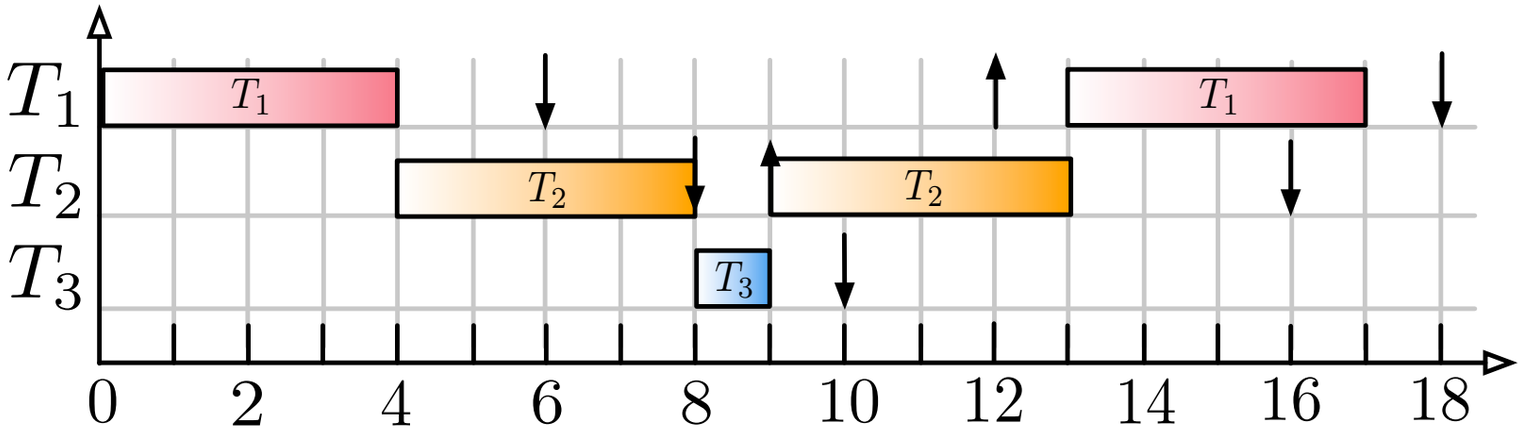
Job set misses no deadline!

Sustainable w.r.t. Execution Times

Original Job Set



Reducing cost of T_3



Uniprocessor scheduling of sporadic tasks is sustainable with respect to execution times.

Alan Burns and Sanjoy K. Baruah. Sustainability in real-time scheduling. Journal of Computing Science and Engineering, 2(1):74–97, 2008.

Definition of Sustainable Policy

[Burns and Baruah, 2008]

Definition 1 *A scheduling policy and/or a schedulability test for a scheduling policy is* sustainable *if any system deemed schedulable by the schedulability test remains schedulable when the parameters of one or more individual tasks[s] are changed in any, some, or all of the following ways*: (i) *decreased execution requirements*; (ii) *larger periods*; (iii) *smaller jitter*; and (iv) *larger relative deadlines.*

Alan Burns and Sanjoy K. Baruah. Sustainability in real-time scheduling.
Journal of Computing Science and Engineering, 2(1):74–97, 2008.

Definition of Sustainable Policy

[Burns and Baruah, 2008]

Definition 1 *A scheduling policy and/or a schedulability test for a scheduling policy is* sustainable *if any system deemed schedulable by the schedulability test remains schedulable when the parameters of one or more individual tasks[s] are changed in any, some, or all of the following ways*: (i) *decreased execution requirements*; (ii) *larger periods*; (iii) *smaller jitter*; and (iv) *larger relative deadlines.*

If a system is **schedulable** with **original parameters,** then it **remains schedulable** when assigned **better parameters.**

e.g., at runtime

Alan Burns and Sanjoy K. Baruah. Sustainability in real-time scheduling.
Journal of Computing Science and Engineering, 2(1):74–97, 2008.

Why Does Sustainability Matter?

Why Does Sustainability Matter?

Analysis	**Minimize/maximize parameters** to **reduce the search space.**

Why Does Sustainability Matter?

Analysis	**Minimize/maximize parameters** to **reduce the search space.**

Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment

C. L. LIU
Project MAC, Massachusetts Institute of Technology
AND
JAMES W. LAYLAND
Jet Propulsion Laboratory, California Institute of Technology

ABSTRACT. The problem of multiprogram scheduling on a single processor is studied from the viewpoint of the characteristics peculiar to the program functions that need guaranteed service. It is shown that an optimum, fixed priority scheduler possesses an upper bound to processor utilization which may be as low as 70 percent for large task sets. A more general scheme is described, in which priorities are assigned dynamically according to the proximity of their current deadlines. A combination of these two scheduling techniques is also discussed.

KEY WORDS AND PHRASES: real-time multiprogramming, scheduling, multiprogram scheduling, dynamic scheduling, priority assignment, processor utilization, deadline driven scheduling

CR CATEGORIES: 3.80, 3.82, 3.83, 4.22

1. *Introduction*

The use of computers for control and monitoring of industrial processes has expanded greatly in recent years, and will probably expand even more dramatically in the near future. Often, the computer used in such an application is shared between a certain number of time-critical control and monitor functions and a non-time-critical batch processing job stream. In other installations, however, no non-time-critical jobs exist, and efficient use of the computer can only be achieved by a careful scheduling of the time-critical control and monitor functions themselves. This latter group might be termed "pure process control" and provides the background for the combinatoric scheduling analyses presented in this paper. Two

This paper presents the results of one phase of research carried out at the Jet Propulsion Laboratory, California Institute of Technology, under Contract No. NAS7-100, sponsored by the National Aeronautics and Space Administration.

Authors' present addresses: C. L. Liu, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801; James W. Layland, Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91103.

Journal of the Association for Computing Machinery, Vol. 20, No. 1, January 1973, pp. 46–61.

Liu and Layland in 1973 already
assumed maximum execution times

(A4) Run-time for each task is constant for that task and does not vary with time. Run-time here refers to the time which is taken by a processor to execute the task without interruption.

Chung Laung Liu and James W Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. Journal of the ACM, 20(1):46–61, 1973.

Why Does Sustainability Matter?

Analysis	**Minimize/maximize parameters** to **reduce the search space.**

Why Does Sustainability Matter?

Analysis	**Minimize/maximize parameters** to **reduce the search space.**
Design	Engineers can **assume worst-case parameters** for **design space exploration.**

Why Does Sustainability Matter?

Analysis	**Minimize/maximize parameters** to **reduce the search space.**
Design	Engineers can **assume worst-case parameters** for **design space exploration.**
Deployment	Ensure **system safety at runtime** under non-worst-case conditions.

Definition of Sustainability is Ambiguous

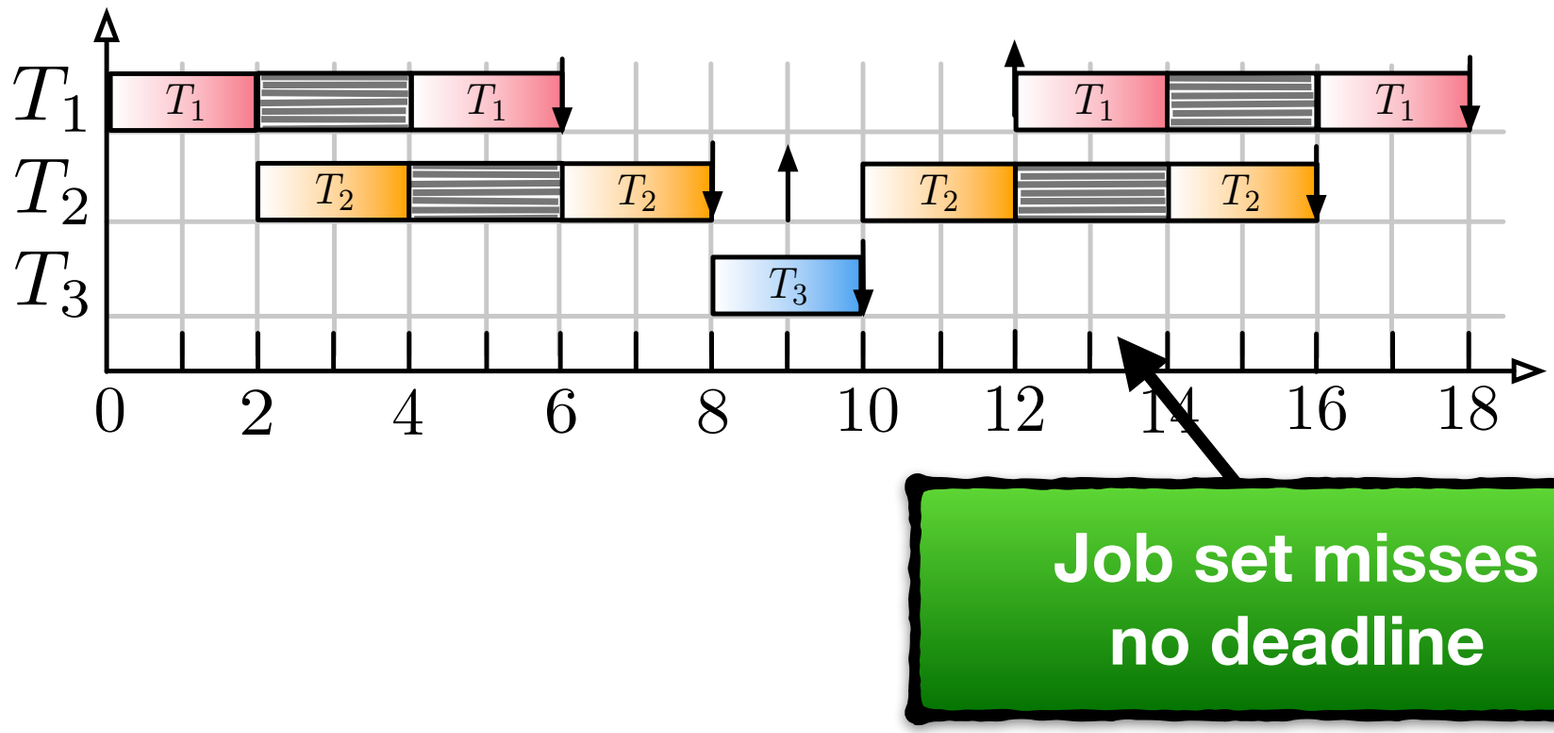
EDF Scheduling of Self-Suspending Tasks is Not Sustainable w.r.t. Execution Times

[Example by Abdeddaïm and Masson, 2012]

EDF Scheduling of Self-Suspending Tasks is Not Sustainable w.r.t. Execution Times

[Example by Abdeddaïm and Masson, 2012]

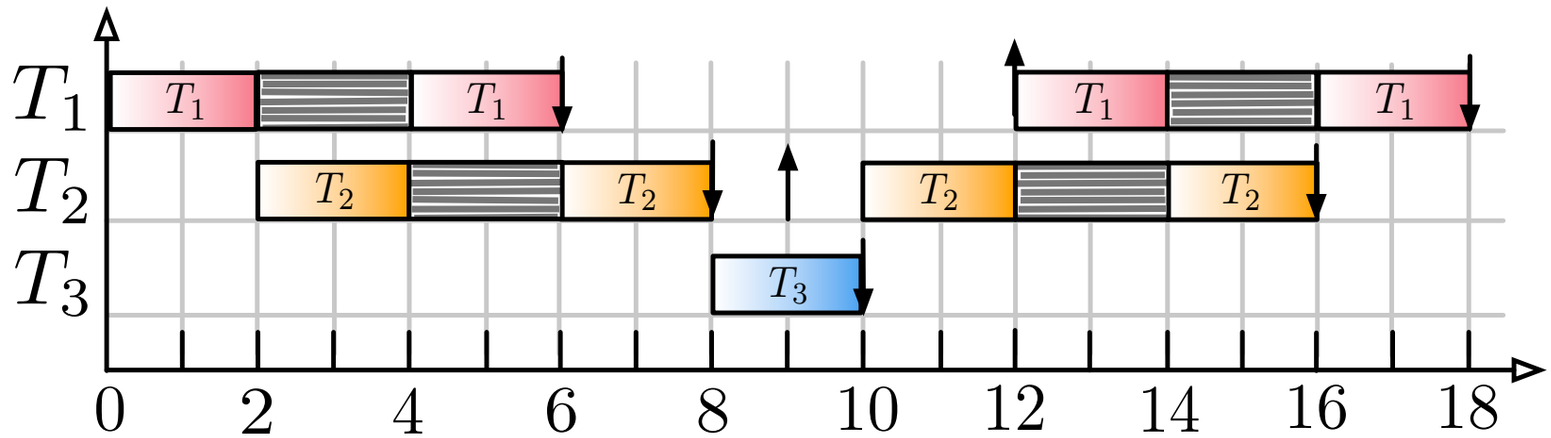
Original Job Set



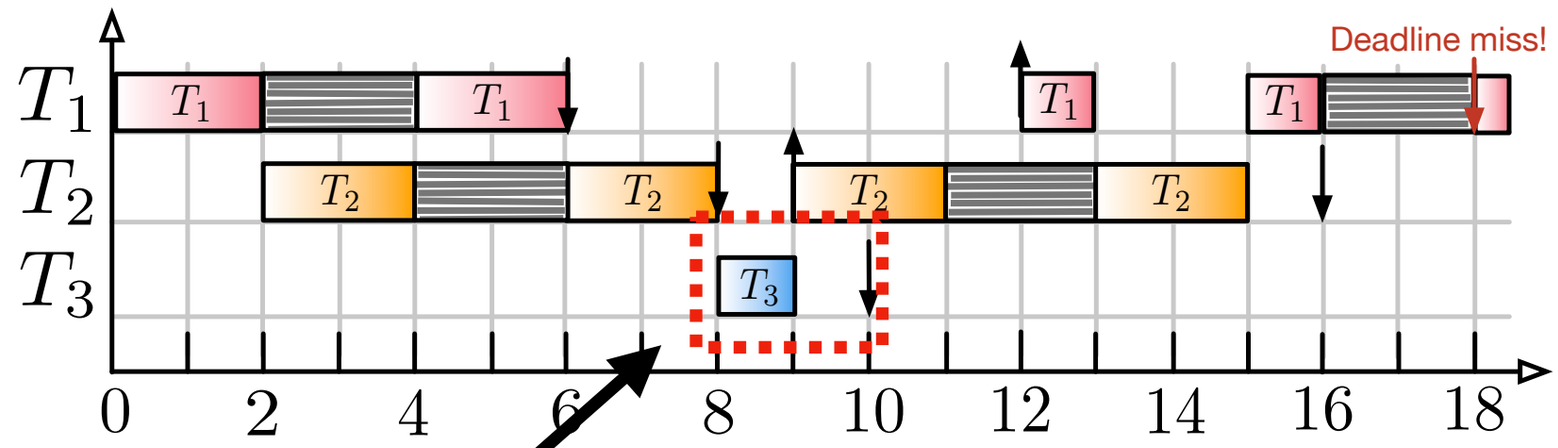
EDF Scheduling of Self-Suspending Tasks is Not Sustainable w.r.t. Execution Times

[Example by Abdeddaïm and Masson, 2012]

Original Job Set



Reducing cost of T_3

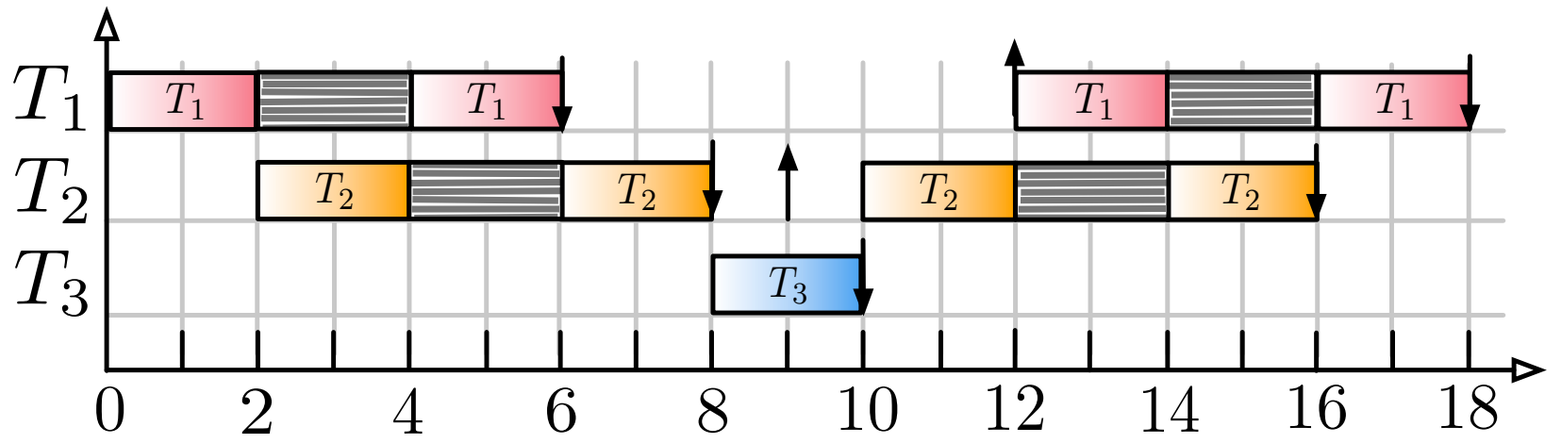


Assume T_3 completes one unit earlier

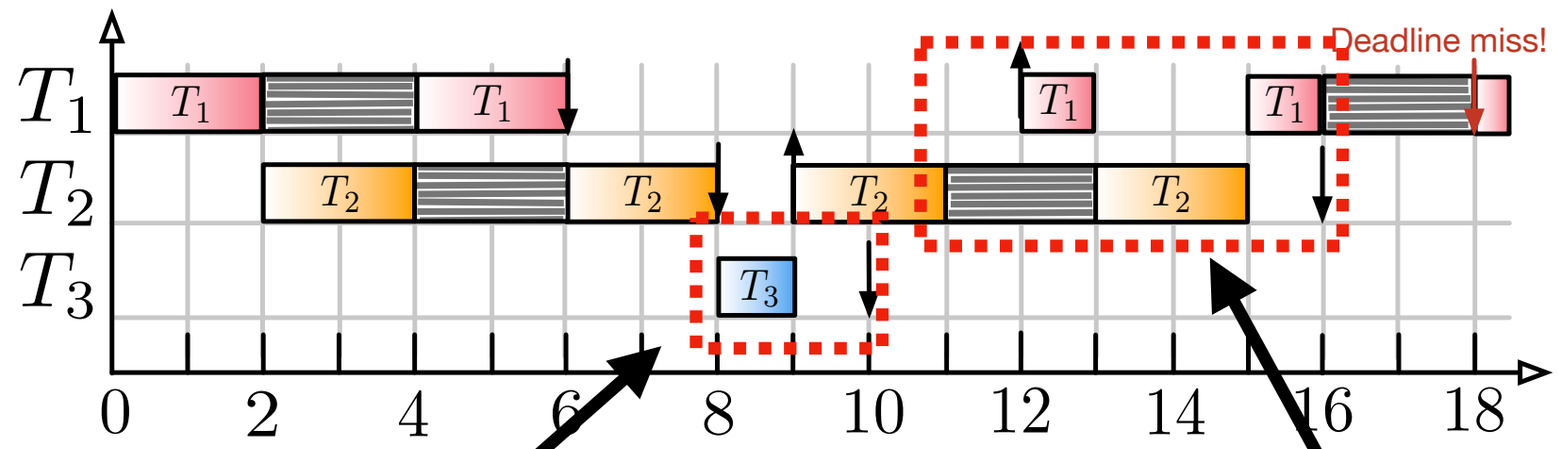
EDF Scheduling of Self-Suspending Tasks is Not Sustainable w.r.t. Execution Times

[Example by Abdeddaïm and Masson, 2012]

Original Job Set



Reducing cost of T_3



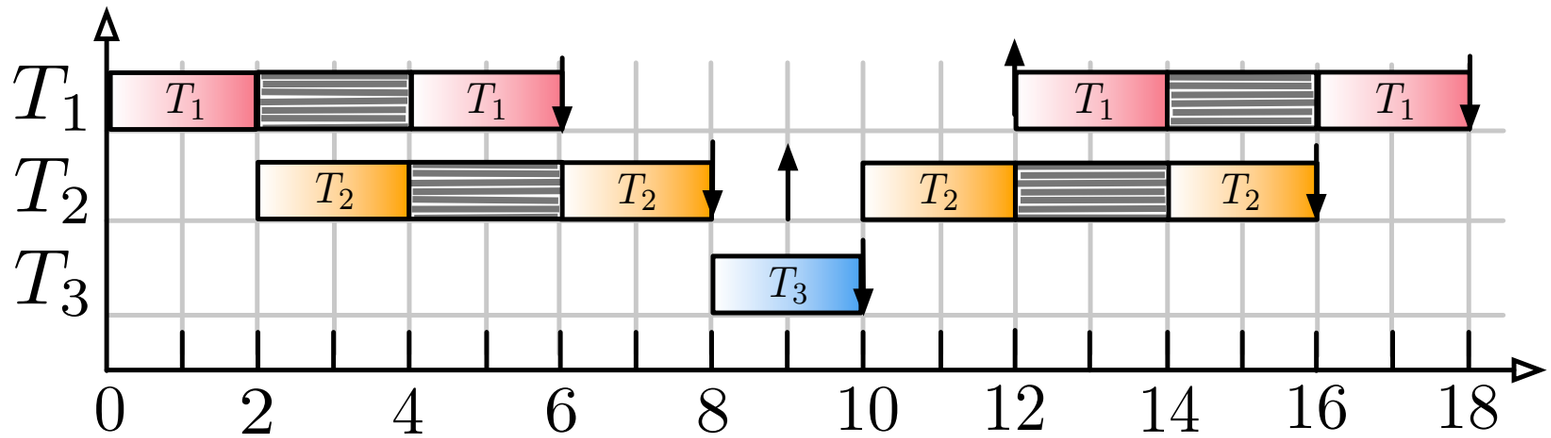
Assume T_3 completes one unit earlier

T_2 suspends earlier and interferes more with T_1

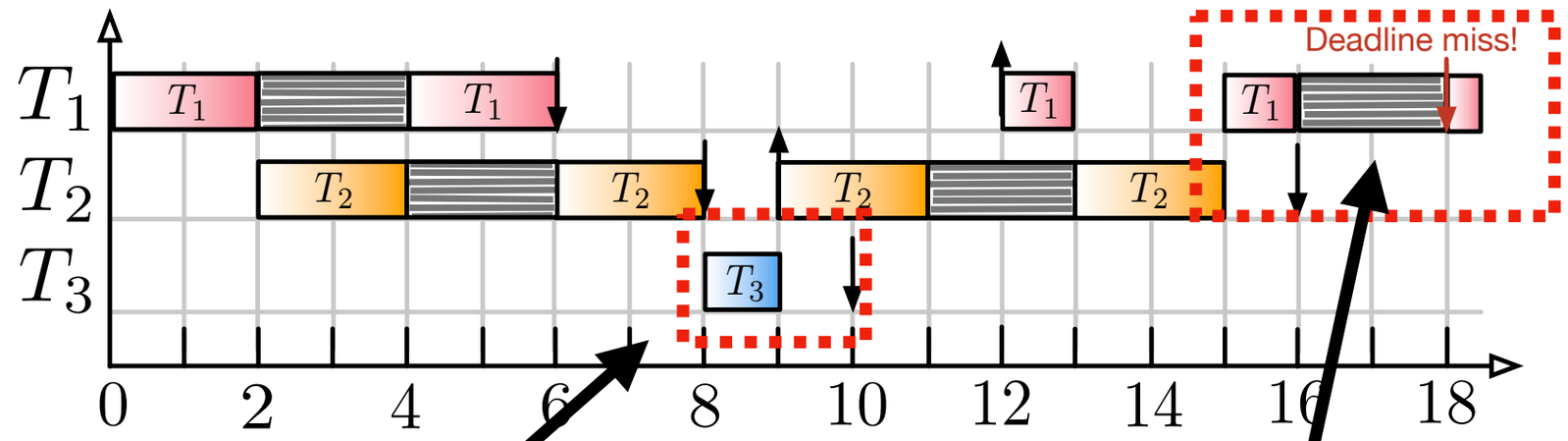
EDF Scheduling of Self-Suspending Tasks is Not Sustainable w.r.t. Execution Times

[Example by Abdeddaïm and Masson, 2012]

Original Job Set



Reducing cost of T_3

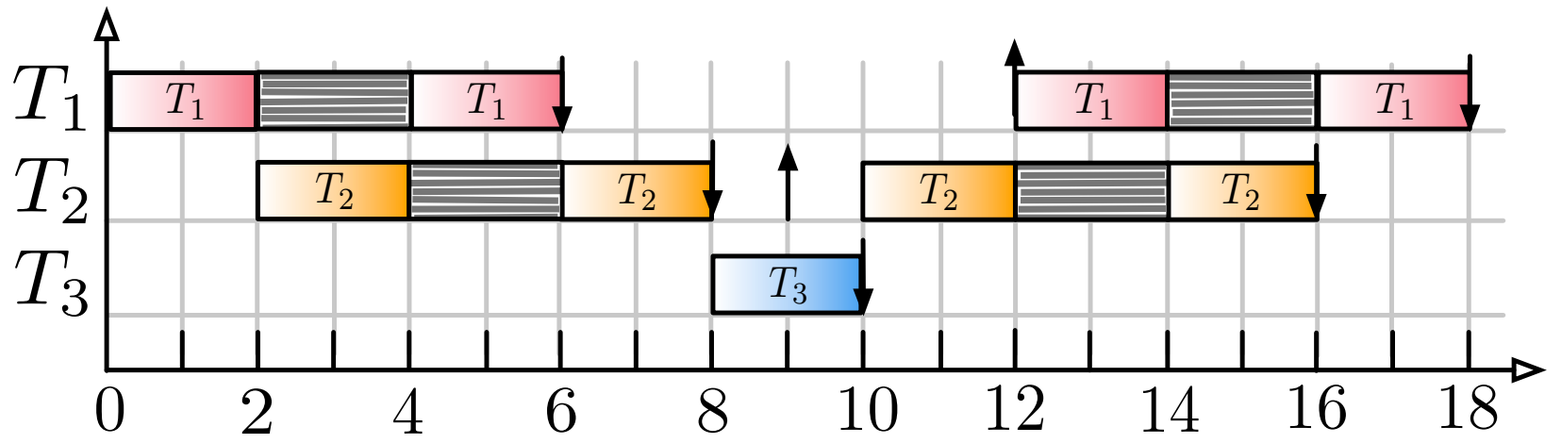


Assume T_3 completes one unit earlier

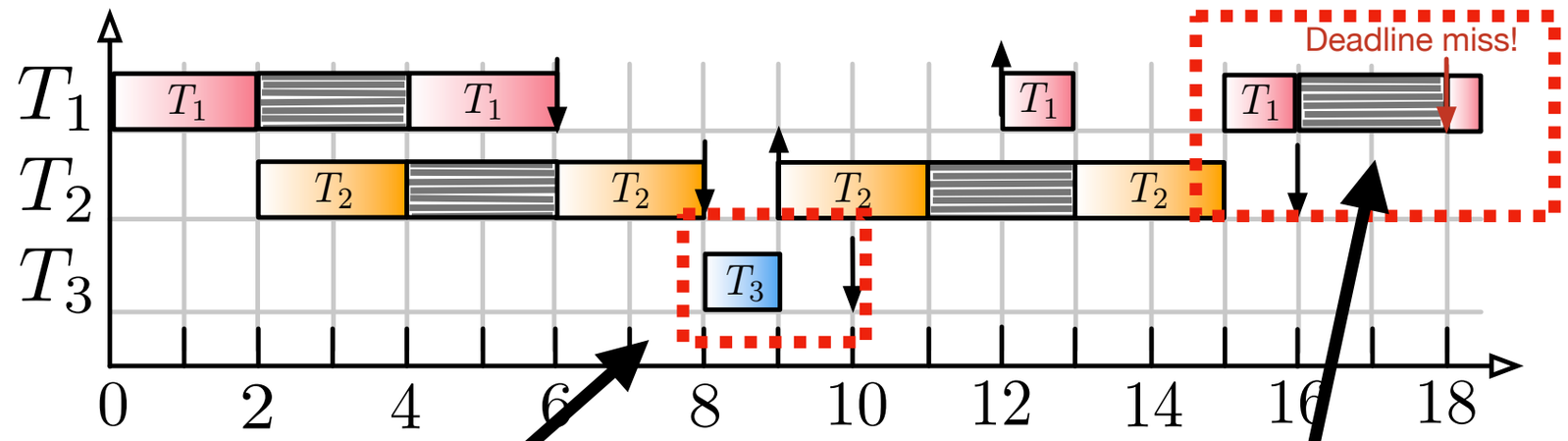
Interference causes deadline miss for T_1

Abdeddaïm and Masson: EDF scheduling of self-suspending tasks is not sustainable w.r.t. execution times.

Original Job Set



Reducing cost of T_3



Assume T_3 completes one unit earlier

Interference causes deadline miss for T_1

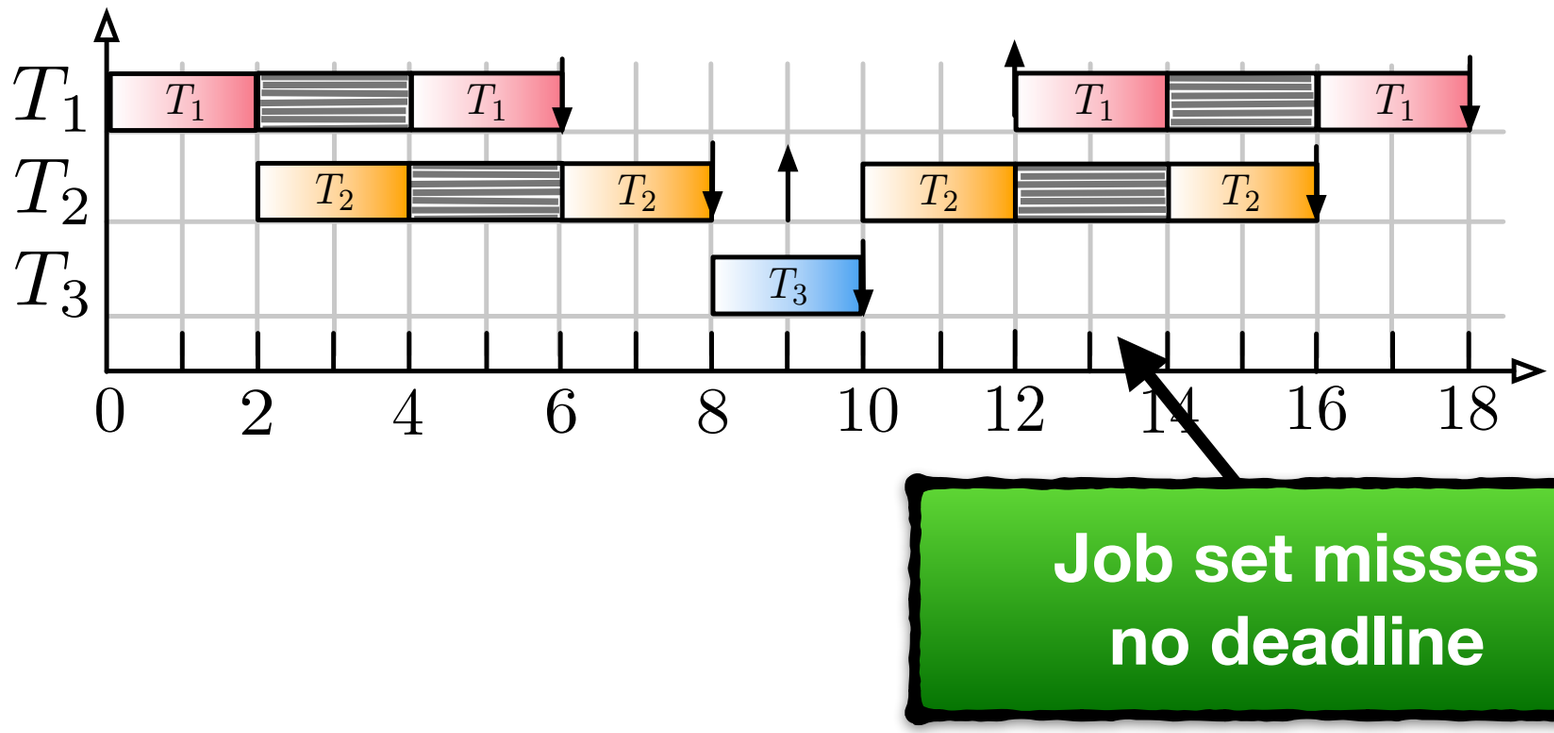
What About Suspension Times?

[Example by Abdeddaïm and Masson, 2012]

What About Suspension Times?

[Example by Abdeddaïm and Masson, 2012]

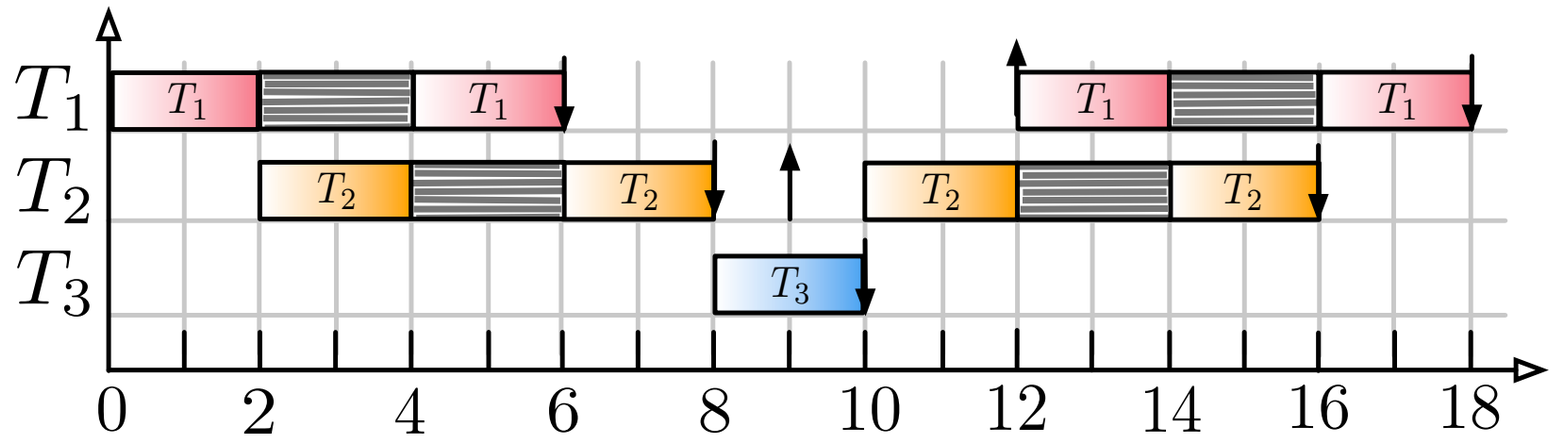
Original Job Set



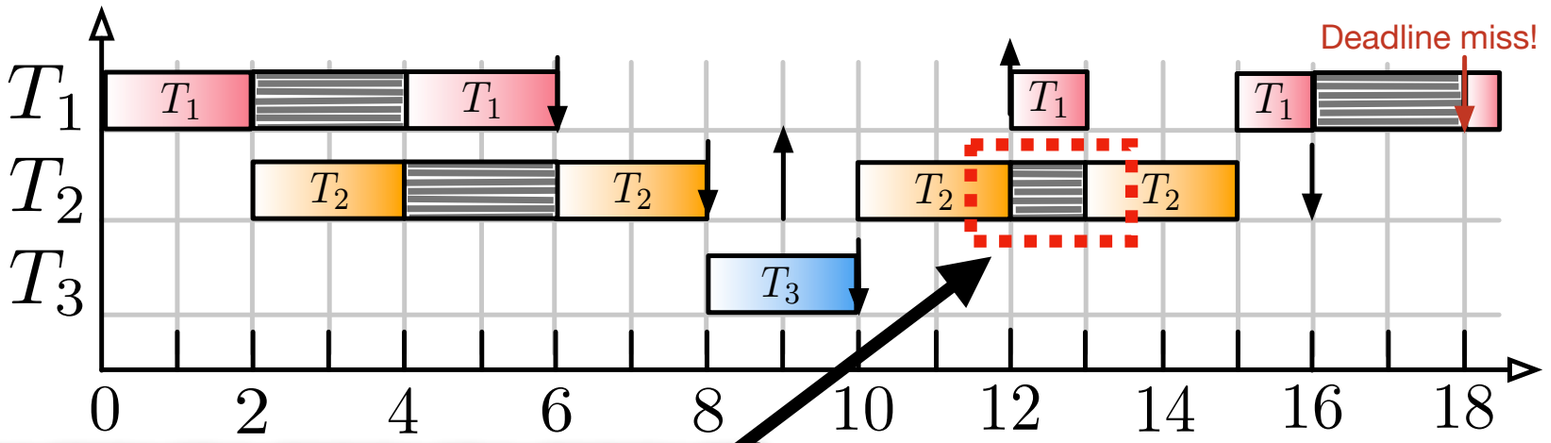
What About Suspension Times?

[Example by Abdeddaïm and Masson, 2012]

Original Job Set



Reducing Suspension Time of T_2

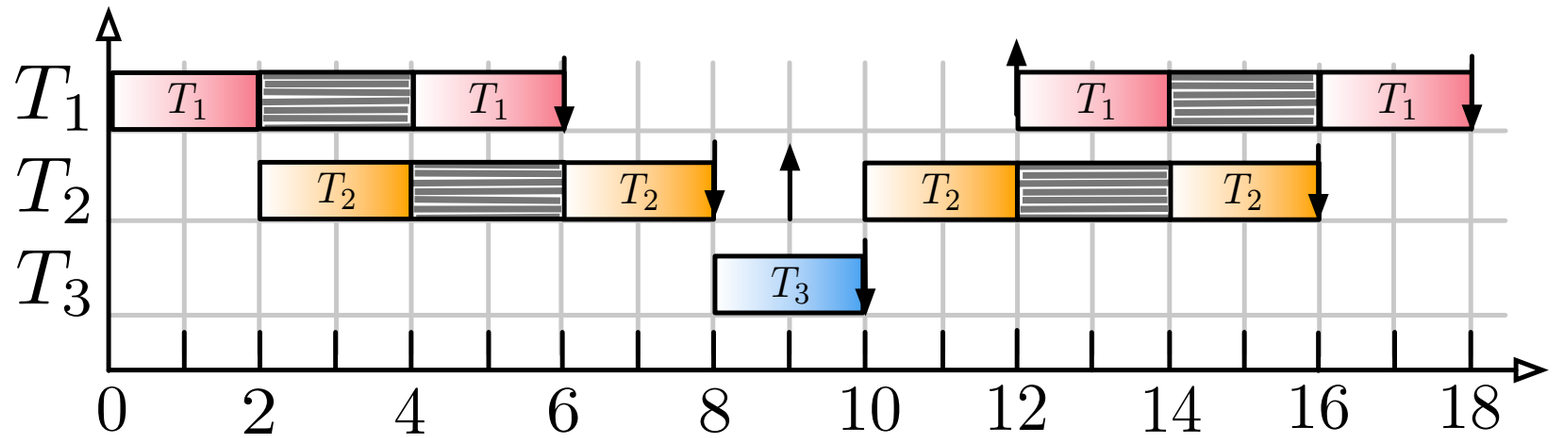


Assume T_2 suspends for only one time unit

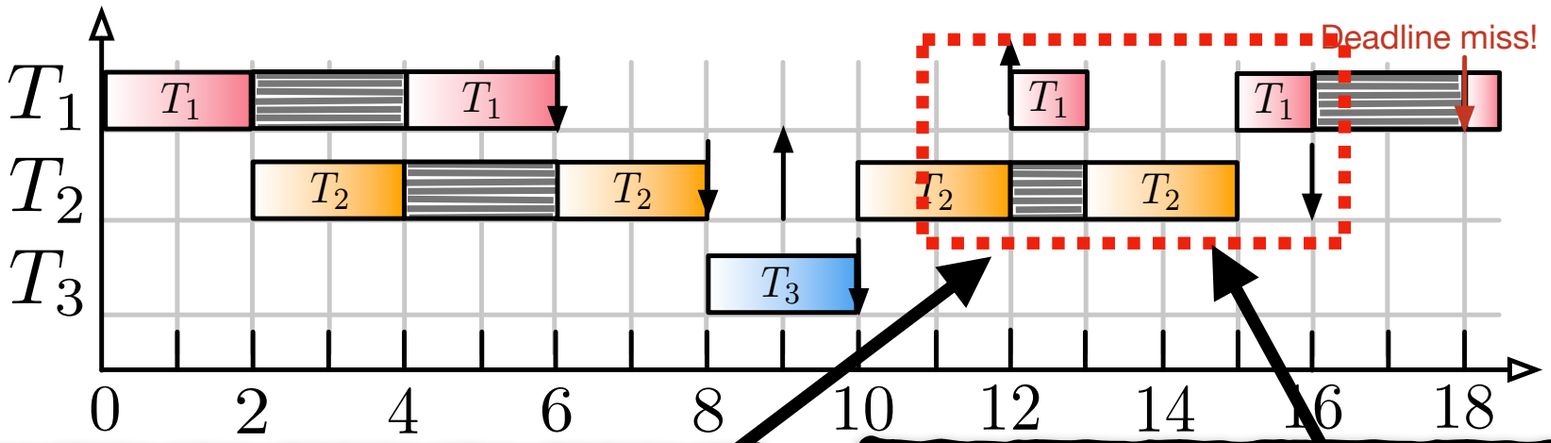
What About Suspension Times?

[Example by Abdeddaïm and Masson, 2012]

Original Job Set



Reducing Suspension Time of T_2



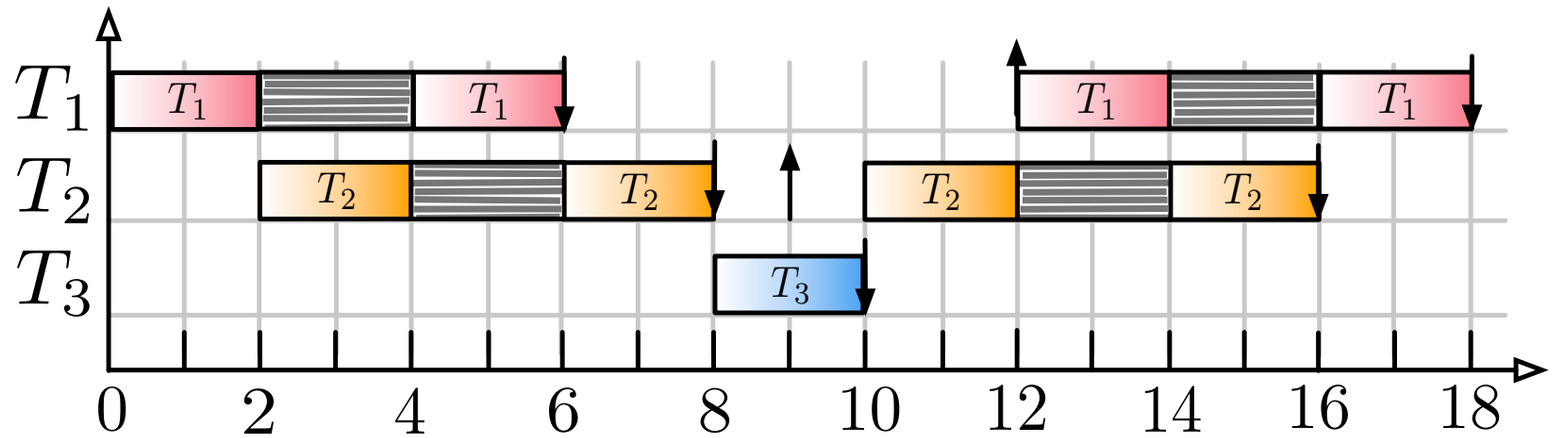
Assume T_2 suspends for only one time unit

T_2 suspends less and interferes more with T_1

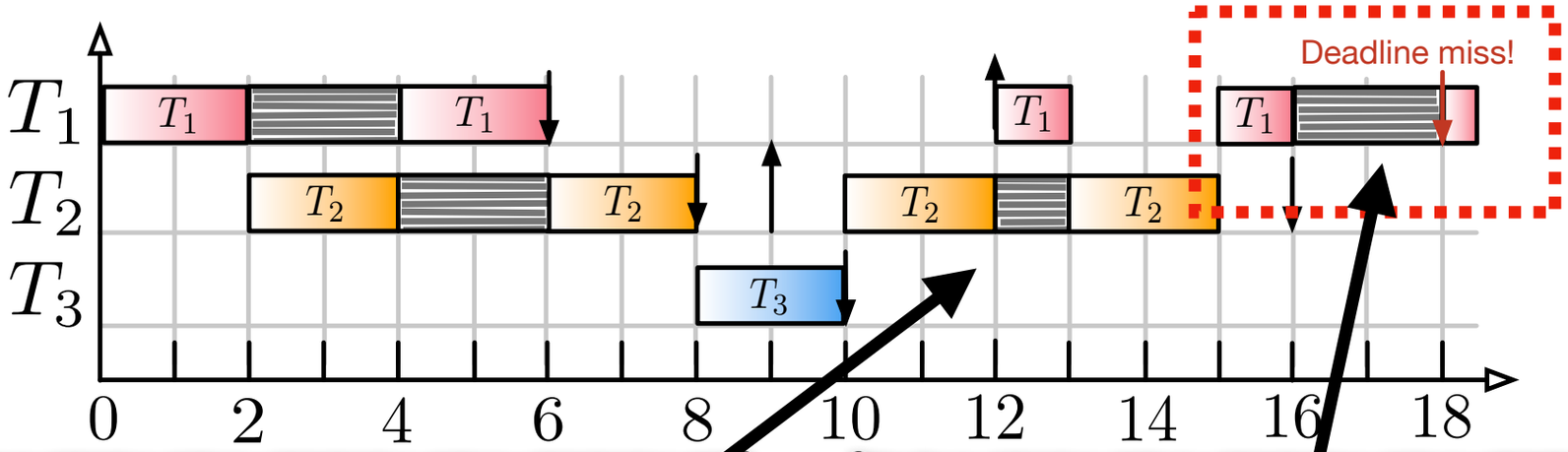
What About Suspension Times?

[Example by Abdeddaïm and Masson, 2012]

Original Job Set



Reducing Suspension Time of T_2

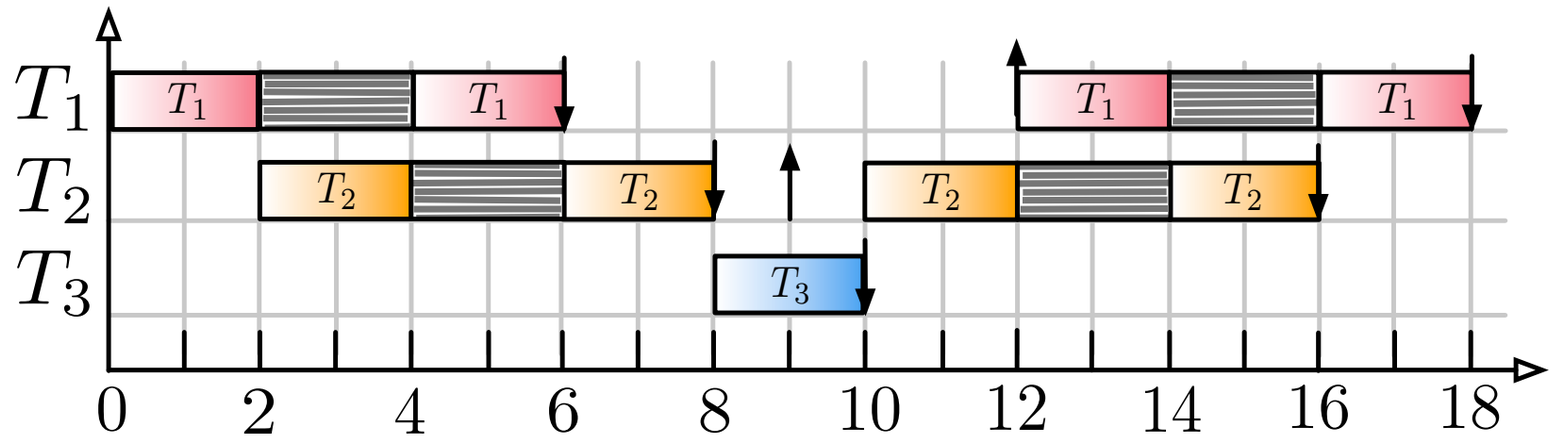


Assume T_2 suspends for only one time unit

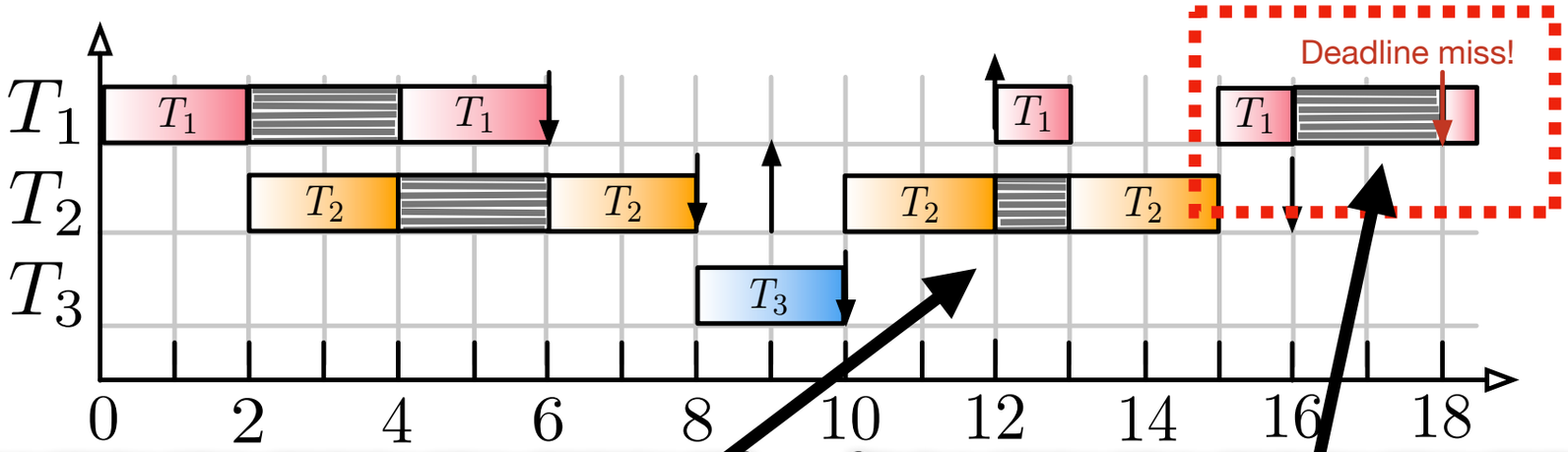
Interference causes deadline miss for T_1

Abdeddaïm and Masson: JLFP scheduling of self-suspending tasks is not sustainable with respect to both execution and suspension times.

Original Job Set



Reducing Suspension Time of T_2

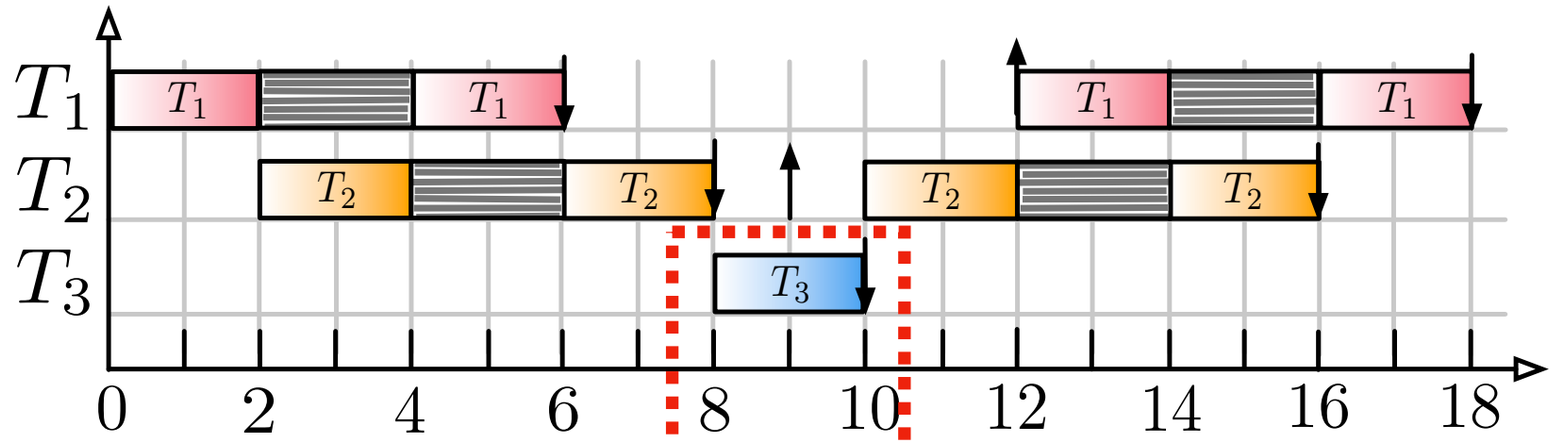


Assume T_2 suspends for only one time unit

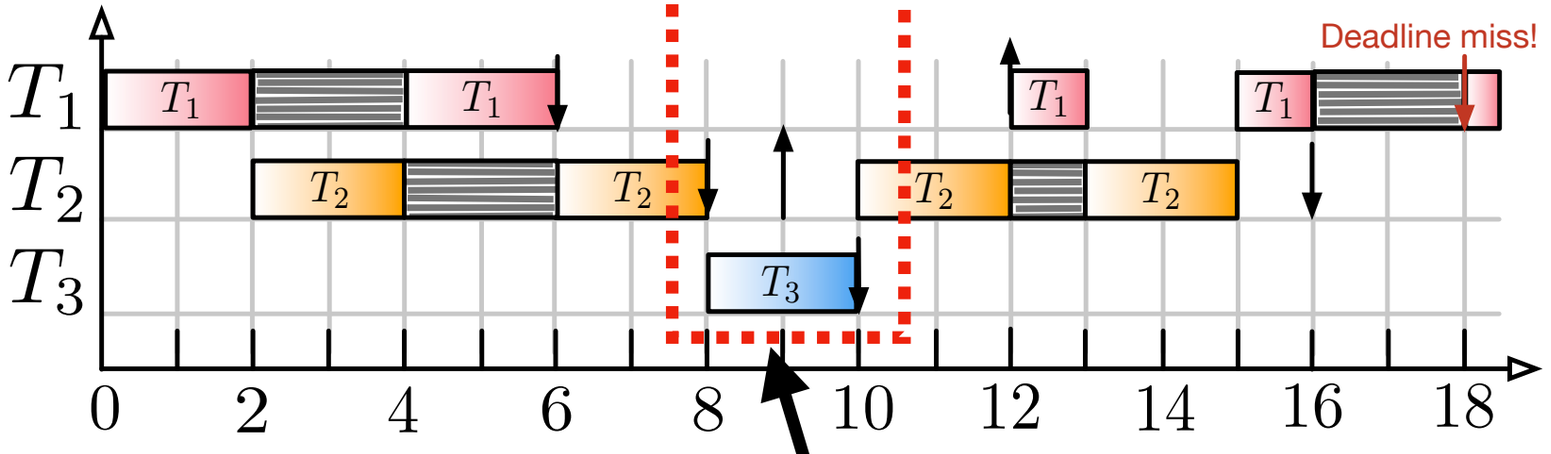
Interference causes deadline miss for T_1

Is the System Originally Schedulable?

Original Job Set



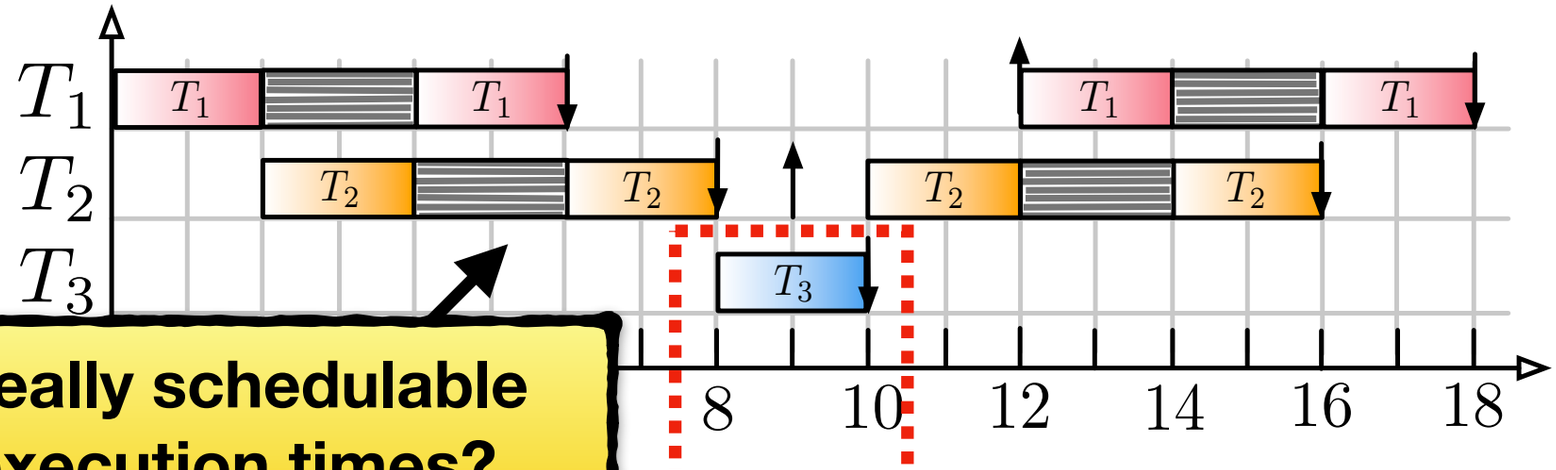
Reducing Suspension Time of T_2



The new job set has the original execution times but misses a deadline

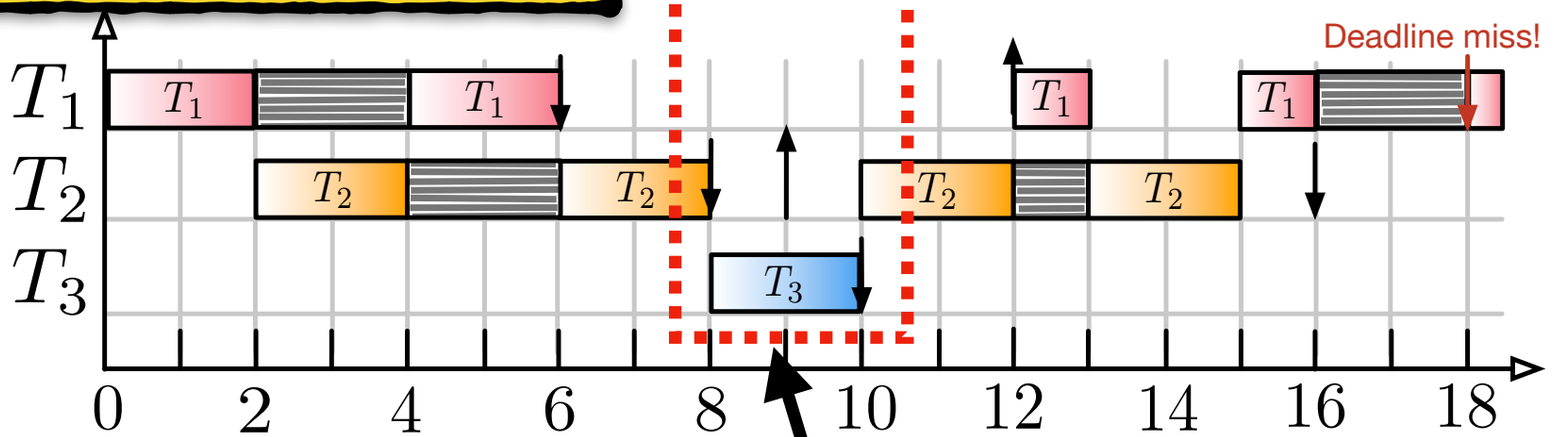
Is the System Originally Schedulable?

Original Job Set



Is the system really schedulable with original execution times?

Reducing Suspension Time of T_2



The new job set has the original execution times but misses a deadline

Two Possible Interpretations

If a system is schedulable with original parameters, then it remains schedulable when assigned better parameters.

	Baseline System	Sustainability w.r.t. Execution Times
Burns and Baruah, 2008	"For any **schedulable job set**…"	**Not Sustainable**
Baker and Baruah, 2009	"For **any job set of a schedulable task set**…"	**Example is not a counterexample**

Did We Pick a Bad Example?

Did We Pick a Bad Example?

NO!

Definitions Need Clarification

Definition 1 (Sustainable scheduling policy) *Let A denote a scheduling policy. Let τ denote any sporadic task system that is A-schedulable. Let $\mathcal{J}$ denote a collection of jobs generated by τ. Scheduling policy A is said to be* sustainable *if and only if A meets all deadlines when scheduling any collection of jobs obtained from $\mathcal{J}$ by changing the parameters of one or more individual jobs in any, some, or all of the following ways: (i) decreased execution requirements; (ii) larger relative deadlines; and (iii) later arrival times with the restriction that successive jobs of any task $\tau_i \in \tau$ arrive at least T_i time units apart.*

Theodore P. Baker and Sanjoy K. Baruah. Sustainable multiprocessor scheduling of sporadic task systems. In Proceedings of the 21st Euromicro Conference on Real-Time Systems, ECRTS '09, pages 141–150, 2009.

Definitions Need Clarification

Definition 1 (Sustainable scheduling policy) *Let A denote a scheduling policy. Let τ denote any sporadic task system that is A-schedulable. Let $\mathcal{J}$ denote a collection of jobs generated by τ. Scheduling policy A is said to be* sustainable *if and only if A meets all deadlines when scheduling any collection of jobs obtained from $\mathcal{J}$ by changing the parameters of one or more individual jobs in any, some, or all of the following ways: (i) decreased execution requirements; (ii) larger relative deadlines; and (iii) later arrival times with the restriction that successive jobs of any task $\tau_i \in \tau$ arrive at least T_i time units apart.*

1) Assume arbitrary job sets or job sets from a schedulable task set?

Theodore P. Baker and Sanjoy K. Baruah. Sustainable multiprocessor scheduling of sporadic task systems. In Proceedings of the 21st Euromicro Conference on Real-Time Systems, ECRTS '09, pages 141–150, 2009.

Definitions Need Clarification

Definition 1 (Sustainable scheduling policy) *Let A denote a scheduling policy. Let τ denote any sporadic task system that is A-schedulable. Let $\mathcal{J}$ denote a collection of jobs generated by τ. Scheduling policy A is said to be* sustainable *if and only if A meets all deadlines when scheduling any collection of jobs obtained from $\mathcal{J}$ by changing the parameters of one or more individual jobs in any, some, or all of the following ways: (i) decreased execution requirements; (ii) larger relative deadlines; and (iii) later arrival times with the restriction that successive jobs of any task $\tau_i \in \tau$ arrive at least T_i time units apart.*

1) Assume arbitrary job sets or job sets from a schedulable task set?

2) Should other parameters remain fixed, or they are allowed to vary?

Theodore P. Baker and Sanjoy K. Baruah. Sustainable multiprocessor scheduling of sporadic task systems. In Proceedings of the 21st Euromicro Conference on Real-Time Systems, ECRTS '09, pages 141–150, 2009.

Definitions Need Clarification

Definition 1 (Sustainable scheduling policy) *Let A denote a scheduling policy. Let τ denote any sporadic task system that is A-schedulable. Let $\mathcal{J}$ denote a collection of jobs generated by τ. Scheduling policy A is said to be* sustainable *if and only if A meets all deadlines when scheduling any collection of jobs obtained from $\mathcal{J}$ by changing the parameters of one or more individual jobs in any, some, or all of the following ways: (i) decreased execution requirements; (ii) larger relative deadlines; and (iii) later arrival times with the restriction that successive jobs of any task $\tau_i \in \tau$ arrive at least T_i time units apart.*

1) Assume arbitrary job sets or job sets from a schedulable task set?

2) Should other parameters remain fixed, or they are allowed to vary?

3) Why are we limited to a specific set of parameters?

Theodore P. Baker and Sanjoy K. Baruah. Sustainable multiprocessor scheduling of sporadic task systems. In Proceedings of the 21st Euromicro Conference on Real-Time Systems, ECRTS '09, pages 141–150, 2009.

This Talk

Ambiguity in the Current Definitions

Strong and Weak Sustainability

Weak Sustainability of Self-Suspending Tasks

This Talk

Ambiguity in the Current Definitions

Strong and Weak Sustainability

Weak Sustainability of Self-Suspending Tasks

Strongly Sustainable w.r.t. Parameter P

Strongly Sustainable w.r.t. Parameter P

If a job set is **schedulable,** then it **remains schedulable** for **better values of parameter P,** assuming **all parameters other than P remain constant.**

Strongly Sustainable w.r.t. Parameter P

If a job set is **schedulable,** then it **remains schedulable** for **better values of parameter P,** assuming **all parameters other than P remain constant.**

This matches the result by Abdeddaïm and Masson.
The examples show that EDF scheduling of self-suspending tasks is not strongly sustainable w.r.t. both execution and suspension times.

What we Know about Sustainability

Strongly Sustainable w.r.t. Parameter P

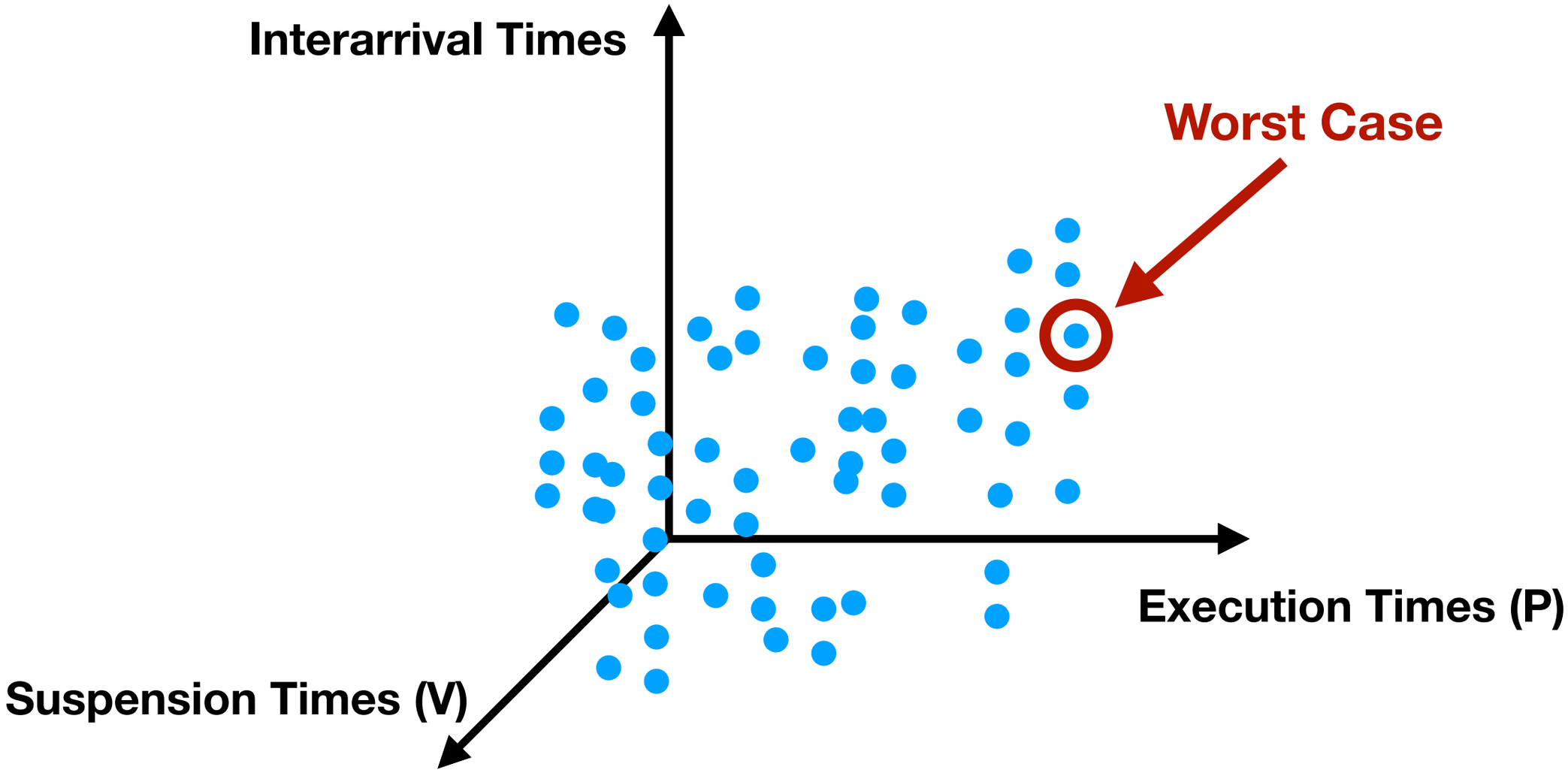
vs.

Unsustainable

Allows fixing worst-case values for parameter P

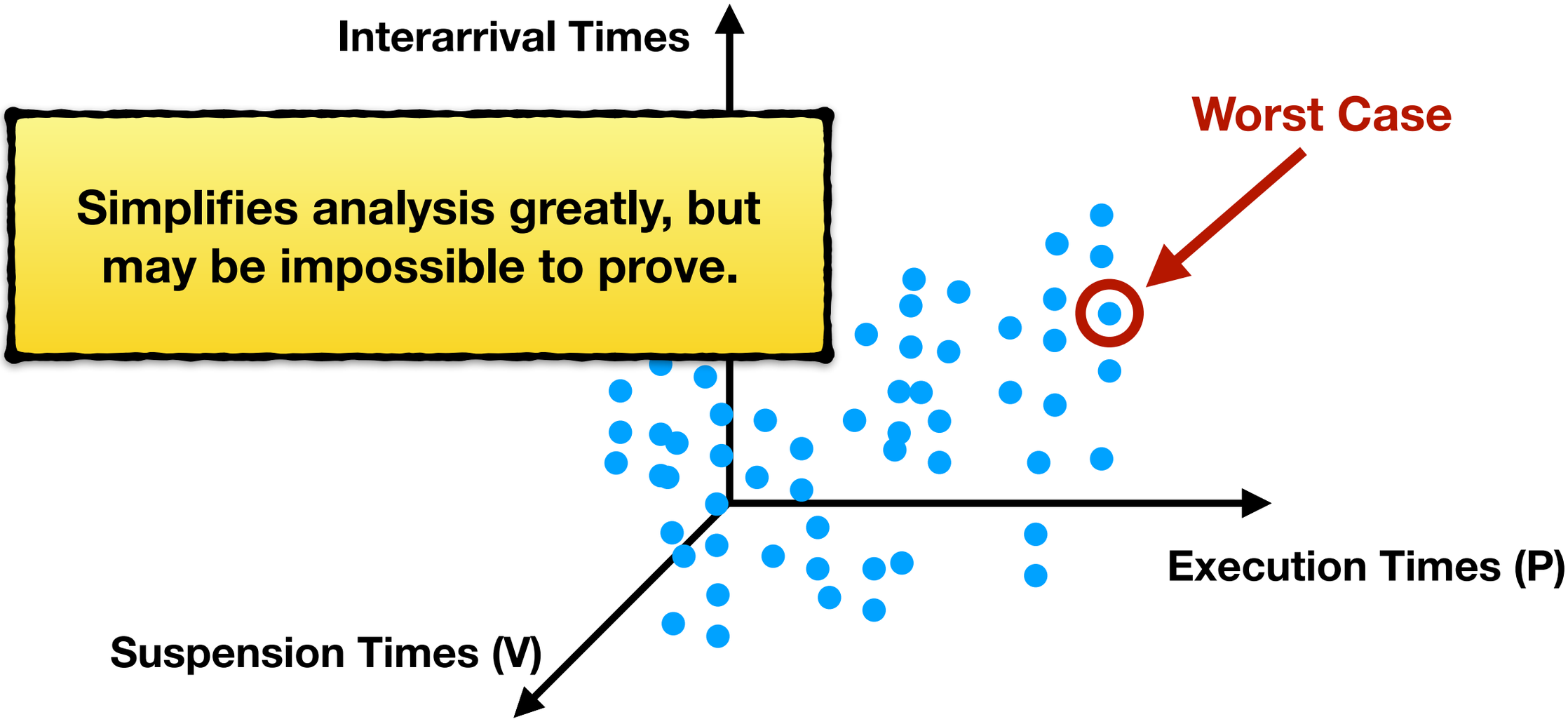
All combinations of parameter values must be checked by the analysis

Strong Sustainability: Ideal but Hard to Prove



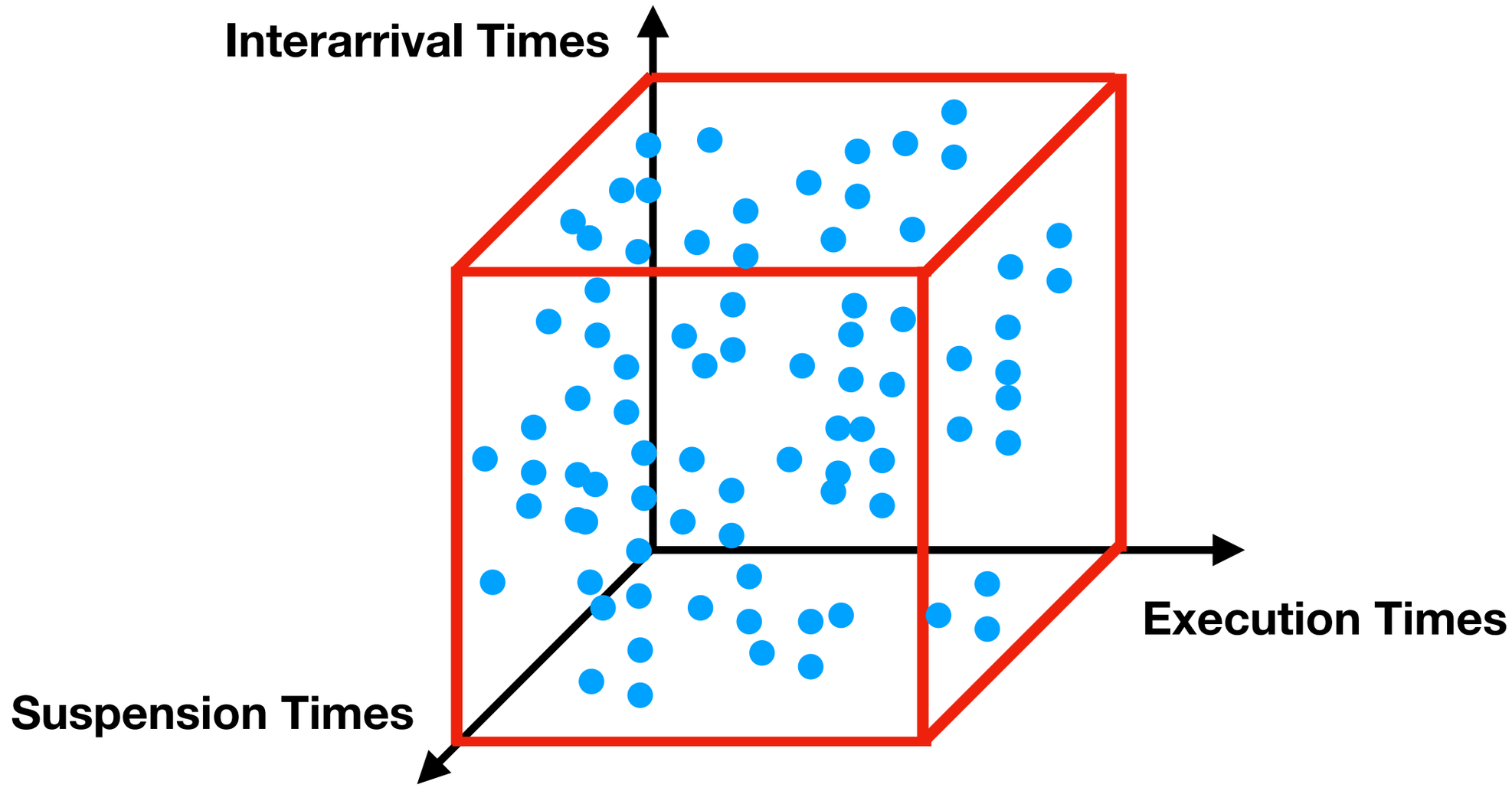
If all parameters are strongly sustainable, the worst-case scenario lies in a single point of the parameter space.

Strong Sustainability: Ideal but Hard to Prove



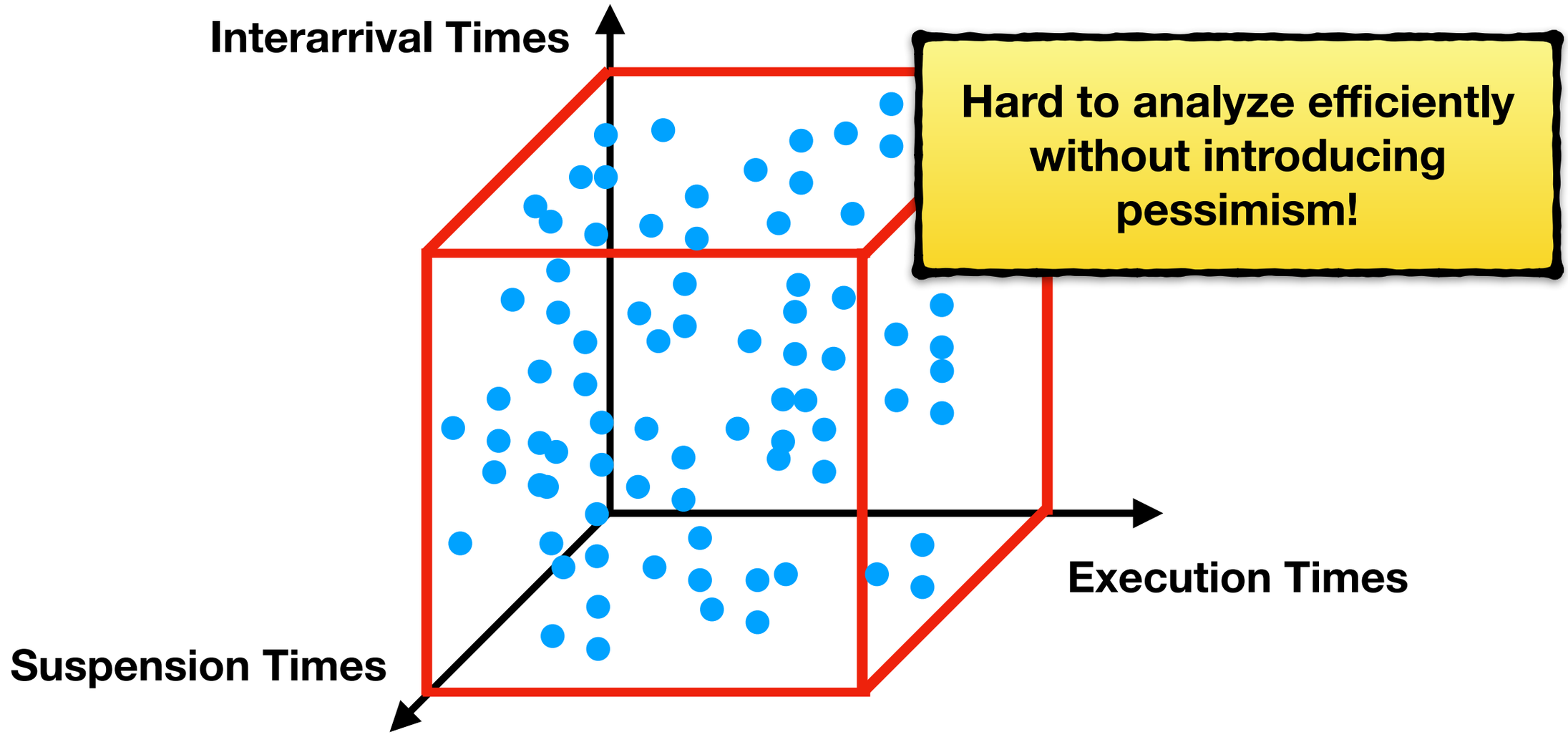
If all parameters are strongly sustainable, the worst-case scenario lies in a single point of the parameter space.

Unsustainability: Difficult to Analyze



All possible combinations of parameters must be checked to determine the worst-case scenario.

Unsustainability: Difficult to Analyze



All possible combinations of parameters must be checked to determine the worst-case scenario.

How to Find a Middle Ground?

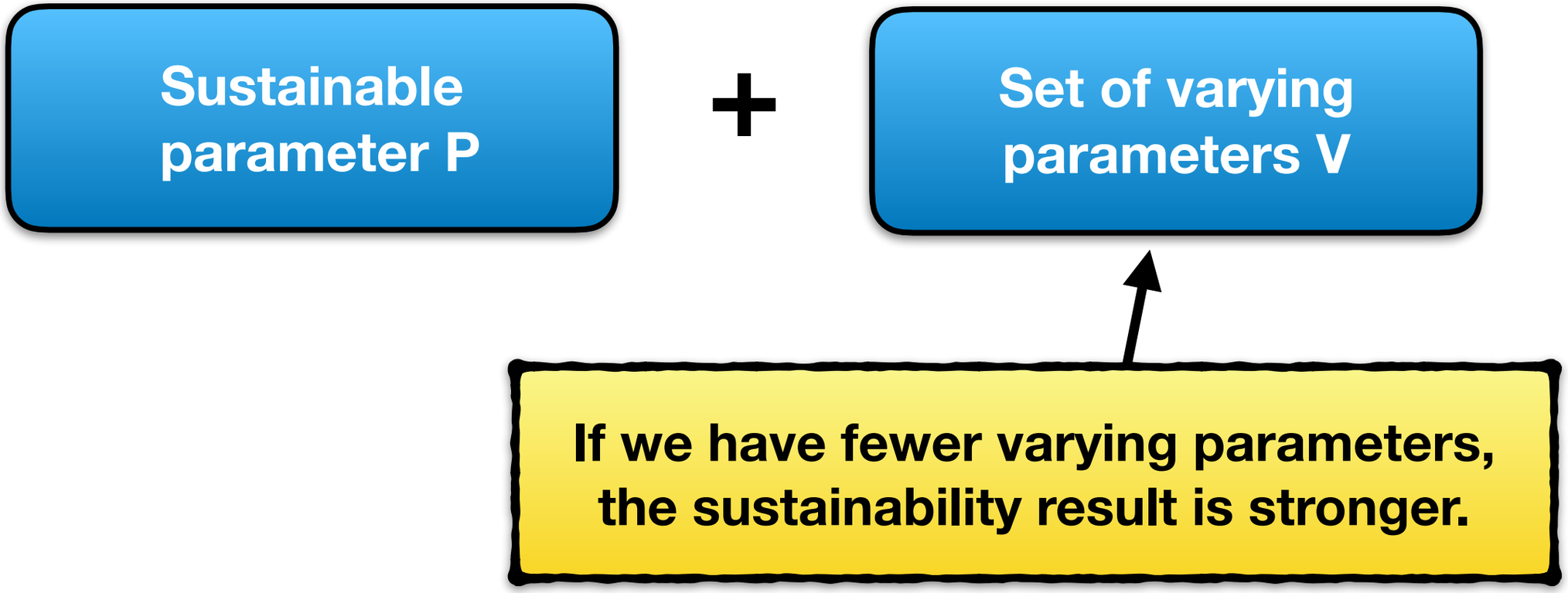
How to Find a Middle Ground?

Core idea: make explicit which parameters are allowed to vary.

Sustainable parameter P + **Set of varying parameters V**

How to Find a Middle Ground?

Core idea: make explicit which parameters are allowed to vary.

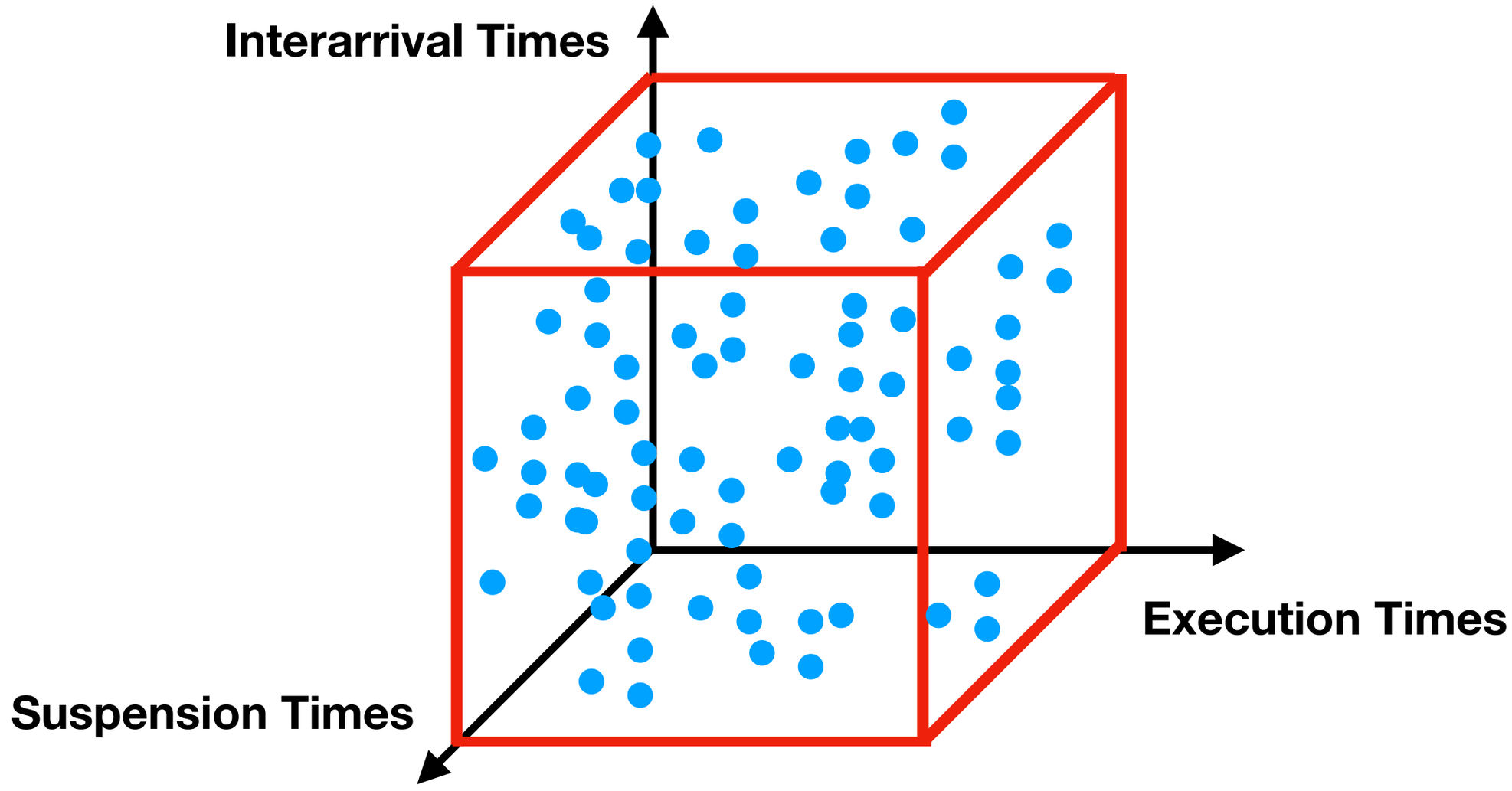


Weakly Sustainable w.r.t. Parameter P and Variable Parameters V

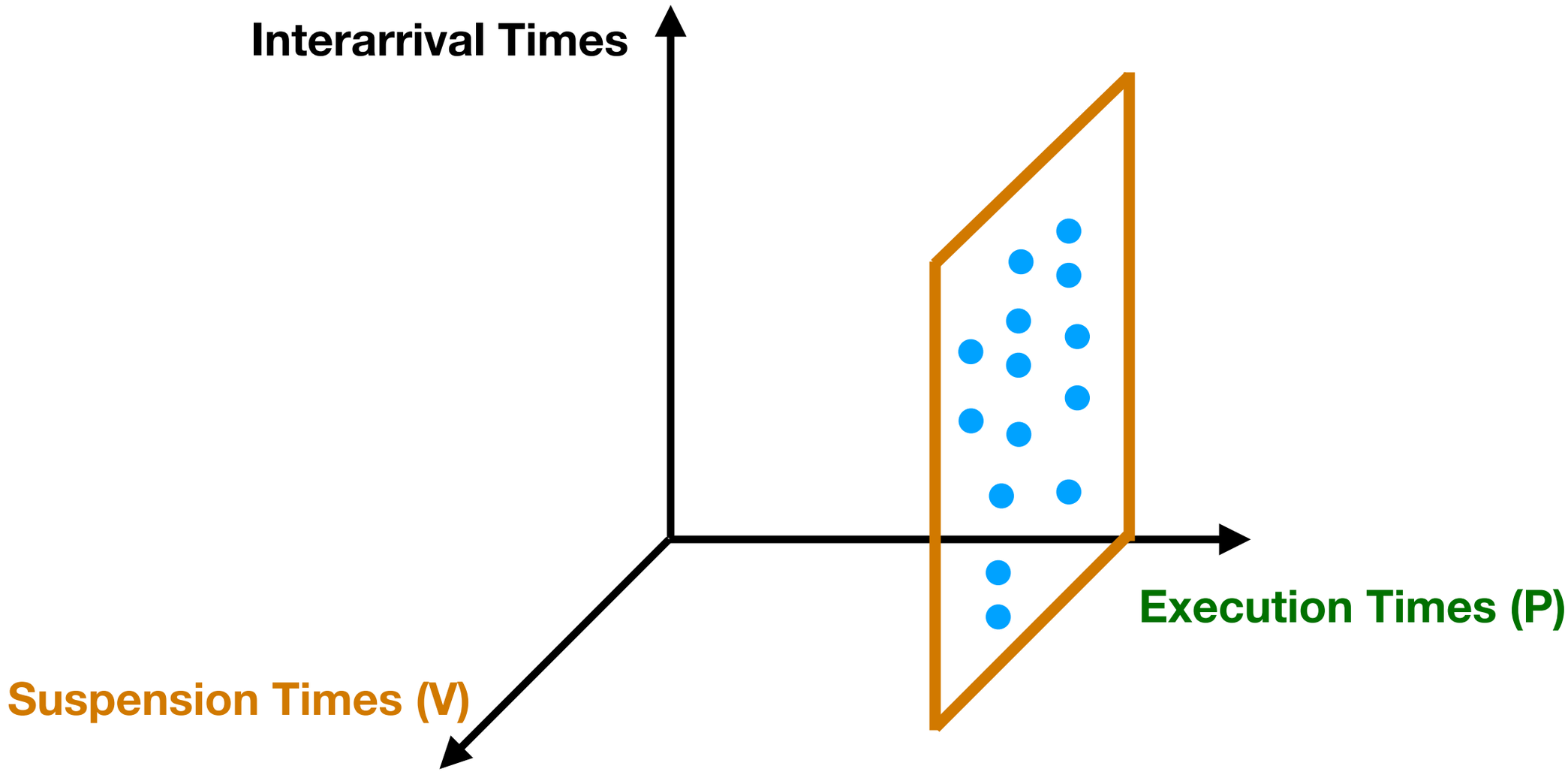
Weakly Sustainable w.r.t. Parameter P and Variable Parameters V

If a job set is **schedulable**
for all possible combinations of parameter values in V,
then it **remains schedulable** for **better values of parameter P,**
assuming **all parameters other than P and those in V remain constant.**

Unsustainability: Difficult to Analyze

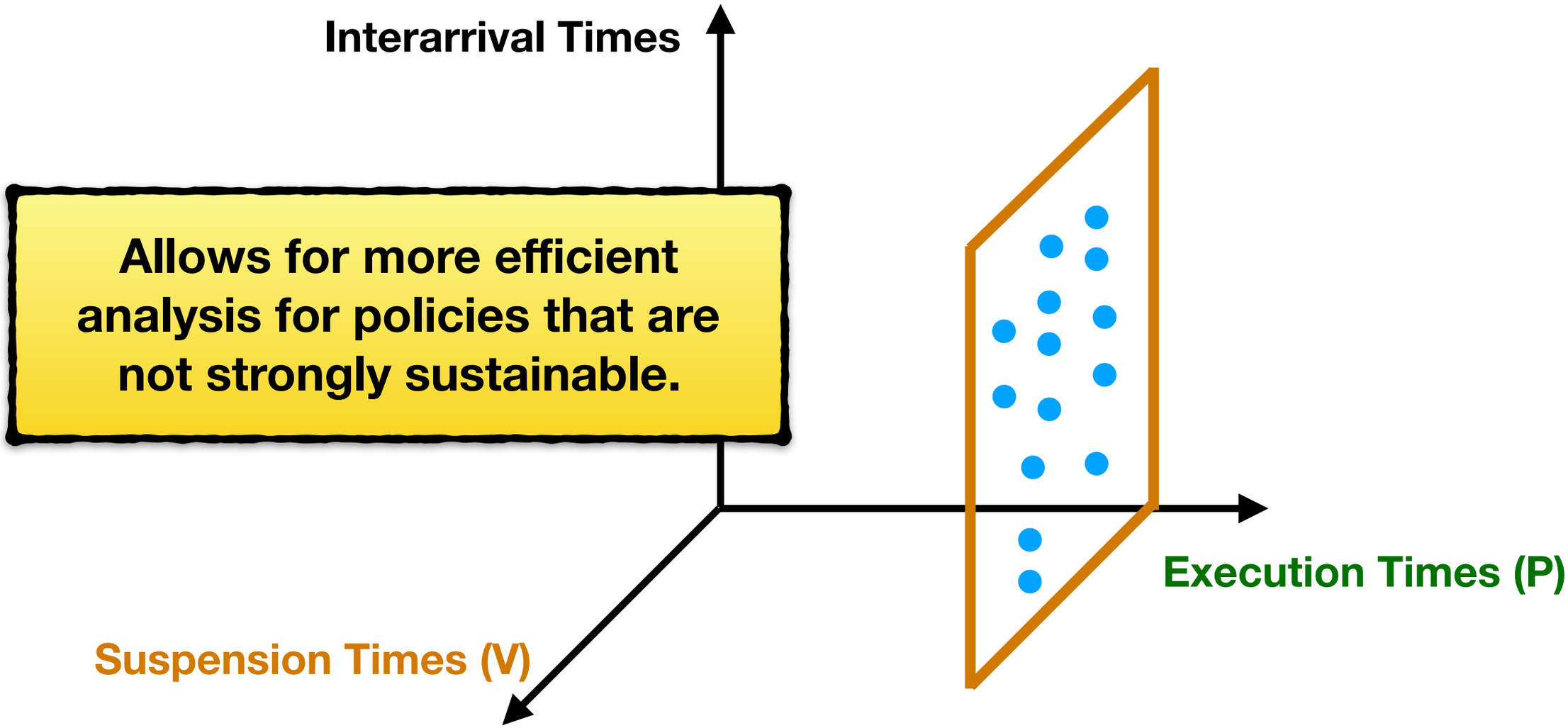


Weak Sustainability: Search Space Reduction



We can maximize/minimize the sustainable parameter P,
as long as the analysis covers all possible values in V.

Weak Sustainability: Search Space Reduction



We can maximize/minimize the sustainable parameter P, as long as the analysis covers all possible values in V.

This Talk

Ambiguity in the Current Definitions

Strong and Weak Sustainability

Weak Sustainability of Self-Suspending Tasks

This Talk

Ambiguity in the Current Definitions

Strong and Weak Sustainability

Weak Sustainability of Self-Suspending Tasks

Recall Result by Abdeddaïm and Masson

EDF scheduling of self-suspending tasks is not (strongly) sustainable w.r.t. both execution and suspension times.

The Policy is Actually Sustainable

EDF scheduling of self-suspending tasks is not (strongly) sustainable w.r.t. both execution and suspension times.

We proved that uniprocessor JLFP scheduling of self-suspending tasks is weakly sustainable w.r.t. execution times and variable suspension times.

The Policy is Actually Sustainable

EDF scheduling of self-suspending tasks is not (strongly) sustainable w.r.t. both execution and suspension times.

We proved that uniprocessor JLFP scheduling of self-suspending tasks is weakly sustainable w.r.t. execution times and variable suspension times.

machine-checked with

formally proven schedulability analysis | PROSA

prosa.mpi-sws.org/releases/sustainability

What Else is in the Paper?

What Else is in the Paper?

Formal theory
of sustainability

What Else is in the Paper?

Formal theory
of sustainability

What really is a job parameter? How to define “better” or “worse”?

We formalize sustainable policy, sustainable analysis and self-sustainable analysis.

What Else is in the Paper?

Formal theory
of sustainability

What Else is in the Paper?

Formal theory
of sustainability

$$A + B = C$$

Composition rules
for weak and
strong sustainability

What Else is in the Paper?

Formal theory
of sustainability

$$A + B = C$$

Composition rules
for weak and
strong sustainability

How to combine sustainability proofs with different values of P and V?

We need certain assumptions on the parameters!

What Else is in the Paper?

Formal theory
of sustainability

$$A + B = C$$

Composition rules
for weak and
strong sustainability

What Else is in the Paper?

Formal theory
of sustainability

$$A + B = C$$

Composition rules
for weak and
strong sustainability

Proof strategy
for weak sustainability

Schedule construction with two proof obligations:
(a) service invariant + (b) validity of the new schedule

Takeaways

Takeaways

With the rigor imposed by a proof assistant, we were able to clarify an important concept in real-time scheduling.

Takeaways

With the rigor imposed by a proof assistant, we were able to clarify an important concept in real-time scheduling.

We proposed weak sustainability, which enables more efficient analysis for policies that are not strongly sustainable.

Takeaways

With the rigor imposed by a proof assistant, we were able to clarify an important concept in real-time scheduling.

We proposed weak sustainability, which enables more efficient analysis for policies that are not strongly sustainable.

We proved that uniprocessor JLFP scheduling of self-suspending tasks is weakly sustainable w.r.t. execution times and variable suspension times.

formally proven schedulability analysis | PROSA

For more details, visit prosa.mpi-sws.org/releases/sustainability