Exact Schedulability Analysis of Global Fixed Priority Scheduling by Using Linear Hybrid Automata

Youcheng Sun[1], Giuseppe Lipari[1 2]

[1]Scuola Superiore Sant'Anna

[2]LSV - ENS Cachan and CNRS

October 8, 2014

Outline

1. Introduction
2. LHA models for multiprocessor G-FP scheduling
3. Weak Simulation Relation
   - Concrete state space
   - Symbolic state space
4. Experiments

Outline

Problem description

- *m* processors
- *n* sporadic tasks
- Global (task level) Fixed-Priority (G-FP) Preemptive Scheduling

Problem description

- *m* processors
- *n* sporadic tasks
- Global (task level) Fixed-Priority (G-FP) Preemptive Scheduling
- Is the taskset schedulable?

System Model

- A sporadic task τ_is specified by a tuple (C, D, T)
  - C is the worst-case execution time
  - D is the relative deadline
  - T is the minimum time interval between two successive job releases of the task
  - $T \sim D$ with $\sim \in \{<, =, >\}$
- Fully Preemptive
- Tasks can migrate among different processors

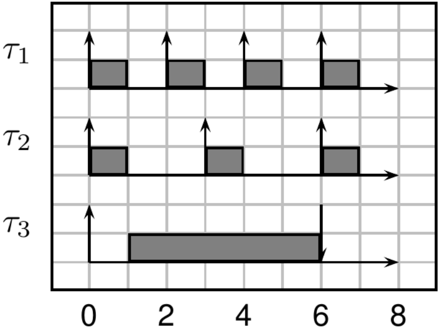
Critical instant for G-FP scheduling?

Critical instant for G-FP scheduling?

- Unknown

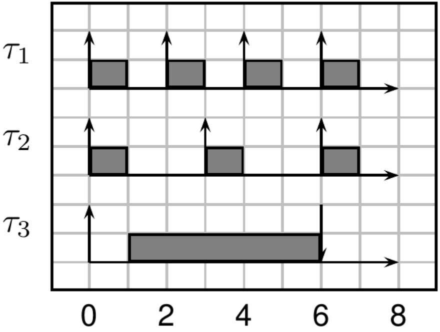
Critical instant for G-FP scheduling?

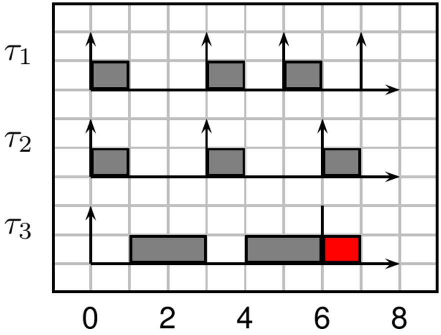
- Unknown
- An example (Baruah, 2007): $\tau_1 = (1,2,2)$, $\tau_2 = (1,3,3)$, and $\tau_3 = (5,6,6)$



Critical instant for G-FP scheduling?

- Unknown
- An example (Baruah, 2007): $\tau_1 = (1, 2, 2)$, $\tau_2 = (1, 3, 3)$, and $\tau_3 = (5, 6, 6)$





State-of-the-art

- Analytical (**over-approximate**) solutions: RTA-CE (Sun et al., 2014), RTA-LC (Guan et al., 2009)

State-of-the-art

- Analytical (**over-approximate**) solutions: RTA-CE (Sun et al., 2014), RTA-LC (Guan et al., 2009)
- Model-based (**exact**) solutions : Geeraerts, Goossens, and Lindstrom, 2013 and Baker and Cirinei, 2007

Our contributions

- A Linear Hybrid Automaton (LHA) model for **exact** G-FP scheduling

Our contributions

- A Linear Hybrid Automaton (LHA) model for **exact** G-FP scheduling
- A weak simulation relation to simplify the state space exploration

Our contributions

- A Linear Hybrid Automaton (LHA) model for **exact** G-FP scheduling
- A weak simulation relation to simplify the state space exploration
- An evaluation on the pessimism of the state-of-the-art analytical G-FP schedulability analysis

Linear Hybrid Automata(LHA)

A Linear Hybrid Automaton is a tuple

$$H = \{V, D, L, init, Lab, T, Invar\}$$

Linear Hybrid Automata(LHA)

A Linear Hybrid Automaton is a tuple

$$H = \{V, D, L, init, Lab, T, Invar\}$$

1. A finite set $V = \{x_1, \ldots, x_n\}$ of continuous variables.
2. A labeling function D which linearly constrains variables' rate ($\dot{V} = \{\dot{x}_1, \ldots, \dot{x}_N\}$) in each location.

Linear Hybrid Automata(LHA)

A Linear Hybrid Automaton is a tuple

$$H = \{V, D, L, init, Lab, T, Invar\}$$

1. A finite set $V = \{x_1, \ldots, x_n\}$ of continuous variables.
2. A labeling function D which linearly constrains variables' rate ($\dot{V} = \{\dot{x}_1, \ldots, \dot{x}_N\}$) in each location.
3. A finite set L of locations.
4. An initial function *init*.
5. A finite set *Lab* of synchronisation labels.
6. A finite set T of transitions (every location has an outgoing stutter transition to itself).
7. A labeling function *Invar* which assigns each location l an *invariant*.

Concrete states and Symbolic States

- A concrete state $s = (l, \nu)$: l is a location and ν is a valuation over V

Concrete states and Symbolic States

- A concrete state $s = (l, \nu)$: l is a location and ν is a valuation over V
  - A transition $s_1 \rightarrow s_2$; a sequence of transitions $s_1 \Rightarrow s_2$
  - The concrete state space of LHA : `space`

Concrete states and Symbolic States

- A concrete state $s = (l, \nu)$: l is a location and ν is a valuation over V
  - A transition $s_1 \rightarrow s_2$; a sequence of transitions $s_1 \Rightarrow s_2$
  - The concrete state space of LHA : `space`
- A symbolic state $S = (l, \mathcal{C})$: l is a location and $\mathcal{C}$ is a linear constraint and can be represented by a convex region

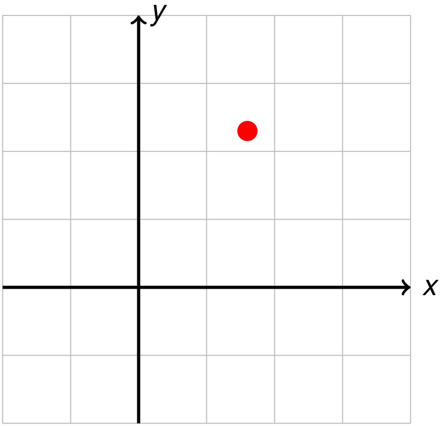
Concrete states and Symbolic States

- A concrete state $s = (l, \nu)$: l is a location and ν is a valuation over V
  - A transition $s_1 \rightarrow s_2$; a sequence of transitions $s_1 \Rightarrow s_2$
  - The concrete state space of LHA : `space`
- A symbolic state $S = (l, \mathcal{C})$: l is a location and $\mathcal{C}$ is a linear constraint and can be represented by a convex region
  - A transition $S_1 \rightarrow S_2$; a sequence of transitions $S_1 \Rightarrow S_2$
  - The symbolic state space of LHA : `Space`

Examples

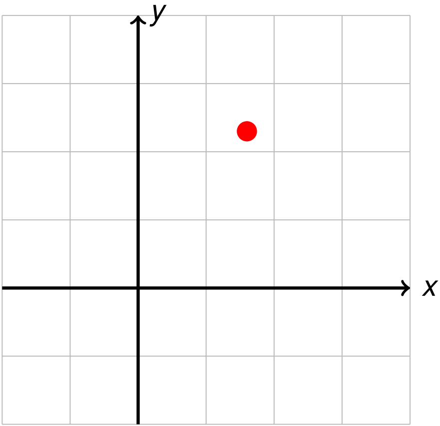
Examples

$s = (l, v)$ with
$\nu = (1.6, 2.3)$

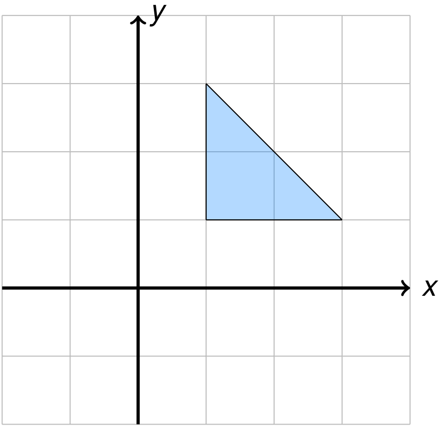


Examples

$s = (l, v)$ with
$\nu = (1.6, 2.3)$



$S = (l, \mathcal{C})$ with
$\mathcal{C} = \{1 \leq x \leq 3 \wedge x + y \leq 4\}$



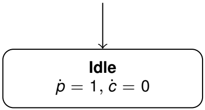
Outline

Task automaton

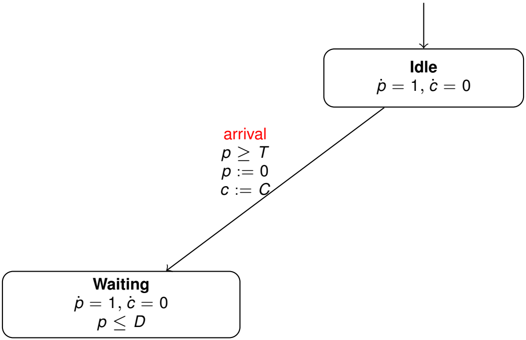
A task (C, D, T) is modeled by a LHA called `Task Automaton`(TA)

- Two continuous variables :
  1. p : the time passed since the latest task activation
  2. c : remaining computation time that needs to be executed

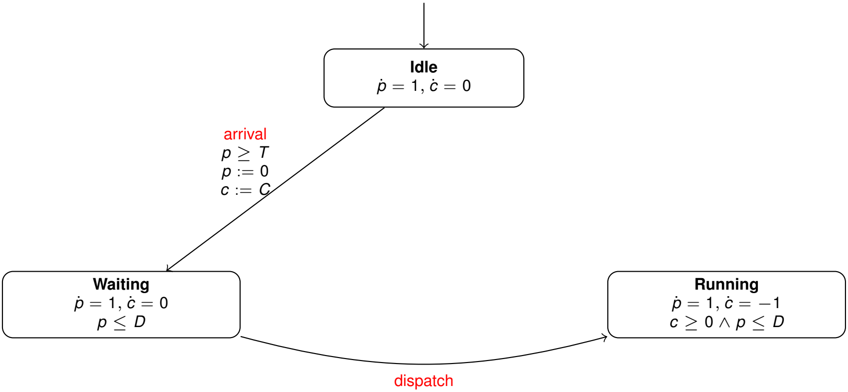
Task automaton



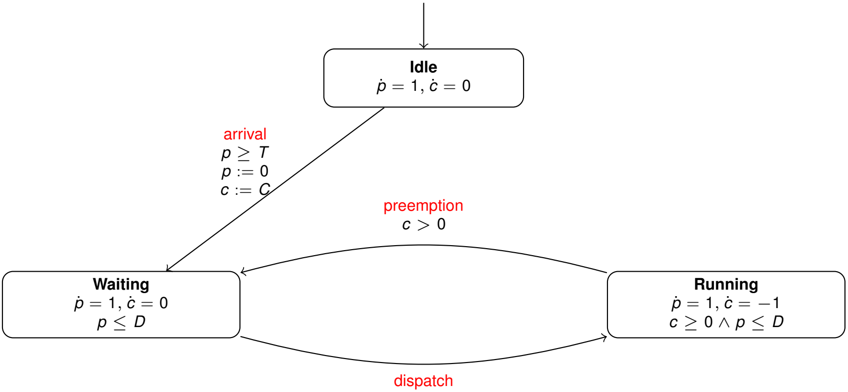
Task automaton



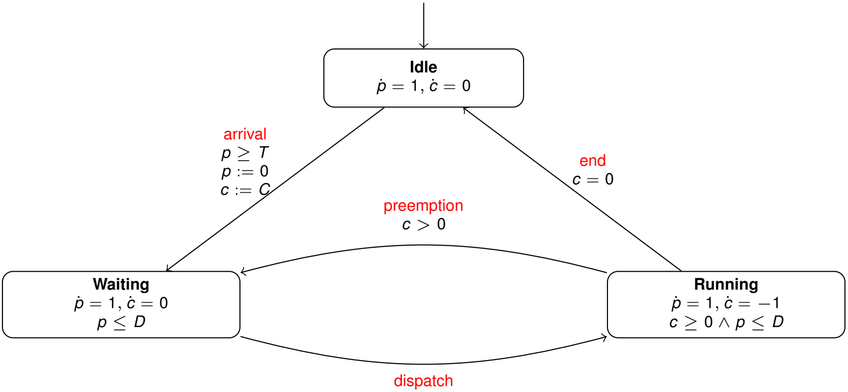
Task automaton



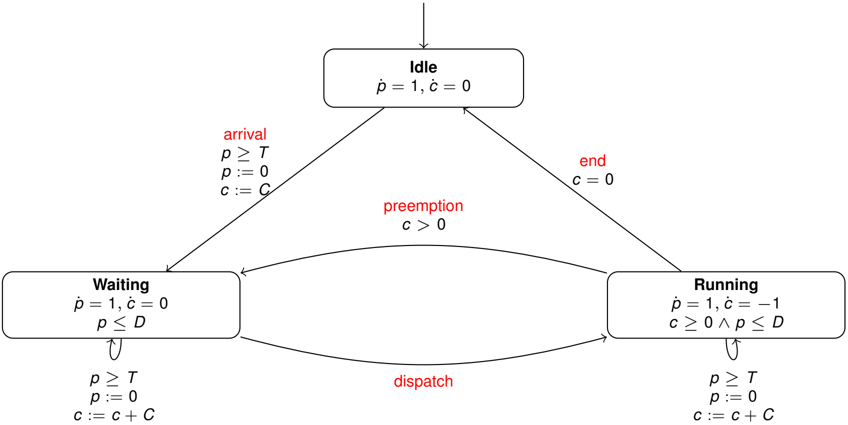
Task automaton



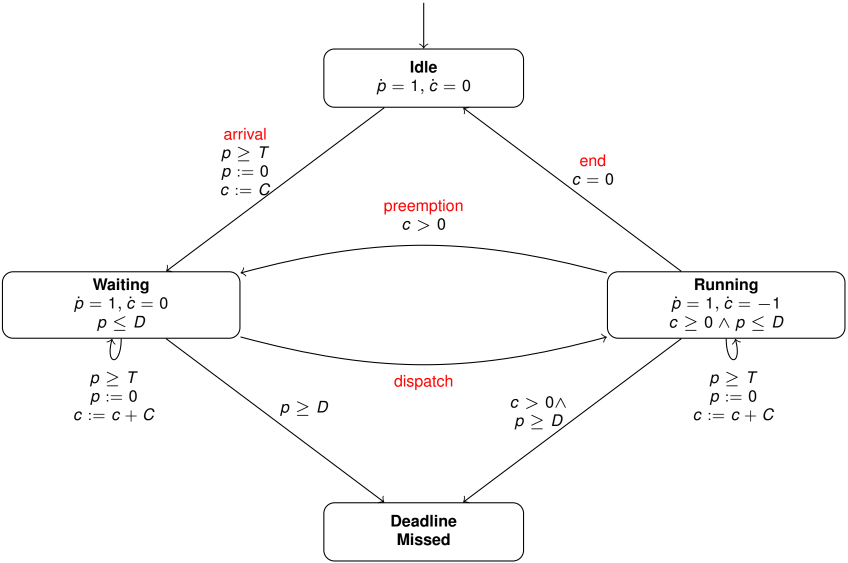
Task automaton



Task automaton



Task automaton



Scheduling Automaton and System Automaton

- The `Scheduling Automaton`(Sched)
  - It synchronises with TAs and decides which tasks to run and which tasks to wait.
  - It is a G-FP preemptive scheduler.
- The `System Automaton`(SA)
  - Composition of TAs and Sched : $Sched \times TA_1 \times \cdots \times TA_n$

Scheduling Automaton and System Automaton

- The `Scheduling Automaton`(Sched)
  - It synchronises with TAs and decides which tasks to run and which tasks to wait.
  - It is a G-FP preemptive scheduler.
- The `System Automaton`(SA)
  - Composition of TAs and Sched : $Sched \times TA_1 \times \cdots \times TA_n$
- The schedulability problem is now the reachability problem of `DeadlineMissed` in SA.

Outline

Outline

Definition

A weak simulation relation in concrete state space of SA is a preorder $\succeq \subseteq$ space $\times$ space such that :

1. $\forall s_1, s_2, s_4$ s.t. $s_1 \succeq s_2, s_2 \rightarrow s_4$ there exists s_3 s.t.
$$s_1 \Rightarrow s_3 \text{ and } s_3 \succeq s_4.$$
2. $\forall s_1, s_2$ s.t. $s_1 \succeq s_2$:
$$s_2 \text{ in DeadlineMissed implies } s_1 \text{ in DeadlineMissed}$$

Whenever $s_1 \succeq s_2$, we say that s_1 (weak) simulates s_2.

The slack-time pre-order relation $\succeq_{st}$

Definition

For the SA with a G-FP preemptive scheduler, its slack-time pre-order relation $\succeq_{st} \subseteq space \times space$ is defined such that $\forall s_1, s_2,\ s_1 \succeq_{st} s_2$ iff

$$\forall \tau_i : \quad s_1.p_i \geq s_2.p_i \quad \wedge \quad s_1.c_i \geq s_2.c_i$$

The slack-time pre-order relation $\succeq_{st}$

Definition

For the SA with a G-FP preemptive scheduler, its slack-time pre-order relation $\succeq_{st} \subseteq$ space $\times$ space is defined such that $\forall s_1, s_2$, $s_1 \succeq_{st} s_2$ iff

$$\forall \tau_i : \quad s_1.p_i \geq s_2.p_i \quad \wedge \quad s_1.c_i \geq s_2.c_i$$

Theorem

$\succeq_{st}$ is indeed a weak simulation relation in SA.

Outline

From `space` to `Space`

- A symbolic state $S = (I, \mathcal{C})$ abstracts a set of concrete states.
- For two symbolic states S_1 and S_2, we say S_1 simulates S_2 if

$$\forall s_2 \in S_2 \ , \ \exists s_1 \in S_1 \quad s.t. \quad s_1 \succeq s_2$$

A preliminary concept : convex region domination

- Assume a N-dimensional space
- Given two valuations $\nu = (\nu_1, \ldots, \nu_N)$ and $\nu' = (\nu'_1, \ldots, \nu'_N)$, we say ν dominates ν', denoted as $\nu \geq \nu'$, if $\forall i \in [1, N], \nu_i \geq \nu'_i$.

A preliminary concept : convex region domination

- Assume a N-dimensional space
- Given two valuations $\nu = (\nu_1, \ldots, \nu_N)$ and $\nu' = (\nu'_1, \ldots, \nu'_N)$, we say ν dominates ν', denoted as $\nu \geq \nu'$, if $\forall i \in [1, N], \nu_i \geq \nu'_i$.
- Given two convex regions $\mathcal{C}_1$ and $\mathcal{C}_2$, we say $\mathcal{C}_1$ dominates $\mathcal{C}_2$, denoted as $\mathcal{C}_1 \geq \mathcal{C}_2$, if for any valuation ν' in $\mathcal{C}_2$, there exists a valuation $\nu \in \mathcal{C}_1$ such that $\nu \geq \nu'$.

$\succeq_{st}$ in `Space`

Definition

For the SA with a G-FP preemptive scheduler, its slack-time pre-order relation $\succeq_{st} \subseteq \mathtt{Space} \times \mathtt{Space}$ is defined such that $\forall S_1, S_2$, $S_1 \succeq_{st} S_2$ iff $S_1.\mathcal{C}$ dominates $S_2.\mathcal{C}$.

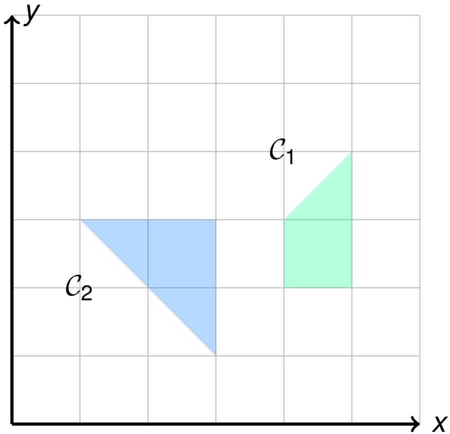
Definition

For the SA with a G-FP preemptive scheduler, its slack-time pre-order relation $\succeq_{st} \subseteq \mathtt{Space} \times \mathtt{Space}$ is defined such that $\forall S_1, S_2$, $S_1 \succeq_{st} S_2$ iff $S_1.\mathcal{C}$ dominates $S_2.\mathcal{C}$.

Theorem

$\succeq_{st} \subseteq \mathtt{Space} \times \mathtt{Space}$ is a weak simulation relation.

How to decide $\mathcal{C}_1 \geq \mathcal{C}_2$?



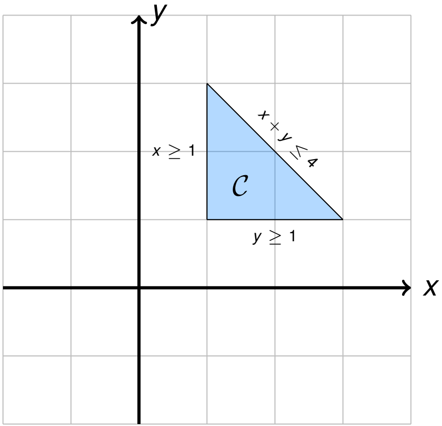
A widening operator ∇

- Given a convex region $\mathcal{C}$
- It's windening region $\nabla(\mathcal{C})$ is constructed as follows:
  1. Construct linear constraints $\mathcal{C}'$ in $2 \times N$ dimensional space $(x_1, \ldots, x_N, y_1, \ldots, y_N)$ such that

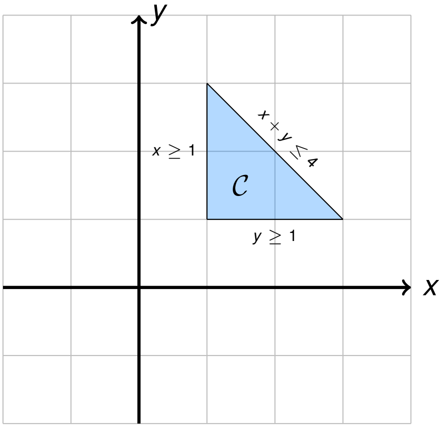
$$(y_1, \ldots, y_N) \models \mathcal{C} \quad \wedge \quad \forall i,\ x_i \leq y_i$$

  2. Remove the space dimensions higher than N in $\mathcal{C}'$.

An example of widening



An example of widening



Lemma

Lemma

Given two convex regions $\mathcal{C}_1$ *and* $\mathcal{C}_2$, $\mathcal{C}_1 \geq \mathcal{C}_2$ *if and only if* $\nabla(\mathcal{C}_1)$ *includes* $\nabla(\mathcal{C}_2)$.

Proof.

$\square$

Schedulability Analysis in System Automaton (SA-SA)

Algorithm 1 Schedulability Analysis in SA (SA-SA)

$R \leftarrow \{S_0\}$
while true **do**
    $P \leftarrow \text{Post}(R)$
    if $P \cap F \neq \emptyset$ **then**
        return NOT schedulable
    end if
    $R' \leftarrow R \cup P$
    $R' \leftarrow \text{Max}^{\succeq}(R')$
    if $R' = R$ **then**
        return schedulable
    else
        $R \leftarrow R'$
    end if
end while

Outline

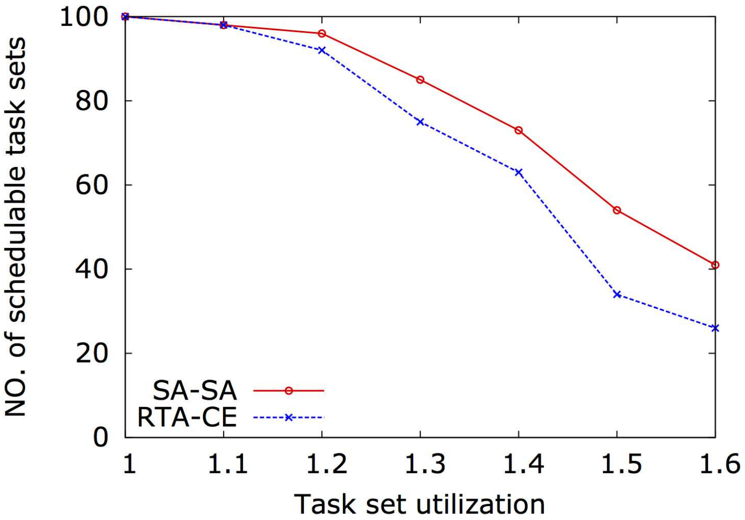
SA-SA vs. RTA-CE

- RTA-CE : Response Time Analysis with Carry-in Enumeration (Sun et al., 2014)

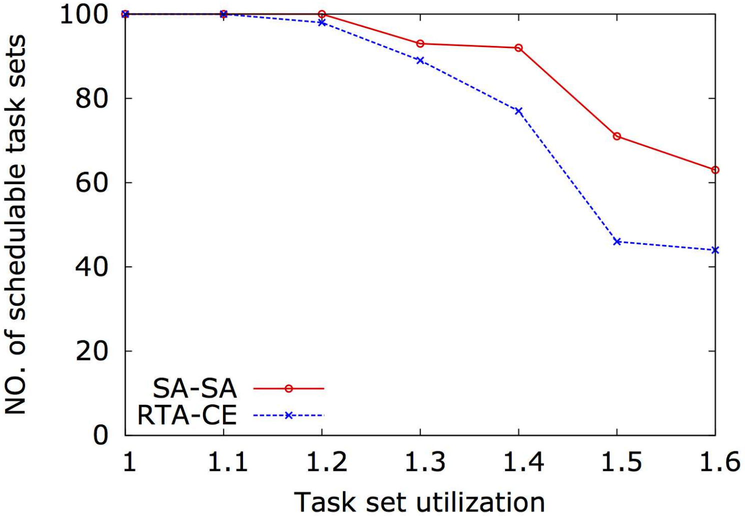
SA-SA vs. RTA-CE

- RTA-CE : Response Time Analysis with Carry-in Enumeration (Sun et al., 2014)
- $m \in \{2, 3\}$ and $n = 5$
- $T_i \in [100, 1000]$ and a series of taskset utilisation levels U seperated by 0.1
- For each (m, n, U) configuration 100 tasksets are generated by Randomfixedsum (Emberson et al., 2010)

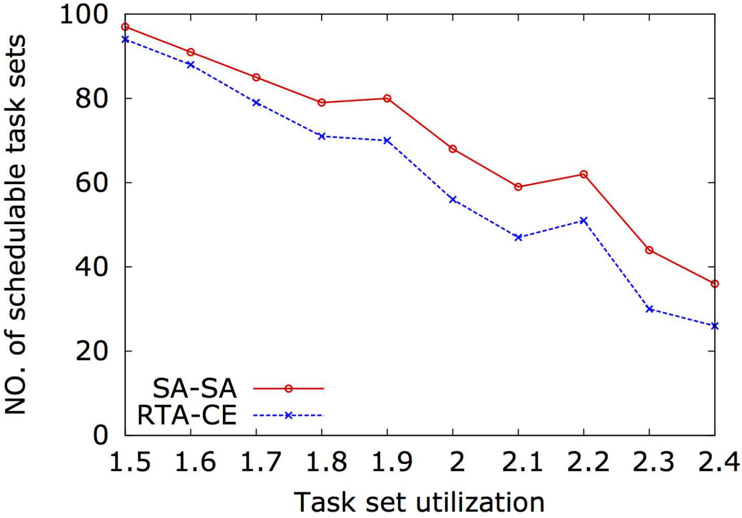
Results I: $m = 2, n = 5, \frac{D_i}{T_i} \in [0.8, 1]$



Results II: $m = 2, n = 5, \frac{D_i}{T_i} \in [0.8, 1.2]$



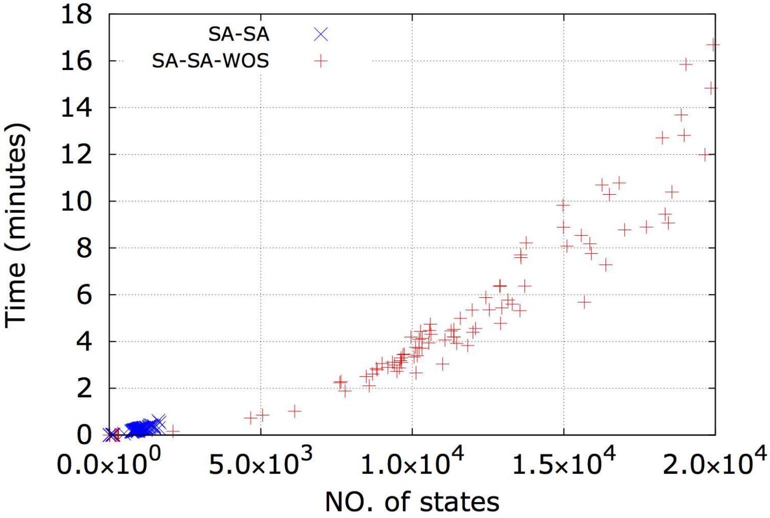
Results IV: $m = 3, n = 5, \frac{D_i}{T_i} \in [0.8, 1]$



SA-SA vs. SA-SA-WOS

- SA-SA-WOS: SA-SA WithOut Simulation
- $m = 2$, $n = 5$, $\frac{D_i}{T_i} \in [0.8, 1]$, and $U \in [1, 1.6]$
- 100 task sets

Results

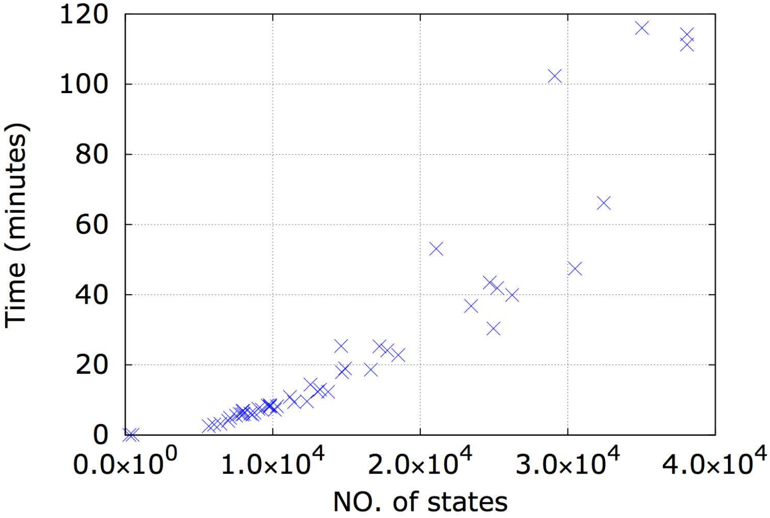


SA-SA
SA-SA-WOS
Time (minutes)
18
16
14
12
10
8
6
4
2
0
0.0×10^0
5.0×10^3
1.0×10^4
1.5×10^4
2.0×10^4
NO. of states

SA-SA: another experiment on complexity

- $m = 2$, $n = 6$, $\frac{D_i}{T_i} \in [0.8, 2]$, and $U \in [1, 2]$
- 50 task sets

Results


120
100
80
60
40
20
0
Time (minutes)
0.0×10^0
1.0×10^4
2.0×10^4
3.0×10^4
4.0×10^4
NO. of states

Questions?