Solving Temporal Problems using SMT: Strong Controllability

Alessandro Cimatti **Andrea Micheli** Marco Roveri

Embedded Systems Unit, Fondazione Bruno Kessler
Trento, Italy
amicheli@fbk.eu

12 October 2012

Constraint Programming 2012

Outline

1. The Strong Controllability Problem
2. SMT-based encodings
   - DTPU encodings
   - TCSPU specific encodings
3. Experimental Evaluation
4. Conclusion

Outline

Scheduling for planning applications

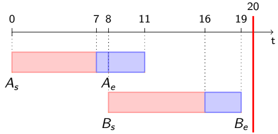
The motivating problem

Planning subject to temporal constraints, when the agent cannot control on the actual duration of all the activities.

Scheduling for planning applications

The motivating problem

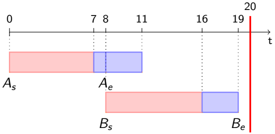
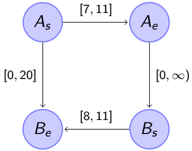
Planning subject to temporal constraints, when the agent cannot control on the actual duration of all the activities.

<table>
<tr><td></td><td colspan="2">Uncertainty Type</td></tr>
<tr><td></td><td>No Uncertainty</td><td>Uncertainty</td></tr>
<tr><td>Activities</td><td></td><td></td></tr>
<tr><td>TimePoints</td><td></td><td></td></tr>
</table>

Scheduling for planning applications

The motivating problem

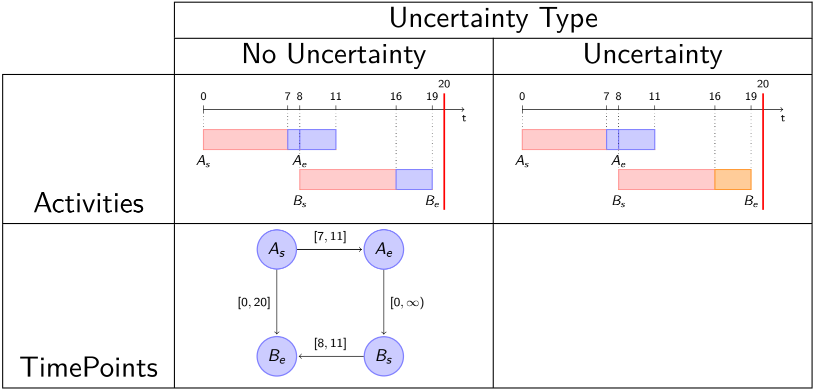
Planning subject to temporal constraints, when the agent cannot control on the actual duration of all the activities.

	Uncertainty Type	
	No Uncertainty	Uncertainty
Activities		
TimePoints		

Scheduling for planning applications

The motivating problem

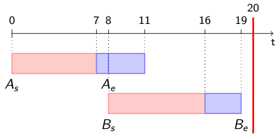
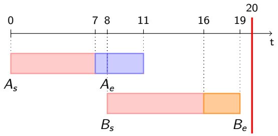
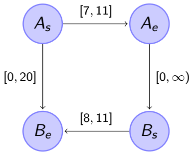
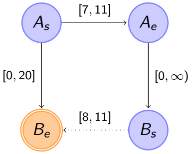
Planning subject to temporal constraints, when the agent cannot control on the actual duration of all the activities.



Scheduling for planning applications

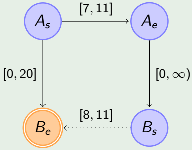
The motivating problem

Planning subject to temporal constraints, when the agent cannot control on the actual duration of all the activities.

	Uncertainty Type	
	No Uncertainty	Uncertainty
Activities		
TimePoints		

Temporal Problems with Uncertainty

Example

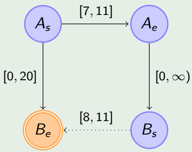


A_s, A_e, B_s are **Controllable Time Points (X_c)**
B_e is an **Uncontrollable Time Point (X_u)**

$\longrightarrow$ represents **Free Constraints (C_f)**
$\cdots\rightarrow$ represents **Contingent Constraints (C_c)**

Temporal Problems with Uncertainty

Example



A_s, A_e, B_s are **Controllable Time Points** **(**X_c**)**
B_e is an **Uncontrollable Time Point** **(**X_u**)**

$\longrightarrow$ represents **Free Constraints** **(**C_f**)**
$\cdots\rightarrow$ represents **Contingent Constraints** **(**C_c**)**

Taxonomy

Let $\{x_1, \ldots, x_k\} \doteq X_c \cup X_u$.

STPU	TCSPU	DTPU
No disjunctions $(x_i - x_j) \in [l, u]$	Interval disjunctions $(x_i - x_j) \in \bigcup_w [l_w, u_w]$	Arbitrary disjunctions $\bigvee_w ((x_{i_w} - x_{j_w}) \in [l_w, u_w])$

Strong Controllability

Intuition

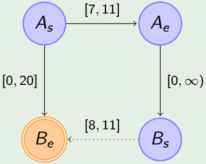
Search for a **Fixed Schedule** that fulfills all free the constraints in every situation.

Strong Controllability

Intuition

Search for a **Fixed Schedule** that fulfills all free the constraints in every situation.

Example



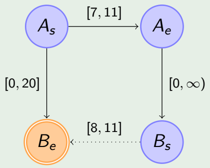
Var	Time
A_s	0
A_e	8
B_s	9

Strong Controllability

Intuition

Search for a **Fixed Schedule** that fulfills all free the constraints in every situation.

Example



Var	Time
A_s	0
A_e	8
B_s	9

Definition

A temporal problem with uncertainty is **Strongly Controllable** if

$$\exists \vec{X}_c . \forall \vec{X}_u . (C_c(\vec{X}_c, \vec{X}_u) \rightarrow C_f(\vec{X}_c, \vec{X}_u))$$

where $\vec{X}_c$ and $\vec{X}_u$ are the vectors of controllable and uncontrollable time points respectively, $C_c(\vec{X}_c, \vec{X}_u)$ are the contingent constraints and $C_f(\vec{X}_c, \vec{X}_u)$ are the free constraints.

Contributions

First comprehensive implemented solver for Strong Controllability

- Logic-based framework for Temporal Problems with Uncertainty
- Efficient encodings of Strong Controllability problems in SMT
- Extensive experimental evaluation of the approach

Outline

Satisfiability Modulo Theory (*SMT*)

SMT is the problem of deciding satisfiability of a first-order Boolean combination of theory atoms in a given theory T.

Given a formula ϕ, ϕ is satisfiable if there exists a model μ such that $\mu \models \phi$.

Satisfiability Modulo Theory (*SMT*)

SMT is the problem of deciding satisfiability of a first-order Boolean combination of theory atoms in a given theory T.

Given a formula ϕ, ϕ is satisfiable if there exists a model μ such that $\mu \models \phi$.

Example

$\phi \doteq (\forall x.(x > 0) \vee (y \geq x)) \wedge (z \geq y)$ is satisfiable in the theory of real arithmetic because

$$\mu = \{(y, 6),\ (z, 8)\}$$

is a model that satisfies ϕ.

Satisfiability Modulo Theory (*SMT*)

SMT is the problem of deciding satisfiability of a first-order Boolean combination of theory atoms in a given theory T.

Given a formula ϕ, ϕ is satisfiable if there exists a model μ such that $\mu \models \phi$.

Example

$\phi \doteq (\forall x.(x > 0) \vee (y \geq x)) \wedge (z \geq y)$ is satisfiable in the theory of real arithmetic because

$$\mu = \{(y, 6), (z, 8)\}$$

is a model that satisfies ϕ.

Theories

Various theories can be used.

In this work:

- *LRA* (*Linear Real Arithmetic*)
- *QF_LRA* (*Quantifier-Free Linear Real Arithmetic*)

Quantifier Elimination in *LRA*

Quantifier Elimination Definition

A theory T has quantifier elimination if for every formula Φ , there exists another formula Φ_{QF} without quantifiers which is *equivalent* to it (modulo the theory T)

Quantifier Elimination in *LRA*

Quantifier Elimination Definition

A theory T has quantifier elimination if for every formula Φ , there exists another formula Φ_{QF} without quantifiers which is *equivalent* to it (modulo the theory T)

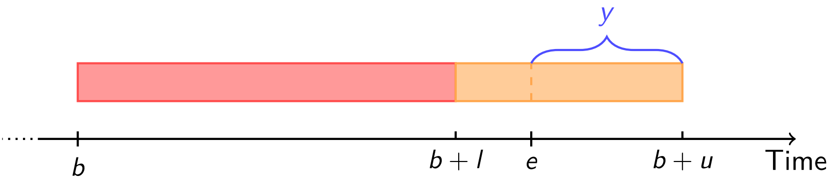
LRA theory admits quantifier elimination, but elimination algorithms are very costly (doubly exponential in the size of the original formula).

Example

$$(\exists x.(x \geq 2y + z) \wedge (x \leq 3z + 5)) \leftrightarrow (2y - 2z - 5 \leq 0)$$

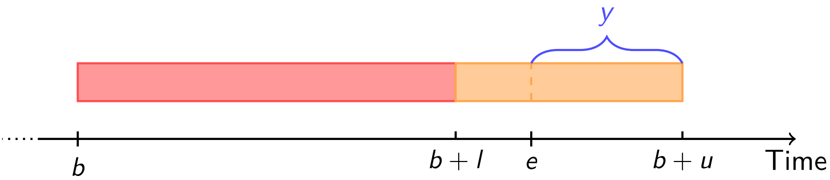
First step: Uncontrollability Isolation

Let $e \in X_u$ and $b \in X_c$.
For every contingent constraint $(e - b) \in [l, u]$, we introduce an offset $y \doteq b + u - e$.



First step: Uncontrollability Isolation

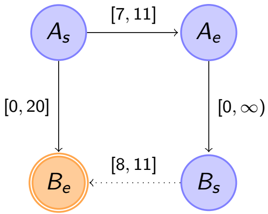
Let $e \in X_u$ and $b \in X_c$.
For every contingent constraint $(e - b) \in [l, u]$, we introduce an offset $y \doteq b + u - e$.



Definition

- Let $\vec{Y}_u$ be the offsets for a given Temporal Problem with Uncertainty
- Let $\Gamma(\vec{Y}_u)$ be the rewritten Contingent Constraints
- Let $\Psi(\vec{X}_c, \vec{Y}_u)$ the rewritten Free Constraints.

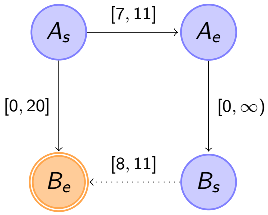
Uncontrollability Isolation: example



Original formulation

$$
\begin{aligned}
&\exists A_s, A_e, B_s.\, \forall B_e. \\
&\quad ((B_e - B_s) \in [8, 11]) \rightarrow (((A_e - A_s) \in [7, 11]) \\
&\qquad\qquad \wedge ((B_e - A_s) \in [0, 20]) \\
&\qquad\qquad \wedge ((B_s - A_e) \in [0, \infty)))
\end{aligned}
$$

Uncontrollability Isolation: example



Original formulation

$$\exists A_s, A_e, B_s. \forall B_e.$$
$$((B_e - B_s) \in [8, 11]) \rightarrow (((A_e - A_s) \in [7, 11]) \wedge ((B_e - A_s) \in [0, 20]) \wedge ((B_s - A_e) \in [0, \infty)))$$

Rewritten formulation with Y_{B_e} offset

$$\exists A_s, A_e, B_s. \forall Y_{B_e}.$$
$$(Y_{B_e} \in [0, 3]) \rightarrow (((A_e - A_s) \in [7, 11]) \wedge (((B_s + 11 - Y_{B_e}) - A_s) \in [0, 20]) \wedge ((B_s - A_e) \in [0, \infty)))$$

- $\vec{Y}_u = [Y_{B_e}]$
- $\Gamma(\vec{Y}_u) = (Y_{B_e} \in [0, 3])$
- $\Psi(\vec{X}_c, \vec{Y}_u) = (((A_e - A_s) \in [7, 11]) \wedge \ldots \in [0, \infty)))$

Outline

Direct and Naïve encodings

Direct Encoding

Strong Controllability definition is by itself an encoding in SMT(LRA)

$$\exists \vec{X}_c . \forall \vec{X}_u . (C_c(\vec{X}_c, \vec{X}_u) \rightarrow C_f(\vec{X}_c, \vec{X}_u))$$

Direct and Naïve encodings

Direct Encoding

Strong Controllability definition is by itself an encoding in SMT(LRA)

$$\exists \vec{X}_c . \forall \vec{X}_u . (C_c(\vec{X}_c, \vec{X}_u) \rightarrow C_f(\vec{X}_c, \vec{X}_u))$$

Naïve Encoding

Thanks to uncontrollability isolation, Strong Controllability can be rewritten as follows.

$$\exists \vec{X}_c . \forall \vec{Y}_u . (\Gamma(\vec{Y}_u) \rightarrow \Psi(\vec{X}_c, \vec{Y}_u))$$

Distributed Encoding

Idea: because of the cost of quantifier elimination, many small quantifications can be solved more efficiently than a big single one.

Distributed Encoding

Idea: because of the cost of quantifier elimination, many small quantifications can be solved more efficiently than a big single one.

Starting Point

We assume $\Psi(\vec{X}_c, \vec{Y}_u)$

$$\Psi(\vec{X}_c, \vec{Y}_u) \doteq \bigwedge_h \psi_h(\vec{X}_{c_h}, \vec{Y}_{u_h})$$

Distributed Encoding

Idea: because of the cost of quantifier elimination, many small quantifications can be solved more efficiently than a big single one.

Starting Point

We assume $\Psi(\vec{X}_c, \vec{Y}_u)$

$$\Psi(\vec{X}_c, \vec{Y}_u) \doteq \bigwedge_h \psi_h(\vec{X}_{c_h}, \vec{Y}_{u_h})$$

Distributed Encoding

From the Naïve Encoding we can derive a Distributed Encoding, by pushing the quantifications:

$$\exists \vec{X}_c. \bigwedge_h \forall \vec{Y}_{u_h}.(\neg\Gamma(\vec{Y}_u)|_{Y_{u_h}} \vee \psi_h(\vec{X}_{c_h}, \vec{Y}_{u_h}))$$

Eager ∀ Elimination Encoding

Idea: Starting from *Distributed Encoding*, we can eliminate quantifiers during the encoding, producing a *QF_LRA* formula.

Eager ∀ Elimination Encoding

Idea: Starting from *Distributed Encoding*, we can eliminate quantifiers during the encoding, producing a *QF_LRA* formula.

Encoding

Let

$$\psi_h^{\Gamma}(\vec{X}_{c_h}) \doteq \neg\exists \vec{Y}_{u_h}.(\Gamma(\vec{Y}_{u_h})|_{Y_{u_h}} \wedge \neg\psi_h(\vec{X}_{c_h}, \vec{Y}_{u_h}))$$

1. Resolve $\psi_h^{\Gamma}(\vec{X}_{c_h})$ for every clause independently using a quantifier elimination procedure
2. Solve the *QF_LRA* encoding:

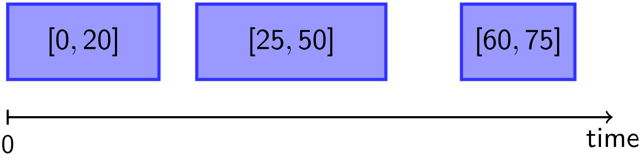
$$\exists \vec{X}_c. \bigwedge_h \psi_h^{\Gamma}(\vec{X}_{c_h})$$

Outline

Exploit *TCSPU* structure

Consider a single *TCSPU* constraint:

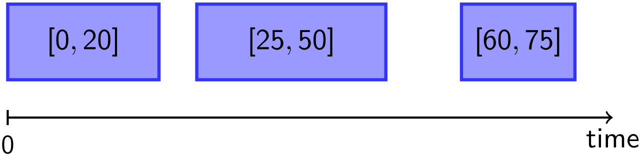
$B - A \in$



Exploit *TCSPU* structure

Consider a single *TCSPU* constraint:

$B - A \in$



Encoding TCSPU constraints in 2-CNF (Hole Encoding)

$$
\begin{aligned}
&((B - A) > 0) \\
\wedge\ &((B - A) < 20) \vee ((B - A) > 25) \\
\wedge\ &((B - A) < 50) \vee ((B - A) > 60) \\
\wedge\ &((B - A) < 75)
\end{aligned}
$$

Static quantification *TCSPU*

Idea: Exploit Hole Encoding for *TCSPU* to statically resolve quantifiers in the Eager ∀ elimination encoding.

Static quantification *TCSPU*

Idea: Exploit Hole Encoding for *TCSPU* to statically resolve quantifiers in the Eager ∀ elimination encoding.

Approach

Hole Encoding gives us a 2-CNF formula. We can enumerate all the possible (8) cases and statically resolve the quantification.

Static quantification *TCSPU*

Idea: Exploit Hole Encoding for *TCSPU* to statically resolve quantifiers in the Eager $\forall$ elimination encoding.

Approach

Hole Encoding gives us a 2-CNF formula. We can enumerate all the possible (8) cases and statically resolve the quantification.

Cases

Let $b_i, b_j \in X_c$, $e_i, e_j \in X_u$.
The only possible clauses in the Hole Encoding are in the form:

- $(b_i - b_j) \leq k$
- $(e_i - b_j) \leq k$
- $(b_i - e_j) \leq k$
- $(e_i - e_j) \leq k$
- $(b_i - b_j) \leq k_1 \vee (b_i - b_j) \geq k_2$
- $(e_i - b_j) \leq k_1 \vee (e_i - b_j) \geq k_2$
- $(b_i - e_j) \leq k_1 \vee (b_i - e_j) \geq k_2$
- $(e_i - e_j) \leq k_1 \vee (e_i - e_j) \geq k_2$

Static quantification *TCSPU* (Example)

Let $b \in X_c$, $e \in X_u$ and let y_e be the offset for e.
Let C be a hole-encoded clause of the *TCSPU* problem.

$$C \doteq (b - e) \leq u \vee (b - e) \geq l$$

Static quantification *TCSPU* (Example)

Let $b \in X_c$, $e \in X_u$ and let y_e be the offset for e.
Let C be a hole-encoded clause of the *TCSPU* problem.

$$C \doteq (b - e) \leq u \vee (b - e) \geq l$$

In the eager $\forall$ elimination encoding we have

$$\neg \exists y_e.((y \geq 0) \wedge (y \leq u_e - l_e) \wedge \\ \neg(((b - (b_e + u - y_e)) \leq u) \vee ((b - (b_e + u - y_e)) \geq l)).$$

Static quantification *TCSPU* (Example)

Let $b \in X_c$, $e \in X_u$ and let y_e be the offset for e.
Let C be a hole-encoded clause of the *TCSPU* problem.

$$C \doteq (b - e) \leq u \vee (b - e) \geq l$$

In the eager $\forall$ elimination encoding we have

$$\neg \exists y_e.((y \geq 0) \wedge (y \leq u_e - l_e) \wedge \\ \neg(((b - (b_e + u - y_e)) \leq u) \vee ((b - (b_e + u - y_e)) \geq l)).$$

The formula can be **statically** simplified

$$R \doteq ((l - b + b_e + u_e \leq 0) \vee (l - b + b_e + l_e > 0)) \wedge \\ ((l - b + b_e + l_e < 0) \vee (b - b_e - u - l_e \leq 0))$$

Static quantification *TCSPU* (Example)

Let $b \in X_c$, $e \in X_u$ and let y_e be the offset for e.
Let C be a hole-encoded clause of the *TCSPU* problem.

$$C \doteq (b - e) \leq u \vee (b - e) \geq l$$

In the eager $\forall$ elimination encoding we have

$$\neg \exists y_e.((y \geq 0) \wedge (y \leq u_e - l_e) \wedge \\ \neg(((b - (b_e + u - y_e)) \leq u) \vee ((b - (b_e + u - y_e)) \geq l)).$$

The formula can be **statically** simplified

$$R \doteq ((l - b + b_e + u_e \leq 0) \vee (l - b + b_e + l_e > 0)) \wedge \\ ((l - b + b_e + l_e < 0) \vee (b - b_e - u - l_e \leq 0))$$

Whenever a clause matches the structure of C we can derive $\psi_h^{\Gamma}(\vec{X}_{c_h})$ by substituting appropriate values for l, u, b_e, l_e and u_e in R.

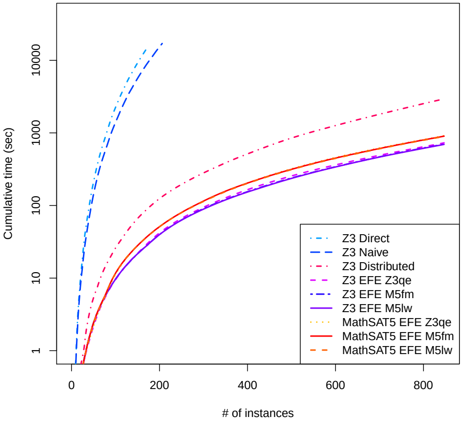
Outline

Strong Controllability Results

- Random instance generator
- *SMT* solvers:
  - Z3 (QF_LRA, LRA)
  - MathSAT5 (QF_LRA)
- Quantification techniques:
  - Z3 simplifier
  - Fourier-Motzkin
  - Loos-Weispfenning
  - Static quantification for *TCSPU*

Strong Controllability Results

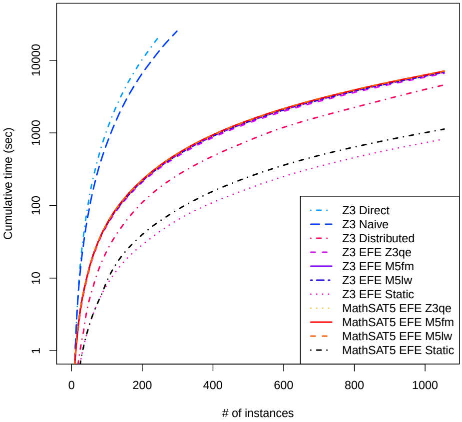
STPU Results



- Random instance generator
- *SMT* solvers:
  - Z3 (QF_LRA, LRA)
  - MathSAT5 (QF_LRA)
- Quantification techniques:
  - Z3 simplifier
  - Fourier-Motzkin
  - Loos-Weispfenning
  - Static quantification for *TCSPU*

Strong Controllability Results

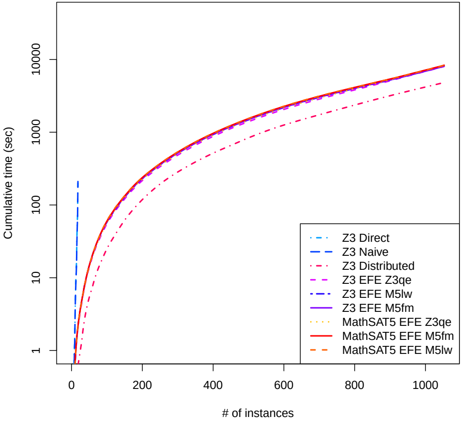
***TCSPU* Results**



- Random instance generator
- *SMT* solvers:
  - Z3 (QF_LRA, LRA)
  - MathSAT5 (QF_LRA)
- Quantification techniques:
  - Z3 simplifier
  - Fourier-Motzkin
  - Loos-Weispfenning
  - Static quantification for *TCSPU*

Strong Controllability Results

DTPU Results



- Random instance generator
- *SMT* solvers:
  - Z3 (QF_LRA, LRA)
  - MathSAT5 (QF_LRA)
- Quantification techniques:
  - Z3 simplifier
  - Fourier-Motzkin
  - Loos-Weispfenning
  - Static quantification for *TCSPU*

Outline

Conclusions

Contributions

- First comprehensive implemented solver for *DTPU* Strong Controllability
- Efficient encodings of Strong Controllability problems in SMT framework
- Tailored constant-time quantification technique for *TCSPU*
- Extensive experimental evaluation of the approach

Conclusions

Contributions

- First comprehensive implemented solver for *DTPU* Strong Controllability
- Efficient encodings of Strong Controllability problems in SMT framework
- Tailored constant-time quantification technique for *TCSPU*
- Extensive experimental evaluation of the approach

Future works

- Dynamic Controllability
- Cost function optimization
- Incrementality

Thanks

Thanks for your attention!

Bibliography

Bart Peintner, Kristen Brent Venable, and Neil Yorke-Smith. Strong controllability of disjunctive temporal problems with uncertainty. In *Principles and Practice of Constraint Programming - CP*, pages 856–863, 2007.

Thierry Vidal and Hélène Fargier. Handling contingency in temporal constraint networks: from consistency to controllabilities. *Journal of Experimental Theoretical Artificial Intelligence*, 11(1):23–45, 1999.

Backup Slides

Related work

- The strong controllability problem has been introduced in Vidal and Fargier [1999].

Related work

- The strong controllability problem has been introduced in Vidal and Fargier [1999].
- Strong Controllability of DTPUs has been theoretically tackled in Peintner et al. [2007] using Meta-CSP techniques.

Related work

- The strong controllability problem has been introduced in Vidal and Fargier [1999].
- Strong Controllability of DTPUs has been theoretically tackled in Peintner et al. [2007] using Meta-CSP techniques.
- We developed SMT-based encodings also for Weak Controllability decision problem, and a portfolio of SMT-based algorithms for strategy extraction.

Temporal Problems with Uncertainty

Definition

A Temporal Problem with Uncertainty is a tuple (X_c, X_u, C_c, C_f).

- $X_c \doteq \{b_1, ..., b_n\}$ is the set of *controllable time points*
- $X_u \doteq \{e_1, ..., e_m\}$ is the set of *uncontrollable time points*
- $C_c \doteq \{cc_1, ..., cc_m\}$ is the set of *contingent constraints*
- $C_f \doteq \{cf_1, ..., cf_h\}$ is the set of *free constraints*

$cc_i \doteq (e_i - b_{j_i}) \in [l_i, u_i]$ $\qquad$ $cf_i \doteq \bigvee_{j=1}^{D_i} (x_{i,j} - y_{i,j}) \in [l_{i,j}, u_{i,j}]$

- $j_i \in [1 \ldots n]$
- $l_i, u_i \in \mathbb{R}$
- $l_i \leq u_i$
- $l_{i,j}, u_{i,j} \in \mathbb{R} \cup \{+\infty, -\infty\}$
- $l_{i,j} \leq u_{i,j}$
- D_i is the number of disjuncts for the i-th constraint
- $x_{i,j}, y_{i,j} \in X_c \cup X_u$

Temporal Uncertainty Formalization

Uncertainty can be seen as a **game** between an *Agent* and the adversarial *Nature*.

Temporal Uncertainty Formalization

Uncertainty can be seen as a **game** between an *Agent* and the adversarial *Nature*.

Rules

- The *Agent* schedules a set of **Controllable Time Points (X_c)**

Temporal Uncertainty Formalization

Uncertainty can be seen as a **game** between an *Agent* and the adversarial *Nature*.

Rules

- The *Agent* schedules a set of **Controllable Time Points (X_c)**
- The *Agent* must fulfill a set of temporal constraints called **Free Constraints (C_f)**

Temporal Uncertainty Formalization

Uncertainty can be seen as a **game** between an *Agent* and the adversarial *Nature*.

Rules

- The *Agent* schedules a set of **Controllable Time Points (X_c)**
- The *Agent* must fulfill a set of temporal constraints called **Free Constraints (C_f)**
- The *Nature* tries to prevent the success of the agent scheduling a set of **Uncontrollable Time Points (X_u)**

Temporal Uncertainty Formalization

Uncertainty can be seen as a **game** between an *Agent* and the adversarial *Nature*.

Rules

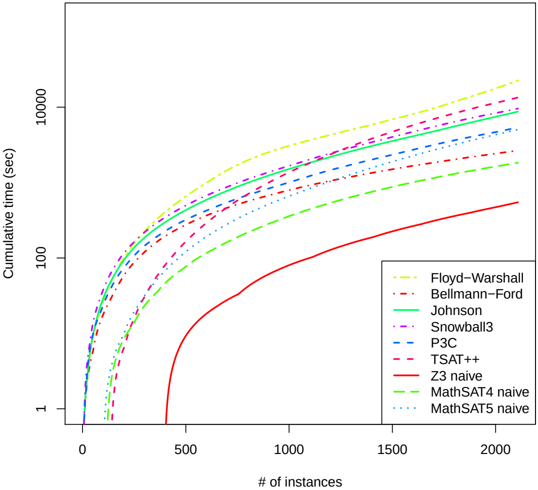
- The *Agent* schedules a set of **Controllable Time Points (X_c)**
- The *Agent* must fulfill a set of temporal constraints called **Free Constraints (C_f)**
- The *Nature* tries to prevent the success of the agent scheduling a set of **Uncontrollable Time Points (X_u)**
- The *Nature* must fulfill a set of temporal constraints called **Contingent Constraints (C_c)**

Extended Temporal Problem Taxonomy

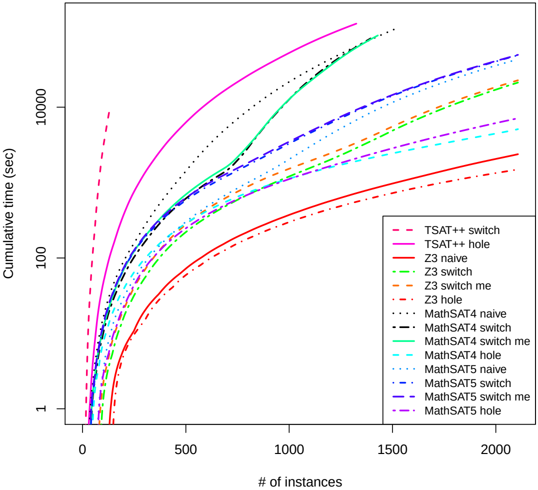
Let $\{x_1, ..., x_k\}$ be the set of all time points of the temporal problem (with uncertainty).

		Uncertainty Type	
		No Uncertainty	Uncertainty
Constraint Type	No disjunctions $(x_i - x_j) \in [l, u]$	STP	STPU
	Interval disjunctions $(x_i - x_j) \in \bigcup_w [l_w, u_w]$	TCSP	TCSPU
	Arbitrary disjunctions $\bigvee_w ((x_{i_w} - x_{j_w}) \in [l_w, u_w])$	DTP	DTPU

Consistency of *STP*



Consistency of *TCSP*



Consistency of *DTP*