Chance-Constrained Probabilistic Simple Temporal Problems

Citation: Fang, Cheng, Peng Yu, and Brian C. Williams. "Chance-Constrained Probabilistic Simple Temporal Problems." in The Twenty-Sixth Annual Conference on Innovative Applications of Artificial Intelligence (IAAI-14), July 29–31, 2014, Québec City, Québec, Canada.

As Published: http://www.aaai.org/Conferences/AAAI/2014/aaai14accepts.php

Publisher: Association for the Advancement of Artificial Intelligence (AAAI)

Persistent URL: http://hdl.handle.net/1721.1/94526

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Chance-constrained Probabilistic Simple Temporal Problems

Cheng Fang and **Peng Yu** and **Brian C. Williams** *
Massachusetts Institute of Technology
Computer Science and Artificial Intelligence Laboratory
32 Vassar Street, Cambridge, MA 02139
{cfang,yupeng,williams}@mit.edu

Abstract

Scheduling under uncertainty is essential to many autonomous systems and logistics tasks. Probabilistic methods for solving temporal problems exist which quantify and attempt to minimize the probability of schedule failure. These methods are overly conservative, resulting in a loss in schedule utility. Chance constrained formalism address over-conservatism by imposing bounds on risk, while maximizing utility subject to these risk bounds.

In this paper we present the probabilistic Simple Temporal Network (pSTN), a probabilistic formalism for representing temporal problems with bounded risk and a utility over event timing. We introduce a constrained optimisation algorithm for pSTNs that achieves compactness and efficiency through a problem encoding in terms of a parameterised STNU and its reformulation as a parameterised STN. We demonstrate through a car sharing application that our chance-constrained approach runs in the same time as the previous probabilistic approach, yields solutions with utility improvements of at least 5% over previous arts, while guaranteeing operation within the specified risk bound.

Introduction

On a Woods Hole Oceanographic Institute mission, a vehicle may be required to sample a methane plume occurring at randomly distributed times of the day. Alternatively, when scheduling for a car sharing network, the uncertain durations of traversal through traffic should be considered when determining reservation lengths for cars.

In such applications, the cost of failing to meet timing constraints is often difficult to quantify. The high investment into such operations means that we must instead provide probabilistic guarantees for timeliness, accounting for the uncertain durations. In addition to robustness against constraint violation, the desirability of schedules may also depend on the time assignments: the quality of shallow water data may depend on the collection time due to the tidal cycle, and car sharing networks may require the inactive times of the cars to be low to maximise use of assets.

Descriptions and corresponding solution methods for such problems must thus have the following characteristics. First, the description must allow the specification of an utility function to be optimised. Second, the description must recognise stochasticity in durations with a probabilistic representation. Further, the problem description must allow rich expressions of constraints. For example, we must be able to describe requirements between the timing of two uncertain events when we schedule a traversal with uncertain duration to observe natural phenomena with uncertain timing. The scheduler must thus maximise utility while providing probabilistic guarantees of compliance with requirements.

Temporal planning has been extensively studied in the operations research and artificial intelligence communities, with early works such as (Baker and Baker 1974; Allen 1983; Valdes-Perez 1986). A Simple Temporal Network (STN) (Dechter, Meiri, and Pearl 1991) is a variant of these earlier works that strikes an effective balance between tractability, expressiveness and simplicity of formulation. While there exists efficient methods for solving corresponding Simple Temporal Problems (STPs), such problems are unable to capture uncertainty in timing for events.

The Simple Temporal Network with Uncertainty (STNU) extends the STN (Vidal and Fargier 1999), introducing set bounded uncertainty. Events may be controllable, such that the timing can be scheduled. Events may also be uncontrollable, in which case there exists an unknown but set bounded difference between the timing of the uncontrollable event and the timing of an controllable event.

The STNU has also been extended with probabilistic representation of the uncertainty (Tsamardinos 2002). In the probabilistic extension, information regarding the distribution of uncontrollable events allows planning for outcomes which are more likely. However, the previous probabilistic formulation does not meet our requirements for two reasons:

1. The existing formulation takes a risk minimisation approach, leading to conservative solutions. In contrast, in real-world applications, it is common to accept a bound on the probability of failure and choose actions maximising an objective function; and

2. The existing formulation disallows constraints between two uncontrollable timepoints. We are thus unable to schedule a rendezvous between multiple agents with un-

*This work was supported by the National Science Foundation under Grant No. IIS-1017992.

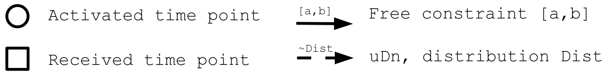


Figure 1: Graphical representations of elements in pSTNs.

controllable travel times, unable to schedule a traversal to observe an event with uncertain timing, and unable to coordinate start times of calibration procedures in time for the arrival of collected samples.

We introduce a chance-constrained probabilistic STP (cc-pSTP). Rather than requiring consistency for all outcomes of the probabilistic durations, we optimise schedules for which the probability of failure can be bounded. We develop a solution method for the cc-pSTP, with a new probabilistic representation which addresses problems involving constraints between uncertain events. We present theoretical results for the soundness of the solution, as well as empirical results for scalability, and improved utility.

Chance-Constrained pSTP

We first define the constraints and variables in our probabilistic STN (pSTN). following earlier conventions (Vidal and Fargier 1999). The definition is measure-theoretic to allow discussion of stochasticity without the details of the distributions. Thus, the discussion applies both to uncontrollable durations with independent distributions, for example the traversal time of a vehicle and the occurrence of natural phenomena, and those with joint distributions, for example two vehicles traveling along the same route at the same time.

Definition 1. *(Probabilistic STN) Let:*

- **activated** *time-points $b_i \in \mathbb{R}$ be those assigned by the agent;*
- **received** *time-points $e_i \in \mathbb{R}$ be those assigned by the external world;*
- **free constraints** *c_{xy} (**Free**) be constraints of type $(y - x) \in [l_{xy}, u_{xy}]$, where x, y are time points; and*
- **uncertain duration** *(**uDn**) $d_{xy} : \Omega \to \mathbb{R}$ be random variables describing the difference $(y - x) = d_{xy}(\omega)$, where y is a received time point and x is an activated time point, for $(\Omega, \mathcal{F}, P)$ a probability space with sample space Ω, σ-algebra $\mathcal{F}$ and measure P.*

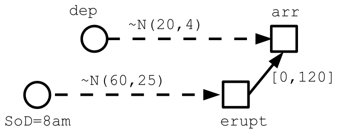
Then, $\mathcal{N}^+ = \langle X_b, X_e, R_c, R_d \rangle$ defines a pSTN, with

- *$X_b = \{b_1, ..., b_B\}$ the set of $B \in \mathbb{N}$ activated time-points;*
- *$X_e = \{e_1, ..., e_E\}$ the set of $E \in \mathbb{N}$ received time-points;*
- *$R_c = \{c_{i_1 j_1}, ..., c_{i_C j_C}\}$ the set of $C \in \mathbb{N}$ Frees; and*
- *$R_d = \{d_{i_1 j_1}, ..., d_{i_G j_G}\}$ the set of $G \in \mathbb{N}$ uDns;*

For ease of visualisation in examples, we use the convention set out in Figure . Note that consistency with respect to a Free constraint is dependent on the outcomes of uDns in addition to the choice of activated time-points. The problem is thus initially a stochastic game.

The formulation above is similar to that of (Tsamardinos 2002). The difference lies in the treatment of stochasticity. Previously, the received time points were random variables, with distributions *conditioned* on the activated time points. Instead, we characterise stochasticity by treating the probabilistic durations as random variables. By posing the problem in terms of probabilistic durations, we are able to address problems in which there exist Frees between two received time-points. This was not possible with previous representations, even though such problems are prevalent. As an example, a pSTN of an oceanographic mission is given:

Example 1. *(Example pSTN) An autonomous underwater vehicle (AUV) is tasked with taking water samples after an underwater volcanic eruption. Relative to the Start of Day at 8am, the timing of the eruption is estimated to be normally distributed, with mean 60 minutes and standard deviation 5 minutes. The vehicle departs from base to the volcano, and the traversal duration is normally distributed with mean 20 minutes and standard deviation 2 minutes. The vehicle must arrive after the eruption, but no more than 120 minutes after the event, as the chemicals diffuse in water.*



The problem is encoded as a pSTN $\mathcal{N}^+$, with:

- *$X_b = [SoD, dep]$, SoD the start of the day at 8am, and dep the meeting time;*
- *$X_e = [arr, erupt]$, arr the time of arrival at site, and $errupt$ the eruption time;*
- *$R_c = \{c_{erupt,arr}\}$, $c_{erupt,arr}$ requiring $arr - erupt \in [0, 120]$ such that the vehicle arrives in time;*
- *$R_d = \{d_{dep,arr}, d_{SoD,erupt}, \}$, $d_{dep,arr} \sim N(20, 4)$ the traversal duration, and $d_{SoD,erupt} \sim N(60, 25)$ the eruption time relative to start of day;*

We consider now how pSTNs are used to frame temporal problems whose solutions incur bounded risk of failure. We return to the car sharing problem to motivate the need for bounded risk. We wish to schedule such that all reservations with the vehicle are completed as early as possible, so that maintenance could be performed at a depot. However, consider the effect of one late handover of the vehicle between reservations. The next user must adjust for the loss in allocated time, or risk returning the car late as well. A late return, a very specific instance of temporal inconsistency, may thus be considered a failure. We must thus schedule for the reservations to be completed as early as possible, while avoiding late handovers of the vehicle.

Due to the possibly unbounded range of stochastic outcomes, we may not be able to schedule such that temporal consistency is guaranteed for *all* outcomes. The approach in (Tsamardinos 2002) minimises the risk of temporal inconsistency. However, such an approach would not consider the time at which the vehicle is returned to the depot, and this over-conservatism would mean the vehicles could be out indefinitely. In order to return the vehicles as early as possible, we should accept a bounded probability of late returns, and optimise the return times.

In general, we would wish to optimise the schedule with respect to an *objective function*, subject to *bounds on the probability of temporal inconsistency*: we try to make the best schedule, tolerating rare cases in which the schedule can not be met. This motivates a way to measure the riskiness of a schedule with respect to temporal constraints as follows:

Definition 2. *(pSTN schedule risk) Consider pSTN $\mathcal{N}^+ = \langle X_b, X_e, R_c, R_d \rangle$. Let a* **schedule** *$S \in \mathbb{R}^B$ be a full assignment of values to the controllable time-points X_b.*

The **risk** *of S with respect to some subset of the constraints $A_m \subseteq R_c$ is:*

$$r_{A_m}(S) = P(\Omega_{A_m,S})$$

where $\Omega_{A_m,S} \subseteq \Omega$ is the subset of the sample space such that for every $c_{ij} \in A_m$, if $\omega \in \Omega$ and $x_j(\omega) - x_i(\omega) \notin [l_{ij}, u_{ij}]$, then $\omega \in \Omega_{A_m,S}$.

For any schedule, the risk of the schedule is defined as the proportion of outcomes of the probabilistic durations which result in *at least one free constraint being violated*.

We define chance-constrained pSTP (cc-pSTP), a problem formulation using the pSTN representation. A solution to a cc-pSTP is a schedule which optimises the timing of events with respect to an objective function, while bounding the probability of temporal inconsistency. Objective functions in optimal scheduling, for example (Khatib et al. 2001; Rossi, Venable, and Yorke-Smith 2006), may be used with our algorithm provided that they are continuously differentiable. We define a cc-pSTP as follows.

Definition 3. *(Chance-constrained pSTP)*
Given:

- $\mathcal{N}^+ = \langle X_b, X_e, R_c, R_d \rangle$*, a pSTN;*
- $\Delta_t \in [0, 1]$*, an upper bound on the risk of failure, for the set R_c of Frees; and*
- $V : \mathbb{R}^B \to \mathbb{R}$*, an objective function dependent on assignments to X_b;*

Find:

- $S_B^* \in \mathbb{R}^B$*, a schedule of X_b minimising V;*

Subject to:

- $r_{R_c}(S_B^*) \leq \Delta_t$*, the probability of inconsistency bounded by Δ_t;*

We construct the cc-pSTP for the AUV scenario below:

Example 2. *(Example cc-pSTP) Recall Example 1, with $\mathcal{N}^+$ as before. The departure should be as early as possible, to ensure availability for later missions. We thus set objective $V = dep$, to minimise the time of departure. However, it should neither arrive early nor be more than 2 hours late. The scientist has required the schedule to satisfy both constraints in at least 99% of cases. Thus we set Δ_t to be 0.01.*

In general, we may use arbitrary distributions for the uDns, as long as we are able to evaluate the cumulative distribution functions (cdfs) for each uDn. We do not *require* a description of the joint distribution. This will be elaborated when we discuss the solution method. In this paper, we assume continuously differentiable cdfs (e.g. normal, exponential etc.), allowing the use of existing solvers.

We have thus formulated a problem involving probabilistic durations, in which we attempt to schedule events to optimise an objective function while providing probabilistic guarantees on the satisfaction of free constraints. We show how the cc-pSTP may be solved in the next section.

Solution of cc-pSTPs

Given the problem statement above, we explain the solution method for cc-pSTPs. Intuitively, we choose the *most probable* set of outcomes of the uDns, and schedule activated time points such that the timing constraints are satisfied for any combination of outcomes in this restricted set. The key insight behind our approach is that *by distributing the allowable risk amongst the uncertain durations uDns, we can set-bound the outcomes*. This allows us to convert a pSTN into a simple temporal network with uncertainty (STNU), a well studied structure. We can then make use of results for STNUs to reframe a cc-pSTP as a convex constrained optimisation problem solvable with standard optimisers.

Related work

We begin by noting that a cc-pSTP is an instance of a stochastic optimisation problem. A family of methods reformulate the stochastic problem to a deterministic problem by converting the stochastic constraints into deterministic constraints. For example, optimising controls for stochastic dynamical systems involves bounding the extent of deviations from the mean, either by distributing the risk evenly (Van Hessem and Bosgra 2006) or by optimising the distribution of risk (Ono, Williams, and Blackmore 2013). The process results in bounds on the state of the system, leading to reformulations as deterministic constraint optimisation problems.

We draw inspiration from this literature because these methods allow probabilistic guarantees. Alternatives include sampling-based methods in the same vein as the Pegasus POMDP method (Ng and Jordan 2000), using particles to simulate the effects of disturbances and initial uncertainty with applications in control and robotics (Blackmore et al. 2010). However, the assessment of risk in such methods only *converges in the limit* as the number of particles increase - they can not guarantee bounds on the probability of failure.

We further note that the temporal reasoning community has developed a set bounded approach leveraging the structure of simple temporal problems. The STNU solution methods offer computational efficiency and correctness guarantees, and we leverage these to efficiently solve pSTNs.

Though not probabilistic, STNU is similar to the pSTN. Uncertainty is represented with set-bounded **contingent durations (Ctg)** $g_{xy} \in [l_{xy}, u_{xy}]$. These are differentiated from Frees because they are *variables* with values *assigned by the environment*. Ctgs describe the relationship of $(y - x)$, where y is a received time point and x is an activated time point. A STNU is defined as $\mathcal{N} = \langle X_b, X_e, R_c, R_g \rangle$ with X_b, X_e, R_c as in pSTN and R_g the set of Ctgs such that $R_g = \{g_{i_1 j_1}, ..., g_{i_G j_G}\}$ for some $G \in \mathbb{N}$.

A rich set of methods exist for offline, online and incremental solutions to STNU controllability, e.g. (Vidal and Fargier 1999; Morris 2006; Shah et al. 2007) as well as the disjunctive extensions (Peintner, Venable, and Yorke-Smith 2007; Shah and Williams 2008; Conrad, Shah, and Williams 2009; Venable et al. 2010). By mapping underlying pSTNs into STNUs, following the paradigm in robust stochastic programming, we leverage the literature on offline robust scheduling of STNUs. We do so by distributing the allowable risk. This allows us to consider only a subset of outcomes for the uDns, represented by Ctgs. We then optimise a schedule with respect to the utility, while remaining robust to the restricted set of outcomes.

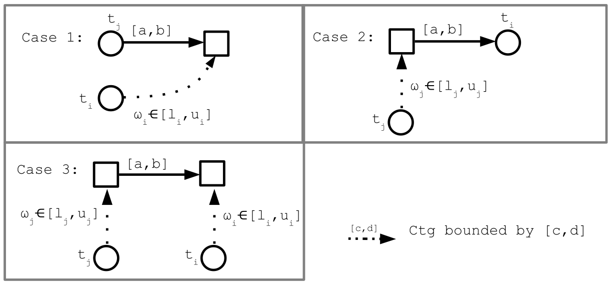

Case 1:
t_j
[a,b]
t_i
$\omega_i \in [l_i, u_i]$
Case 2:
[a,b]
t_i
$\omega_j \in [l_j, u_j]$
t_j
Case 3:
[a,b]
$\omega_j \in [l_j, u_j]$
$\omega_i \in [l_i, u_i]$
t_j
t_i
[c,d]
Ctg bounded by [c,d]

Figure 2: Three cases for reductions.

Solution method

In our approach, we make use of the idea of *strong controllability* for STNUs (Vidal and Fargier 1999). Informally, if a STNU is strongly controllable, then there exists an schedule which is consistent for all outcomes of the contingent durations. Formally:

Definition 4 (Strong Controllability)**.** *Consider STNU* $\mathcal{N} = \langle X_b, X_e, R_c, R_g \rangle$*. Let* $\Omega_g = [l_{i_1 j_1}, u_{i_1 j_1}] \times [l_{i_2 j_2}, u_{i_2 j_2}] \times ... \times [l_{i_G j_G}, u_{i_G j_G}]$*, the space of possible values for the set of contingent durations Ctgs in* R_g*.*

We say that $\mathcal{N}$ *is* **strongly controllable** *if there exists a schedule* S_B *such that for any* $\omega_g \in \Omega_g$*, all constraints in* R_c *are satisfied.*

To determine strong controllability of STNUs, frees are introduced between activated time points to replace frees involving received time points (Vidal and Fargier 1999). The new constraints allow us to reason without dealing directly with the received time points. The problem is reduced from a game against an uncooperative environment to a constrained optimisation problem solvable with existing packages.

We repeat the general reductions from (Vidal and Fargier 1999) to develop our algorithm. Consider STNU $\mathcal{N} = \langle X_b, X_e, R_c, R_g \rangle$. Let $R_c^- \subseteq R_c$ be the set of Frees involving a received time point. These are the only constraints which depend on variables not controlled by the agent, and thus need to be reframed.

We consider the lower and upper bounds separately and first perform our analysis for lower bounds. For each lower bound $c \in R_c^-$, we obtain one of three cases in Figure :

Case 1 (only the end timepoint is a received time point)

$$t_i + \omega_i - t_j \geq a$$

Case 2 (only the start timepoint is a received time point)

$$t_i - t_j - \omega_j \geq a$$

Case 3 (both start and end are received time points)

$$t_i + \omega_i - t_j - \omega_j \geq a$$

where ω_i is the duration of the Ctg starting at time point t_i and bounded by $[l_i, u_i]$, with ω_j similarly defined.

In each case, we replace the original Free with a new Free:

Case 1 $t_i - t_j \geq a - l_i$

Case 2 $t_i - t_j \geq a + u_j$

Case 3 $t_i - t_j \geq a - l_i + u_j$

The corresponding reductions for the upper bounds can be written similarly by multiplying both sides by -1.

Let $\hat{R}_c^-$ be the collection of Frees obtained from the reduction. The importance of the reductions lies in the result, proved in (Vidal and Fargier 1999):

Algorithm 1: Approximating cc-pSTP

```
input  : $\mathcal{N}^+ = \langle X_b, X_e, R_c, R_d \rangle$
output : $X_r$ bounds on $R_d$, $F$ chance-constraint function,
         and $\hat{R}_c^-$ reductions
$X_r \leftarrow \emptyset$, $\hat{R}_c^- \leftarrow \emptyset$;
for each $d_{xy} \in R_d$ do
    $X_r \leftarrow [X_r; l_{xy}; u_{xy}]$ ;
    $F \leftarrow \big[F; F_{d_{xy}}; 1 - F_{d_{xy}}\big]$;
for each $c_{xy} \in R_c$ do
    $a, b \leftarrow$ lower and upper bounds for $c_{xy}$ respectively;
    if $x \in X_e \wedge y \in X_e$ then
        Let $d_{t_i x}, d_{t_j y}$ be uDns ending in $x$ and $y$, $[l_x, u_x]$
        and $[l_y, u_y]$ corresponding bounds on uDns;
        $\hat{R}_c^- \leftarrow \hat{R}_c^- \cup \{t_j - t_i + l_y - u_x \geq a\}$;
        $\hat{R}_c^- \leftarrow \hat{R}_c^- \cup \{-t_j + t_i - u_y + l_x \geq -b\}$;
    else if $y \in X_e$ then
        Let $d_{t_j y}$ be uDn ending in $y$, $[l_y, u_y]$ corresponding
        bounds on uDns ;
        $\hat{R}_c^- \leftarrow \hat{R}_c^- \cup \{t_j - x + l_y \geq a\}$;
        $\hat{R}_c^- \leftarrow \hat{R}_c^- \cup \{-t_j + x - u_y \geq -b\}$;
    else if $x \in X_e$ then
        Let $d_{t_i x}$ be uDn ending in $x$, $[l_x, u_x]$ corresponding
        bounds on uDns;
        $\hat{R}_c^- \leftarrow \hat{R}_c^- \cup \{y - t_i - u_x \geq a\}$;
        $\hat{R}_c^- \leftarrow \hat{R}_c^- \cup \{-y + t_i + l_x \geq -b\}$;
```

Theorem 1. *If there exists a schedule* S_B *satisfying all constraints in* $(R_c \setminus R_c^-) \cup \hat{R}_c^-$*, then the STNU is* strongly controllable*. Further, given any combination of outcomes for* R_g *the set of contingent durations,* S_B *is consistent with respect to all elements in* R_c *the set of Free constraints.*

Theorem 1 allows robust scheduling. If the uncertain durations are set-bounded and the corresponding STNU is strongly controllable, then we can schedule the activated time points to be temporally consistent for all outcomes of the uncertain durations. In general, we need to set the bounds on the uDns ourselves. The process is analogous to deciding the sets of scenarios for which we must prepare.

Recall that in a cc-pSTP, we have an upper bound for the probability of temporal inconsistency. We distribute this allowable risk over the set of uDns. For each uDn, we then consider a subset of its possible outcomes according to the risk allocated, turning each uDn into a corresponding Ctg. This gives us a STNU to check for strong controllability.

For an intuition, start by considering the uDns in the pSTP. For any uDn d_i, we choose a set-bound for the outcome of the uDn, and treat it as a contingent constraint g_i. For example, consider uDn $d \sim N(2, 1)$. With 95% probability, the outcome lies in interval $[0, 4]$. If we found assignments to activated time points consistent for all outcomes in $[0, 4]$, we know the temporal constraints will be satisfied in 95% of cases if d is the only uDn in the pSTN.

We thus derive a system of constraints for a cc-pSTP, summarised in Algorithm 1. The first for-loop of Algorithm 1 adds two decision variables denoting lower and upper bounds for every uDn, and collects the cdf $F_{d_{xy}}$ associated with each uDn. The cdfs are used to calculate the probability mass lost by restricting the outcome of the uDns. Note that both the cdf evaluating the probability mass discarded by the lower bound and the complement cdf evaluating that discarded by the upper bound are recorded.

In the second for-loop, the algorithm applies reductions to the uDns in the pSTN. The reductions are based on those for strong controllability. However, instead of fixed lower and upper bounds, the reductions performed allow the lower and upper bounds to be decided by the solver.

The reductions are similar to those proposed in (Tsamardinos 2002). The key innovation is accounting for free constraints between two received time points, disallowed in previous work. By representing stochasticity with uncertain durations, we can naturally map from the STNU structure and transcribe Case 3 in a probabilistic context.

We can now define an approximate cc-pSTP with the new decision variables and constraints from Algorithm 1.

Definition 5. *(Approximate cc-pSTP) We solve an approximate cc-pSTP as follows:*

Given:

- $\mathcal{N}^+$, $\Delta_t \in [0,1]$, *and* V *as in Definition 3 ; and*
- X_r, F, *and* $\hat{R}_c^-$ *from Algorithm 1*

Find:

- $S_B^* \in \mathbb{R}^B$, *schedule to* X_b *minimising* V*; and*
- $LU \in \mathbb{R}^{2G}$ *lower and upper bound values on uDns*

Subject to:

- $\sum_{F_i \in F} F_i(LU_i) < \Delta_t$, F_i *and* LU_i *the* i^{th} *entries in* F *and* LU *respectively; and*
- S_B^* *and* LU *satisfying constraints* $\left(R_c \setminus R_c^-\right) \cup \hat{R}_c^-$, R_c^- *subset in* R_c *involving received times*

In the approximate cc-pSTP, we deal with the difficulty in bounding the risk of a schedule through the reductions from Algorithm 1. In addition to a schedule to the activated time points, we are required to choose the lower and upper bounds for the uDns. The choice of the schedule and the bounds must satisfy two sets of constraints:

- The chance constraint, ensuring the choice of bounds for uDns do not eliminate too much probability mass. We sum the probability mass beyond the bounds for each uDn, and apply Boole's Inequality to restrict the probability of any uDn outcomes outside our restricted set. Note that Boole's Inequality does not require independence between uDns: this is why we only require the cdfs for each uDn.
- The reduction constraints, enforcing strong controllability when uDns are restricted to set-bounded intervals. The constraints ensure that the bounds for uDns and the solution schedule is chosen together such that, for any outcome of uDns in the restricted intervals, the solution schedule will be valid with respect to the free constraints.

Example 3. *(Approximate solution to cc-pSTP) For Example 2, we bound* $d_{dep,arr}$ *and* $d_{SoD,erupt}$ *and find assignment to dep such that the Free constraint is satisfied. By encoding the problem as a cc-pSTP and solving with SNOPT (Gill, Murray, and Saunders 2005), the bounds are respectively* $[14.421, 29.747]$ *and* $[36.282, 72.196]$*, such that the total probability mass discarded is 1%. The departure is accordingly scheduled for 57.775 minutes after the start of day.*

For the approximate cc-pSTP, we have the result below:

Theorem 2. *Let* S_B^* *be the schedule found by solving the approximate cc-pSTP. Then* $r_{R_c}(S_B^*) \leq \Delta_t$.

Intuitively, a schedule returned by solving the approximate problem will be temporally inconsistent with probability at most Δ_t. A schedule returned by solving the approximate problem is thus also a schedule which satisfies the chance constraint in the original problem.

We present the intuition behind the proof. Observe that a solution schedule to the approximate cc-pSTP is valid for any outcomes of the uDns lying in the selected bounds. Further, the probability mass lost by selecting the particular bounds is less than that allowed by the chance constraint. Thus, the cases in which the solution schedule is not valid lies in the space outside the bounds, which have probability less than Δ_t in the original problem.

Note that Boole's inequality *bounds* the probability of failure, and thus some conservatism is introduced. The solutions are thus not guaranteed to be optimal. Even so, we show empirically that the approach still produces significant improvements in utility over the existing state of the art.

The approximate cc-pSTP is in a form solvable with existing solvers. The reductions comprise the majority of the constraints, and are linear over decision variables S_B^* and LU. The only source of nonlinearity is the chance-constraint.

Experiment and results

In this section, we perform benchmarks on how the difficulty of finding solutions changes with increasing number of uncertain durations and tighter chance-constraints. We further check the correctness of the feasible solutions with respect to the probabilistic guarantees, as well as the quality of solutions with respect to the utility function. Lastly, we examine how the computational runtime scales with increasing number of uncertain durations.

We benchmark our algorithms on scenarios inspired by car sharing, similar to Zipcars (Burkhardt and Millard-Ball 2006). In each scenario we schedule for a 6 hour period, with the number of cars ranging from 1 to 20, each with up to 5 users. For each user, up to three goal locations were generated based on a simplified open source map of Boston.

A pSTN was generated for each scenario. The traversal activities were modelled as normally distributed uncertain durations, with the means of uDns determined by length and speed limits of the roads taken, and standard deviations at 5% of the mean. A total of 1800 pSTNs were generated. To allow comparison to prior art, we ensured that no Free constraints existed between received time-points, as such problems are not handled by previous methods.

For each pSTN, we constructed three cc-pSTPs, with chance-constraints 10%, 20% and 40%. We want to complete all activities as soon as possible, and thus set the time of the last reservation as the objective function in each case.

The cc-pSTPs were solved by three methods: a) evenly distributing the risk analogous to (Van Hessem and Bosgra

		Number of activities to schedule			
		≤ 10	$11-20$	$21-30$	>31
Risk 10%	Even distribution solutions	0	0	0	0
	cc-pSTP solutions	143	16	1	1
Risk 20%	Even distribution solutions	0	0	0	0
	cc-pSTP solutions	146	17	1	1
Risk 40%	Even distribution solutions	28	0	0	0
	cc-pSTP solutions	151	19	1	1
	Risk minimisation solutions	161	22	2	1
	Total number of Scenarios	428	230	165	977

Table 1: Solutions found for different parameters

Method	$P(Success)$ $(\pm 1 - \sigma)$
10% cc-pSTP	0.9012 ± 0.0018
20% cc-pSTP	0.8059 ± 0.0051
40% cc-pSTP	0.6250 ± 0.0198
Min. Risk	0.9372 ± 0.1801

Table 2: Empirical verification of correctness of solution.

2006); b) by our risk allocation method; and c) by the risk minimising method (Tsamardinos 2002) for comparison. Solutions were obtained with SNOPT (Gill, Murray, and Saunders 2005), a nonlinear optimisation solver designed for problems with a large number of linear constraints. Table summarises the difficulty of finding solutions for different numbers of activities.

As expected, the proportion of feasible problems decreases as the number of activities increase. The same amount of risk must be shared amongst a higher number of activities. Thus, as the number of uncertain durations increase, we must be more cautious when bounding each uncertain duration. We must thus consider a larger subset of outcomes, making robust scheduling harder.

Note that even distribution of risk results in almost no solutions. The risk minimisation method has the largest number of solutions: if a solution exists, it will be found regardless of the risk of the solution. Risk allocation find a comparable number of solutions because there is flexibility in how the uncertain durations are bounded, restricted only by the chance-constraint.

The soundness of solutions with respect to the chance-constraints were tested via Monte Carlo sampling. For each, 50000 samples of the joint outcomes of the uDns were tested for consistency with Free constraints, given the assignments to activated times. Table summarises results for the chance-constrained method and the risk minimisation method. Note that the chance-constrained solutions were correct, whereas the variance of the risk minimisation method means no guarantees on the probability of success can be provided for its solutions.

The flip side of robustness is conservatism. For scenarios where solutions are found via the chance-constrained methods and the risk minimisation methods, we compare the utility of solution. On average, the 10%, 20%, and 40% cc-pSTP schedules resulted in last activated time point occurring respectively 5.37%, 6.58% and 6.82% earlier than the risk-minimisation methods. These represent significant savings over the risk-minimisation method, which is too conservative in achieving robustness.

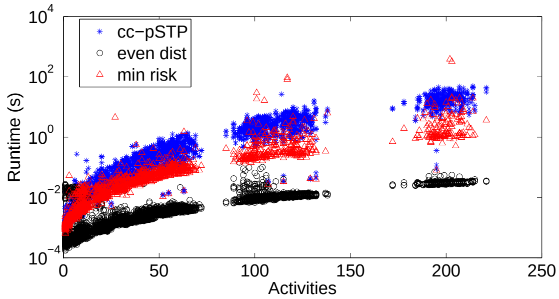


Figure 3: Computation time as a function of the number of activities in the scenarios.

Tests were also performed on scalability, with results summarised in Figure . The runtimes for risk allocation cc-pSTP empirically scale in polynomial time with the increasing number of constraints, although even problems with over 200 activities took less than 90 seconds with a 2.4GHz processor. The risk minimisation method scales similarly to the cc-pSTP method, although the outliers take significantly longer. The polynomial complexity is due to the use of SNOPT for the reduced problems: the sequential quadratic programming method solves a series of quadratic programs, each of which is polynomial time in the number of variables.

The empirical validation confirms the soundness of the cc-pSTP with respect to the chance constraint. Further, the results show that the solution method scales well in time for relatively complicated problems. Lastly, it confirms that, by accepting varying levels of risk, we can derive better solutions than purely risk-averse behaviour.

Contributions

Robust scheduling is crucial in deployable systems. Previous work focused on purely risk averse scheduling, leading to unnecessary conservatism. In this paper, we defined the pSTN structure, as an alternative generalisation of STPUs to that proposed in (Tsamardinos 2002). We further identified the need for a chance-constrained rather than risk minimisation approach to robust execution of pSTNs. By analysis with the new pSTN structure, we leveraged existing work in the STPU literature to provide solution method for static scheduling of cc-pSTNs. We empirically validated the soundness of the method with respect to the chance constraints on real world inspired-problems, and demonstrated the extra utility gained by the approach over the risk-minimisation approach.

References

Allen, J. F. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM* 26(11):832–843.

Baker, K. R., and Baker, K. R. 1974. *Introduction to sequencing and scheduling*, volume 31. Wiley New York.

Blackmore, L.; Ono, M.; Bektassov, A.; and Williams, B. C. 2010. A probabilistic particle-control approximation of chance-constrained stochastic predictive control. *Robotics, IEEE Transactions on* 26(3):502–517.

Burkhardt, J. E., and Millard-Ball, A. 2006. Who is attracted to carsharing? *Transportation Research Record: Journal of the Transportation Research Board* 1986(1):98–105.

Conrad, P. R.; Shah, J. A.; and Williams, B. C. 2009. Flexible execution of plans with choice. In *ICAPS*, volume 9, 74–81.

Dechter, R.; Meiri, I.; and Pearl, J. 1991. Temporal constraint networks. *Artificial intelligence* 49(1):61–95.

Gill, P. E.; Murray, W.; and Saunders, M. A. 2005. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM review* 47(1):99–131.

Khatib, L.; Morris, P.; Morris, R.; Rossi, F.; et al. 2001. Temporal constraint reasoning with preferences. In *IJCAI*, 322–327.

Morris, P. 2006. A structural characterization of temporal dynamic controllability. In *Proceedings of the 12th International Conference on Principles and Practice of Constraint Programming (CP-2006)*, 375–389.

Ng, A. Y., and Jordan, M. 2000. Pegasus: A policy search method for large mdps and pomdps. In *Proceedings of the Sixteenth conference on Uncertainty in Artificial Intelligence*, 406–415. Morgan Kaufmann Publishers Inc.

Ono, M.; Williams, B. C.; and Blackmore, L. 2013. Probabilistic planning for continuous dynamic systems under bounded risk. *Journal of Artificial Intelligence Research* 46:511–577.

Peintner, B.; Venable, K. B.; and Yorke-Smith, N. 2007. Strong controllability of disjunctive temporal problems with uncertainty. In *Principles and Practice of Constraint Programming–CP 2007*. Springer. 856–863.

Rossi, F.; Venable, K. B.; and Yorke-Smith, N. 2006. Uncertainty in soft temporal constraint problems: a general framework and controllability algorithms for the fuzzy case. *Journal of Artificial Intelligence Research* 27(1):617–674.

Shah, J., and Williams, B. 2008. Fast dynamic scheduling of disjunctive temporal constraint networks through incremental compilation. In *Proceedings of the International Conference on Automated Planning and Scheduling*.

Shah, J. A.; Stedl, J.; Williams, B. C.; and Robertson, P. 2007. A fast incremental algorithm for maintaining dispatchability of partially controllable plans. In *ICAPS*, 296–303.

Tsamardinos, I. 2002. A probabilistic approach to robust execution of temporal plans with uncertainty. *Methods and Applications of Artificial Intelligence* 751–751.

Valdes-Perez, R. 1986. Spatio-temporal reasoning and linear inequalitites. *AI Memo*.

Van Hessem, D., and Bosgra, O. 2006. Stochastic closed-loop model predictive control of continuous nonlinear chemical processes. *Journal of Process Control* 16(3):225–241.

Venable, K. B.; Volpato, M.; Peintner, B.; and Yorke-Smith, N. 2010. Weak and dynamic controllability of temporal problems with disjunctions and uncertainty. In *Workshop on Constraint Satisfaction Techniques for Planning & Scheduling*, 50–59.

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