On Expected Value Strong Controllability

Abstract

The Probabilistic Simple Temporal Network (PSTN) generalizes Simple Temporal Networks with Uncertainty (STNUs) by introducing probability distributions over the timing of uncontrollable timepoints. PSTNs are controllable if there is a strategy to execute the controllable timepoints while bounding the risk of violating any constraint to a small value. If this risk bound can't be satisfied, PSTNs are not considered controllable. We introduce the Expected Value Probabilistic Simple Temporal Network (EPSTN), which extends PSTNs by including a benefit to the satisfaction of temporal constraints. We study the problem of Expected Value Strong Controllability (EvSC) of EPSTNs, which seeks a schedule maximizing the expected value of satisfied constraints. We solve the EvSC problem by extending a previously developed linear program, combined with search over constraints to violate at execution time. We describe conditions under which the solution to this linear program is the maximum expected value schedule. We then show how to search for constraints to discard, using the linear program at the core of the search. While the general problem is shown to be exponential, we conclude by providing several methods to bound the complexity of search.

1 Introduction

Since its introduction by (Vidal and Ghallab 1996) and (Vidal and Fargier 1999), there has been considerable research in the area of *controllability* of temporal networks in the presence of *uncertainty*. Controllability asks: can events be scheduled to satisfy temporal constraints in the presence of uncertain outcomes? Many previously studied solutions to this problem use the notion of controllability of Simple Temporal Networks under Uncertainty (STNUs) at their core. The solutions to such problems are *strategies* to execute all events that ensure no constraints are violated, regardless of the outcomes of previously observed uncertain events. More complex problems combine temporal constraints, uncertainty, and *preferences*. Tractability of these problems is ensured due to the simplicity of the constraints and preferences provided as inputs. Uncertainty can be generalized so that algorithms must handle *probabilities* over when events occur. This leads to new *risk-bounded* and *chance-constrained* problems. If the solution is still unsatisfactory, these constraints can be *relaxed* using a cost functions on the constraints and risk bound.

What happens when it is *almost certain* that a constraint will be violated? Current approaches, particularly risk-bounding, do not adequately address this problem. If the risk bound is too low, no strategy can be produced. If a risky strategy is produced, then an undesirable outcome at execution time will cause the executive to 'freeze' when a constraint is violated. When the existing set of constraints cannot be satisfied to produce control strategies, relaxing some constraints up-front may ensure controllability, but with a loss of insight into the original problem.

An alternative solution to such over-constrained problems is to let the execution strategy try to satisfy as many constraints as possible, assuming that at least one constraint will be violated during execution. If some constraints are more important than others, then a natural optimization criteria for the strategy is to maximize the *expected value* of satisfied constraints. This new, unexplored problem blends several notions explored in the controllability literature to date. Accepting risk implies accepting outcomes that violate some constraints. Applying preferences to satisfied constraints suggests control of expected schedule quality based on past information and the probability and cost of future constraint violations.

In this paper we define the Expected Value Probabilistic Simple Temporal Network EPSTN, and begin identifying algorithms to solve this problem. We formalize the problem of finding a schedule maximizing the expected value of satisfied constraints, the Expected Value Strong Controllability (EvSC) problem. We adapt algorithms from the controllability literature that can solve this problem, and provide soundness and completeness results. While the general problem is shown to be exponential, we conclude by providing several methods to bound the complexity of search.

2 Notation and Definitions

Definition 1 (STNU). *(Vidal and Ghallab 1996) (Vidal and Fargier 1999) (Muscettola, Morris, and Vidal 2001) Simple Temporal Networks with Uncertainty (STNUs) consist of* Controllable *time-points,* $A = \cup_i a_i$*, i.e. those assigned by the agent, and* Uncontrollable *time-points,* $R = \cup_i r_i$*, i.e. those assigned by the external world. The set of timepoints* $T = A \cup R$*. The domain of* $t_i \in T = \mathbb{R}$. Requirement constraints $c(t_i, t_j)$ *have the form* $(t_j - t_i) \in [l_{t_i,t_j}, u_{t_i,t_j}]$*. Let* $C = \cup_{t_i,t_j}\ c(t_i, t_j)$. Contingent constraints $g(a_i, r_j)$ *have the* *form* $(r_j - a_i) \in [l_{a_i,r_j}, u_{a_i,r_j}]$ *where* $a_i \in A$, $r_j \in R$; *the semantics is that* $\exists\, v(r_i) \in [l_{a_i,r_j}, u_{a_i,r_j}] \mid r_j - a_i = v(r_i)$, *but* $v(r_i)$ *is only observed during execution. Let* $G = \cup_{a_i,r_j} g(a_i, r_j)$. *An STNU is a 4-tuple* <*A,R,C,G* >.

Definition 2 (Strong Controllability). *(Vidal and Fargier 1999) Let* P *be an STNU. Let* $V = \times_{g_{a_i,r_j}} [l_{a_i,r_j}, u_{a_i,r_j}]$ *(the cross product of all possible outcomes of all contingent constraints). A schedule* s *is an assignment to* $a_i \in A$. *Denote the value of* a_i *in* s *by* $s(a_i)$. P *is* Strongly Controllable *(SC) if there is a schedule* s *such that* $\forall v \in V$, s *satisfies all constraints* $c(t_i, t_j)$.

Definition 3 (PSTN). *(Tsamardinos 2002) Let a* probabilistic duration constraint $d(a_i, r_j)$ *have the form* $r_j - a_i = \omega \in \Omega_{a_i,r_j}$ *where* $a_i \in A$, $r_j \in R$, *and* Ω_{a_i,r_j} *is a random variable with probability distribution function* $P(\Omega_{a_i,r_j})$. *Let* $D = \cup_{a_i,r_j} d(a_i, r_j)$. *(Duration constraints* $d(r_i, r_j)$ *are not permitted for reasons explained later.) A Probabilistic Simple Temporal Networks (PSTN) is a 4-tuple* <*A,R,C,D*>.

In the sequel, we will assume w.l.o.g. that there is a 1-1 mapping between probabilistic duration constraints and controllable timepoints, i.e. $\forall\{d(a_i, r_k), d(a_j, r_m)\}$, $a_i \neq a_j$, allowing us to say $r_j - a_i = \Omega_j$.

Risk as introduced by (Fang, Yu, and Williams 2014) describes the probability that, given a schedule or strategy, an outcome $v \in V$ violates some constraint. Typical approaches transform a PSTN into an STNU and then evaluate controllability. To compute risk for an STNU, we measure how much probability mass on each uncertain duration is not covered after 'squeezing' to transform it into a contingent link, i.e. transforming $d(a_i, r_j)$ to $g(a_i, r_j)$, in a manner similar to (Santana et al. 2016):

Definition 4. *Let* ρ_d:$D \Rightarrow G$ *transform a duration constraint into a contingent link by choosing a compact subset* $[l_{a_i,r_j}, u_{a_i,r_j}] \subset \Omega_j$. *Let* $\rho_D = \{\rho_d\}$. *Let* P *be a PSTN. Then* $\rho_D(P) = U$ *where* U *is the STNU derived from* P.

Definition 5. *Let* P *be a PSTN. Let* $U = \rho_D(P)$ *be an STNU derived from* P. *Let* $\rho_d(d(a_i, r_j)) = g(a_i, r_j)$. *Let* $[l_{a_i,r_j}, u_{a_i,r_j}]$ *be the contingent constraint interval defined by* $g(a_i, r_j)$. *Let* $\Phi_g = \omega \in \Omega_j | \omega \leq l_{a_i,r_j}$. *Let* $\Theta_g = \omega \in \Omega_j | \omega \geq u_{a_i,r_j}$. *The* risk *of* $d(a_i, r_j)$ *relative to* ρ_d, *denoted* $\delta(\rho_d, d(a_i, r_j))$, *is* $\int_{\omega \in \Phi_g \cup \Theta_g} P(\Omega_j)$. *The symmetric case of* $d(r_i, a_j)$ *is similar. The* risk *of* P *relative to* ρ_D, *denoted* $\delta(P, \rho_D)$, *is* $1 - \left(\prod_{d \in D}(1 - \delta(\rho_d, d(t_i, t_j)))\right)$.

Definition 6. P *is SC with risk* Δ *if* $\exists U = \rho_D(P)$, P' *is SC, and* $\delta(P, \rho_D) = \Delta$.

3 A Running Example and Previous Work

Consider a planetary exploration rover that must drive to a location, take an image, and then uplink data. The image takes 10 minutes to collect, and the best time window to take the image is (initially) between 40 and 50 minutes after starting the drive, denoted $c^1(a_1, a_2) = [40, 50]$. The uplink takes 5 minutes, and the orbiter will be visible from 60 to 70 minutes after starting the drive. The rover drive duration is expressed by a probability distribution. The PSTN is shown

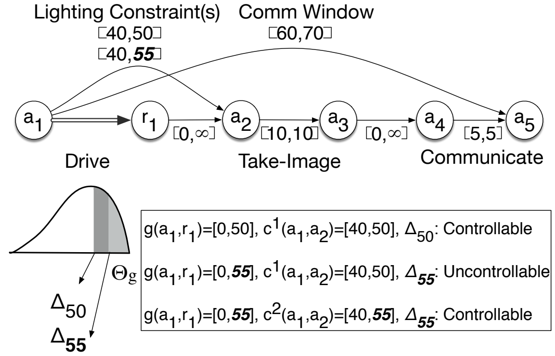

Figure 1: Sample Problem showing how relaxing or sacrificing a constraint can reduce risk.

in Figure 1. In order to transform this PSTN into an STNU, we search over $\rho(d(a_1, r_1)) = g(a_1, r_1)$ to either minimize the risk, or satisfy some risk bound Δ. As in Definition 1, denote the observed value of r_1 by $v(r_1)$. If $g(a_1, r_1) = [0, 50]$ then the resulting STNU is strongly controllable; the schedule $s(a_2)$=$s(a_1)$+50, $s(a_3)$=$s(a_1)$+60, $s(a_4)$=$s(a_1)$+60, $s(a_5)$=$s(a_1)$+65 is valid for any value of $r_1 \in \Omega_1$. Suppose we deem the resulting risk of $v(r_1) > s(a_2)$, which violates $c(a_1, r_2)$, to be too high; we prefer a lower risk option, e.g. $s(a_2)$= 55, requiring $g(a_1, r_1) = [0, 55]$[1]. Unfortunately, if $50 < v(r_1) \leq 55$ then the lighting constraint $c^1(a_1, a_2)$ is guaranteed to be violated; thus, with the lower risk bound, the resulting STNU is not strongly controllable.

One previously explored approach for such problems is to search over *relaxations* for a problem that can be transformed into a controllable STNU with some bounded risk; (Yu, Fang, and Williams 2015) use this approach for conditional STNUs. The search is guided by costs of relaxations of either the requirement constraints or the risk bound. In the example above, the relaxed constraint $c^2(a_1, a_2) = [40, 55]$ combined with $g(a_1, r_1) = [0, 55]$ leads to a controllable STNU. While this approach ensures no *relaxed* constraints in the transformed STNU will be violated at execution time, the original constraints are lost, so there is no information to guide generation of the strategy to avoid violating constraints unnecessarily. Modeling allowable constraint violations could be addressed by first relaxing the bounds on the requirement constraints, and adding *preferences* that value satisfying the original constraint more than the relaxed bounds. This could be done using simple semi-convex preference functions, combined with 'min', to achieve tractability, as in (Rossi, Venable, and Yorke-Smith 2006). This approach is too limiting; in particular, the 'min' function will report the worst preference achieved for any constraint, which could be 0 (representing a 'violated' constraint).

In our example, we would like to trade the likelihood of satisfying the lighting constraint $c(a_1, a_2)$ while ensuring the image is acquired after the drive is complete, i.e. $c(r_1, a_2)$, at execution time. The right strategy depends on the change in risk of satisfying $c(r_1, a_2)$, and the relative im-

[1]Assigning $s(a_2)$> 55 requires violating multiple constraints.

portance of satisfying $c(a_1, a_2)$ and $c(r_1, a_2)$. The expected value formulation is common in MDPs; while the relaxation approach in (Yu, Fang, and Williams 2015) minimizes the cost of relaxations, it does not the expected value of the controllability strategy. The continuous time nature of the state space precludes using formulations such as time-dependent MDPs (Boyan and Littman 2000); the desire to express state spaces representing violated constraints makes other time-based MDP approaches e.g. (Weld and Mausam 2006) inappropriate.

4 The EPSTN

We now formalize the Expected Value Probabilistic Simple Temporal Network (EPSTN) by adding constraint valuations $q_c(t_i, t_j)$, to a PSTN. We then formalize the Expected Value Strong Controllability (EvSC) problem on EPSTNs.

Definition 7 (EPSTN). *Let $q_c(t_i, t_j)$: $c(t_i, t_j) \Rightarrow \mathbb{R}^+$ and let Q be the set of all $q_c(t_i, t_j)$. An Expected Value Probabilistic Simple Temporal Network (EPSTN) is a 5-tuple <A,R,C,D,Q>.*

Definition 8. *Let Pe be an EPSTN. Let s be a schedule. Let $\sigma_c(t_i, t_j, s, v) \Rightarrow \{0, 1\}$ be 1 if $c(t_i, t_j)$ is satisfied by (v, s) (by extracting $v(r_i)$ for $t_i = r_i$ or $s(a_j)$ for $t_i = a_i$ and evaluating the bounds) and 0 otherwise. Then $f_s(s, v) = \sum_{c \in C} q_c(t_i, t_j)\ (\sigma_c(t_i, t_j, s, v))$ is the value of a schedule s combined with a set of outcomes $v \in V$. The expected value of s is then $E(f_s(s, V)) = \int_{v \in V}(P(v) f_s(s, v))$. Given an EPSTN, the* Expected Value Strong Controllability *(EvSC) problem is to find s maximizing $E(f_s(s, V))$.*

While a similar Dynamic Controllability problem can also be formalized, for the remainder of the paper, we will focus on Expected Value Strong Controllability.

We do not assume that the requirement constraints $c(a_i, a_j)$ (those exclusively on controllable timepoints) are all simultaneously satisfiable. EPSTNs also may, in general, have multiple constraints $c^k(t_i, t_j)$ on the same pair of variables t_i,t_j, with different valuations. Consider a variant of Figure 1 with *both* constraints $c^1(a_1, a_2)$ and $c^2(a_1, a_2)$, and $q_c^2(a_1, a_2) \le q_c^1(a_1, a_2)$. Both constraints cannot be satisfied simultaneously; choosing whether to satisfy $c^1(a_1, a_2)$ or $c^2(a_1, a_2)$ will change the expected value of satisfying $c(r_1, a_2)$. Compare this to a different variant of Figure 1, with constraint $c^1(a_1, a_2)$ and constraint $c^3(a_1, a_2)$ with bounds $[40, 55]$. In this case, both constraints $c^1(a_1, a_2)$ and $c^3(a_1, a_2)$ can be satisfied by some schedules and outcomes, leading to a more complex tradeoff analysis.

EPSTNs are a variant of the Disjunctive Temporal Problem with Preferences (DTPP) (Peinter, Moffitt, and Pollack 2005). Each requirement constraint can be expressed as a disjunction where satisfying the 'trivial' constraint has zero value and satisfying the original constraint has value $q_c(t_i, t_j)$. The EPSTN is a *generalization* of the DTPP since the expected value of satisfying $c(r_i, a_j)$ is not crisply expressed as a finite disjunction of constraints with preference values for each disjunction. While the value of satisfying $c(a_i, a_j)$ is captured by $q_c(a_i, a_j)$, the expected value of satisfying $c(r_i, a_j)$ is a nontrivial function of timepoint assignments, rather than a constant associated with the disjunctive decisions. EPSTNs are similar to the Controllable Conditional Temporal Problem with Uncertainty (CCTPU) of (Yu, Fang, and Williams 2015), in that we can choose which constraints to satisfy. EPSTNs are more general than CCTPUs in that they include preferences, but are more limited in that every timepoint of an EPSTN must be scheduled.

We now look deeper at the fundamental tradeoff in EvSC: sacrificing a constraint to improve the overall expected value of a schedule. In Figure 1 above, there is only one constraint over an uncontrollable timepoint, namely $c(r_1, a_2)$. If $50 \le v(r_1) \le 55$ we can construct a schedule violating a single constraint, namely, $c(a_1, a_2)$, in order to satisfy $c(r_1, a_2)$ and ensure all other constraints are satisfied. Committing to a schedule up-front that violates $c(a_1, a_2)$ lets us increase the probability $c(r_1, a_2)$ is satisfied, potentially increasing the expected value of the schedule. We will ignore the other outcomes for now since multiple constraints will be violated, making it harder for increased probability to compensate for the loss of value. We can determine the *relative* values that $q_c(a_1, r_1)$ and $q_c(a_1, a_2)$ must be to make violating $c(a_1, a_2)$ maximize the expected value. Assume $s(a_1)$= 0. Let s be a schedule in which $s(a_2)$= 50 and s' be a schedule in which $s(a_2)$= 55. For s' to be preferred, we would need

$$q_c(r_1, a_2) \int_0^{50} P(\Omega_1) + q_c(a_1, a_2) < q_c(r_1, a_2) \int_0^{55} P(\Omega_1)$$

or, written another way,

$$q_c(a_1, a_2) < q_c(r_1, a_2) \left(\int_0^{55} P(\Omega_1) - \int_0^{50} P(\Omega_1) \right) =$$

$$q_c(a_1, a_2) < q_c(r_1, a_2) \int_{50}^{55} P(\Omega_1)$$

If $\int_{50}^{55} P(\Omega_1)$ is 'small', the inequality above is only satisfied if $q_c(r_1, a_2)$ is 'large'. Put another way, $q_c(r_1, a_2)$ would need to be a factor of $\frac{1}{\int_{50}^{55} P(\Omega_1)}$ larger than $q_c(a_1, a_2)$. Committing to bounds on contingent links that maximize the expected value for a high-value constraint on an uncontrollable may violate other constraints. We may intuitively view this as creating a *cycle* that must be broken by deleting a low-value constraint. It may be necessary to remove multiple constraints to increased coverage of a single high-value uncontrollable duration. Breaking each cycle may reveal another cycle involving another requirement constraint, perhaps even on the same timepoints, as shown in Figure 2 (left). In this example, it is obvious that breaking the minimum cost edge unexpectedly *decreases* the expected value. Even if removing a series of constraints increases the probability of satisfying a contingent link, the expected value can still decrease, until removing enough constraints leads to an eventual *net increase* in the expected value. This scenario is shown in Figure 2 (right). The situation becomes more complex when we consider that removing a single constraint might lead to increased probability, and therefore expected value, of multiple high-value contingent constraints.

Our roadmap for EvSC is as follows. We first describe a restricted EPSTN that can be written as a linear program. We solve the more general EPSTN problem by searching over restricted EPSTNs derived from the most general problem. Along the way, we provide some insights into algorithm complexity, soundness and completeness of these approaches, which shed light on the difficulty of addressing expected controllability.

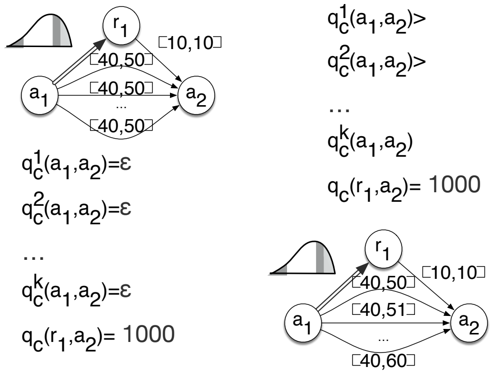

Figure 2: Breaking cycles.

4.1 Semi-Simple EPSTNs

We begin this analysis with some definitions:

Definition 9 (Simple and Semi-Simple EPSTN)**.** *Let P_e be an EPSTN. Denote the STN constraints $c(a_i, a_j)$ by C_s. Denote the At-Risk (AR) constraints $c(r_i, a_j)$ by C_u. P_e is* Semi-Simple *if $\exists$ s that satisfies all constraints in C_s. $P'_e \subset P_e$ defined by C_s' $\subseteq C_s$ is* Simple *if $\exists$ s s.t. $E(f_s(s, V))$ is optimal over all P'_e, and s satisfies all constraints in C_s'.*

Suppose we fix, or are otherwise given, a set of STN constraints over controllable timepoints C_s that can all be satisfied. The resulting EPSTN is semi-simple, but in general will not be simple. We want to search for a schedule to some PSTN that maximizes the expected value, given our EPSTN and this fixed set of constraints. However, we start with an EPSTN P_e and generally don't know the set C_s' that will lead to optimality. Thus it appears *three simultaneous searches* are required: a subset of C_s leading to optimality, a strongly controllable $U' = \rho_D(P')$ given that subset of STN constraints, and the best SC schedule (that is, the one maximizing the expected value of satisfied constraints) for P'.

Fortunately, we have methods for finding $U = \rho_D(P)$ such that P is SC: SREA (Lund et al. 2017) provides one such method. In SREA, a series of LPs are constructed and solved in order to minimize the risk; a feasible LP is guaranteed to be SC. Our approach will use a similar LP to optimize the expected value of some semi-simple EPSTN derived from our problem P_e. Once we know how to do this, we can search over all such EPSTNs to find the optimal, simple EPSTN. However, we can't use the SREA LP formulation directly. First, the LP of SREA does not explicitly represent the risk. Instead, SREA uses an outer-loop search over ever reducing risk bounds on each $d(a_i, r_j)$ to construct candidate transformations $\rho_d(d(a_i, r_j)) = g(a_i, r_j)$ (in our notation), represented as constraints within the LP. In order to compute the expected value, we need some additional LP machinery to represent the actual risk explicitly in the LP.

Second, we don't merely want to minimize the risk that the outcomes don't respect the bounds of the contingent constraints, but rather maximize the expected value of a schedule. Specifically, we need to explicitly represent the probability that an outcome v leads to a violation of each AR constraint. This means we can't directly use $\delta(\rho_d, d(a_i, r_j))$ from Definition 5, because this definition computes the risk by measuring how much of the probability mass of $d(a_i, r_j)$ is covered by $\rho_d(d(a_i, r_j))$.

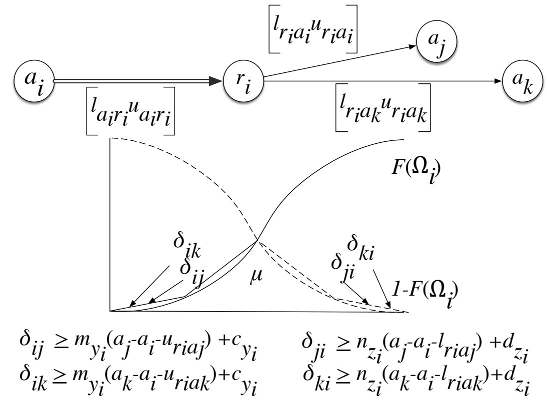

Figure 3: z**fixed fig relative to submitted version** Piecewise linear bounding approximation of the CDF $F(\Omega_i)$ and relevant constraints. δ_{ij} is the risk of violating the upper bound on $a_j - r_i$, δ_{ji} is the risk of violating the lower bound on $a_j - r_i$. δ_{ij} and δ_{ik} are the risks of violating the upper bounds on $a_j - r_i$ and $a_k - r_i$ respectively; both are bounded by the same piecewise linear constraints bounding $F(\Omega_i)$.

When computing the expected values, it is helpful to imagine a triangle of two requirement constraints $c(a_i, a_j)$, $c(r_i, a_j)$, and a duration constraint $d(a_i, r_i)$. Since we have assumed P_e is semi-simple, we know s must satisfy $c(a_i, a_j)$. Ideally, $v(r_i)$ and $s(a_j)$ will satisfy the constraint $c(r_i, a_j)$. But an outcome $\omega_i \in \Omega_i$ may be unlucky, either violating the lower bound l_{r_i,a_j}, because a_i and a_j are scheduled close together to satisfy $c(a_i, a_j)$ and ω_i is too large, or violating the upper bound u_{r_i,a_j}, because a_i and a_j are scheduled far apart and ω_i is too small.

For a given schedule s satisfying all constraints in C_s, we must compute for each constraint $c(r_i, a_j)$ in C_u and the relevant contingent link $g(a_i, r_i)$, $\Phi_c(s) = \omega_i \in \Omega_i |\, s(a_j) - v(r_i) \leq l_{r_i,a_j}$ and $\Theta_c(s) = \omega_i \in \Omega_i |\, s(a_j) - v(r_i) \geq u_{r_i,a_j}$. Recall $v(r_j) = s(a_i) + \omega_i$. We find it is convenient to transform $P(\Omega_i)$ into its Cumulative Distribution Function (CDF); points on the CDF $F(\Omega_i)$ of $P(\Omega_i)$ directly represent the risks $\int_{\omega_i \in \Phi_g} P(\Omega_i)$ or $\int_{\omega_i \in \Theta_g} P(\Omega_i)$.

We now show how to augment the LP in SREA by using a piecewise linear bound on the CDF. We assume $P(\Omega_i)$ is *unimodal* in order for F and $1 - F$ to be *convex* over their respective regions. For the risk of violating the upper bound, $\int_{\omega_i \in \Theta_g} P(\Omega_i)$, we want to compute, in the LP, the probability that the outcome ω_i is too small given the specific schedule s. This becomes

$$\int_{\omega_i | s(a_j) - (s(a_i) + \omega_i) \geq u_{r_i,a_j}} P(\Omega_i) =$$
$$\int_{\omega_i | s(a_j) - s(a_i) - u_{r_i,a_j} \geq \omega_i} P(\Omega_i) =$$
$$\int_{-\infty}^{s(a_j) - s(a_i) - u_{r_i,a_j}} P(\Omega_i).$$

This may appear backwards, but it isn't. The reason is that we are reasoning about the upper bound on the distance a_j

$-r_i$. For fixed $s(a_j)$, the earlier $v(r_i)$ occurs, the larger this distance, until it will exceed the upper bound. Thus, the integral ranges from $-\infty$ to the lowest value u_{r_i,a_i} violating the AR constraint $c(r_i, a_i)$. The CDF $F(\omega_i)$ is approximated by **fixed this relative to submitted version** y_i linear constraints, whose slopes are m_{iz} and intercepts are c_{iz}. We need an auxiliary variable δ_{ij} in the LP representing the candidate value of the upper bound integral, i.e. the risk of violating $c(r_i, a_j)$. Then the relevant constraints on δ_{ij} take the form

$\delta_{ij} \geq m_{y_i}(s(a_j) - s(a_i) - u_{r_i,a_j}) + c_{y_i}$

Thus, δ_{ij} is constrained to a convex zone greater than F in the range $[0, \mu]$ where μ is the mode of $P(\Omega_i)$. We must also ensure that the linear constraints bounding any *pair* of AR constraints $c(r_i, a_j)$, $c(r_i, a_k)$ on the same uncontrollable timepoint r_i are derived from the *same* linear constraints on $F(\Omega_i)$. We see the constraints bounding δ_{ij} above. To bound δ_{ik}, we also have the following constraints:

$\delta_{ik} \geq m_{y_i}(s(a_k) - s(a_i) - u_{r_i,a_k}) + c_{y_i}$

Note the *same* linear coefficients, derived from the shared $F(\Omega_i)$, bound both risks δ_{ij} and δ_{ik}. However, each risk value's location on the linear constraints bounding $F(\Omega_i)$ is computed from different timepoint assignments and the upper bound derived from $c(r_i, a_j)$ as opposed to $c(r_i, a_k)$. (This is comparable to an LP with three free variables constrained by the same linear constraints.) This construction ensures the correct risk is extracted from $F(\Omega_i)$ for each requirement constraint on r_i, and is shown in Figure 3.

We now describe the linear constraints for the risk of violating the lower bound on the separation, $\int_{\omega_i \in \Phi_g} P(\Omega_i)$, corresponding to an outcome in which $s(a_j) - v(r_i)$ is too small because r_i occurs too late:

$\int_{\omega | s(a_j) - (s(a_i) + \omega_i) \leq l_{r_j,a_j}} P(\Omega_i)$
$= \int_{s(a_j) - s(a_i) - l_{r_j,a_i}}^{\infty} P(\Omega_i)$

We need a second auxiliary variable δ_{ji} representing the candidate value of the lower bound integral. Note that δ_{ji} is constrained by by $1 - F$, so we must construct our linear constraints to bound $1 - F$ in this range. **fixed this relative to submitted version** If there are z_i linear constraints, the bounds on δ_{ji} take the form

$\delta_{ji} \geq n_{z_i}(s(a_j) - s(a_i) - l_{r_j,a_i}) + d_{z_i}$

The value of $1 - F$ bounding below δ_{ji} in the region $[\mu, 1]$ is convex. This is also shown in Figure 3.

We must now choose the points at which we construct the slopes. We note δ_{ij} and δ_{ji} are bounded below by the linear constraints in their respective regions. We want to bound above the true risk, represented by F or $1 - F$, in order for solutions to the LP to conservatively bound below the true expected value. For the bound on δ_{ij}, this can be done by selecting **fixed this relative to submitted version** $y_i - 1$ points on F and computing the slopes of the lines between them; for the bound on δ_{ij}, this can be done by selecting $z_i - 1$ points on $1 - F$ and computing the slopes of the lines between them. We can choose both the number $z - 1$ and location of the points to control error and overall size of the LP, trading runtime for accuracy.

Handling the case of $c(r_i, r_j)$ requires computing $P(\Omega_i - \Omega_j)$; even if $P(\Omega_i)$ and $P(\Omega_j)$ are unimodal, the composition in general is not (Ibragimov 1956), and thus free constraints $c(r_i, r_j)$ are not permitted in our EPSTNs [2].

For a given semi-simple EPSTN, it is not initially know what contingent constraint bounds l_{a_i,r_j}, u_{a_i,r_j} result in a schedule of maximum expected value, or even a controllable PSTN. The LP of (Lund et al. 2017) fixes these bounds when solving the LP, and they are adjusted by outer-loop search by SREA in search of a controllable STNU. We transform these bounds into LP variables, allowing the search for the optimal value in the LP to widen the bounds as much as possible. This modified LP has a solution for *any* semi-simple EPSTN by construction. Since, by definition the STN constraints of a semi-simple EPSTN have a feasible solution, setting u_{a_i,r_j} and l_{a_i,r_j} to any values consistent with the STN solution initially produces a controllable STNU, but one whose expected value is certainly suboptimal. The modification preserves the *feasible* solutions of the SREA LP, but now allows search over the contingent constraint bounds within the LP in order to maximize the expected value, thereby producing new *optimal* solutions. In particular, for a given EPSTN, the new LP can find the optimal solution using *any* set of contingent constraint bounds, thereby simultaneously searching over $\rho_d(d(a_i, r_j))$ and schedule.

Finally, we are ready to write the objective function. Boole's Inequality bounds above the risk by $\sum_{d \in D} \delta(\rho_d, d(t_i, t_j))$. For each AR constraint $c(r_i, a_j)$ we want to maximize the quantity $q_c(r_i, a_j)\ (1 - \delta_{ij} - \delta_{ji})$. This objective drives δ_{ij} and δ_{ji} to be as small as possible, pushing both risk variables in the proper direction, along the linear constraints bounding above F and $1 - F$, and towards the extremes.

We now write the full LP, based on the SREA LP of (Lund et al. 2017) combined with the additional variables and constraints discussed above. **fixed this relative to submitted version; changed linear coefs and fixed error in $\delta_j i$ constraints**

$$\max \sum_{c(r_i,a_j),\delta_{ij},\delta_{ji}} (q_c(r_i,a_j)(1 - \delta_{ij} - \delta_{ji})) \tag{1}$$

$$\text{s.t.}\, t_i^+ \geq t_i^- \qquad \forall t_i \in T \tag{2}$$

$$t_j^+ - t_i^- \leq u_{t_i,t_j} \qquad \forall c(t_i,t_j) \in C \tag{3}$$

$$t_i^+ - t_j^- \geq l_{t_i,t_j} \qquad \forall c(t_i,t_j) \in C \tag{4}$$

$$r_j^+ - a_i^+ = u_{a_i,r_j} \qquad \forall g(a_i,r_j) \in G \tag{5}$$

$$r_j^- - a_i^- = l_{a_i,r_j} \qquad \forall g(a_i,r_j) \in G \tag{6}$$

$$a_i^- \leq a_i \leq a_i^+ \qquad \forall a_i \in A \tag{7}$$

$$\delta_{ij} \geq m_{y_i}(a_i - a_j - u_{r_i,a_j}) + c_{iz} \qquad \forall c(r_i,a_j) \in C_u \forall y_i \tag{8}$$

$$\delta_{ji} \geq n_{z_i}(a_i - a_j - l_{r_i,a_j}) + d_{jz} \qquad \forall c(r_i,a_j) \in C_u \forall z_i \tag{9}$$

$$0 \leq \delta_{ij} \leq 1 \qquad \forall (a_i,r_j) \in C_u \tag{10}$$

$$0 \leq \delta_{ji} \leq 1 \qquad \forall (a_i,r_j) \in C_u \tag{11}$$

$$t_0^+ = t_0^- = 0 \tag{12}$$

$$t_i^- \geq t_0^+ \qquad \forall t_i \in T \tag{13}$$

[2] The sum, and thus difference, of two independent normally distributed variables is a normal, and thus unimodal; under these and similar conditions the formulation provided works and $c(r_i, r_j)$ are permitted. Otherwise, constraints $c(r_i, r_j)$ and preferences $q_c(r_i, r_j)$ can be modeled, with some difficulty, by two constraints $c(a_k, r_i)$ and $c(a_k, r_j)$.

The constants in this LP are the $q_c(t_i, t_j)$ and the slopes and intercepts of the linear bounds on F and $1 - F$, and the STN constraint bounds l_{a_i,r_j}, u_{a_i,r_j} and l_{a_i,r_j}, u_{a_i,a_j}. The remaining symbols, including the contingent constraint bounds l_{r_i,a_j}, u_{r_i,a_j}, are the LP variables. We denote upper and lower bounds of timepoints generically by t_i^+, t_i^-, specializing to $a_i^+, a_i^-, r_i^+, r_i^-$ when necessary. Constraints 2, 3 and 4 ensure the STNU requirement constraints are satisfied. Constraints 5 and 6 ensure the STNU contingent constraint bounds are respected. Constraint 7 ensures the timepoint assignment, needed to compute δ_{ij} and δ_{ji}, is consistent with the bounds on the timepoints. Constraints 8 and 9 are the piecewise linear bounds on δ_{ij}, δ_{ji} derived from F and $1 - F$. Observe that constraints 7, 8 and 9 indirectly dictate the values of the δs as a function of the contingent constraint bound choices; the wider the bounds, the more coverage of AR constraints they permit, but the tighter the constraints on the controllable timepoints, which could lead to constraint violations.

4.2 Complexity, Soundness and Incompleteness

The resulting LP has $2|T|$ variables, two per timepoint (upper and lower bounds). It has $|A|$ more variables (controllable assignments). It has $2|G|$ variables, for the lower and upper bounds on each contingent link. It has $2|C|$ constraints, 2 per requirement constraint. It has $2|G|$ constraints, 2 per contingent constraint. **fixed relative to submitted version** Denote $y_* = \max_i y_i$ (similarly (z_*); It has $2(y_* + z_*)|C_u|$ constraints, at most $2(y_* + z_*)$ per AR constraint on an uncontrollable timepoint, handling both the upper bound and lower bound violation risks. We also know $|G| = |R|$ since each uncontrollable timepoint is constrained relative to one controllable timepoint. Thus, Karamakar's Algorithm or interior point methods will take $O(|T|^3)$ to solve the resulting LP. This discussion proves the following:

Theorem 1. *Given a Semi-Simple EPSTN P_e, a schedule maximizing the expected value of the constraints can be found in $O(|T|^3)$.*

While in general there can be many such optimal schedules, and in fact, $\rho_d(d(a_i, r_j))$, leading to optimality, all such combinations with the same expected value are defined by a face of the polytope of the LP.

Given an EPSTN P_e, we know there is a simple EPSTN P'_e with $C_s' \subseteq C_s$. Even if P_e is semi-simple, it is still possible that the optimal $C_s' \subset C_s$. Clearly, the optimal solution can be found by relaxing *every* subset of the $O(|C_s|)$ requirement constraints $c(a_i, a_j)$ and using the LP described above as a sub-solver. If we enumerate each of the exponentially many subsets of requirement constraints and solve the LP for the resulting EPSTN as described above, we can simply keep track of the best solution found for each feasible LP. The objective function in the LP omits the value of the STN constraints, since they must be satisfied by solutions to the LP. It is a simple matter to ensure these values are added to the objective value of the best solution to the LPs. Depth first search requires no more than polynomial space, and has complexity $O(2^{|C_s|}|T|^3)$. This is consistent, in general, with results for DTPPs (Peinter, Moffitt, and Pollack 2005).

Both the LP for solving EPSTNs described in the previous section and the algorithm for the general EPSTN case are *sound*, in that a schedule satisfying the LP constraints has expected value bounded below by the value of the objective function. This is because of the piecewise linear function in the LP bounds above the probability of failure. However, we must conclude that the resulting algorithm is *incomplete*. While we can arbitrarily approximate the CDF F and its inverse with an increasing number of linear constraints, we will in general bound above the risk, and thus bound below the expected value, using this approach. Moreover, we have not explicitly assumed that any pair of probabilities $P(\Omega_i)$ and $P(\Omega_j)$ in an EPSTN are *conditionally independent*. If the uncertain durations are not independent, then the allocation of risk will generally over-constrain the STNU. To see this, note that the STNU captures risk in an n-dimensional box, while correlated risk distributions could be covered by other 'shapes' with less restrictive STNU constraints.

4.3 Testing for Simplicity

The search described above can be implemented as a tree search over sets of requirement constraints to exclude from our EPSTN in the quest for the schedule maximizing the expected value. An inexpensive test for simplicity can be used to terminate the exponential search described above, and potentially reduce search time. This test leverages the existence of an optimal solution for a semi-simple EPSTN. Consider an optimal schedule $s*$ for a semi-simple EPSTN P_e, and consider the triangle formed by constraints $g(a_i, r_j)$, $c(r_i, a_j)$ and $c(a_i, a_j)$. From the construction of the LP, we know the expected value of $c(r_i, a_j)$ in $s*$ is

$q_c(r_i, a_j)\ (1 - \delta_{ij}^{s*} - \delta_{ji}^{s*})$

where

$\delta_{ij}^{s*} = \int_{-\infty}^{s(a_j)-s(a_i)-u_{r_i,a_j}} P(\Omega_i)$.

$\delta_{ji}^{s*} = \int_{s(a_j)-s(a_i)-l_{r_i,a_j}}^{\infty} P(\Omega_i)$.

The sacrifice of any constraint can only improve the expected value of this contingent constraint by at most $q_c(r_i, a_j)\ (\delta_{ij}^{s*} + \delta_{ji}^{s*})$. This leads to the following definition:

Definition 10 (Gain)**.** *Given a semi-simple EPTSN P_e, and a schedule $s*$ maximizing the expected value. Let δ_{ij}^{s*} be the value of δ_{ij} in the optimal solution to the LP defining $s*$. The* gain $\gamma(P_e, s*) = \sum_{c(r_i,a_j)\in C_u} (q_c(r_i, a_j)\ (\delta_{ij}^{s*} + \delta_{ji}^{s*}))$.

By construction, the gain for of each possible optimal solution will be identical. Let $v(s*)$ denote the value of an optimal solution to the LP constructed from a particular semi-simple EPST. The gain is simply

$\sum_{c(r_i,a_j)\in C_u}(q_c(r_i, a_j)\ ) - v(s*)$,

demonstrating that the gain is identical for all optimal $s*$. Even so, the values of δ_{ij}^{s*} and δ_{ji}^{s*} can vary on the polytope surface. As we will now see, it is convenient to define the gain in terms of the values of δ_{ij}^{s*} in the solution defining $s*$. A straightforward test for simplicity follows:

Lemma 1. *Given a semi-simple EPTSN P_e, and a schedule $s*$ maximizing the expected value. Then P_e is simple if $\gamma(P_e, s*) \leq \min_{c(a_i,a_j)\in C_s} q_c(a_i, a_j)$.*

The above test determines whether sacrificing the *least* valuable STN constraint can (optimistically) improve the expected value by *all* of the gain achievable. If even this optimistic tradeoff is not favorable, then s^* is one of the best possible schedules when all constraints are satisfied, therefore P_e is simple. The test is necessary but not sufficient; the test may fail when P_e is simple. In particular, if we look at Figure 1, we see that removing $c(a_4, a_5)$ cannot relax $c(a_1, r_1)$ because the two constraints are not on a cycle. If $q_c(a_4, a_5)$ $\geq \gamma(P_e, s*)$, the test would fail, and search might fruitlessly attempt to relax $c(a_4, a_5)$. If we can restrict the $c(a_i, a_j)$ considered in Lemma 1, or if we can bound the value of $\gamma(P_e, s*)$ that can be 'bought back' by relaxing a constraint, then we will have a more informative test for simplicity.

Consider again a triangle of constraints involving a_i, a_j and r_i, and two requirement constraints $c(a_i, a_j)$, $c(r_i, a_j)$, and a contingent constraint $g(a_i, r_j)$. There are two cycles in the distance graph D, one for the upper bound arc and one for the lower bound arc of $c(r_i, a_j)$. Once the assignment from s^* is found, we see that cycles in D are zero-cost because all timepoints are assigned a fixed time, and thus all timeoints are a fixed distance (upper bound equals lower bound). The edges on these cycles derived from the requirement constraints limit increasing coverage on the AR constraints. Consider any cycle including the directed arc $r_i \rightarrow a_i$. Intuitively, increasing u_{a_i,r_i} by ϵ potentially improves the expected cost, but would reduce the distance of this arc by $-\epsilon$, so any simple cycle including this now has distance $-\epsilon$. Similarly, decreasing l_{a_i,r_i} by ϵ reduces the distance of this arc by $-\epsilon$, so any simple cycle including this arc now has distance $-\epsilon$. This means eliminating edges on these 'negative cycles' is a possible avenue to increase coverage of the AR constraints, and thus the expected cost.

Enumerating cycles in graphs is generally exponential (Valiant 1979). Once we have identified all of the cycles in a graph, we can explore the tradeoffs of removing $c(a_i, a_j)$ by collecting the set of AR constraints $c(r_x, a_y)$ on cycles involving $c(a_i, a_j)$. Removing any STN constraints $c(a_i, a_j)$ on such a cycle can only increase the expected value by relaxing the cycles and allowing the coverage of more probability for these cases. These relaxations may decrease the expected value, as noted in Figure 2. The tradeoffs become complex when multiple uncertain durations are on a single cycle, or when cycles share edges, since relaxing a single requirement constraint could increase coverage of multiple uncertain durations. While cycle-sets could be identified algorithmically by finding the negative cycles in the distance graph, we want to find the STN constraints that could be deleted by search more directly. This motivates the following definition:

Definition 11 (Cycle-Set). *Given a semi-simple EPTSN P_e, let $G(P_e)$ be the undirected graph such that each edge of G is a constraint in P_e. Let $G(P_e, a_i, a_j) \subset G(P_e)$ be the subgraph consisting of all cycles including both edge (a_i,a_j)* and *a second edge (a_k,r_k) (it is possible that $a_k = a_i$). Then the* cycle-set Y_{a_i,a_j} *consists of* $c(a_i, a_j) \in G(P_e, a_i, a_j)$.

The intuition is that the cycle-set Y_{a_i,a_j} captures all AR constraints whose expected value could be improved by eliminating $c(a_i, a_j)$. or other free constraints $c(a_x, a_y) \in Y_{a_i,a_j}$. When multiple AR constraints are present, it is unclear whether the best strategy is to remove individual constraints or the shared constraints. Thus, the collection of all $c(a_x, a_y)$ over all cycle-sets identifies the set of STN constraints whose elimination could improve the expected value of a schedule, and therefore allows us to focus the search. In general, many cycle-sets may be equivalent; we will assume below that Y_{a_i,a_j} refers to the 'canonical' cycle-set for any free constraint $c(a_x, a_y) \in Y_{a_i,a_j}$. We can now give a more refined form of Lemma 1:

Theorem 2. *Given a semi-simple EPTSN P_e, and a schedule $s*$ maximizing the expected value. Let $Y = \cup(Y_{a_i,a_i})$. P_e is simple if $\gamma(P_e, s*) \leq \min_{c(a_i,a_j) \in Y} q_c(a_i, a_j)$.*

Excluding edges that can't help eliminate cycles reduces the number of scenarios in which a low-value STN constraint makes it appear that P_e is not simple, therefore reducing needless search. The next step in refining the test for simplicity is to observe that AR constraints can only be improved upon by deleting STN constraints within their cycle-set; hence, the increase in gain can be computed for each cycle-set *separately*. This is useful because the test now ensures the cost-benefit tradeoff only evaluates the increased expected value of AR constraints that could possibly be relaxed by removal of a low-cost requirement link.

Definition 12 (Cycle-Set-Gain). *Given a semi-simple EPTSN P_e, and a schedule $s*$ maximizing the expected value. Let δ_{ij}^{s*} be the value of δ_{ij} in the optimal solution to the LP defining $s*$. The* Cycle-Set-Gain $\gamma(Y_{a_i,a_j})$ $= \sum_{c(a_i,r_k) \in Y_{a_i,a_j}} (q_c(a_i, r_k)\ (\delta_{ij}^{s*} + \delta_{ji}^{s*}))$.

Corollary 1. *Given a semi-simple EPTSN P_e, and a schedule $s*$ maximizing the expected value. Then P_e is simple if $(\forall\ Y_{a_i,a_j})\ \gamma(Y_{a_i,a_j}) \leq \min_{c(a_i,a_j) \in Y_{a_i,a_j}} q_c(a_i, a_j)$.*

Corollary 1 is still a necessary but not sufficient condition for simplicity. It can, however, be more precise in identifying simplicity than Theorem 2; this is accomplished by bounding the benefit offset by any constraint deletion, as well as removing some low value constraints from consideration in the test, thereby further reducing the false-negative simplicity assessments. Unlike (Yu, Fang, and Williams 2015), removing constraints is not driven by constraint violations, but by identifying opportunities to improve the expected value.

Low-order polynomial time cycle finding algorithms that do not find all cycles can't be used to identify Y_{a_i,a_j} because we might incorrectly pass the simplicity test and prematurely terminate search. Identifying cycles with high run-time cost is also unacceptable as an inner-loop early-termination test. However, if the up-front cost of finding the cycles is paid, Corollary 1 can be used during search with low run-time cost. Notably, we must only store the *edges of the cycles* associated with each Y_{a_i,a_i}, not the actual cycles, meaning that a data structure of $O(|C_s||C_u|)$ size and similarly low order lookup cost is sufficient. As search continues, edges are removed, and cycles are broken. While incremental recomputation of the cycles is too expensive, at a minimum the removed edges can be accounted for, and

$\min_{c(a_i,a_j)\in T_{a_i,a_j}} q_c(a_i,a_j)$ updated for each Y_{a_i,a_i}, if necessary. Ad-hoc analysis of the cycles may allow more edges to be removed at low cost (e.g. single non-nested cycles). Finally, many EPSTNs with the same distance graph, but different $q_c(t_i,t_j)$ and bounds, can benefit from a one-time construction of the cycle-sets, thereby amortizing the up-front exponential cost.

4.4 Bounding the Search Costs

It is tempting to think that exponential search for the best subset of C_s in a semi-simple EPSTN is unnecessary. Unfortunately, as we saw above in the test for simplicity, in general this will not be the case; one might need to search over edges shared between cycles. However, we can use the gain to evaluate the *largest set* of STN constraints whose sacrifice could, possibly, be offset by the gain. The size of this set can be used to bound the search cost.

Theorem 3. *Given a semi-simple EPTSN P_e, and a schedule $s*$ maximizing the expected value. Let $M \subset C_s$ be the largest cardinality set such that $\sum_{c\in M} q_c(a_i,a_j) \leq \gamma(P_e,s*)$. Then the search cost cannot exceed $O\left(\left(\sum_{i=1}^{|M|}\binom{|C_s|}{i}\right)|T|^3\right)$.*

Proof. To compute the largest cardinality set M: sort $q_c(a_i,a_j)$ in increasing order. Begin with an empty set. Add $c(a_i,a_j)$ to the set until the next largest $q_c(a_i,a_j)$ would produce a set M such that $\sum_{c_{a_i}a_j\in M} q_c(a_i,a_j) > \gamma(P_e,s*)$. There are at most $\sum_{i=1}^{|M|}\binom{|C_s|}{i}$ subsets of STN constraints that could be offset by $\gamma(P_e,s*)$. Solving the LP after relaxing each subset costs $O(|T|^3)$. □

Theorem 3 shows that, if the difference between $\gamma(P_e,s*)$ and the 'typical' magnitude of $q_c(a_i,a_j)$ is small, then the resulting search cost will be low, since $|M|$ will tend to be small. On the other hand, if the difference between $\gamma(P_e,s*)$ and the 'typical' magnitude of $q_c(a_i,a_j)$ is large enough that $|M|$ scales as some fraction of $|C_s|$, then $\binom{|C_s|}{|M|}$ will grow exponentially, and search cost will as well. Theorem 3 is simple to use during search, but will generally over-estimate the exponential costs of search. This is because the bound does not take into account whether the requirement constraints' values that contribute to large $|M|$ will actually lead to increased utility by loosening constraints on an uncertain duration.

An alternative bound on the exponent can be computed by only counting STN constraints in cycle-sets; only deleting these constraints can actually lead to relaxing some AR constraints, and hence increase expected utility. This result exploits the independence of the cycle-sets. The AR constraints on each cycle-set can be relaxed independently; the maximum sized cycle-set therefore dominated the exponent in search. This bound is captured in the following result:

Theorem 4. *Given a semi-simple EPTSN P_e, and a schedule $s*$ maximizing the expected value. Let Y_{a_i,a_j} denote the cycle-sets. Let $Y_m = \max_{c(a_i,a_j)}(|Y_{a_i,a_j}|)$. Then the search cost upper bound is $O\left(2^{Y_m}|T|^3\right)$.*

Unfortunately, this result requires incurring the up-front exponential cost of finding cycle-sets.

Theorems 3 and 4 can be combined to further drive down the exponent by identifying, for each cycle set, the largest set of cheap constraints that could offset the gain on that cycleset.

Corollary 2. *Let M_{a_i,a_j} be the maximum cardinality subset of cycleset Y_{a_i,a_j} such that $\sum_{c\in Y_{a_i,a_j}} q_c(a_i,a_j) \leq \gamma(Y_{a_i,a_j})$. The maximum search cost is now $O(\sum_{i=1}^{|M_{a_i,a_j}|}\binom{|Y_{a_i,a_j}|}{i}T^3)$.*

The resulting exponent is guaranteed to be an improvement over using either bound alone. The advantage of using Theorem 3 is that it does not require finding cycles beforehand, but combining the two theorems leads to much stronger bounds than either alone.

5 Conclusions and Future Work

When presented with a control problem on probabilistic simple temporal networks, the usual strategy of establishing controllability may fail when constraints are too stringent. To address this, we formally define a new type of controllability problem, the Expected Value Probabilistic Simple Temporal Network (EPSTN), and address the Expected Value Strong Controllability (EvSC) problem of finding a schedule maximizing the expected value of satisfied constraints. We first extend a previously developed linear programming (LP) formulation to find a schedule for a special case. The expected value of this schedule bounds below the true expected value using a piecewise linear approximation of the probabilities in the EPSTN. This LP must be solved at each branch of a search that discards constraints to allow covering more and more probability mass. While this search may be exponential in the number of simple temporal constraints in the EPSTN, we bound the exponent by reasoning about the tradeoff between the lost value of each constraint and the expected gain. Search can be pruned using termination rules that depend on identifying unfavorable cost-benefit tradeoffs, which may depend on the cycle structure of the constraint graph.

An empirical study of EvSC will require generating problem instances that require trading STN constraints for increased coverage of the AR constraints. The problem classes of (Lund et al. 2017) and (Santana et al. 2016) are good starting points, since many of the problems in the datasets studied in these works are not strongly controllable. However, it isn't obvious how to choose $q_c(t_i,t_j)$ to create interesting problem classes. Performing a sensitivity analysis on $q_c(t_i,t_j)$ will lead to interesting problem class variants. Variations in the probability distributions and their characteristics are also worth investigating, especially asymmetric and heavy-tailed distributions. Once this is done, the best algorithms can be found by implementing and testing the simplicity tests and exponent-bounding concepts described in the theoretical results.

In this paper we have addressed only the problem of EvSC. The next step is to check for Expected Value Dynamic Controllability. Doing this will provide executives with the ability to respond dynamically to unexpected outcomes in order to maximize the expected value of satisfied constraints, which existing risk-bounding and constraint-relaxing strategies simply cannot do. The solution to this problem is likely to be quite different than the techniques described in this work.

References

Boyan, J., and Littman, M. 2000. Exact solutions to time-dependent MDPs. In *NIPS*, 1026–1032.

Fang, C.; Yu, P.; and Williams, B. 2014. Chance-constrained probabilistic simple temporal problems. In *Proceedings of the National Conference on Artificial Intelligence*, 2264 – 2270.

Ibragimov, I. A. 1956. On the composition of unimodal distributions. *Teor. Veroyatnost. i Primenen.* 1(2):283–288.

Lund, K.; Dietrich, S.; Chow, S.; and Boerkoel, J. 2017. Robust execution of temporal plans. In *Proceedings of the National Conference on Artificial Intelligence*, 3597 – 3604.

Muscettola, N.; Morris, P.; and Vidal, T. 2001. Dynamic control of plans with temporal uncertainty. In *Proceedings of the* 17^{th} *International Joint Conference on Artificial Intelligence*.

Peinter, B.; Moffitt, M. D.; and Pollack, M. E. 2005. Solving overconstrained disjunctive temporal problems with preferences. In *Proceedings of the* 15^{th} *International Conference on Automated Planning and Scheduling*.

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* 27:617–674.

Santana, P.; Vaquero, T.; Toledo, C.; Wang, A.; and Williams, B. 2016. Paris: A polynomial-time, risk-sensitive scheduling algorithm for probabilistic simple temporal networks with uncertainty. In *Proceedings of the National Conference on Artificial Intelligence*, 267 – 275.

Tsamardinos, I. 2002. A probabilistic approach to robust execution of temporal plans with uncertainty. In *Methods and Applications of Artificial Intelligence*, 97 – 108.

Valiant, L. G. 1979. The complexity of enumeration and reliability problems. *SIAM Journal of Computing* 8:410–421.

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

Vidal, T., and Ghallab, M. 1996. Dealing with uncertain durations in temporal constraint networks dedicated to planning. In *Proceedings of the* 12^{th} *European Conference on Artificial Intelligence*, 48 – 54.

Weld, D., and Mausam. 2006. Probabilistic temporal planning with uncertain durations. In *Proceedings of the National Conference on Artificial Intelligence*, 880 – 887.

Yu, P.; Fang, C.; and Williams, B. 2015. Resolving over-constrained probabilistic temporal problems through chance constraint relaxation. In *Proceedings of the National Conference on Artificial Intelligence*, 3425 – 3431.